with a useful method to access relevant audio segments quickly.

In the following subsections, a brief overview of the respective approaches is provided. For each of the CSJ core and ASR test sets (199 presentations in total), three subjects were asked to construct summaries according to the following guidelines, where the summarization ratio corresponds to the ratio of the length of the summary to that of the original text:

- a set of key sentences extracted from the transcript (50% and 10% summarization ratios in terms of the number of sentences)
- a summary generated by editing the transcript (50% and 10% summarization ratios in terms of the number of characters)

For the former, the subjects were instructed to extract important sentences, first choosing a set of 50% of the sentences in a talk, and then choosing a subset thereof to obtain an overall 10% summarization ratio.

32.6.2 Key Sentence Extraction

Extracting key sentences is not only an important step toward summarization, but is also useful for indexing speech archives. This is similar to the common human practice of underlining important portions of a text.

The basic scheme is to define a function to measure the importance of the sentences, and train a classifier based on the function. The machine learning approach is effective for many classification tasks in speech and language processing. However, annotation of importance is subjective and often inconsistent, and thus it is sometimes difficult to design the training scheme.

In text-based summarization studies using newspapers [32.100, 104], it is generally known that position information in articles or paragraphs is very useful because key information tends to be placed in the initial portion of the articles or paragraphs. Speech documents differ greatly from text in that they do not contain explicit structural information. Position can be measured only from the start or from the end, and this appears to be too simplistic for long speech materials such as lectures, although it is still useful for summarizing academic presentations at low summarization ratios such as 10% as described later. *Christensen* et al. [32.105] reported that the position information, which was the most effective for key sentence extraction from newspaper articles, is not nearly so effective for extraction from broadcast news.

Table 32.9 Results of key sentence extraction in the CSJ (50% summarization ratio)

Transcript	Segment	Recall	Prec.	F-measure
Manual	Manual	72.4	53.5	61.5
Manual	Auto.	72.7	46.5	56.7
Auto.	Auto.	76.1	45.5	56.9
(cf) by human subject		81.5	60.1	69.2

Kawahara et al. [32.98] proposed the use of discourse markers which suggest discourse boundaries, as described in Sect. 32.5.3, for key sentence indexing. The statistics of the discourse markers are used to determine the importance of sentences, which favors potentially section-initial sentences. This measure is also combined with the conventional tf-idf measure based on content words. Experimental results obtained using the ASR test set of the CSJ are summarized in Table 32.9. In this experiment, 37.5% of the sentences that were agreed upon by two subjects based on the 50% extraction condition were determined to be correct summaries. For reference, agreement by a third subject with this correct set was computed to estimate human performance (last row of Table 32.9). The results show that the performance of the system is only 10% lower than the human judgements. In Table 32.9, the effects of automatic transcription (ASR) and automatic sentence unit detection are also presented. It is observed that the sentence segmentation error of 20% (see Table 32.6) significantly degrades the performance, while the ASR error of 30% has little effect.

Latent semantic analysis (LSA) based on singular value decomposition (SVD) is also useful for extracting important sentences [32.106]. Each singular vector represents a salient topic, and the singular vector with the largest corresponding singular value represents the topic that is the most salient in the speech document. A fixed number of singular vectors having relatively large singular values are selected, and then for each singular vector, the sentence having the largest score is extracted as an important sentence. In this manner, the extracted sentences best describe the topics represented by the singular vectors and are semantically different from each other. The method based on the simple one-to-one mapping, however, may not generate a summary describing key information sufficiently. An enhanced method based on SVD [32.102, 107] defines an importance score in the reduced-dimensional space, and then selects sentences based on the score. Thus, the extracted sentences not only describe the significant topics but also have a latent relationship to each other.

Amazon EX1010 Page 660

Sameer et al. [32.108] investigated the usefulness of lexical, prosodic, structural, and discourse features in selecting extractive summaries from broadcast news stories. The lexical features include counts of named entities. The acoustic/prosodic features include speaking rate, F_0 features, log-energy features, and sentence duration. They are normalized for each speaker. The structural features include normalized sentence position and speaker type (reporter or not). The discourse features include the number of new nouns in each sentence. Experimental results showed that combination of acoustic/prosodic and structural features were the most effective.

32.6.3 Summary Generation

In an early study, summarization of dialogues within limited domains was attempted as part of the VERBMOBIL project [32.109]. *Zechner* and *Waibel* investigated how the accuracy of the summaries changes when methods for WER reduction were applied in summarizing conversations from television shows [32.110]. Recent studies on speech summarization in unrestricted domains have focused almost exclusively on broadcast news [32.101,111]. *Koumpis* and *Renals* investigated the transcription and summarization of voice mails [32.112]. Most of the previous studies on speech summarization used relatively long units, such as sentences or speaker turns, as minimal units for summarization.

Hori et al. [32.113] proposed a sentence-compaction-based summarization method, in which a set of words maximizing a summarization score is extracted from the transcript, according to a target summarization ratio. The extracted set of words is then connected to build a summary. The summarization score consists of:

- a word significance measure based on the tf-idf criterion
- a confidence measure output by ASR
- a linguistic likelihood given by n-gram probability
- a word concatenation probability determined by the dependency structure based on a stochastic dependency context-free grammar (SD-CFG)

The method was further extended to summarize a sequence of utterances, which results in a process of combining sentence extraction and compaction.

Kikuchi et al. [32.103, 114] proposed a two-stage summarization method consisting of important sentence extraction and word-based sentence compaction, as shown in Fig. 32.10. After removing fillers from the ASR output, a set of important sentences is extracted, and sentence compaction is applied to them. In the course of these processes, sentence and word units are extracted from the transcript and concatenated to produce a summary to maximize the summarization score mentioned above. The method was applied to broadcast news as well as oral presentations in the CSJ. It performed effectively for both English and Japanese speech summarization. It was also shown that sentence extraction plays a more important role than sentence compaction in improving summarization performance, especially when the summarization ratio is relatively low, say 10%.

In this context, *Hirohata* et al. [32.107] incorporated a more powerful sentence extraction method based on SVD. Sentence position information was combined to extract important sentences from the introduction and conclusion segments of each presentation. They also investigated the combination of a confidence measure and linguistic likelihood to effectively extract sentences with fewer ASR errors. This method improved the performance in the case of the 10% summarization ratio.

Because it is impossible to conduct human evaluation of automatic summarization results every time methods and parameters are changed, it is indispensable to develop objective evaluation metrics, much as is the case in machine translation research. It is still ideal to use manual summaries as the targets of the summarization system. Since the manual summaries vary according to human subjects, the way in which these variations are handled is an important problem.

Hori et al. [32.115] proposed to merge all of the human summaries into a single word network, which is considered to cover approximately all possible correct summaries. The word accuracy of the automatic

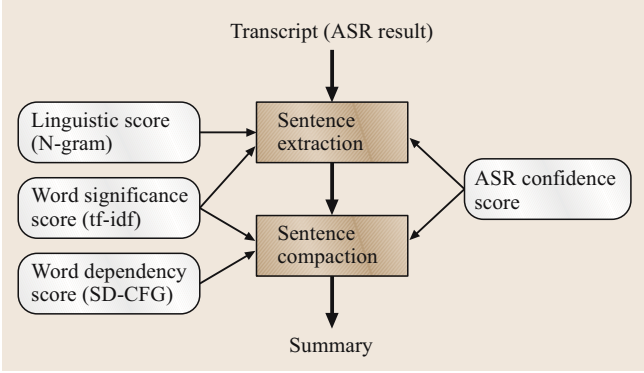


Fig. 32.10 Flow of two-stage speech summarization

Amazon EX1010 Page 661

summary is then measured as the summarization accuracy, SumACCY, by comparing the word sequence with the closest sequence extracted from the word network. This metric works reasonably well at relatively high summarization ratios such as 50%, but has problems at low summarization ratios such as 10%, because the variation between manual summaries is so large that the network accepts inappropriate summaries. Therefore, they have also proposed the use of the word accuracy obtained by using the manual summaries individually, SumACCY-E. In this metric, the largest score of SumACCY-E among all human summaries (SumACCY-E/max) or its average score (SumACCY-E/ave) is used.

ROUGE, which is the most widely used metric, is the *n*-gram recall between an automatic summary and a set of manual summaries [32.116]. The number of co-occurrences of 1-, 2-, or 3-grams in the manual summary and the automatic summary is usually used for this computation.

Hirohata et al. [32.107] investigated and evaluated these objective evaluation metrics in the framework of sentence extraction-based speech summarization for academic presentations with the condition of 10% summarization ratio. Correlation analysis between subjective and objective evaluation scores confirmed that the summarization accuracy and the ROUGE based on 2- or 3-gram recall were effective evaluation metrics.

32.7 Conclusions

We have reviewed the major research efforts concerning spontaneous speech processing, including speech recognition, metadata extraction, and speech summarization. Although many of the results were obtained using the corpus of spontaneous Japanese (CSJ), most of the conclusions should apply to all languages.

Spontaneous speech recognition requires a different paradigm from that of conventional ASR, in that a faithful transcript is not necessarily useful because of the existence of disfluency phenomena and the lack of punctuation and line breaking. Although it is arguable whether or not we should simply remove disfluencies, intelligent transcription, as performed by human transcribers or stenographers, must include the deletion of fillers and cleaning of self-repairs, which are addressed in Sect. 32.5. Moreover, it should also involve the recovery of omitted particles and the correction of colloquial expressions. There is a potentially huge demand for these types of intelligent transcription technologies, but the current level of ASR accuracy, which is approximately 70–80%, is not satisfactory. Thus, more studies on modeling and adaptation of acoustic and language models are needed, by considering the complex factors that affect the ASR performance in spontaneous speech. We should also explore a scheme that integrates the cleaning postprocess, or directly minimizes the edit distance with the cleaned transcript. An approach based on weighted finite-state transducers (WFST) is interesting for this purpose [32.117].

Spontaneous speech processing will expand the application of ASR technologies. First, it will make possible automatic indexing and metadata annotation for content-based access to huge audio archives, enabling text-based information retrieval and extraction techniques to be applied to audio materials. Next, it is also useful as a means of filtering speech documents for effective presentation in various ways, for example, by changing the summarization ratio for a given speech document. It has been shown that current technology can be useful for information retrieval or key-sentence extraction, and that ASR errors of 20–30% do not affect performance greatly. For further improvement, an integrated scheme should also be investigated. Minimum Bayes-risk (MBR) decoding can be formulated by considering the importance of the words in information retrieval or summarization [32.118].

In the future, integration with other media, such as video, should also be studied more intensively. Furthermore, integration with knowledge processing must be explored, because speech is a media for exchanging knowledge and is itself thus a source of knowledge.

References

32.1 D.S. Pallet, J.G. Fiscus, J.S. Garofolo, A. Martin, M.A. Przybocki: Broadcast news benchmark test results, In DARPA Broadcast News Workshop (1999) pp. 5–12

32.2 B. Chen, L.-S. Lee: Spoken document understanding and organization, Signal Process. Mag. **22**(5), 42–60 (2005)

32.3 C. Cieri, D. Graff, M. Liberman, N. Martey, S. Strassel: The TDT-2 text and speech corpus, DARPA Broadcast News Workshop (1999) pp. 57–60

32.4 E. Leeuwis, M. Federico, M. Cettolo: Language modeling and transcription of the TED corpus lectures, Proc. IEEE-ICASSP, Vol. 1 (2003) pp. 232–235

32.5 S. Furui: Recent advances in spontaneous speech recognition and understanding, Proc. ISCA and IEEE Workshop on Spontaneous Speech Processing and Recognition, ed. by (2003) pp. 1–4

32.6 A. Park, T. Hazen, J. Glass: Automatic processing of audio lectures for information retrieval: Vocabulary selection and language modeling, In Proc. IEEE-ICASSP, Vol. 1 (2005) pp. 497–500

32.7 B. Ramabhadran, J. Huang, M. Picheny: Towards automatic transcription of large spoken archives – English ASR for the MALACH project, Proc. IEEE-ICASSP, Vol. 1 (2003) pp. 216–219

32.8 J. Fiscus, W.M. Fisher, A. Martin, M. Przybocki, D.S. Pallett: 2000 NIST evaluation of conversational speech recognition over the telephone, DARPA Speech Transcription Workshop (2000)

32.9 S.F. Chen, B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, G. Zweig: Advances in speech transcription at IBM under the DARPA EARS program, IEEE Trans. Audio, Speech Lang. Process **14**(5), 1596–1608 (2006)

32.10 S. Matsoukas, J.-L. Gauvain, G. Adda, T. Colthurst, C.-L. Kao, O. Kimball, L. Lamel, F. Lefevre, J.Z. Ma, J. Makhoul, L. Nguyen, R. Prasad, R. Schwartz, H. Schwenk, B. Xiang: Advances in transcription of broadcast news and conversational telephone speech within the combined EARS BBN/LIMSI system, IEEE Trans. Audio, Speech Lang. Process **14**(5), 1541–1556 (2006)

32.11 J.J. Godfrey, E.C. Holliman, J. McDaniel: SWITCHBOARD: Telephone speech corpus for research and development, Proc. IEEE-ICASSP, Vol. 1 (1992) pp. 517–520

32.12 H. Soltau, B. Kingsbury, L. Mangu, D. Povey, G. Saon, G. Zweig: The IBM 2004 conversational telephony system for rich transcription, Proc. IEEE-ICASSP, Vol. 1 (2005) pp. 205–208

32.13 J.S. Garofol, C.D. Laprun, J.G. Fiscus: The RT-04 spring meeting recognition evaluation, NIST Meeting Recognition Workshop (2004)

32.14 J. Fiscus, J. Ajot, J.S. Garofol: The rich transcription 2006 evaluation overview and speech-to-text results, NIST Meeting Recognition Workshop (2006)

32.15 C. Gollan, M. Bisani, S. Kanthak, R. Schluter, H. Ney: Cross domain automatic transcription on the TC-STAR EPPS corpus, Proc. IEEE-ICASSP, Vol. 1 (2005) pp. 825–828

32.16 J. Loeoef, M. Bisani, C. Gollan, G. Heigold, B. Hoffmeister, C. Plahl, R. Schlueter, H. Ney: The 2006 RWTH parliamentary speeches transcription system, Proc. TC-STAR Workshop on Speech-to-Speech Translation (2006) pp. 133–138

32.17 Y. Akita, C. Troncoso, T. Kawahara: Automatic transcription of meetings using topic-oriented language model adaptation, Proc. Western Pacific Acoustics Conference (WESPAC) (2006)

32.18 D. McKelvie, A. Isard, A. Mengel, M.B. Moller, M. Grosse, M. Klein: The MATE workbench – an annotation tool for XML coded speech corpora, Speech Commun. **33**, 97–112 (2001)

32.19 S. Bird, M. Liberman: A formal framework for linguistic annotation, Speech Commun. **33**, 23–60 (2001)

32.20 L. Lamel, J. Gauvain, G. Adda: Investigating lightly supervised acoustic model training, Proc. IEEE-ICASSP, Vol. 1 (2001) pp. 477–480

32.21 L. Nguyen, B. Xiang: Light supervision in acoustic model training, Proc. IEEE-ICASSP, Vol. 1 (2004) pp. 185–188

32.22 K. Maekawa: Corpus of Spontaneous Japanese: Its design and evaluation, Proc. ISCA and IEEE Workshop on Spontaneous Speech Processing and Recognition (2003) pp. 7–12

32.23 K. Uchimoto, C. Nobata, A. Yamada, S. Sekine, H. Isahara: Morphological analysis of corpus of spontaneous Japanese, Proc. ISCA and IEEE Workshop on Spontaneous Speech Processing and Recognition (2003) pp. 159–162

32.24 K. Maekawa, H. Kikuchi, Y. Igarashi, J. Venditti: X-JToBI: an Extended J-ToBI for Spontaneous Speech, Proc. ICSLP (2002) pp. 1545–1548

32.25 S. Furui, M. Nakamura, T. Ichiba, K. Iwano: Analysis and recognition of spontaneous speech using Corpus of Spontaneous Japanese, Speech Commun. **47**, 208–219 (2005)

32.26 E. Fosler, M. Weintraub, S. Wegmann, Y.H. Kao, S. Khudanpur, C. Galles, M. Saraclar: Automatic learning of word pronunciation from data, Proc. ICSLP (1996)

32.27 T. Shinozaki, S. Furui: Analysis on individual differences in automatic transcription of spontaneous presentations, Proc. IEEE-ICASSP, Vol. 1 (2002) pp. 729–732

32.28 L. Lee, R.C. Rose: Speaker normalization using efficient frequency warping procedures, Proc. IEEE-ICASSP (1996) pp. 353–356

32.29 S. Wegmann, D. McAllaster, J. Orloff, B. Peskin: Speaker normalization on conversational telephone speech, Proc. IEEE-ICASSP (1996) pp. 339–342

32.30 J.W. McDonough, T. Anastasakos, G. Zavaliagkos, H. Gish: Speaker-adapted training on the switchboard corpus, Proc. IEEE-ICASSP (1997) pp. 1059–1062

32.31 D. Pye, P.C. Woodland: Experiments in speaker normalisation and adaptation for large vocabulary speech recognition, Proc. IEEE-ICASSP (1997) pp. 1047–1050

32.32 M. Ostendorf, V.V. Digalakis, O.A. Kimball: From HMM's to segment models: A unified view of stochastic modeling for speech recognition,

Amazon EX1010 Page 663

IEEE Trans. Speech Audio Process. **4**(5), 360–378 (1996)
32.33 Y. Gong, J.-P. Haton: Stochastic trajectory modeling for speech recognition, Proc. IEEE-ICASSP, Vol. 1 (1994) pp. 57–60
32.34 C. Fugen, I. Rogina: Integrating dynamic speech modalities into context decision trees, Proc. IEEE-ICASSP (2000) pp. 1277–1280
32.35 G. Zweig, S. Russell: Probabilistic modeling with Bayesian networks for automatic speech recognition, Proc. ICSLP (1998) pp. 3011–3014
32.36 J.A. Bilmes: Buried Markov models for speech recognition, Proc. ICASSP, Vol. 2 (1999) pp. 713–716
32.37 J. Bilmes, C. Bartels: Graphical model architectures for speech recognition, Signal Process. Mag. **22**(5), 89–100 (2005)
32.38 J. Zheng, H. Franco, F. Weng: Word-level rate of speech modeling using rate-specific phones and pronunciations, Proc. IEEE-ICASSP (2000) pp. 1775–1778
32.39 N. Morgan, E. Fosler, N. Mirghafori: Speech recognition using on-line estimation of speaking rate, Proc. EUROSPEECH (1997) pp. 2079–2082
32.40 H. Nanjo, K. Kato, T. Kawahara: Speaking rate dependent acoustic modeling for spontaneous lecture speech recognition, Proc. EUROSPEECH (2001) pp. 2531–2534
32.41 J. Nedel, R. Stern: Duration normalization for improved recognition of spontaneous and read speech via missing feature methods, Proc. IEEE-ICASSP, Vol. 1 (2001) pp. 313–316
32.42 M. Richardson, M. Hwang, A. Acero, X.D. Huang: Improvements on speech recognition for fast talkers, Proc. EUROSPEECH (1999) pp. 411–414
32.43 S. Tsuge, T. Fukuda, K. Kita: Frame-period adaptation for speaking rate robust speech recognition, Proc. ICSLP, Vol. 3 (2000) pp. 718–721
32.44 K. Okuda, T. Kawahara, S. Nakamura: Speaking rate compensation based on likelihood criterion in acoustic model training and decoding, Proc. ICSLP (2002) pp. 2589–2592
32.45 T. Shinozaki, S. Furui: Hidden mode HMM using Bayesian network for modeling speaking rate fluctuation, Proc. IEEE Workshop Automatic Speech Recognition and Understanding (2003)
32.46 H. Nanjo, T. Kawahara: Language model and speaking rate adaptation for spontaneous presentation speech recognition, IEEE Trans. Speech and Audio Process. **12**(4), 391–400 (2004)
32.47 T. Holter, T. Svendsen: Combined optimisation of baseforms and model parameters in speech recognition based on acoustic subword units, Proc. IEEE Workshop Automatic Speech Recognition and Understanding (1997) pp. 199–206
32.48 M. Bacchiani, M. Ostendorf: Using automatically-derived acoustic sub-word units in large vocabulary speech recognition, Proc. ICSLP (1998) pp. 1843–1846
32.49 T. Sloboda, A. Waibel: Dictionary learning for spontaneous speech recognition, Proc. ICSLP, Vol. 4 (1996) pp. 2328–2331
32.50 M. Finke, A. Waibel: Speaking mode dependent pronunciation modeling in large vocabulary conversational speech recognition, Proc. EUROSPEECH (1997) pp. 2379–2382
32.51 E. Fosler-Lussier: Multi-level decision trees for static and dynamic pronunciation models, Proc. EUROSPEECH (1999) pp. 463–466
32.52 J.M. Kessens, M. Wester, H. Strik: Improving the performance of a Dutch CSR by modeling within-word and cross-word pronunciation variation, Speech Communication **29**, 193–207 (1999)
32.53 B. Peskin, M. Newman, D. McAllaster, V. Nagesha, H.B. Richards, S. Wegmann, M. Hunt, L. Gillick: Improvements in recognition of conversational telephone speech, Proc. IEEE-ICASSP (1999) pp. 53–56
32.54 H. Schramm, X. Aubert: Efficient integration of multiple pronunciations in a large vocabulary decoder, Proc. IEEE-ICASSP (2000) pp. 1659–1662
32.55 M. Riley, W. Byrne, M. Finke, S. Khudanpur, A. Ljolje, J. McDonough, H. Nock, M. Saraclar, C. Wooters, G. Zavaliagkos: Stochastic pronunciation modelling from hand-labelled phonetic corpora, Speech Commun. **29**, 209–224 (1999)
32.56 T. Fukada, T. Yoshimura, Y. Sagisaka: Automatic generation of multiple pronunciations based on neural networks, Speech Commun. **27**, 63–73 (1999)
32.57 D. Torre, L. Villarrubia, J.M. Elvira, L. Hernandez-Gomez: Automatic alternative transcription generation and vocabulary selection for flexible word recognizers, Proc. IEEE-ICASSP (1997) pp. 1463–1466
32.58 Y. Akita, T. Kawahara: Generalized statistical modeling of pronunciation variations using variable-length phone context, Proc. IEEE-ICASSP, Vol. 1 (2005) pp. 689–692
32.59 X. Zhu, R. Rosenfeld: Improving trigram language modeling with the world wide web, Proc. IEEE-ICASSP, Vol. 1 (2001) pp. 533–536
32.60 R. Sarikaya, A. Gravano, Y. Gao: Rapid language model development using external resources for new spoken dialog domains, Proc. IEEE-ICASSP, Vol. 1 (2005) pp. 573–576
32.61 T. Ng, M. Ostendorf, M.-Y. Hwang, M. Siu, I. Bulyko, X. Lei: Web-data augmented language models for Mandarin conversational speech recognition, Proc. IEEE-ICASSP, Vol. 1 (2005) pp. 589–592
32.62 T. Misu, T. Kawahara: A bootstrapping approach for developing language model of new spoken dialogue systems by selecting web texts, In Proc. INTERSPEECH (2006) pp. 9-12
32.63 Y. Akita, T. Kawahara: Efficient estimation of language model statistics of spontaneous speech via statistical transformation model, Proc. IEEE-ICASSP, Vol. 1 (2006) pp. 1049–1052

Amazon EX1010 Page 664

32.64 R. Khun, R. De Mori: A cache-based natural language model for speech recognition, IEEE Trans. Pattern Analysis and Machine Intelligence **12**(6), 570–583 (1990)

32.65 P. Clarkson, A.J. Robinson: Language model adaptation using mixtures and an exponentially decaying cache, Proc. IEEE-ICASSP (1997) pp. 799–802

32.66 R. Lau, R. Rosenfeld, S. Roukos: Trigger-based language models: A maximum entropy approach, Proc. IEEE-ICASSP, Vol. 2 (1993) pp. 45–48

32.67 C. Troncoso, T. Kawahara: Trigger-based language model adaptation for automatic meeting transcription, Proc. INTERSPEECH (2005) pp. 1297–1300

32.68 T. Yokoyama, T. Shinozaki, K. Iwano, S. Furui: Unsupervised class-based language model adaptation for spontaneous speech recognition, Proc. IEEE-ICASSP, Vol. 1 (2003) pp. 236–239

32.69 G. Moore, S. Young: Class-based language model adaptation using mixtures of weights, Proc. ICSLP, Vol. 4 (2000) pp. 512–515

32.70 L. Lussier, E.W.D. Whittaker, S. Furui: Unsupervised language model adaptation methods for spontaneous speech, Proc. ICSLP (2004) pp. 1981–1984

32.71 L. Chen, J.L. Gauvain, L. Lamel, G. Adda, M. Adda: Using information retrieval methods for language model adaptation, Proc. EUROSPEECH (2001) pp. 255–258

32.72 R. Kneser, V. Steinbiss: On the dynamic adaptation of stochastic language models, Proc. IEEE-ICASSP, Vol. 2 (1993) pp. 586–589

32.73 J.R. Bellegarda: A multispan language modeling framework for large vocabulary speech recognition, IEEE Trans. Speech and Audio Process. **6**(5), 468–475 (1998)

32.74 T. Hoffman: Probabilistic latent semantic indexing, Proc. SIG-IR (1999)

32.75 Y. Akita, T. Kawahara: Language model adaptation based on PLSA of topics and speakers, Proc. ICSLP (2004) pp. 1045–1048

32.76 A. Stolcke, E. Shriberg, R. Bates, M. Ostendorf, D. Hakkani, M. Plauche, G. Tur, Y. Lu: Automatic detection of sentence boundaries and disfluencies based on recognized words, Proc. ICSLP (1998) pp. 2247–2250

32.77 Y. Liu, E. Shriberg, A. Stolcke, B. Peskin, J. Ang, D. Hillard, M. Ostendorf, M. Tomalin, P. Woodland, M. Harper: Structural metadata research in the EARS program, Proc. IEEE-ICASSP, Vol. 5 (2005) pp. 957–960

32.78 E. Shriberg: Spontaneous speech: How people really talk and why engineers should care, Proc. INTERSPEECH (2005) pp. 1781–1784

32.79 T. Kudo, Y. Matsumoto: Chunking with support vector machines, Proc. NAACL (2001)

32.80 A.L. Berger, S.A. Della Pietra, V.J. Della Pietra: A maximum entropy approach to natural language processing, Comput. Ling. **22**(1), 39–71 (1996)

32.81 J. Huang, G. Zweig: Maximum entropy model for punctuation annotation from speech, Proc. ICSLP (2002) pp. 917–920

32.82 J. Lafferty, A. McCallum, F. Pereira: Conditional random fields: Probabilistic models for segmenting and labeling sequence data, Proc. Int. Conf. Machine Learning (2001) pp. 282–289

32.83 E. Shriberg, A. Stolcke, D. Hakkani-Tur, G. Tur: Prosody-based automatic segmentation of speech into sentences and topics, Speech Commun. **32**, 127–154 (2000)

32.84 Y. Akita, M. Saikou, H. Nanjo, T. Kawahara: Sentence boundary detection of spontaneous Japanese using statistical language model and support vector machines, Proc. INTERSPEECH (2006) pp. 1033–1036

32.85 Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, M. Harper: Enriching speech recognition with automatic detection of sentence boundaries and disfluencies, IEEE Trans. Audio, Speech Lang. Process. **14**(5), 1526–1540 (2006)

32.86 Y. Liu, A. Stolcke, E. Shriberg, M. Harper: Comparing and combining generative and posterior probability models: Some advances in sentence boundary detection in speech, Proc. EMNLP (2004) pp. 64–71

32.87 M. Tomalin, P.C. Woodland: Discriminatively trained Gaussian mixture models for sentence boundary detection, Proc. IEEE-ICASSP, Vol. 1 (2006) pp. 549–552

32.88 J.H. Kim, P.C. Woodland: The use of prosody in a combined system for punctuation generation and speech recognition, Proc. EUROSPEECH (2001) pp. 2757–2760

32.89 K. Shitaoka, K. Uchimoto, T. Kawahara, H. Isahara: Dependency structure analysis and sentence boundary detection in spontaneous Japanese, Proc. COLING (2004) pp. 1107–1113

32.90 R. Hamabe, K. Uchimoto, T. Kawahara, H. Isahara: Detection of quotations and inserted clauses and its application to dependency structure analysis in spontaneous Japanese, Proc. COLING-ACL volume Poster Sessions (2006) pp. 324–330

32.91 F.M. Quimbo, T. Kawahara, S. Doshita: Prosodic analysis of fillers and self-repair in Japanese speech, Proc. ICSLP (1998) pp. 3313–3316

32.92 C. Nakatani, J. Hirschberg: A speech first model for repair detectin and correction, Proc. ARPA Human Language Technology Workshop (1993) pp. 329–334

32.93 Y. Liu, E. Shriberg, A. Stolcke, M. Harper: Comparing HMM maximum entropy and conditional random fields for disfluency detection, Proc. INTERSPEECH (2005) pp. 3033–3036

32.94 P.A. Heeman, J.F. Allen: Speech repairs, intonational phrases and discourse markers: Modeling speakers' utterances in spoken dialog, Comput. Ling. **25**(4), 527–571 (1999)

Amazon EX1010 Page 665

32.95 M. Honal, T. Schultz: Correction of disfluencies in spontaneous speech using asnoisy-channel approach, Proc. EUROSPEECH (2003) pp. 2781–2784

32.96 R.J. Passonneau, D.J. Litman: Discourse segmentation by human and automated means, Comput. Ling. **23**(1), 103–139 (1997)

32.97 M. Haase, W. Kriechbaum, G. Mohler, G. Stenzel: Deriving document structure from prosodic cues, Proc. EUROSPEECH (2001) pp. 2157–2160

32.98 T. Kawahara, M. Hasegawa, K. Shitaoka, T. Kitade, H. Nanjo: Automatic indexing of lecture presentations using unsupervised learning of presumed discourse markers, IEEE Trans. Speech Audio Process. **12**(4), 409–419 (2004)

32.99 K. Zechner: Spoken language condensation in the 21st century, Proc. EUROSPEECH (2003) pp. 1989–1992

32.100 I. Mani, M. Maybury: *Advances in Automatic Text Summarization* (MIT Press, Cambridge 1999)

32.101 R. Valenza, T. Robinson, M. Hickey, R. Tucker: Comparing and combining generative and posterior probability models: Some advances in sentence boundary detection in speech, ESCA Workshop Accessing Information in Spoken Audio (1999) pp. 111–116

32.102 G. Murray, S. Renals, J. Carletta: Extractive summarization of meeting recordings, Proc. EUROSPEECH (2005) pp. 593–596

32.103 S. Furui, T. Kikuchi, Y. Shinnaka, C. Hori: Speech-to-text and speech-to-speech summarization of spontaneous speech, IEEE Trans. Speech Audio Process. **12**(4), 401–408 (2004)

32.104 J. Kupiec, J.O. Pedersen, F. Chen: A trainable document summarizer, Proc. SIG-IR (1995)

32.105 H. Christensen, Y. Gotoh, B. Kolluru, S. Renals: Are extractive text summarization techniques portable to broadcast news?, Proc. IEEE Workshop Automatic Speech Recognition and Understanding (2003)

32.106 Y. Gong, X. Liu: Generic text summarization using relevance measure and latent semantic analysis, Proc. SIG-IR (2001)

32.107 M. Hirohata, Y. Shinnaka, K. Iwano, S. Furui: Sentence extraction-based presentation summarization techniques and evaluation metrics, Proc. IEEE-ICASSP, Vol. 1 (2005) pp. 1065–1068

32.108 M. Sameer, J. Hirschberg: Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization, Proc. INTERSPEECH (2005) pp. 621–624

32.109 J. Alexandersson, P. Poller: Towards multilingual protocol generation for spontaneous dialogues, Proc. INLG-98 (1998)

32.110 K. Zechner, A. Waibel: Minimizing word error rate in textual summaries of spoken language, Proc. NAACL (2000)

32.111 J.S. Garofolo, E.M. Voorhees, C.G.P. Auzanne, V.M. Stanford: Spoken document retrieval: 1998 evaluation and investigation of new metrics, ESCA Workshop Accessing Information in Spoken Audio (1999) pp. 1–7

32.112 K. Koumpis, S. Renals: Transcription and summarization of voicemail speech, Proc. ICSLP, Vol. 2 (2000) pp. 688–691

32.113 C. Hori, S. Furui: Advances in speech summarization, Proc. EUROSPEECH (2001) pp. 1771–1774

32.114 T. Kikuchi, S. Furui, C. Hori: Automatic speech summarization based on sentence extraction and compaction, Proc. IEEE-ICASSP, Vol. 1 (2003) pp. 384–387

32.115 C. Hori, T. Hirao, H. Isozaki: Evaluation measures considering sentence concatenation for automatic summarization by sentence or word extraction, Proc. ACL Workshop Text Summarization Branches Out (2004) pp. 82–88

32.116 C.-Y. Lin: ROUGE: A package for automatic evaluation of summaries, Proc. ACL Workshop Text Summarization Branches Out (2004) pp. 74–81

32.117 T. Hori, C. Hori, Y. Minami: Speech summarization using weighted finite-state transducers, Proc. EUROSPEECH (2003) pp. 2817–2820

32.118 H. Nanjo, T. Kawahara: A new ASR evaluation measure and minimum Bayes-risk decoding for open-domain speech understanding, Proc. IEEE-ICASSP, Vol. 1 (2005) pp. 1053–1056

Amazon EX1010 Page 666

33. Environmental Robustness

J. Droppo, A. Acero

When a speech recognition system is deployed outside the laboratory setting, it needs to handle a variety of signal variabilities. These may be due to many factors, including additive noise, acoustic echo, and speaker accent. If the speech recognition accuracy does not degrade very much under these conditions, the system is called *robust*. Even though there are several reasons why real-world speech may differ from clean speech, in this chapter we focus on the influence of the *acoustical environment*, defined as the transformations that affect the speech signal from the time it leaves the mouth until it is in digital format.

Specifically, we discuss strategies for dealing with additive noise. Some of the techniques, like feature normalization, are general enough to provide robustness against several forms of signal degradation. Others, such as feature enhancement, provide superior noise robustness at the expense of being less general. A good system will implement several techniques to provide a strong defense against acoustical variabilities.

33.1 **Noise Robust Speech Recognition** 653
33.1.1 Standard Noise-Robust ASR Tasks .. 654
33.1.2 The Acoustic Mismatch Problem 655
33.1.3 Reducing Acoustic Mismatch 655

33.2 **Model Retraining and Adaptation** 656
33.2.1 Retraining on Corrupted Speech 656
33.2.2 Single-Utterance Retraining 657
33.2.3 Model Adaptation 657

33.3 **Feature Transformation and Normalization** 657
33.3.1 Feature Moment Normalization 658
33.3.2 Voice Activity Detection 662
33.3.3 Cepstral Time Smoothing 662
33.3.4 SPLICE – Normalization Learned from Stereo Data 663

33.4 **A Model of the Environment** 664

33.5 **Structured Model Adaptation** 667
33.5.1 Analysis of Noisy Speech Features .. 667
33.5.2 Log-Normal Parallel Model Combination 667
33.5.3 Vector Taylor-Series Model Adaptation 668
33.5.4 Comparison of VTS and Log-Normal PMC 670
33.5.5 Strategies for Highly Nonstationary Noises 670

33.6 **Structured Feature Enhancement** 671
33.6.1 Spectral Subtraction 671
33.6.2 Vector Taylor-Series Speech Enhancement 673

33.7 **Unifying Model and Feature Techniques** . 675
33.7.1 Noise Adaptive Training 676
33.7.2 Uncertainty Decoding and Missing Feature Techniques ... 676

33.8 **Conclusion** 677

References 677

33.1 Noise Robust Speech Recognition

This chapter addresses the problem of additive noise at the input to an automatic speech recognition (ASR) system. Parts H and I in this Handbook address how to build microphone arrays for superior sound capture, or how to reduce noise for perceptual audio quality. Both of these subjects are orthogonal to the current discussion.

Microphone arrays are useful in that improved audio capture should be the first line of defense against additive noise. However, despite the best efforts of the system designer, there will always be residual additive noise. In general, speech recognition systems prefer linear array algorithms, such as beam forming, to nonlinear techniques. Although nonlinear techniques can achieve better suppression and perceptual quality, the introduced distortions tend to confuse speech recognition systems.

Amazon EX1010 Page 667

Furthermore, speech enhancement algorithms designed for improved human perception do not always help ASR accuracy. Most enhancement algorithms introduce some signal distortion, and the type of distortion that can be tolerated by humans and computers can be quite different.

Additive noise is common in daily life, and can be roughly categorized as either stationary or nonstationary. Stationary noise, such as that made by a computer fan or air conditioning, has a frequency spectrum that does not change over time. In contrast, the spectrum of a nonstationary noise changes over time. Some examples of nonstationary noise are a closing door, music, and other speakers' voices. In practice, no noise is perfectly stationary. Even the noises from a computer fan, an air-conditioning system, or a car will change over a long enough time period.

33.1.1 Standard Noise-Robust ASR Tasks

When building and testing noise-robust automatic speech recognition systems, there are a rich set of standards to test against.

The most popular tasks today were generated by the European Telecommunications Standards Institute's technical committee for Speech, Transmission Planning, and Quality of Service (ETSI STQ). Their AURORA digital speech recognition (DSR) working group was formed to develop and standardize algorithms for distributed and noise-robust speech recognition. As a byproduct of their work, they released a series of standard tasks [33.1] for system evaluation. Each task consists of all the necessary components for running an experiment, including data and recipes for building acoustic and language models, and scripts for running evaluations against different testing scenarios.

The Aurora 2 task is the easiest to set up and use, and is the focus of many results in this chapter. The data was derived from the TIDigits corpus [33.2], which consists of continuous English digit strings of varying lengths, spoken into a close-talking microphone. To simulate noisy telephony environments, these clean utterances were first downsampled to 8 kHz, and then additive and convolutional noise was added. The additive noise is controlled to produce noisy signals with a range of signal-to-noise ratios (SNRs) of −5–20 dB.

The noise types include both stationary and nonstationary noises, and are broken down into three sets: set A (subway, babble, car, and exhibition), set B (restaurant, street, airport, and station), and set C (subway and street). Set C contains one noise from set A, and one from set B, and also includes extra convolutional noise. There are two sets of training data, one is clean and the other is noisy. The noisy training data contains noises similar to set A. This represents the case where the system designers are able to anticipate correctly the types of noises that the system will see in practice. Test set B, on the other hand, has four different types of noises.

The acoustic model training recipe that was originally distributed with Aurora 2 was considered to be too weak. As a result, many researchers built better acoustic models to showcase their techniques. However, this made their results incomparable. To rectify this problem, a standard *complex back-end* recipe [33.3] was proposed, which is the proper model to use when performing new experiments on this task.

The Aurora 3 task is similar in complexity to the Aurora 2 task, but covers four other European languages in real car noise scenarios. Because the data is a subset of the SpeechDat car database [33.4], the noise types between Aurora 2 and Aurora 3 are quite different. The noise types chosen for Aurora 2 can be impulsive and nonstationary, but the car noise in Aurora 3 tends to be well modeled by stationary colored noise. Whereas the noisy utterances in Aurora 2 are artificially mixed, the noisy utterances of Aurora 3 were collected in actual noisy environments. Nevertheless, techniques exhibit a strong correlation in performance between Aurora 2 and Aurora 3, indicating that digitally simulated noisy speech is adequate for system evaluation.

The Aurora 4 task was developed to showcase noise-robust speech recognition for larger-vocabulary systems. Whereas the previous Aurora tasks have 10 or 11 word vocabularies, the Aurora 4 task has a 5000-word vocabulary. In much the same way that Aurora 2 was derived from the clean TIDigits corpus, the Aurora 4 task was derived from the clean Wall Street Journal (WSJ) corpus [33.5]. Noises are digitally mixed at several signal-to-noise ratios. Because of the difficulty in setting up the larger system, the Aurora 4 task is not as commonly cited in the literature. The Aurora 4 task is relevant because some techniques that work well on Aurora 2 either become intractable or fail on larger-vocabulary tasks.

Noisex-92 [33.6] is a set of data that is also useful in evaluating noise robust speech recognition systems. It consists of two CD-ROMS of audio recordings, suitable for use in artificially mixing noise with clean speech to produce noisy speech utterances. There is a great variety of noises available with the data, including voice babble, factory noise, F16 fighter jet noise, M109 tank noise, machine gun noise, and others.

Amazon EX1010 Page 668

Table 33.1 Word accuracy for the Aurora 2 test sets using the clean acoustic model baseline

SNR (dB)	Test set A	Test set B	Test set C	Average
Clean	99.63	99.63	99.60	99.62
20	95.02	91.71	97.02	94.58
15	85.16	78.10	92.38	85.21
10	64.50	55.75	77.78	66.01
5	34.59	29.21	51.36	38.39
0	13.61	9.75	22.82	15.40
−5	5.87	4.08	11.47	7.14
Average (0–20)	58.58	52.90	68.27	58.25

Another common evaluation task for noise robust speech recognition systems is the speech in noisy environments (SPINE) evaluation [33.7]. It was created for the Department of Defense digital voice processing consortium, to support the 2000 SPINE1 evaluation. The corpus contains 9 h 22 min of audio data, collected in simulated noisy environments where users collaborate using realistic handsets and communications channels to seek and shoot targets, similar to the game Battleship.

33.1.2 The Acoustic Mismatch Problem

To understand the extent of the problem of recognizing speech in noise, it is useful to look at a concrete example. Many of the techniques in this chapter are tested on the Aurora 2 task. Table 33.1 contains typical results from the baseline Aurora 2 system. Here, an acoustic model is trained on clean, noise-free data, and tested on data with various digitally simulated noise levels. The accuracy on clean test data averages 99.62%, which may be acceptable for some applications.

As soon as any noise is present in the test data, the system rapidly degrades. Even at a mild 20 dB signal-to-noise ratio (SNR), the system produces more than 14 times as many errors compared to clean data. (The signal-to-noise ratio is defined as the ratio of signal energy to noise energy in the received signal. It is typically measured in decibels (dB), and calculated as $10\log_{10}$[Energy(signal)/Energy(noise)]. An SNR above 30 dB sounds quite noise-free. At 0 dB SNR, the signal and noise are at the same level.) As the SNR decreases further, the problem becomes more intense. Why does an ASR system perform so poorly when presented with even mildly corrupted signals? The answer is deceptively simple. Automatic speech recognition is fundamentally a pattern matching problem. And, when a system is tested on patterns that are unlike anything used to train it, errors are likely to occur. The fundamental problem is the *acoustic mismatch* between the training and testing data.

Figure 33.1 illustrates the severity of the problem. It compares the histograms for C_1 between clean speech and moderately noisy speech. (C_1, the first cepstral coefficient, is typical speech feature used by automatic speech recognition systems.) The two histograms are quite dissimilar. Obviously, a system trained under one condition will fail under the other.

33.1.3 Reducing Acoustic Mismatch

The simplest solution to the acoustic mismatch problem is to build an acoustic model that is a better match for the test data. Techniques that can be helpful in that respect are multistyle training and model adaptation. These types of algorithms are covered in Sect. 33.2.

Another common approach to solving the acoustic mismatch problem is to transform the data so that the training and testing data tend to be more similar. These techniques concentrate on either normalizing

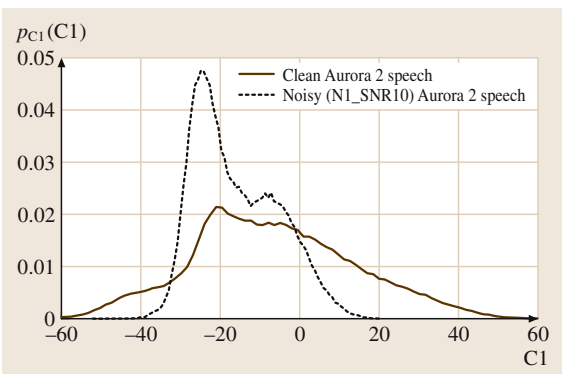


Fig. 33.1 Additive noise creates a mismatch between clean training data and noisy testing data. Here, the histogram for a clean speech feature is strikingly different from the same histogram computed from noisy speech

Amazon EX1010 Page 669

out the effects of noise, or learning transformations that map speech into a canonical noise-free representation. Examples of such techniques are cepstral mean normalization and stereo piecewise linear compensation for environment (SPLICE), which are covered in Sect. 33.3.

The techniques mentioned so far are powerful, but limited. They do not assume any form for the acoustic mismatch, so they can be applied to compensate for a wide range of corrupting influences. But, because they are unstructured, they need a lot of data to handle a new condition. Section 33.4 introduces a model of how noise corrupts clean speech features. Later, the model is used to derive powerful data-thrifty adaptation and normalization algorithms.

Section 33.5 presents the first class of these algorithms, which adapt the parameters of a clean acoustic model to approximate an acoustic model for noisy speech. Parallel model combination with a log-normal approximation is covered, as is vector Taylor-series (VTS) model adaptation.

The model for how additive noise corrupts clean speech features can also be used to do speech feature enhancement. Section 33.6 covers the classic technique of spectral subtraction as it can be integrated into speech recognition. It also covers vector Taylor-series speech enhancement, which has proven to be an easy and economical alternative to full model adaptation.

The last set of techniques discussed in this chapter are hybrid approaches that bridge the space between model and feature based techniques. In general, the former are more powerful, but the latter are easier to compute. Uncertainty decoding and noise adaptive training are two examples presented in Sect. 33.7, which are more powerful than a purely feature-based approach, but without the full cost of model adaptation.

33.2 Model Retraining and Adaptation

The best way to train any pattern recognition system is to train it with examples that are similar to those it will need to recognize later.

One of the worst design decisions to make is to train the acoustic model with data that is dissimilar to the expected testing input. This usually happens when the training data is collected in a quiet room using a close-talking microphone. This data will contain very good speech, with little reverberation or additive noise, but it will not look anything like what a deployed system will collect with its microphone. It's true that robustness algorithms can ameliorate this mismatch, but it is hard to cover up for a fundamental design flaw.

The methods presented in this section demonstrate that designing appropriate training data can greatly improve the accuracy of the final system.

33.2.1 Retraining on Corrupted Speech

To build an automatic speech recognition system that works in a particular noise condition, one of the best solutions is to find training data that matches this condition, train the acoustic model from this data in the normal way, and then decode the noisy speech without further processing. This is known as *matched condition* training.

If the test conditions are not known precisely, *multistyle training* [33.8] is a better choice. Instead of using a single noise condition in the training data, many different kinds of noises are used, each of which would be reasonable to expect in deployment.

Matched condition training can be simulated with sample noise waveforms from the new environments. These noise waveforms are artificially mixed with clean training data to create a synthetically noisy training data set. This method allows us to adapt the model to the new environment with a relatively small amount of data from the new environment, yet use a large amount of training data for the system.

The largest problem with multistyle training is that it makes components in the acoustic model broader and less discriminative. As a result, accuracy in any one condition is slightly worse than if matched condition training were available, but much better than if a mismatched training were used.

Tables 33.1 and 33.2 present the standard clean condition and multistyle training results from the Aurora 2 tasks. In the clean condition experiments, the acoustic model is built with uncorrupted data, even though the test data has additive noise. This is a good example of mismatched training conditions, and the resulting accuracy is quite low.

The multistyle training results in Table 33.2 are much better than the clean training results of Table 33.1. Even though the noises from set B are different from the training set, their accuracy has been improved considerably over the mismatched clean condition training. Also, notice that the word accuracy of the clean test data is lower

Amazon EX1010 Page 670

Table 33.2 Word accuracy for the Aurora 2 test sets using the multistyle acoustic model baseline

SNR (dB)	Test set A	Test set B	Test set C	Average
Clean	99.46	99.46	99.46	99.45
20	98.99	98.59	98.78	98.79
15	98.41	97.56	98.12	98.03
10	96.94	94.94	96.05	95.98
5	91.90	88.38	87.92	89.40
0	70.53	69.36	59.35	66.42
−5	30.26	33.05	25.16	29.49
Average (0–20)	91.36	89.77	88.04	90.06

for the multistyle-trained models. This is typical of multistyle training: whereas before, the clean test data was matched to the clean training data, now every type of test data has a slight mismatch.

33.2.2 Single-Utterance Retraining

Taken to the extreme, the retraining approach outlined above could be used to generate a new acoustic model for every noisy utterance encountered.

The first step would be to extract exemplar noise signals from the current noisy utterance. This is then used to artificially corrupt a clean training corpus. Finally, an utterance specific acoustic model is trained on this corrupted data. Such a model should be a good match to the current utterance, and we would expect excellent recognition performance.

Of course, this approach would only be feasible for small systems where the training data can be kept in memory and where the retraining time is small. It would certainly not be feasible for large-vocabulary speaker-independent systems.

The idea of using an utterance-specific acoustic model can be made more efficient by replacing the recognizer retraining step with a structured adaptation of the acoustic model. Each Gaussian component in the acoustic model is adapted to account for how its parameters would change in the presence of noise. This idea is the basis of the techniques such as parallel model combination (PMC) and VTS model adaptation in Sect. 33.5, where a model for noise is composed with the acoustic model for speech, to build a noisy speech model. Although they are less accurate than the brute-force method just described, they are computationally simpler.

33.2.3 Model Adaptation

In the same way that retraining the acoustic model for each utterance can provide good noise robustness, standard unsupervised adaptation techniques can be used to approximate this effect.

Speaker adaptation algorithms, such as maximum a priori (MAP) or maximum likelihood linear regression (MLLR), are good candidates for robustness adaptation. Since MAP is an unstructured method, it can offer results similar to those of matched conditions, but it requires a significant amount of adaptation data. MLLR can achieve reasonable performance with about a minute of speech for minor mismatches [33.9]. For severe mismatches, MLLR also requires a large number of transformations, which, in turn, require a larger amount of adaptation data.

In [33.10], it was shown how MLLR works on Aurora 2. That paper reports a 9.2% relative error rate reduction over the multistyle baseline, and a 25% relative error rate reduction over the clean condition baseline.

33.3 Feature Transformation and Normalization

This section demonstrates how simple *feature normalization* techniques can be used to reduce the acoustic mismatch problem. Feature normalization works by reducing the mismatch between the training and the testing data, leading to greater robustness and higher recognition accuracies.

It has been demonstrated that feature normalization alone can provide many of the benefits of noise robustness specific algorithms. In fact, some of the best results on the Aurora 2 task have been achieved with feature normalization alone [33.11]. Because these techniques are easy to implement and provide impressive results,

they should be included in every noise-robust speech recognition system.

This section covers the most common feature normalization techniques, including voice activity detection, automatic gain normalization, cepstral mean and variance normalization, cepstral histogram normalization, and cepstral filtering.

33.3.1 Feature Moment Normalization

The goal of feature normalization is to apply a transformation to the incoming observation features. This transformation should eliminate variabilities unrelated to the transcription, while reducing the mismatches between the training and the testing utterances. Even if you do not know how the ASR features have been corrupted, it is possible to normalize them to reduce the effects of the corruption.

With moment normalization, a one-to-one transformation is applied to the data, so that its statistical moments are normalized. Techniques using this approach include cepstral mean normalization, cepstral mean and variance normalization, and cepstral histogram normalization. Respectively, they try to normalize the first, the first two, and all the moments of the data. The more moments that are normalized, the more data is needed to prevent loss of relevant acoustic information.

Another type of normalization affects only the energy-like features of each frame. Automatic gain normalization (AGN) is used to ensure that the speech occurs at the same absolute signal level, regardless of the incoming level of background noise or SNR. The simplest of these AGN schemes subtracts the maximum $C0$ value from every frame for each utterance. With this method, the most energetic frame (which is likely to contain speech) gets a $C0$ value of zero, while every other frame gets a negative $C0$.

When using moment normalization, it is sometimes beneficial to use AGN on the energy-like features, and the more-general moment normalization on the rest. For each of the moment normalization techniques discussed below, the option of treating $C0$ separately with AGN is evaluated.

Cepstral Mean Normalization

Cepstral mean normalization is the simplest feature normalization technique to implement, and should be considered first. It provides many of the benefits available in the more-advanced normalization algorithms.

For our analysis, the received speech signal $x[m]$ is used to calculate a sequence of cepstral vectors $\{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T-1}\}$. In its basic form, cepstral mean normalization (CMN) (*Atal* [33.12]) consists of subtracting the mean feature vector $\boldsymbol{\mu}_{\boldsymbol{x}}$ from each vector $\boldsymbol{x}_t$ to obtain the normalized vector $\hat{\boldsymbol{x}}_t$:

$$\boldsymbol{\mu}_{\boldsymbol{x}} = \frac{1}{T}\sum_t \boldsymbol{x}_t \,, \tag{33.1}$$

$$\hat{\boldsymbol{x}}_t = \boldsymbol{x}_t - \boldsymbol{\mu}_{\boldsymbol{x}} \,. \tag{33.2}$$

As a result, the long-term average of any observation sequence (the first moment) is zero.

It is easy to show that CMN makes the features robust to some linear filtering of the acoustic signal, which might be caused by microphones with different transfer functions, varying distance from user to microphone, the room acoustics, or transmission channels.

To see this, consider a signal $y[m]$, which is the output of passing $x[m]$ through a filter $h[m]$. If the filter $h[m]$ is much shorter than the analysis window used to compute the cepstra, the new cepstral sequence $\{\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{T-1}\}$ will be equal to

$$\boldsymbol{y}_t = \boldsymbol{x}_t + \boldsymbol{h} \,. \tag{33.3}$$

Here, the cepstrum of the filter $\boldsymbol{h}$ is defined as the discrete cosine transform (DCT) of the log power spectrum of the filter coefficients $h[m]$:

$$\boldsymbol{h} = \mathbf{C}\big(\ln|H(\omega_0)|^2 \ldots \ln|H(\omega_M)|^2\big) \,. \tag{33.4}$$

Since the DCT is a linear operation, it is represented here as multiplication by the matrix $\mathbf{C}$.

The sample mean of the filtered cepstra is also offset by $\boldsymbol{h}$, a factor which disappears after mean normalization:

$$\boldsymbol{\mu}_{\boldsymbol{y}} = \frac{1}{T}\sum_{t=1}^{T-1} \boldsymbol{y}_t = \boldsymbol{\mu}_{\boldsymbol{x}} + \boldsymbol{h} \,, \tag{33.5}$$

$$\hat{\boldsymbol{y}}_t = \boldsymbol{y}_t - \boldsymbol{\mu}_{\boldsymbol{y}} = \hat{\boldsymbol{x}}_t \,. \tag{33.6}$$

As long as these convolutional distortions have a time constant that is short with respect to the front end's analysis window length, and does not suppress large regions of the spectrum below the noise floor (e.g., a severe low-pass filter), CMN can virtually eliminate their effects. As the filter length $h[m]$ grows, (33.3) becomes less accurate and CMN is less effective in removing the convolutional distortion. In practice, CMN can normalize the effect of different telephone channels and microphones, but fails with reverberation times that start to approach the analysis window length [33.13].

Tables 33.3 and 33.4 show the effectiveness of whole-utterance CMN on the Aurora 2 task. In the

Amazon EX1010 Page 672

Table 33.3 Word accuracy for Aurora 2, using cepstral mean normalization and an acoustic model trained on clean data. CMN reduces the error rate by 31% relative to the baseline in Table 33.1

Energy normalization	Normalization level	Set A	Set B	Set C	Average
CMN	Static	68.55	73.51	69.48	70.72
AGN	Static	69.46	69.84	74.15	70.55
AGN	Full	70.34	70.74	74.88	71.41
CMN	Full	68.65	73.71	69.69	70.88

Table 33.4 Word accuracy for Aurora 2, using cepstral mean normalization and an acoustic model trained on multistyle data. CMN reduces the error rate by 30% relative to the baseline in Table 33.2

Energy normalization	Normalization level	Set A	Set B	Set C	Average
CMN	Static	92.93	92.73	93.49	92.96
AGN	Static	93.15	92.61	93.52	93.01
AGN	Full	93.11	92.63	93.56	93.01
CMN	Full	92.97	92.62	93.32	92.90

best case, CMN reduces the error rate by 31% relative using a clean acoustic model, and 30% relative using a multistyle acoustic model.

Both tables compare applying CMN on the energy feature to using AGN. In most cases, using AGN is better than applying CMN on the energy term. The failure of CMN on the energy feature is most likely due to the randomness it induces on the energy of noisy speech frames. AGN tends to put noisy speech at the same level regardless of SNR, which helps the recognizer make sharp models. On the other hand, CMN will make the energy term smaller in low-SNR utterances and larger in high-SNR utterances, leading to less-effective speech models.

There are also two different stages in which CMN can be applied. One option is to use CMN on the static cepstra, before computing the dynamic cepstra. Because of the nature of CMN, this is equivalent to leaving the dynamic cepstra untouched. The other option is to use CMN on the full feature vector, after dynamic cepstra have been computed from the unnormalized static cepstra. Tables 33.3 and 33.4 both show that it is slightly better to apply the normalization to the full feature vectors.

Cepstral Variance Normalization

Cepstral variance normalization (CVN) is similar to CMN, and the two are often paired as cepstral mean and variance normalization (CMVN). CMVN uses both the sample mean and standard deviation to normalize the cepstral sequence:

$$\sigma_x{}^2 = \frac{1}{T}\sum_{t=0}^{T-1} x_t^2 - \mu_x^2 \,, \tag{33.7}$$

$$x_t = \frac{x_t - \mu_x}{\sigma_x} \,. \tag{33.8}$$

After normalization, the mean of the cepstral sequence is zero, and it has a variance of one.

Unlike CMN, CVN is not associated with addressing a particular type of distortion. It can, however, be shown empirically that it provides robustness against acoustic channels, speaker variability, and additive noise.

Tables 33.5 and 33.6 show how CMVN affects accuracy on the Aurora 2 task. Adding variance normalization to CMN reduces the error rate 8.7% relative using a clean acoustic model, and by 8.6% relative when using a multistyle acoustic model.

As with CMN, CMVN is best applied to the full feature vector, after the dynamic cepstra have been computed. Unlike CMN, the tables show that applying CMVN to the energy term is often better than using whole-utterance AGN. Because CMVN is both shifting and scaling the energy term, both the noisy speech and the noise are placed at a consistent absolute levels.

Cepstral Histogram Normalization

Cepstral histogram normalization (CHN) [33.14] takes the core ideas behind CMN and CVN, and extends them to their logical conclusion. Instead of only normalizing the first or second central moments, CHN modifies the signal such that all of its moments are normalized. As with CMN and CHN, a one-to-one transformation is independently applied to each dimension of the feature vector.

The first step in CHN is choosing a desired distribution for the data, $p_x(x)$. It is common to choose

Amazon EX1010 Page 673

Table 33.5 Word accuracy for Aurora 2, using cepstral mean and variance normalization and an acoustic model trained on clean data. CMVN reduces the error rate by 8.7% relative to the CMN results in Table 33.3. CMVN is much better than AGN at energy normalization, probably because it provides consistent absolute levels for both speech and noise, whereas AGN only normalizes the speech

Energy normalization	Normalization level	Set A	Set B	Set C	Average
AGN	Static	72.96	72.4	76.48	73.44
CMVN	Static	79.34	79.86	80.8	79.84
CMVN	Full	84.46	85.55	84.84	84.97
AGN	Full	72.77	72.23	77.02	73.40

Table 33.6 Word accuracy for Aurora 2, using cepstral mean and variance normalization and an acoustic model trained on multistyle data. CVMN reduces the error rate by 8.6% relative to the baseline in Table 33.4. The difference between CMVN and AGN for energy normalization is less pronounced than in Table 33.5

Energy normalization	Normalization level	Set A	Set B	Set C	Average
AGN	Static	93.34	92.79	93.62	93.18
CMVN	Static	93.33	92.57	93.24	93.01
CMVN	Full	93.80	93.09	93.70	93.50
AGN	Full	93.37	92.76	93.70	93.19

a Gaussian distribution with zero mean and unit covariance. Let $p_y(y)$ represent the actual distribution of the data to be transformed.

It can be shown that the following function $f(\cdot)$ applied to y produces features with the probability distribution function (PDF) $p_x(x)$:

$$f(y) = F_x^{-1}[F_y(y)] \,. \tag{33.9}$$

Here, $F_y(y)$ is the cumulative distribution function (CDF) of the test data. Applying $F_y(\cdot)$ to y transforms the data distribution from $p_y(y)$ to a uniform distribution. Subsequent application of $F_x^{-1}(\cdot)$ imposes a final distribution of $p_x(x)$. When the target distribution is chosen to be Gaussian as described above, the final sequence has zero mean and unit covariance, just as if CMVN were used. Additionally, every other moment would match the target Gaussian distribution.

Whole-utterance Gaussianization is easy to implement by applying (33.9) independently to each feature dimension.

First, the data is transformed using (33.10) so it has a uniform distribution. The summation counts how many frames have the i-th dimension of $\boldsymbol{y}$ less than the value in frame m, and divides by the number of frames. The resulting sequence of $y_i'[m]$ has a uniform distribution between zero and one:

$$y_i'[m] = \frac{1}{M} \sum_{m'=1}^{M} 1\big(y_i[m'] < y_i[m]\big) \,. \tag{33.10}$$

The second and final step consists of transforming $y_i'[m]$ so that it has a Gaussian distribution. This can be accomplished, as in (33.11), using an inverse Gaussian CDF G_x^{-1}:

$$y_i^{\text{CHN}}[m] = G_x^{-1}\big(y_i'[m]\big) \,. \tag{33.11}$$

Tables 33.7 and 33.8 show the results of applying CHN to the Aurora 2 task. As with CMVN, it is better to apply the normalizing transform to a full feature vector, and to avoid the use of a separate AGN step. In the end, the results are not significantly better than CMVN.

Analysis of Feature Normalization

When implementing feature normalization, it is very important to use enough data to support the chosen technique. In general, with stronger normalization algorithms, it is necessary to process longer segments of speech.

As an example, let us analyze the effect of CMN on a short utterance. Consider an utterance contains a single phoneme, such as the fricative /s/. The mean $\boldsymbol{\mu_x}$ will be very similar to the frames in this phoneme, since /s/ is quite stationary. Thus, after normalization, $\boldsymbol{\mu_x} \approx 0$. A similar result will happen for other fricatives, which means that it would be impossible to distinguish these ultrashort utterances, and the error rate will be very high. If the utterance contains more than one phoneme but is still short, the problem is not insurmountable, but the confusion among phonemes is still higher than if no CMN had been applied.

Amazon EX1010 Page 674

Table 33.7 Word accuracy for Aurora 2, using cepstral histogram normalization and an acoustic model trained on clean data. As with CMVN, CHN is much better than AGN at energy normalization. The CHN results on Aurora 2 are similar to the CMVN results presented in Table 33.5

Energy normalization	Normalization level	Set A	Set B	Set C	Average
AGN	Static	70.34	71.14	73.76	71.34
CHN	Static	82.49	84.06	83.66	83.35
CHN	Full	83.64	85.03	84.59	84.39
AGN	Full	69.75	70.23	74.25	70.84

Table 33.8 Word accuracy for Aurora 2, using cepstral histogram normalization and an acoustic model trained on multistyle data. The CHN results on Aurora 2 are similar to the CMVN results presented in Table 33.8

Energy normalization	Normalization level	Set A	Set B	Set C	Average
AGN	Static	93.19	92.63	93.45	93.02
CHN	Static	93.17	92.69	93.04	92.95
CHN	Full	93.61	93.21	93.49	93.43
AGN	Full	93.29	92.73	93.74	93.16

If test utterances are too short to support the chosen normalization technique, degradation will be most apparent in the clean-speech recognition results. CMVN and CHN, in particular, can significantly degrade the accuracy of the clean speech tests in Aurora 2. In cases where there is not enough data to support CMN, *Rahim* has shown [33.15] that using the recognizer's acoustic model to estimate a maximum-likelihood mean normalization is superior to conventional CMN.

Empirically, it has been found that CMN does not degrade the recognition rate on utterances from the same acoustical environment, as long as there are at least four seconds of speech frames available. CMVN and CHN require even longer segments of speech.

As we have seen, CMN can provide robustness against additive noise. It is also effective in normalizing acoustic channels. For telephone recordings, where each call has a different frequency response, the use of CMN has been shown to provide as much as 30% relative decrease in error rate. When a system is trained on one microphone and tested on another, CMN can provide significant robustness.

Interestingly, it has been found in practice that the error rate for utterances within the same environment can actually be somewhat lower. This is surprising, given that there is no mismatch in channel conditions. One explanation is that, even for the same microphone and room acoustics, the distance between the mouth and the microphone varies for different speakers, which causes slightly different transfer functions. In addition, the cepstral mean characterizes not only the channel transfer function, but also the average frequency response of different speakers. By removing the long-term speaker average, CMN can act as sort of speaker normalization.

One drawback of CMN, CMVN, and CHN is that they do not discriminate between nonspeech and speech frames in computing the utterance mean. As a result, the normalization can be affected by the ratio of speech to nonspeech frames. For instance, the mean cepstrum of an utterance that has 90% nonspeech frames will be significantly different from one that contains only 10% nonspeech frames. As a result, the speech frames will be transformed inconsistently, leading to poorer acoustic models and decreased recognition accuracy.

An extension to CMN that addresses this problem consists in computing different means for noise and speech [33.16]:

$$\boldsymbol{h}^{j+1} = \frac{1}{N_\mathrm{s}} \sum_{t \in q_\mathrm{s}} \boldsymbol{x}_t - \boldsymbol{m}_\mathrm{s} \,, \tag{33.12}$$

$$\boldsymbol{n}^{j+1} = \frac{1}{N_\mathrm{n}} \sum_{t \in q_\mathrm{n}} \boldsymbol{x}_t - \boldsymbol{m}_\mathrm{n} \,, \tag{33.13}$$

i. e., the difference between the average vector for speech frames in the utterance and the average vector $\boldsymbol{m}_\mathrm{s}$ for speech frames in the training data, and similarly for the noise frames $\boldsymbol{m}_\mathrm{n}$. Speech/noise discrimination could be done by classifying frames into speech frames and noise frames, computing the average cepstra for each, and subtracting them from the average in the training data. This procedure works well as long as the speech/noise classification is accurate. This is best done by the recognizer,

Amazon EX1010 Page 675

since other speech detection algorithms can fail in high background noise.

33.3.2 Voice Activity Detection

A voice activity detection (VAD) algorithm can be used to ensure that only a small, consistent percentage of the frames sent to the speech recognizer are nonspeech frames. These algorithms work by finding and eliminating any contiguous segments of nonspeech audio in the input.

VAD is an essential component in any complete noise-robust speech recognition system. It helps to normalize the percentage of nonspeech frames in the utterance, which, as discussed above, helps CMN perform better. It also directly reduces the number of extra words, or *insertion errors* produced by the recognition system. Because these nonspeech frames are never sent to the speech recognizer, they can not be mistaken for speech. As a side-effect, because the decoder does not need to process the eliminated frames, the overall recognition process is more efficient.

Most standard databases, such as Aurora 2, have been presegmented to include only a short pause before and after each utterance. As a result, the benefits of using VAD are not apparent on that data. Other tasks, such as Aurora 3, include longer segments of noise before and after the speech, and need a good VAD for optimal performance. For example, [33.17] shows how a VAD can significantly improve performance on the Aurora 3 task.

When designing a VAD, it is important to notice that the cost of making an error is not symmetric. If a nonspeech frame is mistakenly labeled as speech, the recognizer can still produce a good result because the silence hidden Markov model (HMM) may take care of it. On the other hand, if some speech frames are lost, the recognizer cannot recover from this error.

Some VAD use a single feature, such as energy, together with a threshold. More-sophisticated systems use log-spectra or cepstra, and make decisions based on Gaussian mixture models (GMMs) or neural networks. They can leverage acoustic features that the recognizer may not, such as pitch, zero crossing rate, and duration. As a result, a VAD can do a better job than the general recognition system at rejecting nonspeech segments of the signal.

Another common way to reduce the number of insertion errors is to tune the balance of insertion and deletion errors with the recognizer's *insertion penalty* parameter. In a speech recognition system, the insertion penalty is a fixed cost incurred for each recognized word. For a given set of acoustic observations, increasing this penalty causes fewer words to be recognized. In practice, the number of insertion errors can be reduced significantly while only introducing a moderate number of deletion errors.

33.3.3 Cepstral Time Smoothing

CMN, as originally formulated, requires a complete utterance to compute the cepstral mean; thus, it cannot be used in a real-time system, and an approximation needs to be used. In this section we discuss a modified version of CMN that can address this problem, as well as a set of cepstral filtering techniques that attempt to do the same thing.

Because CMN removes any constant bias from the cepstral time series, it is equivalent to a high-pass filter with a cutoff frequency arbitrarily close to zero. This insight suggests that other types of high-pass filters may also be used. One that has been found to work well in practice is the exponential filter, so the cepstral mean $\boldsymbol{\mu}_{\boldsymbol{x}}[m]$ is a function of time:

$$\boldsymbol{\mu}_{\boldsymbol{x}}[m] = \alpha \boldsymbol{x}[m] + (1-\alpha)\boldsymbol{\mu}_{\boldsymbol{x}}[m-1]\,, \qquad (33.14)$$

where α is chosen so that the filter has a time constant of at least 5 s of speech. For example, when the analysis frame rate is 100 frames per second, an α of 0.999 creates a filter with a time constant of almost 7 s.

This idea of using a filter to normalizing a sequence of cepstral coefficients is quite powerful, and can be extended to provide even better results.

In addition to a high-pass CMN-like filter, it is also beneficial to add a low-pass component to the cepstral filter. This is because the rate of change of the speech spectrum over time is limited by human physiology, but the interfering noise components are not. Abrupt spectral changes are likely to contain more noise than speech. As a result, disallowing the cepstra from changing too quickly increases their effective SNR. This is the central idea behind relative spectral (RASTA) and autoregressive moving average (ARMA) filtering.

The relative spectral processing or RASTA [33.18] combines both high- and low-pass cepstral filtering into a single noncausal infinite impulse response (IIR) transfer function:

$$H(z) = 0.1(z^4)\frac{2 + z^{-1} - z^{-3} - 2z^{-4}}{1 - 0.98z^{-1}}\,. \qquad (33.15)$$

As in CMN, the high-pass portion of the filter is expected to alleviate the effect of convolutional noise introduced in the channel. The low-pass filtering helps to

Amazon EX1010 Page 676

smooth some of the fast frame-to-frame spectral changes present. Empirically, it has been shown that the RASTA filter behaves similarly to the real-time implementation of CMN, albeit with a slightly higher error rate.

ARMA filtering is similar to RASTA, in that a linear time invariant (LTI) filter is applied separately to each cepstral coefficient. The following equation:

$$H(z) = (z^2)\frac{1+z^{-1}+z^{-2}}{5-z^{-1}-z^{-2}} \tag{33.16}$$

is an example of a second-order ARMA filter. Unlike RASTA, ARMA is purely a low-pass filter. As a result, ARMA should be used in conjunction with an additional explicit CMN operation.

It was shown in [33.11] how this simple technique can do better than much more-complex robustness schemes. In spite of its simplicity, those results were the best of their time.

33.3.4 SPLICE – Normalization Learned from Stereo Data

The SPLICE technique was first proposed [33.19] as a brute-force solution to the acoustic mismatch problem. Instead of blindly transforming the data as CMN, CHN, or RASTA, SPLICE learns the joint probability distribution for noisy speech and clean speech, and uses it to map each received cepstra into a clean estimate. Like CHN, SPLICE is a nonlinear transformation technique. However, whereas CHN implicitly assumes the features are uncorrelated, SPLICE learns and uses the correlations naturally present in speech features.

The SPLICE transform is built from a model of the joint distribution of noisy cepstra $\boldsymbol{y}$ and clean cepstra $\boldsymbol{x}$. The model is a Gaussian mixture model containing K mixture components:

$$p(\boldsymbol{y},\boldsymbol{x}) = \sum_{k=1}^{K} p(\boldsymbol{x}|\boldsymbol{y},k)p(\boldsymbol{y},k)\,. \tag{33.17}$$

The distribution $p(\boldsymbol{y},k)$ is itself a Gaussian mixture model on $\boldsymbol{y}$, which takes the form

$$p(\boldsymbol{y},k) = p(\boldsymbol{y}|k)p(k) = N(\boldsymbol{y};\mu_k,\sigma_k)p(k)\,. \tag{33.18}$$

The conditional distribution $p(\boldsymbol{x}|\boldsymbol{y},k)$ predicts the clean feature value given a noisy observation and a Gaussian component index k:

$$p(\boldsymbol{x}|\boldsymbol{y},k) = N(\boldsymbol{x};\,A_k\boldsymbol{y}+b_k,\,\Gamma_k)\,. \tag{33.19}$$

Due to their effect on predicting $\boldsymbol{x}$ from $\boldsymbol{y}$, the matrix A_k is referred to as the *rotation matrix*, and the vector b_k is called the *offset vector*. The matrix Γ_k represents the error incurred in the prediction.

Even though the relationship between $\boldsymbol{x}$ and $\boldsymbol{y}$ is nonlinear, this conditional linear prediction is sufficient. Because the GMM $p(\boldsymbol{y},k)$ effectively partitions the noisy acoustic space into K regions, each $p(\boldsymbol{x}|\boldsymbol{y},k)$ only needs to be accurate in one of these regions.

The SPLICE transform is derived from the joint distribution $p(\boldsymbol{x},\boldsymbol{y},k)$ by finding the expected value of the clean speech $\boldsymbol{x}$, given the current noisy observation $\boldsymbol{y}$. This approach finds the minimum mean-squared error (MMSE) estimate for x under the model, an approach pioneered by the codeword-dependent cepstral normalization (CDCN) [33.20] and multivariate-gaussian-based cepstral normalization (RATZ) [33.21] algorithms:

$$\begin{aligned}\hat{\boldsymbol{x}}_{\text{MMSE}} &= E\{\boldsymbol{x}|\boldsymbol{y}\} = \sum_{k=1}^{K} E\{\boldsymbol{x}|\boldsymbol{y},k\}p(k|\boldsymbol{y}) \\ &= \sum_{k=1}^{K}(A_k\boldsymbol{y}+b_k)p(k|\boldsymbol{y})\,,\end{aligned} \tag{33.20}$$

where the posterior probability $p(k|\boldsymbol{y})$ is given by

$$p(k|\boldsymbol{y}) = \frac{p(\boldsymbol{y},k)}{\sum_{k'=1}^{K} p(\boldsymbol{y},k')}\,. \tag{33.21}$$

Another option is to find an approximate maximum likelihood estimate for $\boldsymbol{x}$ under the model, which is similar to the fixed codeword-dependent cepstral normalization algorithm algorithm [33.20]. A good approximate solution to

$$\hat{\boldsymbol{x}}_{\text{ML}} = \max_x p(\boldsymbol{x}|\boldsymbol{y}) \tag{33.22}$$

is

$$\hat{\boldsymbol{x}}_{\text{ML}} \approx A_{\hat{k}}\boldsymbol{y}+b_{\hat{k}}\,, \tag{33.23}$$

where

$$\hat{k} = \arg\max_k p(\boldsymbol{y},k)\,. \tag{33.24}$$

Whereas the approximate maximum-likelihood (ML) estimate above involves a single affine transformation, the MMSE solution requires K transformations. Even though the MMSE estimate produces more-accurate recognition results, the approximate ML estimate may be substituted when the additional computational cost is prohibitive.

Another popular method for reducing the additional computational cost is to replace the learned rotation matrix A_k with the identity matrix I. This produces systems that are more efficient, with only a modest degradation in performance [33.19].

Amazon EX1010 Page 677

A number of different algorithms [33.20, 21] have been proposed that vary in how the parameters μ_k, Σ_k, A_k, b_k, and Γ_k are estimated.

If stereo recordings are available from both the clean signal and the noisy signal, then we can estimate μ_k and Σ_k by fitting a mixture Gaussian model to $\boldsymbol{y}$ using standard maximum-likelihood training techniques. Then A_k, b_k, and Γ_k can be estimated directly by linear regression of $\boldsymbol{x}$ and $\boldsymbol{y}$. The FCDCN algorithm [33.20, 22] is a variant of this approach when it is assumed that $\Sigma_k = \sigma^2 I$, $\Gamma_k = \gamma^2 I$, and $A_k = I$, so that μ_k and b_k are estimated through a vector quantization (VQ) procedure and b_k is the average difference $(\boldsymbol{y} - \boldsymbol{x})$ for vectors $\boldsymbol{y}$ that belong to mixture component k.

Often, stereo recordings are not available and we need other means of estimating the parameters μ_k, Σ_k, A_k, b_k, and Γ_k. CDCN [33.22] and VTS enhancement [33.21] are examples of algorithms that use a model of the environment (Sect. 33.4). This model defines a nonlinear relationship between $\boldsymbol{x}$, $\boldsymbol{y}$ and the environmental parameters $\boldsymbol{n}$ for the noise. The CDCN method also uses an MMSE approach where the correction vector is a weighted average of the correction vectors for all classes. Other methods that do not require stereo recordings or a model of the environment are presented in [33.21].

Recent improvements in training the transformation parameters have been proposed using discriminative training [33.23–25]. In this case, the noisy-speech GMM is developed from the noisy training data, or from a larger acoustic model developed from that data. The correction parameters are then initialized to zero, which corresponds to the identity transformation. Subsequently, they are trained to maximize a discriminative criterion, such as minimum classification error (MCE) [33.24], maximum mutual information (MMI) [33.23], or minimum phone error (MPE) [33.25]. It has been shown that this style of training produces superior results to the two-channel MMSE approach, and can even increase accuracy in noise-free test cases. The main disadvantage of this approach is that it is easy to overtrain the transformation, which can actually reduce robustness of the system to noise types that do not occur in the training set. This can be easily avoided by the customary use of regularization and separate training, development, and test data.

Although the SPLICE transform is discussed here, there are many alternatives available. The function to map from $\boldsymbol{y}$ to $\boldsymbol{x}$ can be approximated with a neural network [33.26], or as a mixture of Gaussians as in probabilistic optimum filtering (POF) [33.27], FCDCN [33.28], RATZ [33.21], and stochastic vector mapping (SVM) [33.24].

33.4 A Model of the Environment

Sections 33.2 and 33.3 described techniques that address the problem of additive noise by blindly reducing the acoustic mismatch between the acoustic model and the expected test data. These techniques are powerful and general, and are often used to solve problems other than additive noise. However, the more-effective solutions generally require more data to operate properly.

In this section, we use knowledge of the nature of the degradation to derive the relationship between the clean and observed signals in the power-spectrum, log-filterbank, and cepstral domains. Later, Sects. 33.5 and 33.6 show how several related methods leverage this model to produce effective noise robust speech recognition techniques.

In the acoustic environment, the clean speech signal coexists with many other sound sources. They mix linearly in the air, and a mixture of all these signals is picked up by the microphone. For our purposes, the clean speech signal that would have existed in the absence of noise is denoted by the symbol $x[m]$, and all of the other noises that are picked up by the microphone are represented by the symbol $n[m]$. Because of the linear mixing, the observed signal $y[m]$ is simply the sum of the clean speech and noise:

$$y[m] = x[m] + n[m] \,. \tag{33.25}$$

Unfortunately, the additive relationship of (33.25) is destroyed by the nonlinear process of extracting cepstra from $y[m]$. Figure 33.2 demonstrates the path that the noisy signal takes on its way to becoming a cepstral feature vector observation vector. It consists of dividing the received signal into frames, performing a frequency analysis and warping, applying a logarithmic compression, and finally a decorrelation and dimensionality reduction.

The first stage of feature extraction is framing. The signal is split into overlapping segments of about 25 ms each. These segments are short enough that, within each frame of data, the speech signal is approximately stationary. Each frame is passed through a discrete Fourier transform (DFT), where the time-domain signal becomes a complex-valued function of discrete fre-

Amazon EX1010 Page 678

quency k. Concentrating our analysis on a single frame of speech $Y[k]$, clean speech and noise are still additive:

$$Y[k] = X[k] + N[k] \,. \tag{33.26}$$

The next processing step is to turn the observed complex spectra into real-valued power spectra, through the application of a magnitude-squared operation. The power spectrum of the observed signal is

$$|Y[k]|^2 = |X[k]|^2 + |N[k]|^2 + 2|X[k]||N[k]| \cos\theta \,, \tag{33.27}$$

a function of the power spectrum of the clean speech and of the noise, as well as a *cross-term*. This cross-term is a function of the clean speech and noise magnitudes, as well as their relative phase θ. When the clean speech and noise are uncorrelated, the expected value of the cross-term is zero:

$$E\{X[k]N^*[k]\} = 0 \,. \tag{33.28}$$

However, for a particular frame it can have a considerable magnitude.

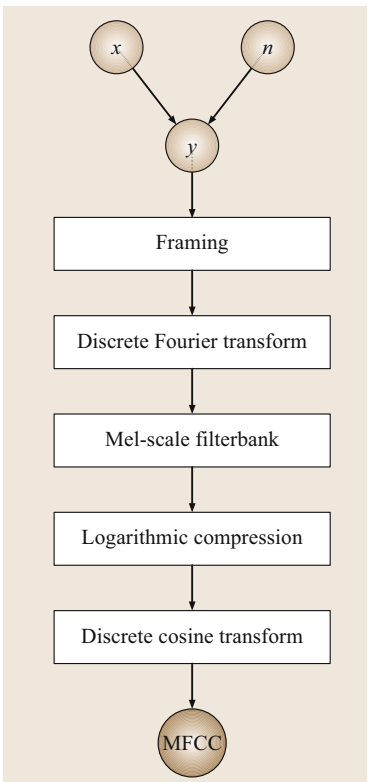


Fig. 33.2 Clean speech x and environmental noise n mix to produce the noisy signal y, which is turned into mel-frequency cepstral coefficients (MFCC) through a sequence of processing steps

It is uncommon to pass the Fourier spectra directly to the speech recognition system. Instead, it is standard to apply a mel-frequency filterbank and a logarithmic compression to create log mel-frequency filterbank (LMFB) features. The mel-frequency filterbank implements dimensionality reduction and frequency warping as a linear projection of the power spectrum. The relationship between the i-th LMFB coefficient y_i and the observed noisy spectra $Y[k]$ is given by

$$y_i = \ln\left(\sum_k w_k^i |Y[k]|^2\right) \,, \tag{33.29}$$

where the scalar w_k^i is the k-th coefficient of the i-th filter in the filterbank.

Figure 33.3 reveals a typical structure of the matrix $\boldsymbol{W}$ containing the scalars w_k^i. It compresses 128 FFT bins into 23 mel-frequency spectral features with a resolution that varies over frequency. At low frequencies, high resolution is preserved by using only a small number of FFT bins for each mel-frequency feature. As the Fourier frequency increases, more FFT bins are used.

Using (33.29) and (33.27), we can deduce the relationship among the LMFB for clean speech, noise, and noisy observation. The noisy LMFB energies are a function of the two unobserved LMFB, and a cross-term that depends on a third nuisance parameter α_i:

$$\exp(y_i) = \exp(x_i) + \exp(n_i) + 2\alpha_i \exp\left(\frac{x_i + n_i}{2}\right) \,, \tag{33.30}$$

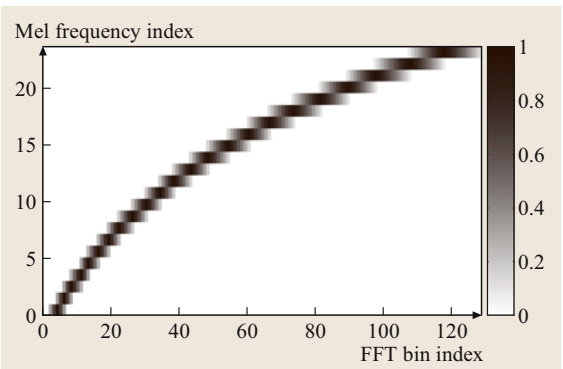


Fig. 33.3 A graphical representation of the mel-frequency filterbank used to compute MFCC features. This filterbank compresses 128 spectral bins into 23 mel-frequency coefficients

where

$$\alpha_i = \frac{\sum_k w_k^i |X[k]N[k]| \cos\theta_k}{\sqrt{\sum_k w_k^i |X[k]|^2}\sqrt{\sum_k w_k^i |N[k]|^2}} \,. \tag{33.31}$$

As a consequence of this model, when we observe y_i there are actually *three* unobserved random variables. The first two are obvious: the clean log spectral energy and the noise log spectral energy that would have been produced in the absence of mixing. The third variable α_i accounts for the unknown phase between the two sources.

If the magnitude spectra are assumed constant over the bandwidth of a particular filterbank, the definition of α_i collapses to a weighted sum of several independent random variables

$$\alpha_i = \frac{\sum_k w_k^i \cos\theta_k}{\sum_j w_j^i} \,. \tag{33.32}$$

According to the central limit theorem, a sum of many independent random variables will tend to be normally distributed. The number of effective terms in (33.32) is controlled by the width of the i-th filterbank. Since filterbanks with higher center frequencies have wider bandwidths, they should be more nearly Gaussian. Figure 33.4 shows the true distributions of α for a range of filterbanks. They were estimated from a joint set of noise and clean speech, and noisy speech taken from the Aurora 2 training data by solving (33.30) for the unknown α_i. As expected, because the higher-frequency

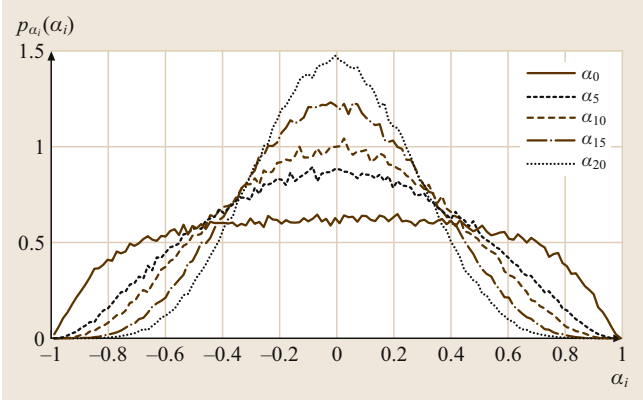


Fig. 33.4 An estimation of the true distribution of α using (33.30) and data from the Aurora 2 corpus. Higher numbered filterbanks cover more frequency bins, are more nearly Gaussian, and have smaller variances

higher-bandwidth filters include more FFT bins, they produce distributions that are more nearly Gaussian.

After some algebraic manipulation, it can be shown that

$$y_i = x_i + \ln\left[1 + \exp(n_i - x_i) + 2\alpha_i \exp\left(\tfrac{n_i - x_i}{2}\right)\right] \,. \tag{33.33}$$

Many current speech enhancement algorithms ignore the cross-term entirely, as in

$$y_i = x_i + \ln[1 + \exp(n_i - x_i)] \,. \tag{33.34}$$

This is entirely appropriate in situations where the observed frequency component is dominated by either noise or speech. In these cases, the cross-term is indeed negligible. But, in cases where the speech and noise components have similar magnitudes, the cross-term can have a considerable effect on the observation.

To create cepstral features from these LMFB features, apply a discrete cosine transform. The j-th cepstral coefficient is calculated with the following sum, where c_{ij} are the DCT coefficients:

$$y_j^{\text{MFCC}} = \sum_i c_{ji} y_i^{\text{LMFB}} \,. \tag{33.35}$$

By convention, this transform includes a truncation of the higher-order DCT coefficients, a process historically referred to as *cepstral liftering*. For example, the Aurora 2 system uses 23 LMFB and keeps only the first 13 cepstral coefficients.

Due to this cepstral truncation, the matrix $\mathbf{C}$ created from the scalars c_{ij} is not square and cannot be inverted. But, we can define a right-inverse matrix $\mathbf{D}$ with elements d_{ij}, such that $\mathbf{CD} = \mathbf{I}$. That is, if $\mathbf{D}$ is applied to a cepstral vector, an approximate spectral vector is produced, from which the projection $\mathbf{C}$ will recreate the original cepstral vector. Using $\mathbf{C}$ and $\mathbf{D}$, (33.34) can be expressed for cepstral vectors $\boldsymbol{x}$, $\boldsymbol{n}$, and $\boldsymbol{y}$ as:

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x} - \boldsymbol{n}) \,, \tag{33.36}$$

$$\boldsymbol{g}(z) = \mathbf{C}\ln[1 + \exp(-\mathbf{D}z)] \,. \tag{33.37}$$

Although $\mathbf{CD} = \mathbf{I}$, it is not the case that $\mathbf{DC} = \mathbf{I}$. In particular, (33.37) is not exactly correct as it needs an extra term $\bar{\mathbf{D}}\bar{\boldsymbol{z}}$ that contains the missing columns from $\mathbf{D}$ and higher-order cepstral coefficients from $\boldsymbol{z}$. If only the truncated cepstrum $\boldsymbol{z}$ is available, then $\bar{\mathbf{D}}\bar{\boldsymbol{z}}$ is a random error term that is generally ignored.

In (33.37), the nonlinearity $\boldsymbol{g}$ was introduced, which maps the signal-to-noise ratio $(\boldsymbol{x} - \boldsymbol{n})$ into the difference between clean and noisy speech $(\boldsymbol{y} - \boldsymbol{x})$. When the noise

Amazon EX1010 Page 680

cepstrum $\boldsymbol{n}$ has significantly less energy than the speech vector $\boldsymbol{x}$, the function $\boldsymbol{g}(\boldsymbol{x}-\boldsymbol{n})$ approaches zero, and $\boldsymbol{y} \approx \boldsymbol{x}$. Conversely, when noise dominates speech, $\boldsymbol{g}(\boldsymbol{x}-\boldsymbol{n}) \approx \boldsymbol{n}-\boldsymbol{x}$, and $\boldsymbol{y} \approx \boldsymbol{n}$.

33.5 Structured Model Adaptation

This section compares two popular methods for structured model adaptation: log-normal parallel model combination and vector Taylor-series adaptation. Both can achieve good adaptation results with only a small amount of data.

By using a set of clean-speech acoustic models and a noise model, both methods approximate the model parameters that would have been obtained by training with corrupted speech.

33.5.1 Analysis of Noisy Speech Features

Figures 33.5 and 33.6 depict how additive noise can distort the spectral features used in ASR systems. In both figures, the simulated clean speech x and noise n are assumed to follow Gaussian distributions:

$$p_x(x) = N(x; \mu_x, \sigma_x) ,$$
$$p_n(n) = N(n; \mu_n, \sigma_n) .$$

After mixing, the noisy speech y distribution is a distorted version of its clean speech counterpart. It is usually shifted, often skewed, and sometimes bimodal. It is clear that a pattern recognition system trained on clean speech will be confused by noisy speech input.

In Fig. 33.5, the clean speech and noise mix to produce a bimodal distribution. We fix $\mu_n = 0\,\text{dB}$, since it is only a relative level, and set $\sigma_n = 2\,\text{dB}$, a typical value. We also set $\mu_x = 25\,\text{dB}$ and see that the resulting distribution is bimodal when σ_x is very large. Fortunately, for modern speech recognition systems that have many Gaussian components, σ_x is never that large and the resulting distribution is often unimodal.

Figure 33.6 demonstrates the more-common skew and offset distortions. Again, the noise parameters are $\mu_n = 0\,\text{dB}$ and $\sigma_n = 2\,\text{dB}$, but a more-realistic value for $\sigma_x = 5\,\text{dB}$ is used. We see that the distribution is always unimodal, although not necessarily symmetric, particularly for low SNR ($\mu_x - \mu_n$).

33.5.2 Log-Normal Parallel Model Combination

Parallel model combination (PMC) [33.29] is a general method to obtain the distribution of noisy speech given the distribution of clean speech and noise as mixtures of Gaussians.

As discussed in Sect. 33.5.1, even if the clean speech and noise cepstra follow Gaussian distributions, the noisy speech will not be Gaussian. The log-normal PMC method nevertheless assumes that the resulting noisy observation is Gaussian. It transforms the clean speech and noise distributions into the linear spectral domain, mixes them, and trans-

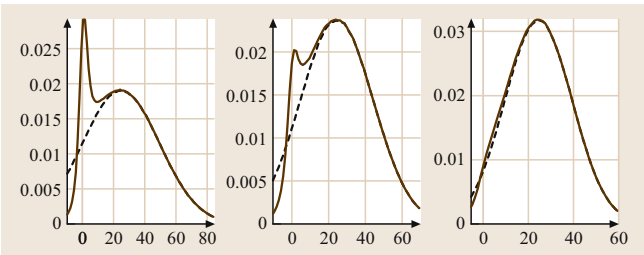


Fig. 33.5 Distributions of the corrupted log-spectra y (*solid lines*) from uncorrupted log-spectra x (*dashed lines*) using simulated data and (33.27). The distribution of the noise log-spectrum n is Gaussian with mean 0 dB and standard deviation of 2 dB. The distribution of the clean log-spectrum x is Gaussian with mean 25 dB and standard deviations of 25, 20, and 15 dB, respectively (the x-axis is expressed in dB). The first two distributions are bimodal, whereas the 15 dB case is more approximately Gaussian

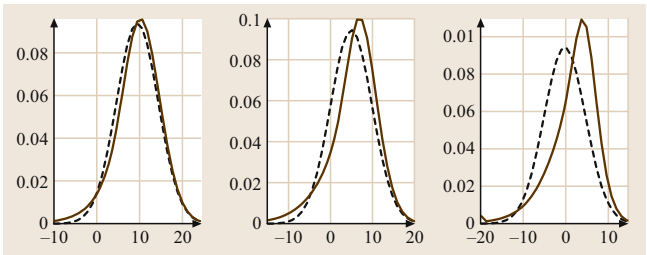


Fig. 33.6 Distributions of the corrupted log-spectra y (*solid lines*) from uncorrupted log-spectra x (*dashed lines*) using simulated data and (33.27). The distribution of the noise log-spectrum n is Gaussian with mean 0 dB and standard deviation of 2 dB. The distribution of the clean log-spectrum is Gaussian with standard deviation of 5 dB and means of 10, 5, and 0 dB, respectively. As the average SNR decreases, the Gaussian experiences more shift and skew

Amazon EX1010 Page 681

forms them back into a distribution on noisy speech cepstra.

If the mean and covariance matrix of the cepstral noise vector $\boldsymbol{n}$ are given by $\mu_{\boldsymbol{n}}^{\mathrm{c}}$ and $\Sigma_{\boldsymbol{n}}^{\mathrm{c}}$, respectively, its mean and covariance matrix in the log spectral domain can be approximated by

$$\mu_{\boldsymbol{n}}^{\mathrm{l}} = \mathbf{D}\mu_{\boldsymbol{n}}^{\mathrm{c}} , \tag{33.38}$$

$$\Sigma_{\boldsymbol{n}}^{\mathrm{l}} = \mathbf{D}\Sigma_{\boldsymbol{n}}^{\mathrm{c}}\mathbf{D}^{\mathrm{T}} . \tag{33.39}$$

As in Sect. 33.4, $\mathbf{D}$ is a right inverse for the noninvertible cepstral rotation $\mathbf{C}$.

In the linear domain $\boldsymbol{N} = \mathrm{e}^{\boldsymbol{n}}$, the noise distribution is log-normal, with a mean vector $\boldsymbol{\mu}_{\boldsymbol{N}}$ and covariance matrix $\Sigma_{\boldsymbol{N}}$ given by

$$\mu_{\boldsymbol{N}}[i] = \exp\left(\mu_{\boldsymbol{N}}^{\mathrm{l}}[i] + \frac{1}{2}\Sigma_{\boldsymbol{N}}^{\mathrm{l}}[i,i]\right) , \tag{33.40}$$

$$\Sigma_{\boldsymbol{N}}[i,j] = \mu_{\boldsymbol{N}}[i]\mu_{\boldsymbol{N}}[j]\left[\exp\left(\Sigma_{\boldsymbol{N}}^{\mathrm{l}}[i,j]\right) - 1\right] . \tag{33.41}$$

As a result, we have the exact parameters for the distribution of noise features in the linear spectral domain. The cepstral Gaussian distribution for the clean speech can be transformed from $\mu_{\boldsymbol{x}}^{\mathrm{c}}$ and $\Sigma_{\boldsymbol{x}}^{\mathrm{c}}$ to $\mu_{\boldsymbol{X}}$ and $\Sigma_{\boldsymbol{X}}$ using expressions similar to (33.38) through (33.41).

Using the basic assumption that the noise and speech waveforms are additive, the spectral vector $\boldsymbol{Y}$ is given by $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{N}$. Without any approximation, the mean and covariance of $\boldsymbol{Y}$ is given by

$$\mu_{\boldsymbol{Y}} = \mu_{\boldsymbol{X}} + \mu_{\boldsymbol{N}} , \tag{33.42}$$

$$\Sigma_{\boldsymbol{Y}} = \Sigma_{\boldsymbol{X}} + \Sigma_{\boldsymbol{N}} . \tag{33.43}$$

Although the sum of two log-normal distributions is not log-normal, the *log-normal approximation* [33.29] consists in assuming that $\boldsymbol{Y}$ is log-normal. In this case we can apply the inverse formulae of (33.40) and (33.41) to obtain the mean and covariance matrix in the log-spectral domain:

$$\Sigma_{\boldsymbol{y}}^{\mathrm{l}} \approx \ln\left(\frac{\Sigma_{\boldsymbol{Y}}[i,j]}{\mu_{\boldsymbol{Y}}[i]\mu_{\boldsymbol{Y}}[j]} + 1\right) , \tag{33.44}$$

$$\mu_{\boldsymbol{y}}^{\mathrm{l}}[i] \approx \ln(\mu_{\boldsymbol{y}}[i]) - \frac{1}{2}\ln\left(\frac{\Sigma_{\boldsymbol{Y}}[i,j]}{\mu_{\boldsymbol{Y}}[i]\mu_{\boldsymbol{Y}}[j]} + 1\right) , \tag{33.45}$$

and finally return to the cepstrum domain applying the inverse of (33.38) and (33.39):

$$\mu_{\boldsymbol{y}}^{\mathrm{c}} = \mathbf{C}\mu_{\boldsymbol{y}}^{\mathrm{l}} , \tag{33.46}$$

$$\Sigma_{\boldsymbol{y}}^{\mathrm{c}} = \mathbf{C}\Sigma_{\boldsymbol{y}}^{\mathrm{l}}\mathbf{C}^{\mathrm{T}} . \tag{33.47}$$

The log-normal approximation cannot be used directly for the delta and delta–delta cepstrum. Another variant that can be used in this case and is more accurate than the log-normal approximation is the *data-driven parallel model combination* (DPMC) [33.29]. DPMC uses Monte Carlo simulation to draw random cepstrum vectors from both the clean-speech HMM and noise distribution to create cepstrum of the noisy speech by applying (33.36) to each sample point. The mean and covariance of these simulated noisy cepstra are then used as adapted HMM parameters. In that respect, it is similar to other model adaptation schemes, but not as accurate as the matched condition training from Sect. 33.2.1 because the distribution is only an approximation.

Although it does not require a lot of memory, DPMC carries with it large computational burden. For each transformed Gaussian component, the recommended number of simulated training vectors is at least 100 [33.29]. A way of reducing the number of random vectors needed to obtain good Monte Carlo simulations is proposed in [33.30].

33.5.3 Vector Taylor-Series Model Adaptation

Vector Taylor-series (VTS) model adaptation is similar in spirit to the log-normal PMC in Sect. 33.5.2, but instead of a log-normal approximation it uses a first-order Taylor-series approximation of the model presented from Sect. 33.4.

This model described the relationship between the cepstral vectors $\boldsymbol{x}$, $\boldsymbol{n}$, and $\boldsymbol{y}$ of the clean speech, noise, and noisy speech, respectively:

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{g}(\boldsymbol{n} - \boldsymbol{x}) , \tag{33.48}$$

where the nonlinear function $\boldsymbol{g}(\boldsymbol{z})$ is given by

$$\boldsymbol{g}(\boldsymbol{z}) = \mathbf{C}\ln[1 + \exp(\mathbf{D}\boldsymbol{z})] . \tag{33.49}$$

As in Sect. 33.4, $\mathbf{C}$ and $\mathbf{D}$ are the cepstral rotation and its right inverse, respectively.

Moreno [33.31] first suggested the use of Taylor series to approximate the nonlinearity in (33.49), though he applies it in the spectral instead of the cepstral domain. Since then, many related techniques have appeared that build upon this core idea [33.32–37]. They mainly differ in how speech and noise are modeled, as well as which assumptions are made in the derivation of the nonlinearity to approximate [33.38].

To apply the vector Taylor-series approximation, first assume that $\boldsymbol{x}$ and $\boldsymbol{n}$ are Gaussian random vectors with means $\{\mu_{\boldsymbol{x}}, \mu_{\boldsymbol{n}}\}$ and covariance matrices $\{\Sigma_{\boldsymbol{x}}, \Sigma_{\boldsymbol{n}}\}$,

Amazon EX1010 Page 682

Table 33.9 Word accuracy for Aurora 2, using VTS model adaptation on an acoustic model trained on clean data

SNR (dB)	Test set A	Test set B	Test set C	Average
Clean	99.63	99.63	99.63	99.63
20	97.86	97.59	98.17	97.88
15	96.47	95.64	96.99	96.37
10	93.47	91.98	93.30	92.91
5	86.87	85.48	85.49	85.94
0	71.68	68.90	68.72	69.77
−5	36.84	33.04	39.04	35.97
Average (0–20)	89.27	87.92	88.53	88.58

and furthermore that $\boldsymbol{x}$ and $\boldsymbol{n}$ are independent. After algebraic manipulation it can be shown that the Jacobian of (33.48) with respect to $\boldsymbol{x}$ and $\boldsymbol{n}$ evaluated at $\{\boldsymbol{x} = \mu_{\boldsymbol{x}}, \boldsymbol{n} = \mu_{\boldsymbol{n}}\}$ can be expressed as:

$$\left.\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}\right|_{(\mu_{\boldsymbol{x}},\mu_{\boldsymbol{n}})} = \mathbf{G}\,, \tag{33.50}$$

$$\left.\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{n}}\right|_{(\mu_{\boldsymbol{x}},\mu_{\boldsymbol{n}})} = \mathbf{I}-\mathbf{G}\,, \tag{33.51}$$

where the matrix $\mathbf{G}$ is given by

$$\mathbf{G} = \mathbf{CFD}\,. \tag{33.52}$$

In (33.52), $\mathbf{F}$ is a diagonal matrix whose elements are given by vector $\boldsymbol{f}(\mu)$, which in turn is given by

$$\boldsymbol{f}(\mu_{\mathrm{n}} - \mu_x) = \frac{1}{1+\exp[\mathbf{D}(\mu_{\mathrm{n}} - \mu_x)]}\,. \tag{33.53}$$

Using (33.50) and (33.51) we can then approximate (33.48) by a first-order Taylor-series expansion around $\{\mu_{\mathrm{n}}, \mu_x\}$ as

$$\begin{aligned}\boldsymbol{y} \approx{}& \mu_x + \boldsymbol{g}(\mu_{\mathrm{n}} - \mu_x) + \mathbf{G}(\boldsymbol{x} - \mu_x)\\ &+ (\mathbf{I}-\mathbf{G})(\boldsymbol{n} - \mu_{\mathrm{n}})\,.\end{aligned} \tag{33.54}$$

The mean of $\boldsymbol{y}$, μ_y, can be obtained from (33.54) as

$$\mu_y \approx \mu_x + \boldsymbol{g}(\mu_{\mathrm{n}} - \mu_x)\,, \tag{33.55}$$

and its covariance matrix Σ_y by

$$\Sigma_y \approx \mathbf{G}(\Sigma_x)\mathbf{G}^{\mathrm{T}} + (\mathbf{I}-\mathbf{G})\Sigma_n(\mathbf{I}-\mathbf{G})^{\mathrm{T}}\,. \tag{33.56}$$

Note that, even if Σ_x and Σ_n are diagonal, Σ_y is not. Nonetheless, it is common to make a diagonal assumption. That way, we can transform a clean HMM to a corrupted HMM that has the same functional form and use a decoder that has been optimized for diagonal covariance matrices.

To compute the means and covariance matrices of the delta and delta–delta parameters, let us take the derivative of the approximation of $\boldsymbol{y}$ in (33.54) with respect to time:

$$\frac{\partial \boldsymbol{y}}{\partial t} \approx \mathbf{G}\frac{\partial \boldsymbol{x}}{\partial t}\,, \tag{33.57}$$

so that the delta cepstrum computed through $\Delta\boldsymbol{x}_t = \boldsymbol{x}_{t+2} - \boldsymbol{x}_{t-2}$, is related to the derivative [33.39] by

$$\Delta\boldsymbol{x} \approx 4\frac{\partial \boldsymbol{x}}{\partial t}\,, \tag{33.58}$$

so that

$$\mu_{\Delta\boldsymbol{y}} \approx \mathbf{G}\mu_{\Delta\boldsymbol{x}}\,, \tag{33.59}$$

and similarly

$$\Sigma_{\Delta\boldsymbol{y}} \approx \mathbf{G}\Sigma_{\Delta\boldsymbol{x}}\mathbf{G}^{\mathrm{T}} + (\mathbf{I}-\mathbf{G})\Sigma_{\Delta\boldsymbol{n}}(\mathbf{I}-\mathbf{G})^{\mathrm{T}}\,. \tag{33.60}$$

Similarly, for the delta–delta cepstrum, the mean is given by

$$\mu_{\Delta^2\boldsymbol{y}} \approx \mathbf{G}\mu_{\Delta^2\boldsymbol{x}}\,, \tag{33.61}$$

and the covariance matrix by

$$\Sigma_{\Delta^2\boldsymbol{y}} \approx \mathbf{G}\Sigma_{\Delta^2\boldsymbol{x}}\mathbf{G}^{\mathrm{T}} + (\mathbf{I}-\mathbf{G})\Sigma_{\Delta^2\boldsymbol{n}}(\mathbf{I}-\mathbf{G})^{\mathrm{T}}\,. \tag{33.62}$$

Table 33.9 demonstrates the effectiveness of VTS model adaptation on the Aurora 2 task. For each test noise condition, a diagonal Gaussian noise model was estimated from the first and last 200 ms of all the test data. Then, each Gaussian component in the acoustic model was transformed according to the algorithm presented above to create a noise condition specific acoustic model. In theory, better results could have been obtained by estimating a new noise model from each utterance and creating an utterance specific acoustic model. However, the computational cost would be much greater.

Amazon EX1010 Page 683

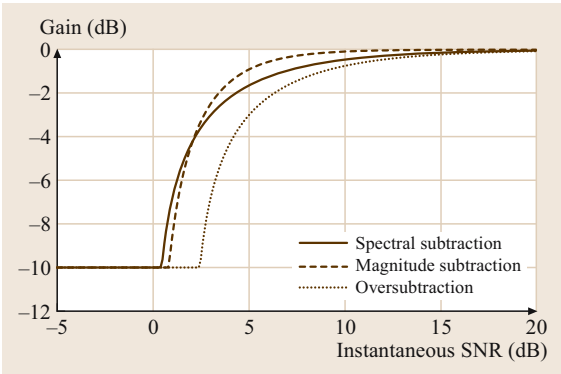


Fig. 33.7 Magnitude of the spectral subtraction filter gain as a function of the input instantaneous SNR for $A = 10$ dB, for the spectral subtraction of (33.68), magnitude subtraction of (33.71), and over-subtraction of (33.72) with $\beta = 2$ dB

Compared to the unadapted results of Table 33.1, the performance of the VTS adapted models is improved at each SNR level on every test set. At 20 dB SNR, the VTS model adaptation reduces the number of errors by 61%. Averaged over 0–20 dB, the adapted system has an average of 73% fewer errors.

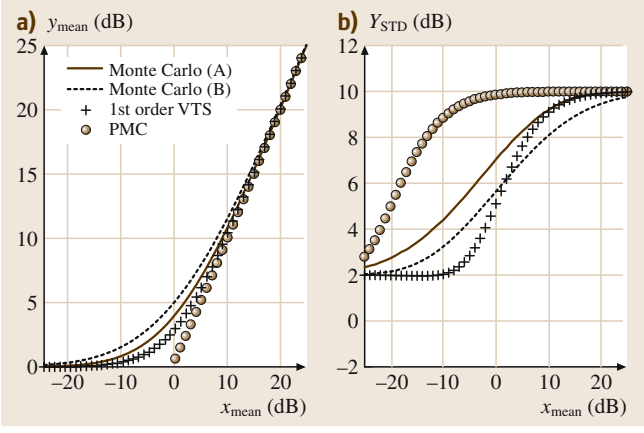


Fig. 33.8a,b Mean and standard deviation of noisy speech y in dB. The distribution of the noise log-spectrum n is Gaussian with mean 0 dB and standard deviation of 2 dB. The distribution of the clean speech log-spectrum x is Gaussian, having a standard deviation of 10 dB and a mean varying from −25 to 25 dB. The first-order VTS approximation is a better estimate for a Monte Carlo simulation of (33.27) when the cross-term is ignored **(b)**, although both VTS and lognormal PMC underestimate the standard deviation compared to when the cross-term is included **(a)**

The VTS model adaptation results also better than any feature normalization technique using clean training data from Sect. 33.3. This is a direct result of using the model from Sect. 33.4 to make better use of the available adaptation data.

33.5.4 Comparison of VTS and Log-Normal PMC

It is difficult to visualize how good the VTS approximation is, given the nonlinearity involved. To provide some insight, Figs. 33.8 and 33.9 provide a comparison between the log-normal approximation, the VTS approximation, and Monte Carlo simulations of (33.27). For simplicity, only a single dimension of the log spectral features are shown.

Figure 33.8 shows the mean and standard deviation of the noisy log-spectral energy y in dB as a function of the mean of the clean log-spectral energy x with a standard deviation of 10 dB. The log-spectral energy of the noise n is Gaussian with mean 0 dB and standard deviation 2 dB.

We see that the VTS approximation is more accurate than the log-normal approximation for the mean, especially in the region around 0 dB SNR. For the standard deviation, neither approximation is very accurate. Because the VTS models fail to account for the cross-term of (33.27), both tend to underestimate the true noisy standard deviation.

Figure 33.9 is similar to Fig. 33.8 except that the standard deviation of the clean log-energy x is only 5 dB, a more-realistic number for a speech recognition system. In this case, both the log-normal approximation and the first-order VTS approximation are good estimates of the mean of y. Again, mostly because they ignore the cross-term, neither approximation gives a reliable estimate for the standard deviation of y in the region between −20 dB and 20 dB.

33.5.5 Strategies for Highly Nonstationary Noises

So far, this section has dealt only with stationary noise. That is, it assumed the noise cepstra for a given utterance are well modeled by a Gaussian distribution. In practice, though, there are many nonstationary noises that do not fit that model. What is worse, nonstationary noises can match a random word in the system's lexicon better than the silence model. In this case, the benefit of using speech recognition vanishes quickly.

Amazon EX1010 Page 684

One solution to the problem of nonstationary noise is to use a more-complex noise model with standard model adaptation algorithms. For instance, the Gaussian noise model can be replaced with a Gaussian mixture model (GMM) or hidden Markov model (HMM).

With a GMM noise model, the decoding becomes more computationally intensive. Before, a Gaussian noise model transformed each component of the clean speech model into one component in the adapted noisy speech model. Now, assume that the noise is independent of speech and is modeled by a mixture of M Gaussian components. Each Gaussian component in the clean speech model will mix independently with every component of the noise GMM. As a result, the acoustic model will grow by a factor of M.

An HMM noise model can provide a much better fit to nonstationary noises [33.40, 41]. However, to efficiently use an HMM noise model, the decoder needs to be modified to perform a three-dimensional Viterbi search which evaluates every possible speech state and noise state at every frame. The computational complexity of performing this *speech/noise decomposition* is very large, though in theory it can handle nonstationary noises quite well.

Alternatively, dedicated whole-word garbage models can bring some of the advantages of an HMM noise model without the additional cost of a three-dimensional Viterbi search. In this technique [33.42], new words are created in the acoustic and language models to cover nonstationary noises such as lip smacks, throat clearings, coughs, and filler words such as *uhm* and *uh*. These nuisance words can be successfully recognized and ignored during non-speech regions, where they tend to cause the most damage.

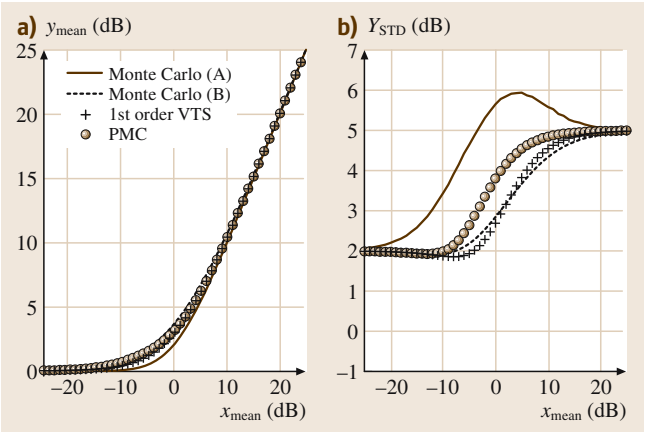


Fig. 33.9a,b Mean and standard deviation of noisy speech y in dB. The distribution of the noise log-spectrum n is Gaussian with mean of 0 dB and standard deviation of 2 dB. The distribution of the clean log-spectrum x is Gaussian with a standard deviation of 5 dB and a mean varying from −25 dB to 25 dB. Both log-normal PMC and the first-order VTS approximation make good estimates compared to a Monte Carlo simulation of (33.27) when the cross-term is ignored **(b)**, although the standard deviation is revealed as an underestimate when the cross-term is included **(a)**

33.6 Structured Feature Enhancement

This section presents several popular methods for structured feature enhancement. Whereas the techniques of Sect. 33.5 adapt the recognizer's acoustic model parameters, the techniques discussed here use mathematical models of the noise corruption to enhance the features before they are presented to the recognizer.

Methods in this class have a rich history. *Boll* [33.43] pioneered the use of spectral subtraction for speech enhancement almost thirty years ago, and it is still found in many systems today. *Ephraim* and *Malah* [33.44] invented their logarithmic minimum mean-squared error short-time spectral-amplitude (logMMSE STSA) estimator shortly afterwards, which forms the basis of many of today's advanced systems. Whereas these earlier approaches use weak speech models, today's most promising systems use stronger models, coupled with vector Taylor series speech enhancement [33.45].

Some of these techniques were developed for speech enhancement for human consumption. Unfortunately, what sounds good to a human can confuse an automatic speech recognition system, and automatic systems can tolerate distortions that are unacceptable to human listeners.

33.6.1 Spectral Subtraction

Spectral subtraction is built on the assumption that the observed power spectrum is approximately the sum of the power spectra for the clean signal and the noise. Although this assumption holds in the expected sense, as we have seen before, in any given frame it is only a rough approximation (Sect. 33.4):

$$|Y(f)|^2 \approx |X(f)|^2 + |N(f)|^2 \,. \qquad (33.63)$$

Using (33.63), the clean power spectrum can be estimated by subtracting an estimate of the noise power spectrum from the noisy power spectrum:

$$|\hat{X}(f)|^2 = |Y(f)|^2 - |\hat{N}(f)|^2 = |Y(f)|^2 H_{\text{ss}}^2(f)\,, \tag{33.64}$$

where the equivalent spectral subtraction filter in the power spectral domain is

$$H_{\text{ss}}^2(f) = 1 - \frac{1}{\text{SNR}(f)}\,, \tag{33.65}$$

and the frequency-dependent signal-to-noise ratio $\text{SNR}(f)$ is

$$\text{SNR}(f) = \frac{|Y(f)|^2}{|\hat{N}(f)|^2}\,. \tag{33.66}$$

The weakest point in many speech enhancement algorithms is the method for estimating the noise power spectrum. More advanced enhancement algorithms can be less sensitive to this estimate, but spectral subtraction is quite sensitive. The easiest option is to assume the noise is stationary, and obtain an estimate using the average periodogram over M frames that are known to be just noise

$$|\hat{N}(f)|^2 = \frac{1}{M}\sum_{i=0}^{M-1} |Y_i(f)|^2\,. \tag{33.67}$$

In practice, there is no guarantee that the spectral subtraction filter in (33.65) is nonnegative, which violates the fundamental nature of power spectra. In fact, it is easy to see that noise frames do not comply. To enforce this constraint, *Boll* [33.43] suggested modifying the filter as

$$H_{\text{ss}}^2(f) = \max\left(1 - \frac{1}{\text{SNR}(f)}, a\right) \tag{33.68}$$

with $a \geq 0$, so that the filter is always positive.

This implementation results in output speech that has significantly less noise, though it exhibits what is called *musical noise* [33.46]. This is caused by frequency bands f for which $|Y(f)|^2 \approx |\hat{N}(f)|^2$. As shown in Fig. 33.7, a frequency f_0 for which $|Y(f_0)|^2 < |\hat{N}(f_0)|^2$ is attenuated by A dB, whereas a neighboring frequency f_1, where $|Y(f_1)|^2 > |\hat{N}(f_1)|^2$, has a much smaller attenuation. These rapid changes of attenuation over frequency introduce tones at varying frequencies that appear and disappear rapidly.

The main reason for the presence of musical noise is that the estimates of $\text{SNR}(f)$ are poor. This is partly because the $\text{SNR}(f)$ are computed independently for every time and frequency, even though it would be more reasonable to assume that nearby values are correlated. One possibility is to smooth the filter in (33.68) over time, frequency, or both. This approach suppresses a smaller amount of noise, but it does not distort the signal as much, and thus may be preferred. An easy way to smooth over time is to mix a small portion of the previous SNR estimate into the current frame:

$$\text{SNR}(f,t) = \gamma\,\text{SNR}(f,t-1) + (1-\gamma)\frac{|Y(f)|^2}{|\hat{N}(f)|^2}\,. \tag{33.69}$$

This smoothing yields more-accurate SNR measurements and thus less distortion, at the expense of reduced noise suppression.

Other enhancements to the basic algorithm have been proposed to reduce the musical noise. Sometimes (33.68) is generalized to

$$f_{\text{ms}}(x) = \left[\max\left(1 - \frac{1}{x^{\frac{\alpha}{2}}}, a\right)\right]^{\frac{1}{\alpha}}\,, \tag{33.70}$$

where $\alpha = 2$ corresponds to the *power spectral subtraction rule* in (33.64), and $\alpha = 1$ corresponds to the *magnitude subtraction rule* (plotted in Fig. 33.7 for $A = 10\,\text{dB}$):

$$g_{\text{ms}}(\overline{x}) = \max\left[20\log_{10}\left(1 - 10^{-\frac{\overline{x}}{5}}\right), -A\right]\,. \tag{33.71}$$

Another variation, called oversubtraction, consists of multiplying the estimate of the noise power spectral density $|\hat{N}(f)|^2$ in (33.67) by a constant $10^{\frac{\beta}{10}}$, where $\beta > 0$, which causes the power spectral subtraction rule to be transformed to another function

$$g_{\text{ms}}(\overline{x}) = \max\left\{10\log_{10}\left[1 - 10^{-\frac{(\overline{x}-\beta)}{10}}\right], -A\right\}\,. \tag{33.72}$$

This causes $|Y(f)|^2 < |\hat{N}(f)|^2$ to occur more often than $|Y(f)|^2 > |\hat{N}(f)|^2$ for frames for which $|Y(f)|^2 \approx |\hat{N}(f)|^2$, and thus reduces the musical noise.

Since the normal cepstral processing for speech recognition includes finding the power spectrum for each frame of data, spectral subtraction can be performed directly in the feature extraction module, without the need for resynthesis. Instead of operating directly on the full-resolution power spectrum, a popular alternative is to use the mel-frequency power spectrum. This does not change the motivation or derivation for spectral subtraction, but the mel-frequency power spectrum is more stable and less susceptible to musical noise.

Because spectral techniques like spectral subtraction can be implemented with very little computational cost, they are popular in embedded applications. In particular, the advanced front-end (AFE) standard produced by the ETSI DSW working group [33.47, 48]

Amazon EX1010 Page 686

Table 33.10 ETSI advanced front-end word accuracy for Aurora 2. This low-resource front end produces respectable results on the task

Acoustic model	Test set A	Test set B	Test set C	Average
Clean	89.27	87.92	88.53	88.58
Multistyle	93.74	93.26	92.21	93.24

uses a variation of this technique in its noise reduction module. The complete AFE noise reduction process consists of a two-stage time-smoothed frequency-domain filtering [33.49], time-domain noise reduction [33.50], SNR-dependent waveform processing [33.51], and an online variant of CMN [33.52].

Table 33.10 presents the performance of the ETSI AFE on the Aurora 2 clean and multistyle tasks. Despite its low computational cost, the multistyle AFE has only 10% more errors than the best system described in this chapter (Table 33.15). It achieves a average word accuracy of 88.19% when trained with clean data, and 93.24% when trained with multistyle data.

33.6.2 Vector Taylor-Series Speech Enhancement

As shown in Sect. 33.5.3, the VTS approximation can be used to create a noisy speech model from a clean speech model. But, adapting each Gaussian component in a large speech model can be quite computationally expensive.

A lightweight alternative is to leverage a smaller model for speech into a separate enhancement step. The VTS approximation is applied to this smaller model, which is then used to estimate the enhanced features that are sent to an unmodified recognition system. A good comprehensive summary of this approach can be found in [33.45], and the literature is rich with implementations [33.38, 53].

Typically, the clean speech model is a multivariate Gaussian mixture model, and the noise model is a single Gaussian component. The parameters of this prior model include the state-conditional means and variances, $\mu_s^{\boldsymbol{x}}$, $\sigma_s^{\boldsymbol{x}}$, $\mu^{\boldsymbol{n}}$ and $\sigma^{\boldsymbol{n}}$, as well as the mixture component weights $p(s)$:

$$p(\boldsymbol{x}) = \sum_s N\left(\boldsymbol{x}; \mu_s^{\boldsymbol{x}}, \sigma_s^{\boldsymbol{x}}\right) p(s) \,, \tag{33.73}$$

$$p(\boldsymbol{n}) = N(\boldsymbol{n}; \mu^{\boldsymbol{n}}, \sigma^{\boldsymbol{n}}) \,. \tag{33.74}$$

These models are tied together by the nonlinear mixing represented by (33.36), recast as a probability distribution:

$$p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{n}) \approx N[\boldsymbol{y}; \boldsymbol{x} + \boldsymbol{g}(\boldsymbol{x} - \boldsymbol{n}), \Psi^2] \,. \tag{33.75}$$

Here, the variance Ψ^2 chiefly represents the error incurred in ignoring the cross-term produced by mixing the speech and noise (Sect. 33.4), and is in general a function of $(\boldsymbol{x} - \boldsymbol{n})$. For an overview of three reasonable approximations for Ψ^2 [33.38].

This Chapter uses the approximation $\Psi^2 = 0$, which is equivalent to ignoring the effects of the cross-term entirely. We call this approximation the zero-variance model (ZVM). This yields a good fit to the data at the extreme SNR regions, and a slight mismatch in the $\boldsymbol{x} \approx \boldsymbol{n}$ region.

If the VTS techniques from Sect. 33.5.3 were applied directly to the variables $\boldsymbol{x}$ and $\boldsymbol{n}$ for VTS enhancement, the result would be quite unstable [33.38]. Instead, the problem is reformulated in terms of $\boldsymbol{r}$, the instantaneous signal-to-noise ratio, defined as

$$\boldsymbol{r} = \boldsymbol{x} - \boldsymbol{n} \,.$$

By performing VTS on the new variable $\boldsymbol{r}$, the stability problems are circumvented. In the end, an estimate for the instantaneous SNR can be mapped back into estimates of $\boldsymbol{x}$ and $\boldsymbol{n}$ through

$$\boldsymbol{x} = \boldsymbol{y} - \boldsymbol{g}(\boldsymbol{r}) \,, \tag{33.76}$$

$$\boldsymbol{n} = \boldsymbol{y} - \boldsymbol{g}(\boldsymbol{r}) - \boldsymbol{r} \,. \tag{33.77}$$

These formulas satisfy the intuition that as the SNR $\boldsymbol{r}$ becomes more positive, $\boldsymbol{x}$ approaches $\boldsymbol{y}$ from below. As the SNR $\boldsymbol{r}$ becomes more negative, $\boldsymbol{n}$ approaches $\boldsymbol{y}$ from below.

The joint PDF for the ZVM is a distribution over the clean speech $\boldsymbol{x}$, the noise $\boldsymbol{n}$, the observation $\boldsymbol{y}$, the SNR $\boldsymbol{r}$, and the speech state s:

$$p(\boldsymbol{y}, \boldsymbol{r}, \boldsymbol{x}, \boldsymbol{n}, s) = p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{n}) p(\boldsymbol{r}|\boldsymbol{x}, \boldsymbol{n}) p(\boldsymbol{x}, s) p(\boldsymbol{n}) \,.$$

The observation and SNR are both deterministic functions of $\boldsymbol{x}$ and $\boldsymbol{n}$. As a result, the conditional probabilities $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{n})$ and $p(\boldsymbol{r}|\boldsymbol{x}, \boldsymbol{n})$ can be represented by Dirac delta functions:

$$p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{n}) = \delta[\ln(\mathrm{e}^{\boldsymbol{x}} + \mathrm{e}^{\boldsymbol{n}}) - \boldsymbol{y}] \,, \tag{33.78}$$

$$p(\boldsymbol{r}|\boldsymbol{x}, \boldsymbol{n}) = \delta(\boldsymbol{x} - \boldsymbol{n} - \boldsymbol{r}) \,. \tag{33.79}$$

Amazon EX1010 Page 687

Table 33.11 Word accuracy for Aurora 2, using VTS speech enhancement and an acoustic model trained on clean data. Enhancement is performed on static cepstra, which are then used to compute the dynamic coefficients used in recognition. This enhancement technique does better than any of the normalization techniques

Acoustic model	Iterations	Set A	Set B	Set C	Average
Clean	0	88.59	88.06	86.92	88.04
Clean	1	89.67	89.05	87.37	88.96
Clean	2	89.92	89.39	87.28	89.18
Clean	3	89.99	89.48	87.05	89.20

Table 33.12 Word accuracy for Aurora 2, using VTS speech enhancement and an acoustic model trained on enhanced multistyle data. As in Table 33.11, the dynamic coefficients are calculated from enhanced static coefficients. In the end, the result is not as good as simple feature normalization alone on a multistyle acoustic model

Acoustic model	Iterations	Set A	Set B	Set C	Average
Multistyle	0	92.58	91.14	92.52	91.99
Multistyle	1	92.79	91.27	92.24	92.07
Multistyle	2	92.86	91.35	92.10	92.11
Multistyle	3	92.82	91.35	91.94	92.06

This allows us to marginalize the continuous variables $\boldsymbol{x}$ and $\boldsymbol{n}$, as in

$$\begin{aligned} p(\boldsymbol{y},\boldsymbol{r},s) &= \int \mathrm{d}\boldsymbol{x} \int \mathrm{d}\boldsymbol{n}\, p(\boldsymbol{y},\boldsymbol{r},\boldsymbol{x},\boldsymbol{n},s) \\ &= N[\boldsymbol{y}-\boldsymbol{g}(\boldsymbol{r});\mu_s^{\boldsymbol{x}},\sigma_s^{\boldsymbol{x}}]p(s) \\ &\quad \times N[\boldsymbol{y}-\boldsymbol{g}(\boldsymbol{r})-\boldsymbol{r};\mu^{\boldsymbol{n}},\sigma^{\boldsymbol{n}}]\,. \end{aligned} \tag{33.80}$$

After marginalization, the only remaining continuous hidden variable is $\boldsymbol{r}$, the instantaneous SNR. The behavior of this joint PDF is intuitive. At high SNR,

$$p(\boldsymbol{y},\boldsymbol{r},s) \approx N(\boldsymbol{y};\mu_s^{\boldsymbol{x}},\sigma_s^{\boldsymbol{x}})p(s)N(\boldsymbol{y}-\boldsymbol{r};\mu^{\boldsymbol{n}},\sigma^{\boldsymbol{n}})\,.$$

That is, the observation is modeled as clean speech, and the noise is at a level $\boldsymbol{r}$ units below the observation. The converse is true for low SNR:

$$p(\boldsymbol{y},\boldsymbol{r},s) \approx N(\boldsymbol{y}-\boldsymbol{r};\mu_s^{\boldsymbol{x}},\sigma_s^{\boldsymbol{x}})p(s)N(\boldsymbol{y};\mu^{\boldsymbol{n}},\sigma^{\boldsymbol{n}})\,.$$

To solve the MMSE estimation problem, the nonlinear function $\boldsymbol{g}(\boldsymbol{r})$ in (33.80) is replaced by its Taylor-series approximation, as in Sect. 33.5.3. For now, the expansion point is the expected a priori mean of $\boldsymbol{r}$:

$$\boldsymbol{r}_s^0 = E\{\boldsymbol{r}|s\} = E\{\boldsymbol{x}-\boldsymbol{n}|s\} = \mu_s^{\boldsymbol{x}} - \mu^{\boldsymbol{n}}\,.$$

Using (33.80) and the VTS approximation, it can be shown that $p(\boldsymbol{y}|s)$ has the same form derived in Sect. 33.5.3. Essentially, we perform VTS adaptation of the clean speech GMM to produce a GMM for noisy speech:

$$p(\boldsymbol{y}|s) = N(\boldsymbol{y};\mu_{\boldsymbol{y}|s},\sigma_{\boldsymbol{y}|s})\,, \tag{33.81}$$

$$\mu_{\boldsymbol{y}|s} = \mu_s^{\boldsymbol{x}} + \boldsymbol{g}\left(-\boldsymbol{r}_s^0\right) + \mathbf{G}_s\left(\mu_s^{\boldsymbol{x}} - \mu_s^{\boldsymbol{n}} - \boldsymbol{r}_s^0\right)\,, \tag{33.82}$$

$$\sigma_{\boldsymbol{y}|s} = \mathbf{G}_s\sigma_s^{\boldsymbol{x}}\mathbf{G}_s^{\mathrm{T}} + (\mathbf{I}-\mathbf{G}_s)\sigma^{\boldsymbol{n}}(\mathbf{I}-\mathbf{G}_s)^{\mathrm{T}}\,. \tag{33.83}$$

When using the recommended expansion point, (33.82) simplifies to

$$\mu_{\boldsymbol{y}|s} = \mu^{\boldsymbol{x}} + \boldsymbol{g}\left(\mu^{\boldsymbol{n}} - \mu_s^{\boldsymbol{x}}\right)\,.$$

It is also straightforward to derive the conditional posterior $p(\boldsymbol{r}|\boldsymbol{y},s)$, as in (33.84). As in the other iterative VTS algorithms, we can use the expected value $E[\boldsymbol{r}|\boldsymbol{y},s] = \mu_s^{\boldsymbol{r}}$ as a new expansion point for the vector Taylor-series parameters and iterate.

$$\begin{aligned} p(\boldsymbol{r}|\boldsymbol{y},s) &= N\left(\boldsymbol{r};\mu_s^{\boldsymbol{r}},\sigma_s^{\boldsymbol{r}}\right)\,, \\ \left(\sigma_s^{\boldsymbol{r}}\right)^{-1} &= (\mathbf{I}-\mathbf{G}_s)^{\mathrm{T}}\left(\sigma_s^{\boldsymbol{x}}\right)^{-1}(\mathbf{I}-\mathbf{G}_s) \\ &\quad + (\mathbf{G}_s)^{\mathrm{T}}(\sigma^{\boldsymbol{n}})^{-1}\mathbf{G}_s\,, \\ \mu_s^{\boldsymbol{r}} &= \mu_s^{\boldsymbol{x}} - \mu^{\boldsymbol{n}} + \sigma_s^{\boldsymbol{r}}\left[(\mathbf{G}_s-\mathbf{I})^{\mathrm{T}}\left(\sigma_s^{\boldsymbol{x}}\right)^{-1}\right. \\ &\quad \left. + (\mathbf{G}_s)^{\mathrm{T}}(\sigma^{\boldsymbol{n}})^{-1}\right](\boldsymbol{y}-\mu_{\boldsymbol{y}|s})\,. \end{aligned} \tag{33.84}$$

After convergence, we compute an estimate of x from the parameters of the approximate model:

$$\hat{\boldsymbol{x}} = \sum_s E[\boldsymbol{x}|\boldsymbol{y},s]p(s|\boldsymbol{y})\,,$$

$$E[\boldsymbol{x}|\boldsymbol{y},s] \approx \boldsymbol{y} - \ln(\mathrm{e}^{\mu_s^{\boldsymbol{r}}}+1) + \mu_s^{\boldsymbol{r}}\,.$$

Here, (33.76) has been used to map $E[\boldsymbol{r}|\boldsymbol{y},s] = \mu_s^{\boldsymbol{r}}$ to $E[\boldsymbol{x}|\boldsymbol{y},s]$. Since the transformation is nonlinear, our estimate for $\hat{\boldsymbol{x}}$ is not the optimal MMSE estimator.

Tables 33.11–33.14 evaluate accuracy on Aurora 2 for several different VTS speech enhancement configurations. For all experiments, the clean speech GMM consisted of 32 diagonal components trained on the clean Aurora 2 training data.

Amazon EX1010 Page 688

Table 33.13 Word accuracy for Aurora 2, using VTS speech enhancement and an acoustic model trained on clean data. Unlike Table 33.11, the static cepstral coefficients are enhanced and then concatenated with noisy dynamic coefficients. The end result is worse than computing the dynamic coefficients from enhanced static coefficients

Acoustic model	Iterations	Set A	Set B	Set C	Average
Clean	0	83.56	84.60	82.52	83.77
Clean	1	84.38	85.48	82.82	84.51
Clean	2	84.60	85.72	82.76	84.68
Clean	3	84.68	85.81	82.65	84.73

Table 33.14 Word accuracy for Aurora 2, using VTS speech enhancement and an acoustic model trained on enhanced multistyle data. Unlike Table 33.12, the static cepstral coefficients are enhanced and then concatenated with noisy dynamic coefficients. The end result is better than computing the dynamic coefficients from enhanced static coefficients

Acoustic model	Iterations	Set A	Set B	Set C	Average
Multistyle	0	93.64	93.05	93.05	93.29
Multistyle	1	93.76	93.14	92.81	93.32
Multistyle	2	93.72	93.19	92.79	93.32
Multistyle	3	93.57	92.95	92.70	93.15

Table 33.15 Word accuracy for Aurora 2, adding CMVN after the VTS speech enhancement of Table 33.14. The result is a very high recognition accuracy

Acoustic model	Normalization	Set A	Set B	Set C	Average
Multistyle	None	93.72	93.19	92.79	93.32
Multistyle	CMVN	94.09	93.46	94.01	93.83

Of course, using the multistyle training data produces better accuracy. Comparing Tables 33.11 and 33.12, it is apparent that multistyle data should be used whenever possible. Note that the result with the clean acoustic model is better than all of the feature normalization techniques explored in this chapter, and better than the VTS adaptation result of Table 33.9. Because VTS enhancement is fast enough to compute new parameters specific to each utterance, it is able to better adapt to the changing conditions within each noise type.

Although the speech recognition features consist of static and dynamic cepstra, the VTS enhancement is only defined on the static cepstra. As a result, there are two options for computing the dynamic cepstra. In Tables 33.11 and 33.12, the dynamic cepstra were computed from the enhanced static cepstra. In the corresponding Tables 33.13 and 33.14, the dynamic cepstra were computed from the noisy static cepstra. In the former case, the entire feature vector is affected by the enhancement algorithm, and in the latter, only the static cepstra are modified.

Using the noisy dynamic cepstra turns out to be better for the multistyle acoustic model, but worse for the clean acoustic model. Under the clean acoustic model, the benefit of the enhancement outweighs the distortion it introduces. However, the multistyle acoustic model is already able to learn and generalize the dynamic coefficients from noisy speech. Table 33.14 shows that the best strategy is to only enhance the static coefficients.

Finally, consider the effect of adding feature normalization to the system. Normalization occurs just after the full static and dynamic feature vector is created, and before it is used by the recognizer. Table 33.15 presents the final accuracy achieved by adding CMVN to the result of Table 33.14. Even though the accuracy of the multistyle model was already quite good, CMVN reduces the error rate by another 7.6% relative.

33.7 Unifying Model and Feature Techniques

The front- and back-end methods discussed so far can be mixed and matched to good performance. This section introduces two techniques that achieve better accuracy through a tighter in-

tegration of the front- and back-end robustness techniques.

33.7.1 Noise Adaptive Training

Section 33.2 discussed how the recognizer's HMMs can be adapted to a new acoustical environment. Section 33.6 dealt with cleaning the noisy feature without retraining the HMMs. It is logical to consider a combination of both, where the features are cleaned to remove noise and channel effects and then the HMMs are retrained to take into account that this processing stage is not perfect.

It was shown in [33.19] that a combination of feature enhancement and matched condition training can achieve a lower word error rate than feature enhancement alone. This paper demonstrated how, by introducing a variant of the enhancement algorithm from Sect. 33.3.4, very low error rates could be achieved.

These low error rates are hard to obtain in practice, because they assume preknowledge of the exact noise type and level, which in general is difficult to obtain. On the other hand, this technique can be effectively combined with the multistyle training discussed in Sect. 33.2.1.

33.7.2 Uncertainty Decoding and Missing Feature Techniques

Traditionally, the speech enhancement algorithms from Sect. 33.6 output an estimate of the clean speech to be used by the speech recognition system. However, the accuracy of the noise removal process can vary from frame to frame and from dimension to dimension in the feature stream.

Uncertainty decoding [33.54] is a technique where the feature enhancement algorithm associates a confidence with each value that it outputs. In frames with a high signal-to-noise ratio, the enhancement can be very accurate and would be associated with high confidence. Other frames, where some or all of the speech has been buried in noise, would have low confidence.

The technique is implemented at the heart of the speech recognition engine, where many Gaussian mixture components are evaluated. When recognizing uncorrupted speech cepstra, the purpose of these evaluations is to discover the probability of each clean observation vector, conditioned on the mixture index, $p_{\boldsymbol{x}|m}(\boldsymbol{x}|m)$, for each Gaussian component in the speech model used by the recognizer.

If the training and testing conditions do not match, as is the case in noise-corrupted speech recognition, one option is to ignore the imperfections of the noise removal, and evaluate $p_{\boldsymbol{x}|m}(\hat{\boldsymbol{x}}|m)$. This is the classic case of passing the output of the noise removal algorithm directly to the recognizer.

With the uncertainty decoding technique, the joint conditional PDF $p(\boldsymbol{x}, \boldsymbol{y}|m)$ is generated and marginalized over all possible unseen clean-speech cepstra:

$$p(\boldsymbol{y}|m) = \int_{-\infty}^{\infty} p(\boldsymbol{y}, \boldsymbol{x}|m)\,\mathrm{d}\boldsymbol{x}\,. \tag{33.85}$$

Under this framework, instead of just providing cleaned cepstra, the speech enhancement process also estimates the conditional distribution $p(\boldsymbol{y}|\boldsymbol{x}, m)$, as a function of $\boldsymbol{x}$. For ease of implementation, it is generally assumed that $p(\boldsymbol{y}|\boldsymbol{x})$ is independent of m:

$$p(\boldsymbol{y}|\boldsymbol{x}, m) \approx p(\boldsymbol{y}|\boldsymbol{x}) = \alpha N\big(\boldsymbol{x}; \hat{\boldsymbol{x}}(\boldsymbol{y}), \sigma^2_{\hat{\boldsymbol{x}}}(\boldsymbol{y})\big)\,, \tag{33.86}$$

where α is independent of $\boldsymbol{x}$, and therefore can be ignored by the decoder. Note that $\hat{\boldsymbol{x}}$ and $\sigma^2_{\hat{\boldsymbol{x}}}$ are always functions of $\boldsymbol{y}$; the cumbersome notation is dropped for the remainder of this discussion.

Finally, the probability for the observation $\boldsymbol{y}$, conditioned on each acoustic model Gaussian mixture component m, can be calculated:

$$\begin{aligned} p(\boldsymbol{y}|m) &= \int_{-\infty}^{\infty} p(\boldsymbol{y}|\boldsymbol{x}, m)p(\boldsymbol{x}|m)\,\mathrm{d}\boldsymbol{x} \\ &\propto \int_{-\infty}^{\infty} N\big(\hat{\boldsymbol{x}}; \boldsymbol{x}, \boldsymbol{\sigma}^2_{\hat{\boldsymbol{x}}}\big) N\big(\boldsymbol{x}; \boldsymbol{\mu}_m, \boldsymbol{\sigma}^2_m\big)\,\mathrm{d}\boldsymbol{x} \\ &= N\big(\hat{\boldsymbol{x}}; \boldsymbol{\mu}_m, \boldsymbol{\sigma}^2_m + \boldsymbol{\sigma}^2_{\hat{\boldsymbol{x}}}\big)\,. \end{aligned} \tag{33.87}$$

This formula is evaluated for each Gaussian mixture component in the decoder, $p(\boldsymbol{x}|m) = N(\boldsymbol{x}, \boldsymbol{\mu}_m, \boldsymbol{\sigma}^2_m)$.

As can be observed in (33.87), the uncertainty output from the front end increases the variance of the Gaussian mixture component, producing an effective smoothing in cases where the front end is uncertain of the true value of the cleaned cepstra.

Two special cases exist for uncertainty decoding. In the absence of uncertainty information from the noise removal process, we can either assume that there is no uncertainty or that there is complete uncertainty.

If there were no uncertainty, then $\boldsymbol{\sigma}^2_{\hat{\boldsymbol{x}}} = 0$. The probability of the observation $\boldsymbol{y}$ for each acoustic model Gaussian mixture component m simplifies to:

$$p(\boldsymbol{y}|m) = p(\hat{\boldsymbol{x}}|m) = N\big(\hat{\boldsymbol{x}}; \boldsymbol{\mu}_m, \boldsymbol{\sigma}^2_m\big)\,. \tag{33.88}$$

This is the traditional method of passing features directly from the noise removal algorithm to the decoder.

Amazon EX1010 Page 690

If there were complete uncertainty of any of the cepstral coefficients, the corresponding $\sigma_{\hat{x}}^2$ would approach infinity. That coefficient would have no effect on the calculation of $p(\boldsymbol{y}|m)$. This is desirable behavior, under the assumption that the coefficient could not contribute to discrimination.

Both of these extreme cases are similar to the computations performed when using hard thresholds with missing-feature techniques [33.55]. There has been some success in incorporating heuristic soft thresholds with missing-feature techniques [33.56], but without the benefits of a rigorous probabilistic framework.

33.8 Conclusion

To prove a newly developed system, it can be tested on any one of a number of standard noise-robust speech recognition tasks. Because a great number of researchers are publishing systematic results on the same tasks, the relative value of complete solutions can be easily assessed.

When building a noise-robust speech recognition system, there exist several simple techniques that should be tried before more-complex strategies are invoked. These include the feature normalization techniques of Sect. 33.3, as well as the model retraining methods of Sect. 33.2.

If state-of-the-art performance is required with a small amount of adaptation data, then the structured techniques of Sects. 33.5 and 33.6 can be implemented. Structured model adaptation carries with it an expensive computational burden, whereas structured feature enhancement is more lightweight but less accurate.

Finally, good compromises between the accuracy of model adaptation and the speed of feature enhancement can be achieved through a tighter integration of the front and back end, as shown in Sect. 33.7.

References

33.1 H.G. Hirsch, D. Pearce: The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions, ISCA ITRW ASR2000 "Automatic Speech Recognition: Challenges for the Next Millennium" (2000)

33.2 R.G. Leonard, G. Doddington: *Tidigits* (Linguistic Data Consortium, Philadelphia 1993)

33.3 D. Pierce, A. Gunawardana: Aurora 2.0 speech recognition in noise: Update 2. Complex backend definition for Aurora 2.0, http://icslp2002.colorado.edu/special_sessions/aurora (2002)

33.4 A. Moreno, B. Lindberg, C. Draxler, G. Richard, K. Choukri, J. Allen, S. Euler: SpeechDat-Car: A large speech database for automotive environments, Proc. 2nd Int. Conf. Language Resources and Evaluation (2000)

33.5 J. Garofalo, D. Graff, D. Paul, D. Pallett: *CSR-I (WSJ0) Complete* (Linguistic Data Consortium, Philadelphia 1993)

33.6 A. Varga, H.J.M. Steeneken, M. Tomlinson, D. Jones: The NOISEX-92 study on the effect of additive noise on automatic speech recognition. Tech. Rep. Defence Evaluation and Research Agency (DERA) (Speech Research Unit, Malvern 1992)

33.7 A. Schmidt-Nielsen: *Speech in Noisy Environments (SPINE) Evaluation Audio* (Linguistic Data Consortium, Philadelphia 2000)

33.8 R.P. Lippmann, E.A. Martin, D.P. Paul: Multi-style training for robust isolated-word speech recognition, Proc. IEEE ICASSP (1987) pp. 709–712

33.9 M. Matassoni, M. Omologo, D. Giuliani: Hands-free speech recognition using a filtered clean corpus and incremental HMM adaptation, Proc. IEEE ICASSP (2000) pp. 1407–1410

33.10 G. Saon, J.M. Huerta, E.-E. Jan: Robust digit recognition in noisy environments: The Aurora 2 system, Proc. Eurospeech **2001** (2001)

33.11 C.-P. Chen, K. Filali, J.A. Bilmes: Frontend post-processing and backend model enhancement on the Aurora 2.0/3.0 databases, Int. Conf. Spoken Language Process. (2002)

33.12 B.S. Atal: Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification, J. Acoust. Soc. Am. **55**(6), 1304–1312 (1974)

33.13 B.W. Gillespie, L.E. Atlas: Acoustic diversity for improved speech recognition in reverberant environments, Proc. IEEE ICASSP **I**, 557–560 (2002)

33.14 A. de la Torre, A.M. Peinado, J.C. Segura, J.L. Perez-Cordoba, M.C. Benítez, A.J. Rubio: Histogram equalization of speech representation for robust speech recognition, IEEE Trans. Speech Audio Process. **13**(3), 355–366 (2005)

Amazon EX1010 Page 691

33.15 M.G. Rahim, B.H. Juang: Signal bias removal by maximum likelihood estimation for robust telephone speech recognition, IEEE Trans. Speech Audio Process. **4**(1), 19–30 (1996)

33.16 A. Acero, X.D. Huang: Augmented cepstral normalization for robust speech recognition, Proc. IEEE Workshop on Automatic Speech Recognition (1995)

33.17 J. Ramírez, J.C. Segura, C. Benítez, L. García, A. Rubio: Statistical voice activity detection using a multiple observation likelihood ratio test, IEEE Signal Proc. Lett. **12**(10), 689–692 (2005)

33.18 H. Hermansky, N. Morgan: RASTA processing of speech, IEEE Trans. Speech Audio Process. **2**(4), 578–589 (1994)

33.19 L. Deng, A. Acero, M. Plumpe, X.D. Huang: Large-vocabulary speech recognition under adverse acoustic environments, Int. Conf. Spoken Language Process. (2000)

33.20 A. Acero: *Acoustical and Environmental Robustness in Automatic Speech Recognition* (Kluwer Academic, Boston 1993)

33.21 P. Moreno: *Speech Recognition in Noisy Environments*, Ph.D. Thesis (Carnegie Mellon University, Pittsburgh 1996)

33.22 A. Acero, R.M. Stern: Environmental robustness in automatic speech recognition, Proc. IEEE ICASSP (1990) pp. 849–852

33.23 J. Droppo, A. Acero: Maximum mutual information SPLICE transform for seen and unseen conditions, Proc. Interspeech Conf. (2005)

33.24 J. Wu, Q. Huo: An environment compensated minimum classification error training approach and its evaluation on Aurora 2 database, Proc. ICSLP **1**, 453–456 (2002)

33.25 D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, G. Zweig: fMPE: Discriminatively trained features for speech recognition, Proc. IEEE ICASSP (2005)

33.26 S. Tamura, A. Waibel: Noise reduction using connectionist models, Proc. IEEE ICASSP (1988) pp. 553–556

33.27 L. Neumeyer, M. Weintraub: Probabilistic optimum filtering for robust speech recognition, Proc. IEEE ICASSP **1**, 417–420 (1994)

33.28 A. Acero, R.M. Stern: Robust speech recognition by normalization of the acoustic space, Proc. IEEE ICASSP **2**, 893–896 (1991)

33.29 M.J. Gales: *Model Based Techniques for Noise Robust Speech Recognition*, Ph.D. Thesis (Cambridge University, Cambridge 1995)

33.30 E.A. Wan, R.V.D. Merwe, A.T. Nelson: Dual estimation and the unscented transformation. In: *Advances in Neural Information Processing Systems*, ed. by S.A. Solla, T.K. Leen, K.R. Muller (MIT Press, Cambridge 2000) pp. 666–672

33.31 P.J. Moreno, B. Raj, R.M. Stern: A vector taylor series approach for environment independent speech recognition, Proc. IEEE ICASSP (1996) pp. 733–736

33.32 B.J. Frey, L. Deng, A. Acero, T. Kristjansson: ALGONQUIN: Iterating Laplace's method to remove multiple types of acoustic distortion for robust speech recognition, Proc. Eurospeech (2001)

33.33 J. Droppo, A. Acero, L. Deng: A nonlinear observation model for removing noise from corrupted speech log mel-spectral energies, Proc. Int. Conf. Spoken Language Process. (2002)

33.34 C. Couvreur, H. Van Hamme: Model-based feature enhancement for noisy speech recognition, Proc. IEEE ICASSP **3**, 1719–1722 (2000)

33.35 J. Droppo, A. Acero: Noise robust speech recognition with a switching linear dynamic model, Proc. IEEE ICASSP (2004)

33.36 B. Raj, R. Singh, R. Stern: On tracking noise with linear dynamical system models, Proc. IEEE ICASSP **1**, 965–968 (2004)

33.37 H. Shimodaira, N. Sakai, M. Nakai, S. Sagayama: Jacobian joint adaptation to noise, channel and vocal tract length, Proc. IEEE ICASSP **1**, 197–200 (2002)

33.38 J. Droppo, L. Deng, A. Acero: A comparison of three non-linear observation models for noisy speech features, Proc. Eurospeech Conf. (2003)

33.39 R.A. Gopinath, M.J.F. Gales, P.S. Gopalakrishnan, S. Balakrishnan-Aiyer, M.A. Picheny: Robust speech recognition in noise – performance of the IBM continuous speech recognizer on the ARPA noise spoke task, Proc. ARPA Workshop on Spoken Language Systems Technology (1995) pp. 127–133

33.40 A.P. Varga, R.K. Moore: Hidden markov model decomposition of speech and noise, Proc. IEEE ICASSP (1990) pp. 845–848

33.41 A. Acero, L. Deng, T. Kristjansson, J. Zhang: HMM adaptation using vector taylor series for noisy speech recognition, Int. Conf. Spoken Language Processing (2000)

33.42 W. Ward: Modeling non-verbal sounds for speech recognition, Proc. Speech and Natural Language Workshop (1989) pp. 311–318

33.43 S.F. Boll: Suppression of acoustic noise in speech using spectral subtraction, IEEE T. Acoust. Speech **24**(April), 113–120 (1979)

33.44 Y. Ephraim, D. Malah: Speech enhancement using a minimum mean-square error log-spectral amplitude estimator, IEEE Trans. Acoust. Speech Signal Process., Vol. ASSP-33 (1985) pp. 443–445

33.45 V. Stouten: *Robust Automatic Speech Recognition in Time-varying Environments*, Ph.D. Thesis (K. U. Leuven, Leuven 2006)

33.46 M. Berouti, R. Schwartz, J. Makhoul: Enhancement of speech corrupted by acoustic noise, Proc. IEEE ICASSP (1979) pp. 208–211

33.47 ETSI ES 2002 050 Recommendation: Speech processing, transmission and quality aspects (STQ);

Amazon EX1010 Page 692

distributed speech recognition; advanced front-end feature extraction algorithm (2002)
33.48 D. Macho, L. Mauuary, B. Noê, Y.M. Cheng, D. Ealey, D. Jouvet, H. Kelleher, D. Pearce, F. Saadoun: Evaluation of a noise-robust DSR front-end on Aurora databases, Proc. ICSLP (2002) pp. 17–20
33.49 A. Agarwal, Y.M. Cheng: Two-stage mel-warped Wiener filter for robust speech recognition, Proc. ASRU (1999)
33.50 B. Noê, J. Sienel, D. Jouvet, L. Mauuary, L. Boves, J. de Veth, F. de Wet: Noise reduction for noise robust feature extraction for distributed speech recognition, Proc. Eurospeech (2001) pp. 201–204
33.51 D. Macho, Y.M. Cheng: SNR-Dependent waveform processing for improving the robustness of ASR front-end, Proc. IEEE ICASSP (2001) pp. 305–308
33.52 L. Mauuary: Blind equalization in the cepstral domain for robust telephone based speech recognition, Proc. EUSPICO **1**, 359–363 (1998)
33.53 M. Afify, O. Siohan: Sequential estimation with optimal forgetting for robust speech recognition, IEEE Trans. Speech Audio Process. **12**(1), 19–26 (2004)
33.54 J. Droppo, A. Acero, L. Deng: Uncertainty decoding with SPLICE for noise robust speech recognition, Proc. IEEE ICASSP (2002)
33.55 M. Cooke, P. Green, L. Josifovski, A. Vizinho: Robust automatic speech recognition with missing and unreliable acoustic data, Speech Commun. **34**(3), 267–285 (2001)
33.56 J.P. Barker, M. Cooke, P. Green: Robust ASR based on clean speech models: An evaluation of missing data techniques for connected digit recognition in noise, Proc. Eurospeech **2001**, 213–216 (2001)

Amazon EX1010 Page 693

34. The Business of Speech Technologies

J. Wilpon, M. E. Gilbert, J. Cohen

With the fast pace of developments of communications networks and devices, immediate and easy access to information and services is now the expected norm. Several critical technologies have entered the marketplace as key enablers to help make this a reality. In particular, speech technologies, such as speech recognition and natural language understanding, have changed the landscape of how services are provided by businesses to consumers forever. In 30 short years, speech has progressed from an idea in research laboratories across the world, to a multibillion-dollar industry of software, hardware, service hosting, and professional services. Speech is now almost ubiquitous in cell phones. Yet, the industry is still very much in its infancy with its focus being on simple *low hanging fruit* applications of the technologies where the current state of technology actually fits a specific market need, such as voice enabling of call center services or voice dialing over a cell phone.

With broadband access to networks (and therefore data), anywhere, anytime, and using any device, almost a reality, speech technologies will continue to be essential for unlocking the potential that such access provides. However, to unlock this potential, advances in basic speech technologies beyond the current state of the art are essential. In this chapter, we review the *business* of speech technologies and its development since the 1980s. How did it start? What were the key inventions that got us where we are, and the services innovations that supported the industry over the past few decades? What are the future trends on how speech technologies will be used? And what are the key technical challenges researchers must address and resolve for the industry to move forward to meet this vision of the future? This chapter is by no means meant to be exhaustive, but it gives the reader an understanding of speech technologies, the speech business, and areas where continued technical invention and innovation will be needed before the ubiquitous use of speech technologies can be seen in the marketplace.

34.1 **Introduction** 682
34.1.1 Economic Value of Network-Based Speech Services 682
34.1.2 Economic Value of Device-Based Speech Applications 683
34.1.3 Technology Overview 683

34.2 **Network-Based Speech Services** 686
34.2.1 The Industry 686
34.2.2 The Service Paradigm and Historical View of Service Deployments 688
34.2.3 Paradigm Shift from Directed-Dialog- to Open-Dialog-Based Services 690
34.2.4 Technical Challenges that Lay Ahead for Network-Based Services 691

34.3 **Device-Based Speech Applications** 692
34.3.1 The Industry 692
34.3.2 The Device-Based Speech Application Marketplace 692
34.3.3 Technical Challenges that Enabled Mass Deployment 693
34.3.4 History of Device-Based ASR 694
34.3.5 Modern Use of ASR 695
34.3.6 Government Applications of Speech Recognition 695

34.4 **Vision/Predications of Future Services – Fueling the Trends** 697
34.4.1 Multimodal-Based Speech Services 697
34.4.2 Increased Automation of Service Development Process 699
34.4.3 Complex Problem Solving 699
34.4.4 Speech Mining 700
34.4.5 Mobile Devices 701

34.5 **Conclusion** 701

References 702

Amazon EX1010 Page 694

34.1 Introduction

In 1969, the influential John Pierce wrote an article questioning the prospects of speech recognition, criticizing the *mad inventors and unreliable engineers* working in the field. In his article, entitled *Whither speech recognition?*, Pierce argued that speech recognition was futile because the task of speech understanding is too difficult for any machine [34.1]. Such a speech understanding system would require tremendous advances in linguistics, natural language, and knowledge of everyday human experiences. In this prediction he was completely correct. Furthermore, this ultimate goal is still not within sight in 2006. Pierce went on to describe the motivation for speech recognition research: 'The attraction [of speech recognition] is perhaps similar to the attraction of schemes for turning water into gasoline, extracting gold from the sea, curing cancer, or going to the moon.' His influential article was successful in curtailing, but not stopping, speech recognition research.

What Pierce failed to foretell was that even limited success in speech recognition would have interesting and important applications, especially within the telecommunications industry. In the decades since Pierce's article, tens of thousands of speech recognition systems have been put into use, and have proven to be important enabling technologies. From speech research to speech engine products, delivery platforms, application development, application tuning and maintenance, and service hosting – speech technologies are responsible for generating new revenue streams and vast cost savings. Players range from small startups and universities to Fortune 500 companies – each with its own particular focus on the revenue chain created by a group of technologies.

In this chapter, we present a view on the applications and services that are driving the need for speech recognition and natural language understanding technologies with a focus on those opportunities that provide the broadest value to people and businesses. Current and future needs are described and examined in terms of their strengths, limitations, and the degree to which stakeholder needs have been or have yet to be met. We start with brief high-level business and technical overviews. Next we drill down into the specific opportunity spaces in several markets – Sect. 34.1 network-based, Sect. 34.2 mobile devices, Sect. 34.3 government, and Sect. 34.4 new emerging markets including multimodal and multimedia applications. And finally, we present our view on where we believe the speech industry is heading. This chapter is not meant to be exhaustive, but to give the reader an understanding of speech technologies, the speech business, and areas where continued technical invention and innovation will be needed before the ubiquitous use of speech technologies can be seen in the marketplace.

34.1.1 Economic Value of Network-Based Speech Services

America's increasing mobility on and off the job lead users to demand access to automated services anytime and anywhere using any access device. The selection of user interface mode or combination of modes depends on the device, the task, the environment, and the user's abilities or preferences. As such, industry demand has been increasingly pointed toward a wide range of voice-enabled services, such as:

- Consumer communication (voice dialing, unified messaging, voice portals, and access to services for people with disabilities)
- E-commerce (business-to-business and business-to-consumer)
- Call center automation (help lines and customer care)
- Enterprise communication (unified messaging and enterprise sales)

Network-based voice-enabled services imply scale. That is, the ability to automate parts of or entire transactions from billions of calls per year provides the opportunity to radically change the cost structure of companies' telecommunications and call center operations. Speech-enabled automation for network services can vastly enhance user experience while optimizing revenue potential and cost savings for businesses; hence their increased popularity.

Decreased Cost

Robust automation of call center functions radically reduces agent counts and operational overhead, while innovative, scalable technology reduces servicing costs. Current call centers, for example, are characterized by fragmentation of channels, limited customer insight, and high dependency on live agent interactions, resulting in high costs, and inconsistent customer experiences. Dramatic success can be achieved by utilizing speech-based services. For example, call center human agents can cost

5.50–12.00 $ per call , while speech-enabled automation can cost less than 0.20 $ per call and are 40% faster on average.

Network-hosted services reduce barriers to entry by limiting the capital spending and the ongoing maintenance costs for companies. In addition, given the complexity and maturity level of speech technologies and services, it is increasingly more difficult and expensive for individual companies:

1. to be responsible for maintaining up-to-date technologies on their platform, and
2. to be able to design ever more-complex speech solutions to meet their call center needs.

Improved Customer Experience

As customers experience more-effective natural, conversational, and personalized interaction that understands and fulfills their requests, they begin to accept automation. Wait and call times are reduced even during peak periods because of improved correct routing rates, increased call completion rates, and a reduction in the rate of agent transfers. This results in happier customers.

Revenue Generation

Redefined business models exploit new revenue opportunities with personal and customized services. New services such as information search and retrieval services (e.g., access to information available on the Internet), are not viable in a mobile environment without the use of speech technologies. Additionally, as will be discussed below, speech technologies provide the ability to simplify complex interactive voice response (IVR) menu structures. Therefore, businesses can offer an increased number of automated service opportunities (e.g., e-commerce) than could otherwise be achieved using touch-tone interfaces.

34.1.2 Economic Value of Device-Based Speech Applications

Small devices have been a challenging platform for speech systems because of their limited computing capability, short-lived batteries, imperfect audio systems, and small audiences. Starting with the advent of the cell phone is the mid 1990s, all of these constraints are lifting.

The number of cell phone users has exploded worldwide. The cell phone is an essential part of today's communication infrastructure in the developed world, and the cell-phone infrastructure is invading the developing world because of the ease of deployment and the economies of scale due to tremendous volumes. Audio systems are carefully crafted in these devices because of the essential part that voice quality plays in telephony. Battery life has increased both because of technical advances, and due to engineering discipline caused by consumer-led product analysis. Finally, computational abilities are growing rapidly due to the demands for high-bandwidth connectivity, multimodal use of devices, and applications ranging from browsers to music players.

While there are improvements in mobile enterprise systems and personal digital assistants (PDAs), the economic gorilla in this space is, and will remain, the cell phone and associated smart phones and feature phones. (Smart phones have open operating systems, while feature phones appear to have the capabilities of smart phones while generally retaining the proprietary operating systems of their consumer cell-phone counterparts.)

The value of the cell-phone market is immense. Gartner predicts that in 2009, 1 billion cell phones will be sold, at an average price (not necessarily to the consumer) of 174 $. Given that speech services will be available on about half that number (say 500 million), and that second-tier (nonprimary) software services can command 1–2% of the value of the devices in which they are embedded, we estimate the economic value of the speech market for embedded devices to be at least 870 million $ per year.

Note that cell phones are replaceable items with an expected 2.5 year lifespan, so this economic market will continue. Specialized services (games, dictation, or other high-value applications) can add substantially to the value of speech services, so our estimate should be treated as a lower limit. In short, the embedded speech market is about a billion dollars a year.

34.1.3 Technology Overview

Technology Landscape

Progress in speech and language technologies along with advances in computing and electronic devices over the past 50 years have resulted in a spectrum of innovative human/computer applications and services. The field has been transformed from the simple speaker-dependent isolated-digit recognition systems that were demonstrated in 1952 by *Davis*, *Biddulph*, and *Balashek* of Bell Labs [34.2], to today's very large-vocabulary speaker-independent context-dependent, IBM continuous speech recognition and understanding systems as demonstrated by AT&T, BBN,

Cambridge, Carnegie Mellon University (CMU), among others [34.3–7]. This steady progress in acoustic and language modeling, robust feature extraction, search and network optimization has enabled speech-based applications to advance from the simplest command-and-control applications deployed in the 1980s, to those in use today for network services and speech data mining that require upwards of 1 M word vocabularies. A few of the key technology innovations include:

1. Acoustic/phonetic: initially studied by *Davis* et al. [34.2] to extract salient spectral features from filter-bank analysis that can better correlate phoneme sets, such as vowels.
2. Linear predictive coding (LPC): developed by Itakura and Atal in the late 1960s for speech coding, and has been shown to be central for extracting features, such as cepstrum, from speech [34.8].
3. Dynamic programming (DP): proposed by *Vintsyuk* [34.9] for the alignment of two speech utterances of different lengths.
4. Hidden Markov models (HMMs): proven to converge by Baum, Petrie, Soules and Weiss, and later developed by *Jellink* and *Baker* at IBM [34.8,10] and popularized by Larry *Rabiner* at Bell Labs [34.11] for modeling speech – a departure from template-based methods that were previously proposed by Bell Labs. The combination of HMMs and Gaussian mixture models (GMMs) are the heart and soul of most of today's speech recognition systems.
5. Stochastic language models (SLMs): proposed by *Jellink* at IBM [34.12] for language modeling in continuous speech recognition.
6. Discriminative training (DT): methods including maximum mutual information (MMI) and minimum classification error training (MCE) that aim to minimize the expected recognition error rate [34.13].
7. Large-margin classifiers: methods including support vector machines (SVMs) and boosting, which attempt to maximize the margins between correct and incorrect classes, have proved to be essential for spoken language understanding [34.14].
8. Neural networks: developed for speech recognition at Institut Dalle Molle d'Intelligence Artificielle Perceptive (IDIAP), International Computer Science Institute (ICSI) and Oregon Graduate Institute of Science and Technology (OGI) by Bourlard, Morgan, and Hermansky, these nonlinear classifiers proved useful in providing rapid probability estimates for use in HMM decoders, and allowed the use of a posterior probabilities rather than likelihoods in search.
9. Digital signal processing: widely practiced by electrical and communications engineers worldwide, this opened the field of speech for moderate-computation front-end processing using both fast Fourier transform (FFT) and multiscale front-end processors.
10. JAVA, Windows Mobile CE, and BREW: these operating system extensions allowed third-party applications to be added to both standard handsets and smart phones, thereby accelerating innovation in mobile platform offerings.
11. VTLN (vocal-tract-length normalization): a one-parameter transformation which accounts, to first order, for varying vocal-tract lengths between speakers, was first noted by Helmholtz, but was shown viable during the Rutgers summer workshop in speech recognition by Andreou, Cohen, and Kamm in 1994.

From Knowledge-Based to Data-Driven Systems

Creating the underlying acoustic, language, and understanding models for speech applications has historically been a task of finding a set of rules that maps distinct features, such as spectral features, key phrases, etc., into predefined classes, such as words, phonemes, etc. For example, to map acoustic cues into digits, *Davis* et al. [34.2] provided a set of rules that relate filter-bank spectral features into vowels; to train a language model for speech recognition, a set of rules was identified in the form of regular expressions that describe all the possible variations on how users may be expected to interact with the application; to build language understanding models, *Gorin* et al. [34.15] provided a method for routing calls by mapping a set of phrases into meanings. If a user says 'I want to register', for example, then the salient phrase 'want to register' maps the input request into a target class *registration*, enabling the caller to register [34.16].

Although knowledge-based systems have led to a series of successful applications including directory assistance and travel reservation, these systems have been found to be both brittle, since they reflect what the designer is expecting the user to say, and expensive to create and maintain, since they require extensive manual effort to craft the grammars and rules. Over the past two decades, there has been a migration from rule-based systems to data-driven, or statistical, systems. Data-driven systems rely on machine learning algorithms to model a set of features into a set of classes. Therefore, the models can reflect what the users say as opposed to

what designers believe what users may say. Statistical models have been shown in both government applications and large-scale industrial applications to be both robust, since they provide a rich representation for modeling various effects, as well as efficient, since they are easy to build once data becomes available.

The migration to statistical models led by the availability of speech and textual data has revolutionized speech recognition and understanding research. The famous saying *there is no data like more data* has been a key driver for improved speech recognition performance. State-of-the-art recognizers are currently trained with over 1000 hours of speech and several tens of millions of words, compared with just a few hours of speech over a decade ago. This migration to statistical models has also led to a new wave of advanced speech and language processing applications. IBM and Dragon [34.17] captured the speech dictation market in the 1980s by demonstrating speaker-dependent systems based on stochastic language models and hidden Markov models that can recognize several thousands of words. AT&T began a trial in the mid 1990s which applied statistical classification models for contact center call routing [34.18]. Companies such as *Nexidia* and *Verint* [34.19, 20] are today providing solutions for contact center analytics, applying statistical models for audio search of large volumes of conversational speech.

The State of the Art

Over the past decade, there have been numerous studies on improving speech recognition accuracy [34.4,21–23]. Better training and adaptation methods have been on the forefront of research. Several research institutions have begun to apply discriminative approaches that can indirectly reduce the word error rate as opposed to the traditional maximum-likelihood methods that aim to increase a likelihood function. Examples of such methods include linear discriminative analysis (LDA), which is essentially a linear transformation of the acoustic features (or the model parameters) in such a way to increase the ratio of inter- to intraclass distance. Other methods include MCE and MMI training methods. These discriminative model training methods have been fairly successful in improving word discrimination by increasing the separation between the correct hypothesis and an alternative hypothesis based on N-best methods or some form of a word lattice [34.6].

Although word accuracy has been the most dominant measure used to evaluate speech recognizers, deploying speech recognition engines typically imposes several other technical challenges. These challenges, which have been the subject of significant research over the past two decades, are detailed below.

Efficiency. Having faster decoders with a small memory footprint is essential for supporting large-scale speech recognition applications in the network and limited vocabulary speaker-dependent speech applications on handheld devices. Advances in this area have been possible due to a numerous set of algorithms including the use of finite state transducers that enable efficient representation of the recognition network through general composition and determinization algorithms in both time and space [34.24]. Other methods include those that attempt to reduce the likelihood computation by minimizing the number of Gaussian computations. Such methods include Gaussian selection [34.25], which works by ignoring the Gaussians that contribute minimally to the state likelihood.

Robustness. Creating speech recognition systems that are invariant to that extraneous background noise, channel conditions as well as speaker and accent variations that are inherent in large-scale deployments of speech services in a major challenge and has been the subject of significant research over the past two decades. The basic problem is that adverse conditions as well as speaker differences cause deterioration in recognition performance. For example, applying models trained on wire-line speech to wireless applications typically results in 10–20% absolute degradation in word accuracy. This level of degradation causes many applications of speech technology to be unusable. Advances in this area include the use of cepstral normalization methods, such as cepstral mean subtraction or signal bias removal [34.25], which aim to remove the component of the cepstrum that is mostly sensitive to channel variations. Other methods include vocal-tract-length normalization (VTLN) [34.22], which reduces speaker differences attributed to variations in their vocal-tract length. VTLN works by computing a factor that enables compression/expansion of the spectrum. In theory, this is equivalent to downsampling/upsampling the speech signal. Other robustness methods include maximum-likelihood linear regression (MLLR), which adapts model parameters towards a speaker or an environment.

Operational Performance. Although many speech recognition applications demand a trade-off between speed and accuracy, the operational performance of these applications depends on additional measures such

as latency, end-pointing, barge-in, rejection, and confidence scoring. These measures affect the application user experience. Barge-in, which is based on using statistical methods to determine whether speech is present or not, enables users to speak while the prompt is playing. Rejection and confidence scoring provide measures of *goodness* of the recognized speech. These methods are essential for rejecting out-of-vocabulary words and for providing the application with a reliability measure of the recognition output. Although methods for improving the operational performance of speech recognition applications have traditionally been rule based, state-of-the-art systems rely on statistical models to provide an additional degree of robustness.

Multilingual Processing. The migration of speech recognition technology towards statistical methods has resulted in portable methods for multilingual processing. LIMSI's research shows that methods that work well for English seem to work just as well for other languages, especially for European languages. Nevertheless, achieving robust high-accuracy multilingual speech recognition requires continued research to handle limited training data, to deal with tonal languages, and to provide word pronunciation dictionaries. These advances are essential to provide both industry and the military with necessary technologies to support various applications including surveillance, audio mining, contact center automation, and speech translation [34.26].

Learning. The availability of large amounts of data is a key driver for the timely creation of cost-effective speech-based applications. This data, whether in audio or text form, is not only an essential source for improving the underlying speech and language technologies, but it is also critical for identifying problematic areas within a service that may need immediate attention. Logging, transcribing, storing, and managing this data, however, is often one of the most difficult tasks when scaling voice-enabled IVR services.

As technologies continue to transition to becoming more statistical, relying more heavily on speech data as opposed to knowledge rules, the quality and transcription accuracy of this data will continue to play a central role in the success of deployed services. In addition, improving the speed at which a dialog service is developed, deployed, and maintained is essential not only for scaling of these applications, but also for providing product differentiation.

AT&T has created a general learning framework based on daily feeds of speech service data to enable them to perform three learning operations [34.27, 28]:

Active Learning. Minimize the effort needed to label data by automatically identifying only those interesting or problematic dialogs that, when manually transcribed and labeled, would result in a significant improvement in system performance.

Active Labeling. Automatically identify inconsistencies in the manually labeled data. This process is applied for detecting outliers and incorrectly labeled data.

Active Evaluation. Automatically tracks trends in the performance of voice-enabled services to generate automatic alerts to indicate when to update or adapt a service.

In addition to these research advancements that have enabled widespread deployment of speech recognition applications, it is important to note that an additional important factor in creating usable speech applications is the user experience (UE) design [34.29]. UE can be considered as the technology front-end and plays an essential role in the success of voice applications, especially during problematic situations caused by system failures or by users not providing concise or accurate information. Poor UE design can very easily overshadow the best technologies. Conversely, good UE design can make up for deficiencies in poor technologies. The combination of statistical and robust methods for voice recognition and understanding as well as good UE design is what results in natural and intelligent voice applications.

34.2 Network-Based Speech Services

34.2.1 The Industry

Little did Alexander Graham Bell and Thomas A. Watson know in 1875 that the *harmonic telegraph* would spawn industry after industry, revolutionizing the way people do business.

One such broad industry is interactive voice response (IVR) systems for business call centers and consumer

services, which provides an automated communication mechanism between people (customers, employees, suppliers, partners, stakeholders, etc.) and an organization (businesses, enterprises, government agencies, etc.) to accomplish or facilitate a task (customer care, information request, and access to messaging and telephony services such as voice dialing).

Initially dominated from the 1970s through the 1990s by telecommunication and computer giants worldwide (e.g., AT&T, Lucent, ITT, Nortel, NTT, BBN/GTE, IBM), starting in the 1980s small niche startups that were nimble and singularly focused began to emerge in the marketplace. The characteristics of these companies were that they could react more quickly to market demand than big industry could. The main focus of these companies was to provide simple voice-enabled services to the call center industry. Voice automation of call center services and old-fashioned touch-tone-based IVR services began to increase their acceptability in the marketplace.

The industry has now matured into a highly diversified *food chain* that encompasses hundreds of equipment suppliers (for example, automatic call distributor (ACDs), private branch eXchange (PBXs), media servers, routers, phones, PCs, application servers), software suppliers (for example, operating systems, browsers, agent management systems, routing engines, speech engines, and computer telephony integration systems), system integrators, service providers, application developers, and user-interface designers.

The first voice-enabled service, automatic answer network system for electrical request (ANSER), was offer by Nippon Telephone and Telegraph (NTT) in Japan [34.30]. The service was deployed in 1981 and used very small-vocabulary isolated-word speech recognition to automate customer access to banking services typically handled through personal contact with human agents. Today, it has been estimated that US companies spend well over 120 B $ annually on business calls centers to deliver services to their customers, suppliers and employees. According to the Yankee Group, approximately 80% of that cost is spent on agent personnel costs, including direct expenses such as salaries, and indirect expenses such as benefits, administration, and buildings (Fig. 34.1). Automating even a fraction of the calls currently handled by live agents will generate tremendous cost savings.

On the consumer side, network services began entering the marketplace in the 1980s, enabling people to better access information and personal communications services. Since then, voice-enabled personalized communications services have expanded to include voice dialing, voice access to messaging (both voice mail and e-mail), and access to general information such as news, weather, sports, stocks, directory assistance, etc. In today's web-based nomenclature, we could call this group of services – *voice portal* services. These services provide consumers with simple, easy-to-use access to the services and information they use everyday from wherever they are.

Seizing a real business opportunity for speech technologies, many companies rushed into the market to provide speech recognition engine components. Such companies included, Verbex, Threshold, Centigram, Kurzweil, L&H, Interstate, Scott Instruments, Phillips, VCS, and Dragon. These products sold for hundreds and even thousands of dollars per engine, and were capable of recognizing isolated words with vocabularies ranging from two to a few thousands of words. While services based on speech technologies generated large cost savings for businesses that deploy them, few, if any of the speech vendors generated profits from selling speech engines. Therefore, over the years most of these companies merged with other companies or just disappeared.

In the mid 1990s, during the Internet boom, two new speech vendors came to the marketplace that provided much needed energy and marketing savvy to the industry – Speechworks (started as *ALTech* by speech scientists from MIT), and Nuance (started as *Corona*, a spin-off from SRI). As these companies grew, so did the voice-enabled services industry as a whole. Their focus, as with the early pioneers, was on providing speech technologies to the call center business market. However, both companies realized early on that providing only speech engine products to their customers was not a sufficient business model. While speech science has come a long way, there was still much to do to support a viable industry. In particular, the process of creating a voice-enabled service that is acceptable to consumers today is as much an *art* as a science. Hence, both companies invested heavily on providing a wide range of service development and monitoring tools, reusable service templates, and professional services (e.g., user-interface design support) that helped their clients build and maintain world-class services. This was, and continues to be, absolutely essential to the widespread success of speech recognition applications within the business community.

Over the decade since their entry into the market, Speechworks (latter merged with ScanSoft) and Nuance have been responsible for hundreds, if not thousands, of successful network-based voice-enabled services. While both of these companies succeeded greatly in helping

Amazon EX1010 Page 700

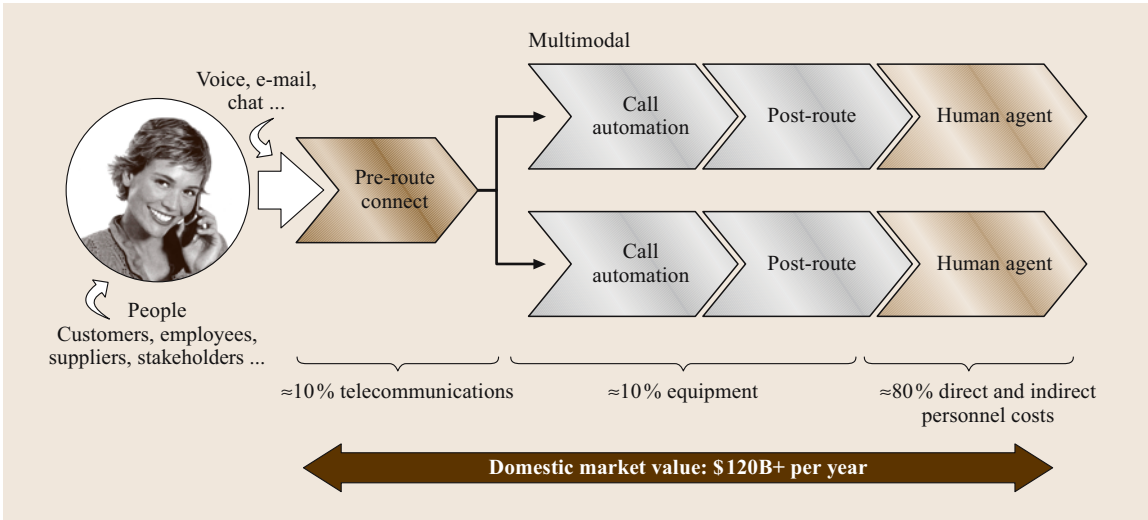


Fig. 34.1 Call center costs

make speech recognition a household name, in 2005 both were merged into a single company – called Nuance [34.31].

Over the past decade the market has differentiated further with the creation of companies that host services, but do not create or sell speech engines or platforms (e.g., Tellme , beVocal, TelUreka, Convergys, and West) [34.32–36]. These companies are very flexible in meeting market needs for quick service creation and deployment, yet are completely dependent on others (e.g., Nuance, IBM, Microsoft, and AT&T) for speech technology innovation.

34.2.2 The Service Paradigm and Historical View of Service Deployments

It is important to bear in mind that today the speech technologies described above and elsewhere in this Handbook remain error-prone and will remain so for the foreseeable future. For this reason, successful network-based voice-enabled services have been, and continue to be, those that have the following important characteristics.

Simplicity. Successful voice-enabled services are natural, easy to use, and intuitive to the user. Additionally, they provide real value to the businesses that deploy them.

Evolutionary Growth. Most applications introducing speech recognition were extensions of pre-existing IVR services. For example, speech recognition as a touch-tone replacement for IVR menus. Therefore, the users were familiar with the services and generally accepted them.

Tolerant of Errors. Given that speech recognizer will make mistakes, the inconvenience to the user must be minimized. This means that careful design of human–machine interaction, (e.g., the call flow) is *absolutely* essential in providing reliable and acceptable services. Hence, a demand for skilled user interface designers has occurred over the past decade.

This being said, there are some general questions that one must ask when considering an application using current speech technologies. The answers to these questions will help determine whether it is advisable or possible to design a quality application using speech technology. Some of these questions include:

- Are the potential users predominantly *power or novice users*? Power (or repeat) users generally desire a very simple design, i. e., the customers wants to get in and get out as fast as possible. Examples, in this space would include stock quotes and voice dialing. However, novice users require hand-holding (e.g., more instructions) and error-handling capabilities to enable them to achieve their goal effectively. This would be the case for most business-related customer-care IVR services.
- What environment will the user be speaking in (e.g., a noisy airport, quiet home, VoIP on a PC)? While advances to the robustness of speaker recognition systems have occurs over the past decade, robustness

still remains one of the key challenges for speech scientists today.
- What performance will a user of this service tolerate, and is speech recognition accurate enough to meet those expectations? This is critical in setting the expectations of all stakeholders, e.g., the business deploying the service and the customer using the service.
- Does the benefit of using speech technology in this application outweigh its cost, compared to alternative technologies? This is obviously critical in developing viable business cases for deploying a voice-enabled service.

Over the years, as more and more voice-enabled services have been development and deployed, speech scientists and engineers have gained a much richer understanding of the way people need to use speech technologies to positively affect the usability of many network-based services. For example, when people call a call center, there is a problem that they want solved or some information that they want. In addition, they often do not even know the *name* of the problem they are trying to solve. They generally only know the symptoms of the problem – for example, *I got a new bill and there is something on it that I don't understand*, or *I want instructions on faxing prescriptions in to a pharmacist*, or *Can I claim my daughter as a dependent on my tax return?* So, when we think of truly automating services with speech technologies there are two necessary goals that must be achieved:

- Determining *why* the customer is calling,
- Fulfilling the request to complete a transaction.

From the 1980s to today, given the state of the art in speech technologies, addressing the *why* question was generally achieved by asking the caller to provide the answer by navigating (via spoken command words or touch tones) a sequence of menu choices. In these system-controlled *directed dialog services* (i. e., guiding the caller step-by-step for information), the burden is placed squarely on the users to convert the thoughts in their mind to the physical actions presented to them at every level of the IVR menu. Users had to correctly navigate through a maze of complex menus to get the service they want. That is, users had to tailor their interaction to fit the capabilities of the service and technologies, not the other way around.

The first class of rudimentary or primitive directed-dialog-based applications that began to appear in the late 1980s and early 1990s were those services that used speech technologies to replace functions that human operators/agents carried out within call centers, thus reducing the cost of doing business for companies. Such *low-hanging-fruit* applications were able to make use of the then state-of-the-art capabilities in small-vocabulary isolated-word speech recognition techniques, and innovations in word spotting and barge-in technologies [34.37,38] to provide value to businesses – saving billions of dollars by reducing costs of providing services to customers. Examples of such pioneering applications included:

- Automation of operator services, deployed by many telephone companies including AT&T's voice recognition call processing (VRCP) service (1992) [34.39], Northern Telecom, Ameritech (now AT&T) and New Jersey Bell (now Verizon),
- Automation of directory assistance by Nynex (now Verizon) and Northern Telecom
- Control of network fraud developed by Texas Instruments (TI), Sprint and AT&T,
- Automation of banking services offered in 1981 by NTT in Japan,
- Reverse directory assistance in which a user enters a telephone number to retrieve a name and address, provided by Ameritech/AT&T and Bellcore (now Telcordia) in 1992,
- Information access services, such as a stock quotation services trialed at BNR in 1993.

Throughout the mid 1990s and early 2000s, network-based voice-enabled services established themselves as mainstream in the IVR industry. Applications that aimed to reduce the cost of doing business have completely dominated the industry and provided much of the justification for continued research and development for speech technologies. Over this time period, advances in processing and memory capabilities and in speech technologies themselves have enabled new applications to be deployed that require larger vocabularies (i. e., thousands of words), complex grammatical structures with continuous speech recognition, and more-complex directed dialog design. Such applications, which have provided real value to both business and their consumers, are now pervasive throughout the IVR industry in several business sectors including:

- Telecommunications
- Banking, finance, and insurance
- Retail sales
- Energy
- Education

- Travel
- Healthcare
- Government

Across these industries there has been some common themes on the types of applications deployed that provide both a value-added service and that can be supported given the limitations in capabilities within speech technologies today. This commonality has brought a high degree of reuse of knowledge across industries, and thus has enabled quicker and lower-cost development of voice-enabled services. Examples of horizontal applications that cut across industries include:

- Customer care routing (i. e., routing a customer's call to the proper subapplication via speech commands)
- Corporate and white-pages directory assistance (e.g., speech-based *411* for name and address listing retrieval)
- Corporate information access (e.g., account balance, order status, flight arrival/departure information, tracing shipments, locating retail stores, resetting passwords)
- Personal information access (e.g., voice dialing, access to messaging, stock quotes, news, weather, sports)

34.2.3 Paradigm Shift from Directed-Dialog- to Open-Dialog-Based Services

Directed-dialog services have been the mainstay of the voice-enabled services market since the mid 1980s. The main characteristics and assumptions of successful directed-dialog-based services have been:

- They consist of a small number of easily understood user options with a simple easy-to-follow call flow.
- The customer is not vague with his/her spoken input.
- The application does not need to combine statements of intent with a lot of additional information within the same utterance (that is, extra information that goes along with an intent, for example, *Gee, I'm having a problem with my March bill*).

However, directed-dialog design imposes several limitations on network services that are obviously designed to work for millions of consumers. The biggest is that people tend to use their own words or phrases in describing a problem, and they are often vague in their request and need the opportunity to clarify their intentions. They sometimes change their minds in the course of a dialog, mix all kinds of information together, and their meaning often depends on the context of where there are within the conversation. In the end, depending on the specific service, a business's end-users *self-misroute* themselves through their existing directed-dialog-based IVR 20–60% of the time. Simply stated, people find it hard to navigate menus. Users cannot change their minds/context, recover from mistakes, skip steps, or add relevant information. In the end, customers are frustrated. This should not be a surprise to anyone who has used an IVR service. So, while directed-dialog-based services have been adequate to create the voice-enabled services market over the past couple of decades, they are too inflexible and limiting to allow the market to move forward and become ubiquitous in network applications.

In the 1990s, the Defense Advanced Research Projects Agency (DARPA) began a program to support open-dialog research. The air travel information system (ATIS) was an extensive project aimed at extending our knowledge in speech recognition for *simple* services to those requiring more-complex interactions. Universities, such as MIT and industrial labs such as BBN, IBM, and AT&T were the first to use natural-language understanding and simple dialog management to enable services to have simpler, easy-to-use interfaces to services. Research in open-dialog-based call routing services, led by AT&T, over the past decade has focused on spoken natural-language understanding (NLU) and dialog management (DM) technologies for call classification, whose function, along with large-vocabulary speech recognition, is to determine first, *why* a person is calling, and second to fulfill their needs through a conversation. The innovation of open-dialog-based services enables a natural interaction between a customer and an automated service, therefore simulating human-to-human-like interactions. That is, open dialog services shift the burden of determining how to use a service from the user to the service and associated speech technologies. These technologies, while challenging to the speech and AI Research communities, will enable the voice-enabled market to explode – breaking through the barriers created by menu-based services.

AT&T's research formed the basis for AT&T's customer care system called *How May I Help You* (HMIHY), which was deployed in the network initially in 2000 by AT&T consumer services [34.18]. The HMIHY service, when fully operational throughout the US, handled about 3.5 million voice dialogues with customers every month. This service was the first open-dialog spoken natural-language system to be fully deployed in the world. It significantly improved cus-

tomer satisfaction as represented by four measures, namely:

- Decreased repeat calls by 37% (customers who cannot navigate touch-tone menus hang up and redial)
- Decreased the rate of customers switching to another long-distance company by 18%
- Decreased customer call agent times by 10%
- Decreased customer complaints by 78%

Following this success, in 2002 AT&T launched its VoiceTone IVR service, which took advantage of the new advances in natural-language understanding and dialog management. Since then, many open-dialog-based services have been developed and deployed. Successful characteristics of these services fall into several categories, including:

- Automation of new customer transaction services that previously had been *off-limits* due to limitations in traditional menu-based touch-tone and simple speech recognition technologies.
- Increased *self-service* completion rate.
- *Improved navigation and content delivery* and a reduction of misdirected calls, as well as substantially shortened call times.
- A reduction in customer callbacks resulting in increased customer satisfaction. This will positively impact the client's ability to meet their objectives and improve their customer retention rates.
- Customer perception of a more-personal experience through the use of both external sources of information (for example, account information and attributes, offer information and attributes, and business-targeted outcomes and goals) and through flexible interaction with the customer (for example, contact conditions and attributes, caller intent, and response).

The following interaction is an example of an open dialog interaction possible through VoiceTone.

VoiceTone:	*Hello, this is Barney Health Care, how may I help you?*
User:	My medicine ran out last week.
VoiceTone:	*Would you like to refill your prescription?*
User:	Yep uh my refill I.D. number is PB14720.
VoiceTone:	*Okay, I have ordered for you a new prescription. Is there anything else I can help you with?*
User:	No thanks.
VoiceTone:	*Thank you for calling Barney Health Care. Goodbye.*

34.2.4 Technical Challenges that Lay Ahead for Network-Based Services

History aside, the true vision of automating network-based services is to provide the user with access to services independent of the user's communication device or location. That is where speech technologies become necessary. Advances of speech technologies in several dimensions have been and will continue to be essential to the viability of deploying automated services in the future. That is, *we have not solved the problem*. Amongst the key challenges that must be addressed are

1. speech technologies are still too fragile to use
2. too expensive to deploy
3. too hard for developers to use to design network services

Improving Performance. Performance is important not only for meeting the business success criteria but also for ensuring high-quality customer experience.

In the area of performance, improving the robustness and accuracy of speech recognition, the breath of natural-language understanding beyond call classification, the naturalness of text-to-speech synthesis, and the flexibility of dialog management are subjects of current focus. This area represents the bulk of the research over the past few decades and is important for meeting business success criteria and ensuring high-quality customer experience.

Speeding up Service Deployment Through Automation and Reuse. Automation is critical for reducing the development effort during both creation and maintenance of these services.

For increased automation, novel self-learning algorithms for reducing overall development resources during the creation of natural-language applications are being applied. In particular, they use the *active* family of methods, which includes *active learning* for reducing transcription and labeling effort, *active labeling* for ensuring high-quality labeled data necessary to create natural-language understanding models, and *active evaluation* for evaluating and adapting models automatically with no human intervention. Automation in these technical areas is critical for reducing the development effort during creation and maintenance of these services.

Increased Scalability. Scalability is a key component of the mass deployment of cost-effective natural-language dialog services. Improving the efficiency of speech algorithms and their implementation (i. e., speed, memory, and size), leveraging standards, and exploiting reusable data and libraries is the focus of much research.

Adoption of Technology and Application Standards. Understanding the need to scale application development, in 1995, AT&T created the phone markup language (PML) application program interface (API) for rapid creation of interaction voice response services. In 1999, PML was released with the founding of the voice extensible mark-up language (VXML) forum by AT&T, Lucent, Motorola, and IBM, responsible for developing and promoting the VXML standard currently used industry wide. Today, VXML is the standard environment for development of most IVR services worldwide. Going forward the W3C voice browser group is working on expanding the capabilities of VXML to include support open-dialog services, and especially services that require multimodalities of the input devices, such as voice, touch, writing, etc. This will drive the effort of unifying network-based telecommunications services with web- and device-based services into a single environment that makes it easy for application developers to create new and innovative services [34.40, 41].

Additionally, standard interfaces, such as the media resource control protocol (MRCP) [34.42], that allow for rapid integration of new speech engines into service delivery platforms have begun to emerge into the marketplace. Such standards enable improvements and advances in speech technologies to find their way more quickly into the field.

34.3 Device-Based Speech Applications

34.3.1 The Industry

Portable devices such as cell phones offer a unique opportunity for speech recognition technologies, either by themselves or as part of a multimodal interface. If these devices communicate wirelessly, they tend to support both IP networking and standard cell-telephone communications, and thus the devices support both telephone interchanges and web-based data exchange.

It has always been challenging to design an interface on a small device that enables fluent use of the device, while allowing full access to its capabilities. For a cell phone, the 12-button interface is a copy of the touch-tone phone keypad, which itself is a representation of the 10-hole *dial* ring of the original telephones. This interface was designed in a time when the phone number was the only critical information that the system required, and it was very efficient at delivering that information to the network.

With the advent of new services, a user often wants to input text, video, pictures, or other nondigit information. While the keypad is useable for many purposes, typing a *7* four times to produce an 'S' is tedious for all but the most dedicated teenage text messenger.

The interface on a standard PDA device is even more limited, although it can be assisted by the touch-sensitive screen, which often accepts handwriting or stylus inputs. It is possible to build a completely graphical operating system on this type of device, and early devices were button-poor boxes with touch-sensitive screens. While these devices were successful in the market as personal information managers (Palm controlled this market for many years), new devices have become less successful because they violate the *one-hand* usability criterion for mobile devices – that is – if you cannot use it with one hand, then you cannot use it while walking, driving, or carrying something, and thus the device is much less useable than it would otherwise be.

Speech has always been a natural command-and-control information exchange, at first between people, and then for people to control machines. This approach has come late to mobile devices because of the limited capabilities of the devices, the abominable design of early PDA acoustic systems, and the limited insight of the technological community.

34.3.2 The Device-Based Speech Application Marketplace

Embedded speech recognition is a relatively recent phenomenon. The markets for these devices range widely, but are evolving as devices evolve.

Early applications were mobile-phone dialing. AT&T was instrumental in designing and fielding a speaker-dependent digit and name recognizer with their automobile-based mobile telephone in the mid 1980s [34.42]. While the design included both digits

and names, the devices which were built and marketed contained only name recognition. The adventure was not a marketing success.

Mobile phones (remember the *bag phone*?) were relatively rapidly supplanted by smaller cell phones in the late 1990s and early 2000s. As computation became possible in smaller platforms, and battery technology improved, it became possible to build digital cell phones that were *luggable*, and, in the past half decade, which have become pocket devices. These communications instruments now have substantial computing capability, and are often multiply connected, using the digital cell-phone channel, a side-channel for text (SMS), and often a separate IP network and even a wireless fidelity (WIFI) network. While early cell phones had limited capacity, as their computing capabilities improved, speech researchers discovered that dynamic programming methods allowed digit and name recognition to be integrated into these devices [34.43]. These techniques were offered by AT&T, Motorola, NEC, Siemens, Qualcomm, and many other laboratories. This plethora of *speaker-dependent* applications were widely deployed, but failed the market usability test, as they were too complicated for the blinking-12 generation [34.44]. Name recognition for voice dialing is embedded in more than 40 million cell phones, and is widely ignored by the user population.

Since 2002, speaker-independent recognition has been available on many mobile phones, and it is now available on close to 100 million devices. This technology allows speakers to simply pick up and use digit and name recognition for dialing and command-and-control applications on cell phones. It meets the requirements for simple and robust applications, and has become a common application used by technological sophisticates. Microsoft offered both AutoPC and PocketPC speech solutions, but neither achieved traction in the market place. The PocketPC phone voice command became the number-one application seller for the CE platform, but that fact only attests to the small market which was served. It appears that the population at large has yet to discover this capability in their devices (a failure of marketing), and thus this modern technology is only now gaining attention. Media review has been enthusiastic.

There are other markets for embedded speech technology, although most remain skeletal. Games and toys can sometimes benefit from a speech interface, and Hasbro and other toy manufacturers have fielded a series of small toys and plush animals that both respond to voice and talk to the user (in this class is a very capable R2D2 robot, which is being reissued in the mid-2006 time frame). These devices have not risen above the novelty level, and remain uncommon.

Navigation and automobile command and control is a relatively successful market for speech recognition on noncomputing platforms. The advent on On-Star in the US market delayed sophisticated speech recognition, as the system was fielded with a very crude discrete speech recognition system, and remained in that state for several years. Only in the past year has On-Star begun to field continuous speaker-independent recognition (from IBM). The market success of this venture will be interesting to watch, but is far from assured. Automobile systems (including navigation) use speech as output, but speech recognition has not yet become a common input.

PDAs would seem to be ideal for speech systems, but for several years the audio systems in IPAQs and other PDA devices offered audio quality ranging from inconsistent to abominable. (It is easy to mix audio and digital signals in computing platforms, and this is especially true of very small devices.) Continuous speech recognition was demonstrated on an IPAQ platform in 2003 by Voice Signal in English, and was simultaneously announced by Phillips for Mandarin. Neither company has fielded a commercial product on these devices for dictation, and the field remains unserved. (In the case of the Voice Signal demonstration, an outboard audio system was used to mitigate the IPAQ native systems.) Microsoft now offers a voice dialing system on its Windows CE systems that combine phones with PDAs.

The continuing increase in both memory and computing capability in cell phones allows dictation recognition in recent devices. Discrete large-vocabulary recognition in cell phones was announced in a commercial device (Samsung P-207) by Voice Signal in 2005, and continuous recognition was demonstrated in 2006. Nuance has also announced offerings, both embedded and distributed speech recognition (DSR)-server based. While text creation for email, messaging, and internal annotations seems an obvious requirement for communicating platforms, the market remains small. It is unclear whether communications (SMS), mobile search, location-based services, or some other application will ultimately become the market maker for this technology.

34.3.3 Technical Challenges that Enabled Mass Deployment

There have been many technical challenges to successful speech recognition in mobile devices, although some of these are being addressed by continuing technical advancement. Additionally, the rather limited

processing and memory resources that are available on mobile devices, compared to the seemingly unlimited resources available for network applications, imposes unique technical challenges for even the simplest mobile applications, let alone the complex applications that can be deployed within a telecommunications network.

Speech recognition algorithms range from the very simple – dynamic-programming-based speaker-independent small-vocabulary recognition systems – to the extraordinarily sophisticated HMM-based large-vocabulary continuous recognition.

Dynamic programming may be used to create an *acoustic pattern matching* algorithm, as described in *Sakoe* and *Chiba* [34.43]. While this pattern matching works well for a single speaker in a consistent environment, it produces a brittle matching algorithm that fails in moderate noise or for other speakers. On the other hand, the basic algorithm simply captures the speech to be modeled, derives a straightforward time-based model, and allows this model to be compared with a later spoken utterance. Because it is an acoustic matching algorithm, and must be trained by the user, the initial memory cost is very low. However, because the algorithm knows very little about speech or the signal that it is to model, the reference models tend to be memory intensive. Thus these algorithms can run in small-memory systems (as they are also not very compute intensive), but work only for small vocabularies (digits or 10 names, in typical applications).

Early implementations of dynamic programming algorithms used the digital signal processing (DSP) computation engine that is part of many cell phones. There was ample computation, and the low memory requirements allowed name recognition. Sensory announced toolkits for these technologies in 1997, and software for DSPs in 1999. Major cell-phone handset manufacturers (Nokia, Siemens, Motorola, NEC, Qualcomm, and others) supported internal speech laboratories that produced similar software throughout the 1980s and 1990s.

Modern cell phones have two separate processing systems – a digital signal processor and a second which is the user interface (UI) processor. The market leader for the UI processor is ARM, which first was designed in 1985, and has become dominant in the field. Modern ARM processors run at speeds from tens to hundred of MHz. In cell-phone systems designed in 2005 and 2006, these ARM processors can support applications independent of the cell-phone functionality of the devices, and often allow after-the-fact programming using BREW, Microsoft CE, or JAVA.

The UI processor has become a sophisticated platform in cell phones. While the DSP has become a closed system dedicated to cell-phone functionality, the user interface processor has become an *open* computing platform. Memory in modern cell phones ranges from 16 to 500 MB. Thus, these UI processors have computing capabilities ranging from those of the 286 to those of the Pentium I, except that they do not do native floating point. Audio systems in cell phones, with the exception of Bluetooth systems, tend to be well designed, and microphone placement is often optimized to minimize mouth sounds and extraneous noise. There are no remaining technical constraints on the implementation of speech recognition applications in these devices. Successful products now require user applications that are intuitive and discoverable, but these requirements are essentially independent of the speech recognition technology.

In remote-control applications, speech recognition has had mixed success. Standard remote-control devices remain technically difficult for speech applications, as they are generally built around 4 bit or 8 bit processors. (This is sensible if battery life is the primary factor, as it is in these devices). Some advanced remote-control devices (see Sensory for examples) provide limited speech recognition input, but this remains a novelty market.

There has been an industry call for a general solution to the remote control problem for advanced television services, since the blinking-12 generation cannot manage video on demand, the premier cost-plus service in television. Research has been focused on array-microphone solutions centered on cable boxes or other devices, but solutions remain suboptimal. This arena remains in flux.

34.3.4 History of Device-Based ASR

Embedded ASR solutions emerged from the development of *word spotting* as a topic of government interest during the 1970s. The general idea was that one could spot words in running speech by simply building a model of the acoustic signal that was the word or words of interest, and then search incoming speech for that acoustic event. The signal processing was, in general, LPC based, and very efficient. The search algorithm was a generalized dynamic program, which worked by attempting to match a warped version of the speech sample to the incoming speech, starting at all possible start points, and aborting if the *score* became too improbable.

Wide use was made of the Itakura distance, which Itakura claimed was an information-based mea-

sure [34.45]. The computation of the matching distance could be made very efficient, and the task could be carried out in early signal processing chips. On the other hand, the acoustic models of the example speech were quite large, and thus in early cell phones using this technology it was only possible to search for 10 or 20 names, despite the much larger phonebook capabilities.

Voice dialing using the dynamic programming method was available in the mid 1980s on several wired devices. Among these was the Freedom Phone, sold by Southwest Bell in the 1980s, which had a 50 item dynamic time warping (DTW)-based speech recognition system and also a speaker trained digit system. About the same time, Bell Laboratories built a voice dialer (called the Victory Dialer) for a car phone which recognized names.

In 1987, this DTW technology was used by Interstate Electronics to build a car-phone dialer that would dial either names or digits. The digit dialer was speaker independent, and was trained from a small database of digits, while the name recognizer had to be trained by the user. This dialer was plugged into the handset connector on a car phone, and the handset was, in turn, plugged into the dialer. The technology was extremely effective for its time, and the company sold several thousand of these devices. Unfortunately, the *car phone* market (the radiophones) was quickly supplanted by cell phones, and the market for these dialing devices disappeared.

34.3.5 Modern Use of ASR

While DTW technology was used by cell-phone manufacturers and a few other companies (ART – Advanced Recognition Technology, an Israeli company, was the most effective private purveyor) throughout the late 1980s and 1990s, the field was overtaken by much more-robust HMM technology. In 2002, Voice Signal introduced the first HMM-based speaker-independent digit dialer in the A500, a Samsung phone, and a year later introduced speaker-independent name dialing in the A610. From that time onwards, HMM technology has proven to be an effective and robust method of voice recognition in cell-phone platforms, and has even been extended to dictation systems of many thousands of words. The two largest providers of speaker-independent voce dialing were ART (now part of Nuance), and Voice Signal. The market is just emerging, although more than 75 million cell phones contain this technology.

The conjunction of mobile platforms and speech recognition is beginning to have an effect on the assistive technology community. It is now possible to use cell phones with only a single button push, and small-footprint synthesizers allow the use of these devices by visually impaired people. (Some devices *say* the name of any button pushed.) This market is not well established in the United States, but there is government support for such devices in the European Community.

34.3.6 Government Applications of Speech Recognition

General Service Overview

Government interest in speech recognition and in speech processes in general, covers many facets of the industry. Among the most interesting and unique are the uses of speech recognition in command-and-control applications, low-bandwidth communications, and intelligence functions.

Speech recognition interfaces are now being developed for aircraft simulators, and are specified as part of the standard avionics in the Joint Strike Fighter. The continuing DARPA speech programs, largely concentrated on speech and language recognition and interpretation of broadcast and conversational information, can be expected to yield systems that make transcription and analysis of intelligence information easier in the future. Other programs, such as TRANSTEC, are attempting to create mobile platforms that assist communications between speakers of non-intersecting languages. While current devices are concentrated on Arabic and Mandarin, the technology is being created to be easily ported to other languages, and commercial opportunities are awaiting enough technical advances to make the markets financially rewarding.

Technical Challenges to Overcome for Deployment

Technological evolution has made state-of-the-art speech recognition available in many devices and processes, and the availability of memory and computing power follows that availability in the commercial space, delayed by a year or two.

This delay is due to a multiplicity of factors. For field use, systems must be able to work over a large temperature and humidity range, and must tolerate physical abuse found in combat situations. Among the current platforms of choice are the Panasonic Toughbook and others of its type (there is a MIL standard, 810, which specifies insensitivity to temperature, humidity, liquid spills, salt, gunfire, acceleration and other factors). While central processing unit (CPU) speeds tend

Amazon EX1010 Page 708

to lag the commercial space, they tend to catch up within a year or two, so any process available on a commercial laptop today can be expected to run on a hardened processor in a short while.

On the other hand, acoustic requirements for military systems are generally assumed to be extreme, with operating equipment (tanks or airplanes), and military field operations (gunfire, crowds, etc.) forming the most challenging problems. Speech systems for translation are now being tested in Iraq, and have generally met with a positive response, although the technical program managers tend to be very conservative, and system integration time is very long. Speech recognition solutions have been deployed experimentally in the US Navy, Marines, and Air Force, but large-scale deployments have not yet been made.

Initial Uses of Speech Technologies

Two interesting applications of speech recognition are in general intelligence applications and in low-bandwidth communications systems.

With the advent of competent large-vocabulary speech recognition and highly intelligible speech synthesis, a low-bandwidth communications process can be completed by simply recognizing the speech into words, sending the text, and reconstructing the audio at the other end from the word string.

For general intelligence applications, an effort has been underway for many years with (D)ARPA funding of large-vocabulary speech recognition for the Department of Defense. In its original instantiation, the DARPA project of the 1970s defined the goals of a computer-based speech recognition system to recognize continuous speech, using a vocabulary of 1000 words, in real time. This project, called the Speech Understanding Research program, was funded at CMU, Stanford Research Institute (SRI), MIT's Lincoln Laboratory, Systems Development Corporation (SDC), and Bolt, Beranek, and Newman (BBN), and became the basis for speech recognition research worldwide. This project and its follow on allowed Dragon Systems to develop commercial discrete, then continuous speech recognition systems first on plug-in personal computer (PC) boards, and then on the native PC platforms themselves.

DARPA has since funded several follow-on projects in speech recognition, and in 1993 they funded a summer workshop in speech recognition at Rutgers University, where recorded spontaneous speech was, for the first time, the target of the research community. These efforts were aimed at *intelligence* applications, where it was assumed that some entity would transcribe everything available to it and mine that information for intelligence purposes. Recently the focus of DARPA programs has been on applications of speech recognition technology including automatic translation and information extraction systems, in the global autonomous language exploitation (GALE) program. The intelligence applications of speech recognition remain the stuff of fiction and fantasy reporting, while commercial applications grow at a very slow rate.

Speaker Identification/Verification

Over the past few years, with the rise of international terrorism and the concomitant growth of automated financial interactions and identity theft, the government has begun to pay attention to the entire question of biometrics in personal identification. While biometrics is a larger field than speech recognition, the speech portion of a biometric suite has the ability to enhance system performance, while allowing identity to be assessed from audio signals. Audio is the one biometric that does not depend on having the identification target present for either system training, system test, or both.

The United States has fielded a handheld biometrics system for use in military activities, called handheld interagency identity detection equipment (HIIDE) (http://www.securimetrics.com/solutions/hiide.html), but this system does not yet contain speech recogni-

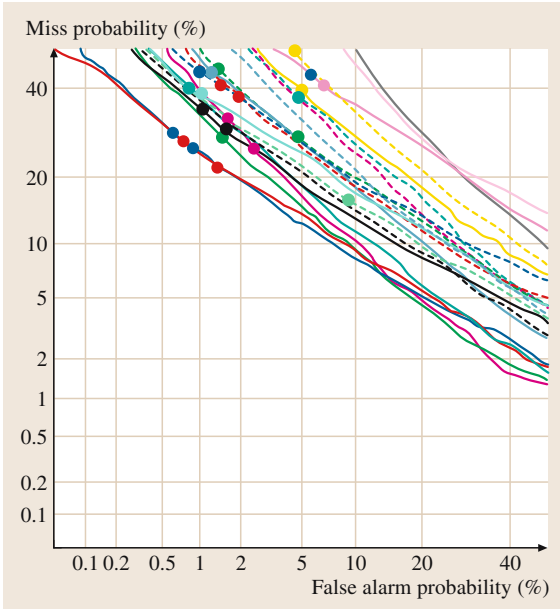


Fig. 34.2 Plot showing results from 21 systems for the 2002 primary condition in the one-speaker detection task

tion. In preparation for its inclusion in such systems, the government has been holding annual technology system performance evaluations with publicly available data and public results (although the identities of the participants are not part of the public results). In Fig. 34.2, one can observe the change in performance from 2002 to 2005 [34.46]. These improvements are due to more-sophisticated statistical modeling, increasing use of support vector machines (SVMs) rather than GMMs, and the extension of the scoring technology from short-term spectral analysis to long-term linguistically motivated features.

Speaker ID can be used in many ways, but the most common two are for speaker identification and speaker verification. While the technology improvements allow improved performance in both regimes, the open questions tend to be different.

Speaker verification assumes that the speaker's biometric identification (in this case, his voice) is accompanied by a claim of the speaker's identity. The task is to verify that the speaker is or is not the claimed person. This is the scenario of the National Institute of Standards and Technology (NIST) tests, where the error rates are computed over hundreds of thousands of trials. In this task, computation tends to be a secondary constraint (think, for instance, of an access system in which the talker speaks several words or sentences into a microphone, and then is granted access to a particular physical installation), while accuracy is crucially important. The acoustic channel is typically not an issue, since in most scenarios the speaker is a cooperative user of a system, which makes his life easier.

In the second scenario, one is typically attempting to do speaker identification (maybe speaker spotting) over a much larger collection of data, and the identity claim is missing. In that case, one must ask for each conversation whether the speaker is one of a known population of speakers, or whether the speaker is *someone else*. In this technology, the accuracy of the system tends to be less good, since the acoustic channel is not controlled (microphone, telephone, cell phone, etc.), but the task is more likely to be sorting data into interesting or uninteresting piles, and it may be assumed that secondary testing will be done on the output.

The reader can imagine more scenarios where speaker identification might be used, both in government and in industrial applications, and the technology is approaching usability for many of these.

34.4 Vision/Predications of Future Services – Fueling the Trends

Several technology trends and new technologies will fuel the growth in capabilities over the next few years. As the wireless network bandwidth pro capita and the speed of CPUs keep growing, rich and interactive multimodal and multimedia applications will continue to migrate from the core network to the periphery hosted by mobile devices, providing ubiquitous multimedia communication services *anywhere* and *anytime* to consumers and *road warrior* professionals. At the same time, broadband access is increasing at impressive rates and reached 65% of the American households by the end of 2005.

Although this scenario seems to suggest that video, data, and voice services are relentlessly converging to a seamless integrated internet protocol (IP) network, it is still unclear what kind of applications and standards will succeed in fostering next-generation human–machine communication services. It is certain, however, that these multimedia services, powered by a new breed of machines, will be operating globally supporting multiple languages, and will provide input access through a variety of different modalities (such as speech, gesture, pen, etc.). With this growth and complexity, improvements in service performance and deployment scalability that require minimal human intervention will be critical. Therefore, automatic learning techniques will become an essential component of these new services. Mining of heterogeneous information and data (call logs, speech, call center information, etc.) will also be an integrated part of these multimedia services to help extract trends and knowledge that can aid in improving business operation.

Predicating the future is always a risking task. That said, in this section we postulate several treads we envision seeing as the industry moves forward as it continues to mature into a mainstream market.

34.4.1 Multimodal-Based Speech Services

The widespread availability of broadband access, VoIP, and powerful mobile devices are fostering a new wave of human–computer interfaces that support multiple modalities, thus bridging and eventually blurring the device and network voice-enabled service markets. These

so-called multimodal user interfaces combine interactive multimedia output where audio, video, graphics, and text are accurately synchronized under time constraints, with several modes of input such as speech, keyboard, pointing devices, handwriting, and gestures. One example is the AT&T's multimodal access to city help (MATCH) multimodal service prototype (Fig. 34.3), which enables users to use speech, gesture, or the combination of the two modalities to access and retrieve information about cities and sites in the US [34.47]. This integration of speech and visual interfaces provide richer, more-versatile, and robust user interfaces than traditional speech input alone. Multimodal applications offer the ability to adapt the user experience to the actual terminal capabilities of the access device.

Although multimodal applications using mobile devices have received significant interest over the past five years, the central idea of using speech to manipulate a computer device goes back to the desktop dictation systems of the early 1980s. Several companies, including Dragon, Microsoft, and IBM, provided desktop solutions for speech dictation supporting speech, mouse, and keyboard input. These systems, which sold hundreds of thousands of copies primarily to the medical and legal industries, allowed users to click and talk. Speech recognition errors were corrected by clicking and either speaking or picking a word from a menu selection. Although early incarnations of these systems, which were essentially speaker dependent, required several training hours of speech and required users to pause between words, later systems that were available in the mid 1990s supported a much larger vocabulary set (> 5000 words), required only a few minutes of training data, and performed continuous speech recognition.



Fig. 34.3 AT&T's MATCH multimodal service prototype

Today, the Internet revolution and the availability of powerful handheld devices are facilitating the widespread of a new wave of multimodal applications. These applications support speech, gesture, and handwriting recognition, imposing new technical challenges to factor in speech recognition, natural-language understanding, and dialog management technologies as well as user-interface design, which is currently restricted to multimedia and video gaming applications. Major players in the industry such as Microsoft and IBM, have recognized the potential of this new multimodal market, proposing limited solutions to extend the concept of web browsing with speech input. Recently, Microsoft integrated its speech server into Microsoft Exchange 2007. Exchange is one of the most widely used messaging platforms for enterprises, thus positioning Microsoft well for providing flexible user designs for messaging services. Microsoft founded the speech application language tags (SALT) forum to promote voice applications as well as multimodal navigation in a web browser environment [34.48]. SALT interoperates with existing web standards, introducing a few new tags for speech control. IBM and Motorola, on the other hand, proposed and alternative approach to multimodal called XHTML + voice (X+V). X+V extends the existing Voice XML standards and complements them with a presentation layer built around XHTML [34.41]. In 2002, W3C launched a multimodal interaction working group with the purpose of finding common agreement upon a general architecture and an approach to define multimodal applications. The next generation of multimodal services will take advantage of integration of modalities to achieve robust performances.

Among the technology challenges when dealing with multimodal interfaces is the need to provide synchronization among the different input modalities. Several organizations, including AT&T, IBM, CMU, and IDIAP, have been actively pursing different strategies for synchronized modality integration. In AT&T's multimodal systems for information access to restaurants and subways, for example, users can circle with a pen, speak directly to a mobile device, and/or write on the screen. Mutual disambiguation, when using speech and gesture inputs simultaneously, is supported using a multimodal finite-state

Amazon EX1010 Page 711

transducer, which outputs a semantic lattice representation [34.47]. The lattice is interpreted by a multimodal dialog manager, which produces the next dialog action. A multimodal generator renders a graphical response with eventual animation together with a text-to-speech synthesizer. Multimodal integration can also be viewed as combing different streams, firing at different rates, using an asynchronous HMM. Whether using finite-state transducers or asynchronous HMMs to support multimodal interfaces, there is an inherent limitation in performance attributed to the lack of multimodal data. Rapid creation of multimodal applications with limited training data is a major technology challenge the research community is addressing today.

Multimodal interfaces will solve many of the current limitations of voice applications. They provide not only a mobile access to information but also a richer selection process through the visual interface. This visual interface significantly improves the user experience when dealing with complex applications such as maps that require special dialog strategies including disambiguation, clarification, relaxation, and constraining. Multimodalities provide several enhancements to a spectrum of applications including:

1. Information access and navigation: speech input alone for retrieving information about restaurants and travel directions, for example, can be significantly enhanced with a multimodal interface, supporting speech, gesture and global positioning systems (GPS).
2. Instant messaging: with the availability of powerful mobile devices, multimodal interfaces provide users with a richer and a more-flexible interface for communication using text messaging, clicking, and/or speaking into handheld devices.
3. Meetings and conferencing: multimodalities provide users engaged in meetings and conference calls the ability to geocollaborate and plan on the same project or event.
4. Gaming: combining gestures, pointing, and speech facilitates new types of video games in which players can speak and gesture simultaneously to perform a task. It will also aid people who have dyslexia (difficulties with movement) and dysgraphia (difficulties with handwriting) to apply speech as a primary modality to play games.
5. Diagnostics: combining speech and gesture provide an easy-to-use interface to diagnose products and services, such as airplanes.

34.4.2 Increased Automation of Service Development Process

The task of creating customized spoken dialog applications is traditionally labor intensive, requiring significant data resources and A tremendous level of expertise. As a result, despite efforts to modularize and reuse components of the dialog, it is not surprising that only a few hundred speech services were actually deployed in 2003 for large business customers. These services tend to be highly customized and are typically designed independently of any other sources of information, such as a website, or human–human conversational data.

AT&T has been pursuing a new research direction to scale this industry by completely automating the process of creating spoken, or chat-based, dialog applications by leveraging the wealth of information on business websites. Given that most businesses maintain a website, the goal has been to leverage that information to design, create, and maintain a new line of automated services requiring no human intervention. A prototype system of these new types of services (WebTalk) has been created by AT&T. WebTalk employs speech, language, and machine learning technologies to enable customers to enter into an automated chat-based or a spoken-language-dialog-based *conversation*. It includes a website analyzer that automatically downloads web pages, analyzes them, and constructs dialog-oriented task knowledge.

Information extraction is a key component of the AT&T WebTalk system. It enables the system to extract key entities, such as a list of products and services, contact information, definitions, questions/answers, etc. This information, along with a general-purpose DM, enables WebTalk to exchange in a natural-language dialog without actually involving a person in the loop either during the creation or the maintenance process.

34.4.3 Complex Problem Solving

At the heart of a spoken dialog system is the *dialog manager*, the component which decides what the system should say at each point in the conversation. Whereas all of the other elements of a spoken dialog system, such as speech recognition, language understanding, and text-to-speech, are based on probabilistic methods and optimized based on data, dialog managers have historically been handcrafted by user-interface experts. Because speech recognition errors are common, this design task is extremely complex and, in prac-

tice, the dialog manager is often *the weakest link* in a spoken dialog system, causing conversations be either be very structured or to go wildly off-track after even a single speech recognition error. As a result, deployed telephone-based spoken dialog systems have been limited to highly structured tasks where user's behavior can be constrained effectively (such as banking or simple travel transactions) or can be predicted accurately using natural-language understanding technologies for call routing.

A family of new approaches to building dialog managers seeks to extend the reach of dialog systems by improving the ability of the dialog manager to cope with speech recognition errors, hence better enabling the automation the class of more-complex problem solving services. Conventional approaches make binary accept/reject decisions about each speech recognition result, and use accepted speech recognition results to form a single hypothesis for the dialog state. Inevitably, a speech recognition error will be accepted into this dialog state, which can cause a system to get *stuck* and lead to an unnatural and unproductive conversation. New techniques in development today are aimed at representing the state of the conversation as a Bayesian network, which allows three important improvements [34.49,50]. First, a Bayesian network maintains *multiple hypotheses for the dialog state*: every speech recognition result is viewed as having some probability of being correct, and these probabilities are aggregated over the course of the dialog, preventing the system from getting *stuck* on a single hypothesis. Second, this aggregation process can incorporate models estimated from data of how users *actually* behave, rather than designer intuitions about how users might behave. Finally, automated planning in the form of partially observable Markov decision processes can be applied in a principled manner to ensure that the system always chooses an optimal action.

For example, consider the troubleshooting domain, in which a user is trying to restore a faulty product or service (such as an Internet connection) to a working state. Dialog systems in this domain today are highly constrained, for example asking a series of yes/no questions. While this approach avoids engendering more-complex speech, a forced regiment of questions can be tedious for callers who can feel trapped when a system cannot understand, for example, *But I've already tried rebooting my computer and that didn't work*. To build better dialog systems in this domain, more *mixed initiative* capabilities are required, which allow users to take control of the conversation, offer new information, or revisit an earlier part of the conversation.

Advanced dialog management techniques are theoretically well-grounded and appear able to deliver these capabilities. That said, several important research challenges remain, and chief among these is scalability: tracking many hypotheses for the current dialog state is computationally difficult, and performing planning using this representation is exceptionally difficult. Even so, early results show that approximate techniques appear to reduce the complexity dramatically with little loss of performance [34.50]. There is more work to do, but advanced dialog management techniques should begin to filter into mainstream commercial applications in the next few years.

34.4.4 Speech Mining

As the volume of accumulated speech data grows, we are faced with the analytical challenge of finding information of interest for the researchers, developers, executives, and customers, that can explain what is happening, and why, within a deployed service. Mining heterogeneous spoken dialog data for the purpose of improving system operation and extracting business intelligence is a fertile growth area for new research initiatives at many research and industrial labs around the world.

Data mining is concerned with the science, technology, and engineering of discovering patterns and extracting potentially useful or interesting information automatically or semiautomatically from data [34.51]. Data mining was introduced in the 1990s and has deep roots in the fields of statistics, artificial intelligence, and machine learning. With the advent of inexpensive storage space and faster processing over the past decade or so, data mining research has started to penetrate new grounds in areas of speech and audio processing as well as spoken language dialog. This has been fueled by the influx of audio data that are becoming more widely available from a variety of multimedia sources including webcasts, conversations, music, meetings, voice messages, lectures, television, and radio. Algorithmic advances in automatic speech recognition have also been a major, enabling technology behind the growth in data mining. Current state-of-the-art large-vocabulary, continuous speech recognizers are now trained on a record amount of data – several hundreds of millions of words and thousands of hours of speech. Pioneering research in robust speech processing, large-scale discriminative training, finite-state automata, and statistical hidden Markov modeling have resulted in

real-time recognizers that are able to transcribe spontaneous speech with a word accuracy exceeding 85%. With this level of accuracy, the technology is now highly attractive for a variety of speech mining applications.

Speech mining research includes many ways of applying machine learning, speech processing, and language processing algorithms to benefit and serve commercial applications. It also raises and addresses several new and interesting fundamental research challenges in areas of prediction, search, explanation, learning, and language understanding. These basic challenges are becoming increasingly important in revolutionizing business processes by providing essential sales and marketing information about services, customers, and product offerings. They are also enabling a new class of learning systems to be created that can infer knowledge and trends automatically from data, analyze and report application performance, and adapt and improve over time with minimal or zero human involvement.

Effective techniques for mining speech, audio, and dialog data can impact numerous business and government applications. The technology for monitoring conversational speech to discover patterns, capture useful trends, and generate alarms is essential for intelligence and law-enforcement organizations as well as for enhancing call centers operations. It is useful for analyzing, monitoring, and tracking customer preferences and interactions to better establish customized sales and technical support strategies. It is also an essential tool in media content management for searching through large volumes of audio warehouses to find information, documents, and news.

34.4.5 Mobile Devices

The mobile devices market continues to expand. While cell phones are predicted to reach steady-state volumes of about a billion devices a year starting in 2009, the sophistication and variety of additional services continually stresses the existing user interface. Many services remain undiscovered by the majority of cell phone users.

Manufacturers continue to produce specialized devices. The IPod music player by Apple has been a tremendous success, as has been the Razor cell phone by Motorola. In both these cases, the device capabilities are secondary to their industrial design. The prediction for the future is that the form of these devices and their user interfaces have become more important than the actual function of these devices. Consumer demand, at least in the United States, seems to be driven by marketing and consumer appeal.

That said, there are new markets opening because of recent advances. The government is exploring handheld electronic platforms for the soldier of the future. Handheld devices are being used in advanced medical facilities for patient information on demand, and for prescription ordering directly from the doctor to the pharmacy. Handheld navigation systems are a rapidly growing market segment, as larger memories, high-resolution designs, and accurate GPS location systems offer outstanding performance value.

All of these devices could incorporate speech interfaces. In navigation systems, speech output is essential to allow hands-free/eyes-free driver directions. Speech recognition is clearly the input method of the future, but the exact course of the introduction of this interface technology remains unclear.

34.5 Conclusion

In this chapter, we have presented a broad view of speech technologies from the viewpoint of the users of the various technologies. We showed how, beginning in the 1980s, even though speech recognition technologies were not very advanced, there were many *lower-hanging-fruit* applications that were deployed and that were tremendously successful. These applications, although quite simple (i. e., command-and-control applications), nonetheless provided real value to the people and/or businesses that deployed them and were the basis of continued research throughout the private and public sectors. Moving into the 1990s, speech services and speech engine products moved into the mainstream of network-based IVR services. Later in the decade and through the first decade of the 2000s, increased innovations and advances in microprocessor technologies enabled speech technologies to be a mainstay on small devices, such as cell phones, and enabled a new class of complex network-based speech services that begin to understand human speech instead of merely recognizing the words, thus enabling more-human-like interactions between people and machines.

Where is the industry heading? We will see much more automation of telecommunications services, and

for individuals, access to information and services at any time, with any device, and from anywhere, as well as in any language will be the norm. New advances in speech technologies will allow people, governments, educational institutions, and businesses to be informed immediately of any important event and to solve problems automatically. In this networked economy of *anytime, anywhere, and any channel communications*, customers continue to demand services that are quick, convenient, comprehensive, and personalized – this is, where speech is a necessary and valuable innovation. Business is done in real time and conducted with high mobility. Call centers and government services must evolve into 24×365 multimedia-based service centers that combine networking with voice interaction, web technologies, and natural-language voice processing. Speech researchers, though advancing the state of art from an academic exercise to over a billion-dollar market in about 30 years, have still only scratched the surface of the algorithms and capabilities that are required for speech to become ubiquitous in all facets of our lives. This is the challenge we all face, and the fun we will all have as we move forward to make this *future* become a reality.

References

34.1 J.R. Pierce: Whither speech recognition?, J. Acoust. Soc. Am. **46**(4), 1029–1051 (1969)

34.2 K.H. Davis, R. Biddulph, S. Balashek: Automatic recognition of spoken digits. In: *Communication Theory*, ed. by W. Jackson (Butterworths, London 1953)

34.3 A. Lolje, M. Riley, D. Hindle, F. Pereira: The AT&T 60000 word speech-to-text system, Proc. Spoken Language Technology Workshop (Morgan Kaufmann, Austin 1995) pp. 162–165

34.4 L. Rabiner, B.-H. Juang: *Fundamentals of Speech Recognition* (Prentice Hall, Englewood Cliffs 1993)

34.5 F.C. Pereira, M. Riley: Speech recognition by composition of weighted finite automata. In: *Finite-State Devices for Natural Language Processing*, ed. by E. Roche, Y. Schabes (MIT Press, Cambridge 1997)

34.6 V. Goffin, C. Allauzen, E. Bocchieri, D. Hakkani-Tur, A. Ljolje, S. Parthasarathy, M. Rahim, G. Riccardi, M. Saraclar: The AT&T watson speech recognizer, Proc. IEEE ICASSP (2005)

34.7 J. Huang, B. Kingsbury, L. Mangu, M. Padmanabhan, G. Saon, G. Zweig: Recent improvements in speech recognition performance on large conversational speech, Proc. ICSLP (2000)

34.8 B. Atal: Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification, J. Acoust. Soc. Am. **55**(6), 1304–1312 (1974)

34.9 K. Vintsyuk: Speech discrimination by dynamic programming, Kibernetika **4**, 81–88 (1968)

34.10 F. Jelinek: Continuous speech recognition by statistical methods, Proc. IEEE **64**(4), 532–556 (1976)

34.11 L.R. Rabiner: A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE **77**(2), 257–286 (1989)

34.12 F. Jelinek: *Statistical Methods for Speech Recognition* (MIT Press, Cambridge 1997)

34.13 L.R. Bahl, P.F. Brown, P.V. De Souza, R.L. Mercer: Maximum mutual information estimation of HMM parameters for speech recognition, Proc. IEEE ICASSP (1986)

34.14 M.H. Cohen, J.P. Giangola, J. Balogh: *Voice User Interface Design* (Addison Wesley, Boston 2004)

34.15 A. Smola, P. Bartlett, B. Scholkopf, D. Schuurmans: *Advances in Large Margin Classifiers* (MIT Press, Cambridge 2000)

34.16 R. Schapire, M. Rochery, M. Rahim, N. Gupta: Incorporating prior knowledge into boosting, Proc. Nineteenth Int. Conf. Machine Learning (2002)

34.17 J. Baker: The Dragon system – an overview, IEEE Trans. ASSP **23**(1), 24–29 (1975)

34.18 A. Gorin, G. Riccardi, J. Wright: How May I Help You?, Speech Commun. **23**, 113–127 (1997)

34.19 http://www.nexidia.com

34.20 http://www.verint.com

34.21 R. Natarajan, B. Prasad, B. Suhm, D. McCarthy: Speech enabled natural language call routing: BBN call director, Proc. Int. Conf. Spoken Language Process. (2002)

34.22 L. Lee, R. Rose: A Frequency Warping Approach to Speaker Normalization, IEEE Trans. Speech Audio Process. **6**, 49–60 (1998)

34.23 D.A. Reynolds, R.C. Rose: Robust text-independent speaker identification using Gaussian mixture speaker models, IEEE Trans. Speech Audio Process. **3**(1), 72–83 (1995)

34.24 X.D. Huang, A. Acero, H.-W. Hon: *Spoken Language Processing* (Prentice Hall, Englewood Cliffs 2001)

34.25 M. Rahim, B.-H. Juang: Signal bias removal by maximum likelihood estimation for robust speech recognition, IEEE Trans. Speech Audio Process. **4**(1), 19–30 (1996)

34.26 S. Bangalore, G. Riccardi: Stochastic finite-state models for spoken language machine translation, Mach. Transl. **17**(3), 165–184 (2002)

Amazon EX1010 Page 715

34.27 N. Gupta, G. Tur, D. Hakkani-Tür, S. Bangalore, G. Riccardi, M. Rahim: The AT&T spoken language understanding system, IEEE Trans. Audio Speech Lang. Process. **14**(1), 213–222 (2006)

34.28 G. Riccardi, D. Hakkani-Tür: Active and unsupervised learning for automatic speech recognition, Proc. 8th European Conf. Speech Commun. and Technol. (2003)

34.29 S. McGlashan: Voice Extensible Markup Language (VoiceXML) Version 2.0 (2004) (http://www.w3.org/TR/2004/PR-voicexml20-20040203)

34.30 R. Nakatsu: Anser – An application of speech technology to the Japanese banking industry, Computer **23**(8), 43–48 (1990)

34.31 http://www.nuance.com

34.32 http://www.tellme.com

34.33 http://www.bevocal.com

34.34 http://www.telureka.com

34.35 http://www.convergys.com

34.36 http://www.west.com

34.37 J. Wilpon, L.R. Rabiner, C.H. Lee, E.R. Goldman: Automatic recognition of keywords in unconstrained speech using hidden Markov models, IEEE Trans. Acoust. Speech Signal Process. **38**(11), 1870–1878 (1990)

34.38 W.T. Hartwell, M.A. Johnson, J. Picone: Automatic speech recognition using echo cancellation, US Patent **4,914,692** (1990)

34.39 V. Franco: Automation of operator services at AT&T, Proc. Voice (1993)

34.40 S. Shanmugham, D. Burnett: Media Resource Control Protocol Version 2 (MRCPv2) (http://tools.ietf.org/wg/speechsc/draft-ietf-speechsc-mrcpv2/draft-ietf-speechsc-mrcpv2-09.txt)

34.41 http://www.w3.org/TR/xhtml+voice

34.42 L.R. Rabiner: Applications of voice processing to telecommunications, Proc. IEEE **82**(2), 199–228 (1994)

34.43 H. Sakoe, C. Chiba: Dynamic programming algorithm optimization for spoken word recognition, IEEE Trans. Acoust. Speech Signal Process. **ASSP-26**, 43–49 (1978)

34.44 J. Cooperstock: From the flashing 12:00 to a usable machine: Applying UbiComp to the VCR (http://acm.org/sigchi/chi97/proceedings/short-talk/jrc.htm)

34.45 A.H. Gray Jr., J.D. Markel: Distance measures for speech processing, IEEE Trans. ASSP **24**(5), 380–391 (1976)

34.46 M. Przybocki, A. Martin: *NIST's Assessment of Text Independent Speaker Recognition Performance* (2005)

34.47 M. Johnston, S. Bangalore, G. Vasireddy, A. Stent, P. Ehlen, M. Walker, S. Whittaker, P. Maloor: MATCH: An architecture for multimodal dialogue systems, Proc. 40th Annual Meeting of the Association for Computational Linguistics (2002)

34.48 http://www.saltforum.org/saltforum/downloads/SALT1.0.pdf

34.49 T. Paek, E. Horvitz: Conversation as action under uncertainty, Proc. Conf. Uncertainty in Artificial Intelligence (UAI) (2000)

34.50 J.D. Williams: *Partially Observable Markov Decision processes for Spoken Dialog Management*, Ph.D. Thesis (University of Cambridge, Cambridge 2006)

34.51 I. Witten, E. Frank: *Data Mining* (Morgan Kaufmann, Austin 1999)

Amazon EX1010 Page 716

35. Spoken Dialogue Systems

V. Zue, S. Seneff

Spoken dialogue systems are a new breed of interfaces that enable humans to communicate with machines naturally and efficiently using a conversational paradigm. Such a system makes use of many human language technology (HLT) components, including speech recognition and synthesis, natural language understanding and generation, discourse modeling, and dialogue management. In this contribution, we introduce the nature of these interfaces, describe the underlying HLTs on which they are based, and discuss some of the development issues. After providing a historical perspective, we outline some new research directions.

35.1 **Technology Components and System Development** 707
35.1.1 System Architecture 707
35.1.2 Spoken Input Processing 708
35.1.3 Spoken Output Processing 710
35.1.4 Dialogue Management 711
35.2 **Development Issues** 712
35.2.1 Data Collection 712
35.2.2 Evaluation 713
35.3 **Historical Perspectives** 714
35.3.1 Large-Scale Government Programs 714
35.3.2 Some Example Systems 715
35.4 **New Directions** 715
35.4.1 User Simulation 715
35.4.2 Machine Learning and Dialogue Management 716
35.4.3 Portability 717
35.4.4 Multimodal, Multidomain, and Multilingual Application Development 717
35.5 **Concluding Remarks** 718
References 718

Speech is the most natural means for humans to communicate; nearly all of us can talk and listen to one another without special training. Speech is flexible; it can free our eyes and hands to attend to other tasks. Speech is also very efficient; one can typically speak several times faster than one can type or write. Nowadays, with the pervasiveness of telephones and cell phones, speech is also one of the most inexpensive ways for us to communicate. It is therefore not surprising that speech-based interfaces are in the minds of every techno-visionary, science fiction writer, and Hollywood producer.

Other authors of this volume have discussed automatic speech recognition, an important and as yet unsolved problem in its own right, with a clear set of applications that include document preparation and audio indexing. However, speech recognition by itself is only a part of the interface solution. Human communication is predicated on the assumption that we *understand* one another. In fact, this understanding is often couched in other parts of the dialogue – how can one possibly make sense out of a sentence *what about that one?* without knowledge of the previous dialogue? Many applications that lend themselves to spoken input/output – inquiring about weather or making travel arrangements – are in fact exercises in information access and/or interactive problem solving. The solution is often built up incrementally, with both the user and the computer playing active roles in the *dialogue*. Therefore, several language-based input and output technologies must be developed and integrated to reach this goal. The resulting spoken dialogue systems are the subject of this contribution.

Many speech-based interfaces can be considered conversational, and they may be differentiated by the degree to which the system maintains an active role in the conversation. At one extreme are *system-initiative*, or directed-dialogue transactions where the computer takes complete control of the interaction by requiring that the user answer a set of prescribed questions, much like the touch-tone implementation of interactive voice response (IVR) systems. In the case of air travel planning, for example, a directed-dialogue system could ask the user to *Please say just the departure city*. Since the user's options are severely restricted,

Table 35.1 Transcript of an actual conversation between an agent (A) and a client (C) over the phone. Typical conversational phenomena are annotated on the right

C:	Yeah, [umm] I'm looking for the Buford Cinema.	Disfluency
A:	OK, and you're wanting to know what's showing there or . . .	Interruption
C:	Yes, please.	Confirmation
A:	Are you looking for a particular movie?	
C:	[umm] What's showing.	Clarification
A:	OK, one moment.	Back channel
	. . .	
A:	They're showing *A Troll In Central Park*.	
C:	No.	Inference
A:	Frankenstein.	Ellipsis
C:	What time is that on?	Co-reference
A:	Seven twenty and nine fifty.	
C:	OK, any others?	Fragment
	. . .	

Table 35.2 Transcript of an actual phone conversation between a spoken dialogue system (S) and a novice user (U), using the system for the second time. The system's capabilities are illustrated on the right

U:	I'd like to fly from Boston to Minneapolis on Tuesday – no, Wednesday November 21.	Understand mid-stream correction
S:	Okay, flights from Boston to Minneapolis Saint Paul on Wednesday November twenty first. Can you provide the approximate departure time or airline preference?	Solicit necessary information from user
U:	Yeah, I'd like to fly United and I'd like to leave in the afternoon.	Permit multiple utterances
S:	Okay, United flights from Boston to Minneapolis Saint Paul departing between twelve p.m. and six p.m. on Wednesday November twenty first. One moment please... There are no nonstop flights. There are more than three flights. The earliest departure leaves at 1:45 p.m. and arrives at 6:07 p.m.	Confirm understanding; summarize alternatives
U:	Do you have something leaving around 4:45?	Allow negotiation
S:	Okay, United flights from Boston to Minneapolis Saint Paul leaving at 4:45 p.m. on Wednesday November twenty first. One moment please... What about a flight connecting in Chicago o'Hare leaving at 4:45 p.m., would that work? I also have a flight connecting in Chicago o'Hare leaving at 5:45 p.m.	Articulate pertinent information
U:	4:45.	Understand fragments in context
S:	United flight 1661 connecting to United flight 1107 in Chicago O'Hare has been added to your itinerary. You depart Boston at 4:45 p.m. and arrive in Minneapolis Saint Paul at 9:08 p.m. What date will you be returning on?	
U:	I'd like to return the following Tuesday.	Understand relative dates
	. . .	

successful completion of such transactions is easier to attain, and indeed some successful demonstrations and deployment of such systems have been made [35.1]. However, system-initiated dialogue may cause users to be annoyed due to its inherent inflexibility when the task becomes more complex, involving many turns or exchanges. In fact, user-experience studies [35.2] show that people much preferred *more-natural* user interfaces over system-initiated ones.

At the other extreme are *user-initiative* systems whereby the user has complete freedom in what they say to the system, (e.g., *I want to visit my grandmother*) while the system remains relatively passive, asking only for clarification when necessary. In this case, the user may feel uncertain as to what capabilities exist, and may, as a result, stray quite far from the domain of competence of the system, leading to great frustration because nothing is understood. Lying between these two extremes are systems that incorporate a *mixed-initiative*, goal-oriented dialogue, in which both the user and the computer participate actively to solve a problem interactively using a conversational paradigm. It is this latter mode of interaction that is the primary focus of this contribution.

Amazon EX1010 Page 718

Of course there is not a black-and-white distinction between mixed- and system-initiative systems. Most system-initiative systems support powerful metacommands such as *back up* and *start over*, which are certainly examples of user initiative. Probably the most effective systems will turn out to be those that can gracefully transition between a more-flexible mode and a mode that guides and restricts the user, based on monitoring user behavior. For instance, when Carnegie Mellon University's *Let's Go!* system [35.3, 4] was deployed to the public, it was changed considerably from its original configuration to increase the amount of user prompting, in order to enhance the yield on successful interactions. It is likely that extensive dialogue context-dependent *help* mechanisms, which would take the form of suggestions or prompts, would be beneficial for transitioning novice users into experts.

One way for developers to discover what form natural human–computer interaction might take is to examine human–*human* interactions during joint problem solving [35.5]. Table 35.1 shows the transcript of a conversation between an agent (A) and a client (C) over the phone. As illustrated by this example, spontaneous dialogue is replete with disfluencies, interruption, confirmation, clarification, ellipsis, co-reference, and sentence fragments. Some of the utterances cannot be understood properly without knowing the context in which they appear. While many of the conversational phenomena illustrated here are still ongoing research challenges, some of the mixed-initiative spoken dialogue systems developed over the past decade can help users solve problems interactively with good success. Table 35.2 shows the transcript of an actual conversation between such a system [35.6] and a novice user calling the system for the second time. This example illustrates the state of the art at the writing of this contribution.

The last decade has witnessed the emergence of some spoken dialogue systems with limited capabilities. Despite this moderate success, the ultimate deployment of such interfaces will require continuing improvement of the core human language technologies (HLTs) and the exploration of many uncharted research territories. In this contribution, we introduce the nature of the spoken dialogue systems, describe the underlying HLTs on which they are based, and discuss some of the development issues. After providing a historical perspective, we outline some of the new research directions. Given space limitations, it is not possible to provide adequate coverage of the entire field. Instead, we will draw from our own experience in developing such systems at MIT since the late 1980s to illustrate our points [35.7–14], and we will provide extensive references. Interested readers are referred to the recent proceedings of the Eurospeech Conference, the International Conference of Spoken Language Processing, the International Conference of Acoustics, Speech, and Signal Processing, the International Symposium on Spoken Dialogue, and other relevant publications [35.15].

35.1 Technology Components and System Development

35.1.1 System Architecture

Figure 35.1 shows the major components of a typical spoken dialogue system. The spoken input is first processed through the speech recognition component. The natural language component, working in concert with the recognizer, produces a meaning representation for the utterance. For the information retrieval application illustrated in this figure, the meaning representation can be used to retrieve the appropriate information in the form of text, tables, and graphics. If the information in the utterance is insufficient or ambiguous, the system may choose to query the user for clarification. If verbal conveyance of the information is desired, then natural language generation and text-to-speech synthesis are utilized to produce the spoken responses. Throughout the process, discourse informa-

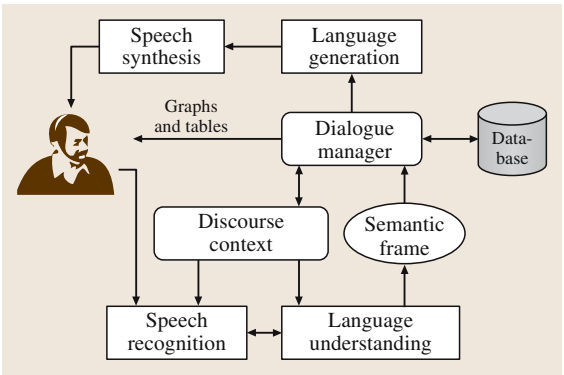


Fig. 35.1 A generic block diagram for a typical spoken dialogue system

tion is maintained and fed back to the speech recognition and language understanding components, so that sentences can be properly understood in context. Finally, a dialogue component manages the interaction between the user and the computer. The nature of the dialogue can vary significantly depending on whether the system is creating or clarifying a query prior to accessing an information database, or perhaps negotiating with the user in a post-information-retrieval phase to relax or somehow modify some aspects of the initial query.

While speech may be the modality of choice, as is the case with phone-based interactions and hands-busy/eyes-busy settings, there are clearly cases where speech is not a good modality, especially, for example, on the output side when the information contains maps, images, or large tables of information that cannot be easily explained verbally. Human communication is inherently multimodal, employing facial, gestural, and other cues to communicate the underlying linguistic message. Thus, speech interfaces should be complemented by visual and sensory motor channels.

Communication can be enhanced by enabling the user to choose among many different modalities, including gesturing, pointing, writing, and typing on the input side [35.16, 17], along with graphics and a talking head on the output side [35.18], to achieve the task in hand in the most natural and efficient manner.

The development of spoken dialogue systems offers a set of new challenges to speech and natural language researchers, and raises several important research issues, some of which will be discussed next.

35.1.2 Spoken Input Processing

Spoken language understanding involves the transformation of the speech signal into a meaning representation that can be used to interact with the specific application back-end. This is typically accomplished in two steps, the conversion of the signal to a set of words (i. e., speech recognition), and the derivation of the meaning from the word hypotheses (i. e., language understanding). A discourse component is often used to properly interpret the meaning of an utterance in the larger context of the interaction.

Often this meaning representation takes the form of a simple list of (attribute: value) pairs encoding user-specified constraints. Thus, the sentence, *Show me flights from San Francisco to Denver leaving in the morning on March ninth* could be represented by the following simple structure:

{ action: display
topic: flights
source: SFO
destination: BOS
departure_interval: morning
date: Mar09 }

Automatic Speech Recognition

Techniques for automatic speech recognition are similar to those described in other contributions in this volume by Young, Mohri et al., and Picheny and Nahamoo. However, some issues that are particular to spoken dialogue input offer additional challenges. Input to spoken dialogue systems is often generated extemporaneously – especially from novice users of these systems. Such spontaneous speech typically contains disfluencies (i. e., unfilled and filled pauses such as *umm* and *aah*, as well as word fragments). In addition, the input utterances are likely to contain words outside the system's working vocabulary – a consequence of the fact that present-day technology can only support the development of systems within constrained domains. Thus far, some attempts have been made to deal with the problem of disfluency. For example, researchers have improved their systems' recognition performance by introducing explicit acoustic models for the filled pauses [35.19, 20]. Similarly, *trash* models have been used to detect the presence of word fragments or unknown words [35.21], and procedures have been devised to learn the new words once they have been detected [35.22, 23]. Suffice it to say, however, that the detection and learning of unknown words continues to be a problem that needs our collective attention. A related topic is utterance- and word-level rejection, in the presence of either out-of-domain queries or unknown words [35.24, 25]. Many systems incorporate some form of confidence scoring to try to identify problematic inputs [35.1, 26, 27]. The system can then try an alternative strategy to either help the user, or back off to a more-directed dialogue and/or one that requires explicit confirmation.

It is perhaps not surprising that some of the first conversational systems available to the general public were accessible via telephone [35.1, 28, 29], in many cases replacing presently existing interactive voice response (IVR) systems. Telephone quality speech is significantly more difficult to recognize than high-quality recordings, both because of the limited bandwidth and the noise and

distortions introduced in the channel [35.30]. The acoustic condition deteriorates further for cellular telephones, either analog or digital.

Natural Language Understanding

Speech recognition systems typically implement linguistic constraints as a statistical language model (i. e., *n*-gram) that specifies the probability of a word given its predecessors. While these language models have been effective in reducing the search space and improving performance, they do not begin to address the issue of speech understanding. On the other hand, most natural language systems are developed with text input in mind; it is usually assumed that the entire word string is known with certainty. This assumption is clearly false for speech input, where many alternative word hypotheses are competing for the same time span in any sentence hypothesis produced by the recognizer (e.g., *euthanasia* and *youth in Asia*,) and some words may be more reliable than others because of varying signal robustness. Furthermore, spoken language is often agrammatical, containing fragments, disfluencies, and partial words. Language understanding systems designed for text input may have to be modified in fundamental ways to accommodate spoken input.

Natural language analysis has traditionally been predominantly syntax-driven – a complete syntactic analysis is performed which attempts to account for *all* words in an utterance. However, when working with spoken material, researchers quickly came to realize that such an approach [35.31–33] can break down dramatically in the presence of unknown words, novel linguistic constructs, recognition errors, and spontaneous speech events.

Due to these problems, many researchers have turned towards more semantic-driven approaches, at least for spoken language tasks in constrained domains. In such approaches, a meaning representation is derived by *spotting* key words and phrases in the utterance [35.34]. While this approach loses the constraint provided by syntax, and may not be able to interpret complex linguistic constructs adequately, the need to accommodate spontaneous speech input has outweighed these potential shortcomings. In fact, many systems have abandoned the notion of achieving a complete syntactic analysis of every input sentence, favoring a more-robust strategy that can still be used to produce an answer when a full parse is not achieved [35.35–37]. This can be accomplished by identifying parsable phrases and clauses, and providing a separate mechanism for gluing them together to form a complete meaning analysis [35.36]. Ideally, the parser includes a probabilistic framework with a smooth transition to parsing fragments when full linguistic analysis is not achievable. Examples of systems that incorporate such *stochastic* modeling techniques can be found in [35.38, 39].

How should the speech recognition component interact with the natural language component in order to obtain the correct meaning representation? One of the most popular strategies is the so-called *N*-best interface [35.40], in which the recognizer proposes its best *N* complete sentence hypotheses one by one, stopping with the first sentence that is successfully analyzed by the natural language component. In this case, the natural language component acts as a filter on *whole sentence* hypotheses. Alternatively, competing recognition hypotheses can be represented in the form of a stochastic word graph or lattice [35.41], which is significantly more compact than an *N*-best list, thus permitting a deeper search if desired. A pruned lattice can easily represent an *N*-best list of several thousand unique utterances. *Sausages* [35.42] provide an even more compact representation, reducing the memory requirement by as much as 100-fold by essentially collapsing a lattice into a sequence of *N*-best word hypotheses.

In an *N*-best list, many of the candidate sentences may differ minimally in regions where the acoustic information is not very robust. While confusions such as *an* and *and* are acoustically reasonable, one of them can often be eliminated on linguistic grounds. In fact, many of the top *N* sentence hypotheses might be eliminated before reaching the end if syntactic and semantic analyses take place early on in the search. One possible solution, therefore, is for the speech recognition and natural language components to be tightly coupled, so that only the acoustically promising hypotheses that are linguistically meaningful are advanced. For example, partial theories can be arranged on a stack, prioritized by score. The most promising partial theories are extended using the natural language component as a predictor of all possible next-word candidates; none of the other word hypotheses are allowed to proceed. Therefore, any theory that completes is guaranteed to parse. Researchers are beginning to find that such a tightly coupled integration strategy can achieve higher performance than an *N*-best interface, often with a considerably smaller stack size [35.43–46]. The future is likely to see increasing use of linguistic analysis at earlier stages in the recognition process.

Discourse Modeling

Human verbal communication is a two-way process involving multiple, active participants. Mutual under-

standing is achieved through direct and indirect speech acts, turn taking, clarification, and pragmatic considerations. Discourse analysis allows a conversational system to understand an utterance in the context of the previous interaction. As such, discourse can be considered to be part of the input processing stage. To communicate naturally, a system must be able to handle phenomena such as deictic (e.g., verbal pointing as in *I'll take the second one*) and anaphoric reference (e.g., using pronouns as in *what's their phone number*) to allow users to refer to items currently in focus efficiently. An effective system should also be able to handle ellipsis and fragments so that a user does not have to fully specify each query. For instance, if a user says, *I want to go from Boston to Denver*, followed with, *show me only United flights*, he/she clearly does not want to see *all* United flights, but rather just the ones that fly from Boston to Denver. The ability to inherit information from preceding utterances is particularly helpful in the face of recognition errors. The user may have asked a complex question involving several restrictions, and the recognizer may have misunderstood a single word, such as a flight number or an arrival time. If a good context model exists, the user can then utter a short correction phrase, and the system will be able to replace just the misunderstood word, preventing the user from having to repeat the entire utterance, running the risk of further errors.

There is a related question of how to *forget* context appropriately. For example, if a user provides a new airline, the remainder of the context can be sensibly preserved. However, if the user provides a new *departure city* not served by the specified airline, pragmatics must play a role in recognizing that the airline in question should be deleted from context.

35.1.3 Spoken Output Processing

On the output side, a spoken dialogue system must be able to convey the information to the user in natural sounding sentences. This task is typically accomplished in two steps. First, the information – whether it be expressed in tabular, database query language, or hypertext forms – must be converted into well-formed natural sentences. If a spoken response is desired, the sentences must then be fed to a text-to-speech (TTS) system to generate the speech signal for playback.

Natural Language Generation

Spoken language generation serves two important roles. First and foremost, it provides a verbal response to the user's queries, which is essential in applications where visual displays are unavailable. In addition, it can provide feedback to the user in the form of a paraphrase, confirming the system's proper understanding of the input query. Although there has been much research on natural language generation (NLG), dealing with the creation of coherent paragraphs [35.47, 48], the language generation component of a spoken dialogue system typically produces the response one sentence at a time, without paragraph-level planning.

Research in language generation for spoken dialogue systems has not received nearly as much attention as has language understanding, especially in the US, perhaps due to the funding priorities set forth by the major government sponsors. In many cases, output sentences are simply word strings, in text or prerecorded acoustic format, that are invoked when appropriate. In some cases, sentences are generated by concatenating templates after filling slots by applying recursive rules along with appropriate constraints [35.49]. There has also been some work using more-corpus-based methods for language generation in order to provide more variation in the surface realization of the utterance [35.50].

Spoken dialogue systems can behave quite differently depending on what input and output modalities are available to the user. In displayless environments such as the telephone, it might be necessary to tailor the dialogue so as not to overwhelm the user with information. When displays are available however, it may be more desirable simply to summarize the information to the user, and to show them a table or image, etc. [35.51]. Similarly, the nature of the interaction will change if alternative input modalities, such as pen or gesture, are available to the user.

Speech Synthesis

The conversion of text to speech is the final stage of output generation (see Part D of this Handbook). TTS systems in the past were primarily rule driven, requiring the system developers to possess extensive acoustic–phonetic and other linguistic knowledge [35.52]. These systems are typically highly intelligible, but suffer greatly in naturalness. In recent years, we have seen the emergence of a new, concatenative approach, brought on by inexpensive computation/storage, and the availability of large corpora [35.53–55]. In this corpus-based approach, units excised from recorded speech are concatenated to form an utterance. The selection of the units is based on a search procedure subject to a predefined distortion measure. The output of these TTS systems is often judged to be more natural than that of rule-based systems [35.56].

The language generation and text-to-speech components of most spoken dialogue systems are not closely coupled; the same text is generated whether it is to be read or spoken. Furthermore, systems typically expect the language generation component to produce a textual surface form of a sentence (throwing away valuable linguistic and prosodic knowledge) and then require the text-to-speech component to produce linguistic analysis anew. In *concept-to-speech* generation [35.57], one *already knows the parse* [35.58], and this knowledge can simplify the synthesizer's task. Such a close coupling can also potentially produce higher-quality output speech than could be achieved with a decoupled system, since it permits finer control of prosody. Whether language generation and speech synthesis components should be tightly integrated, or can remain modular but effectively coupled by augmenting text output with a markup language (e.g., SABLE [35.59]) remains to be seen. Clearly however, these two components would benefit from a shared knowledge base.

35.1.4 Dialogue Management

The dialogue management component of a spoken dialogue system manages the interaction between the user and the computer. The technology for building this component is one of the least developed in the HLT repertoire, especially for the mixed-initiative dialogue systems considered in this contribution. Although there has been some theoretical work on the structure of human–human dialogue [35.60], this has not led to effective insights for building human–machine interactive systems. As mentioned previously, there is also considerable debate in the speech and language research communities about whether modeling human–machine interactions after human–human dialogues is necessary or appropriate [35.61–63].

Dialogue modeling means different things to different people. For some, it includes the *planning* and *problem solving* aspects of human–computer interactions [35.64]. In the context of this chapter, we define dialogue modeling as the planning, for each turn, of the system's side of the conversation, including verbal, tabular, and graphical response, as well as any clarification requests.

Dialogue modeling and management serves many roles [35.65–68]. In the early stages of the conversation the role of the dialogue manager might be to gather information from the user, possibly clarifying ambiguous input along the way, so that, for example, a complete query can be produced for the application database. The dialogue manager must be able to resolve ambiguities that arise due to recognition error (e.g., *Did you say Boston or Austin?*) or incomplete specification (e.g., *On what day would you like to travel?*).

In later stages of the conversation, after information has been accessed from the database, the dialogue manager might be involved in some negotiation with the user. For example, if there were too many items returned from the database, the system might suggest additional constraints to help narrow down the number of choices. Pragmatically, the system must be able to initiate requests so that the information can be reduced to digestible chunks (e.g., *I found ten flights; do you have a preferred airline or connecting city?*).

In addition to these two fundamental operations, the dialogue manager must also inform and guide the user by suggesting subsequent subgoals (e.g., *Would you like me to price your itinerary?*), offer assistance upon request, help relax constraints or provide plausible alternatives when the requested information is not available (e.g., *I don't have sunrise information for Oakland, but in San Francisco ...*), and initiate clarification subdialogues for confirmation. In general the overall goal of the dialogue manager is to take an active role in directing the conversation towards a successful conclusion for the user.

The dialogue manager can influence other system components by, for example, dynamically making dialogue context-dependent adjustments to language models or discourse history. At the highest level, it can help detect the appropriate broad subdomain (e.g., weather, air travel, or urban navigation). Within a particular domain, certain queries could introduce a focus of attention on a subset of the lexicon. For instance, in a dialogue about a trip to France, the initial user utterance, *I'm planning a trip to France*, would allow the system to greatly enhance the probabilities on all the French destinations. Finally, whenever the system asks a directed question, the language model probabilities can be altered so as to favor appropriate responses to the question. For example, when the system asks the user to provide a date of travel, the system could temporarily enhance the probabilities of date expressions in the response.

A challenging area of research is recovery from the inevitable misunderstandings that a system will make [35.69]. Errors could be due to many different phenomena (e.g., acoustics, speaking style, disfluencies, out-of-vocabulary words, parse coverage, or understanding gaps), and it can be difficult to detect that there is a problem, determine what the problem is caused by,

Amazon EX1010 Page 723

and convey to the user an appropriate response that will fix the problem. This is another responsibility of dialogue management. One of the most challenging aspects of spoken dialogue system development is the design of an appropriate framework for managing the dialogue interaction. For system-initiative designs, it is usually possible to define the dialogue flow in terms of a finite interconnected graph encoding a dialogue state sequence, often referred to as a call-flow diagram. Each state encodes an action to be taken and conditions under which to move to a set of possible next states. However, developers are realizing that even for such scripted systems an explicit enumeration of all possible states and state transitions can quickly become unwieldy for systems with any degree of complexity. Modest extensions to this basic design to support augmented transition networks (ATNs) can lead to a great deal more sharing of common subdialogues among multiple situations, somewhat analogous to subroutine calls in programming languages [35.70].

To handle flexible dialogue interaction, the most straightforward technique is a so-called *electronic-form* (e-form)-based dialogue management strategy. The e-form specifies a set of slots to be filled in, and an agenda specifies the order in which to prompt the user for any missing values [35.71]. An important step towards further generalization can be achieved by decomposing the problem into a static declarative table describing the domain knowledge and a temporally ordered script controlling the turn-by-turn operations [35.72, 73]. The declarative table specifies the entities, attributes, and goals of the domain, while the script executes a sequence of actions based on a specification of Boolean or arithmetic conditions. The conditions test variables in a dialogue state table, which are updated at each turn based on the user query and the system reply. Research at Carnegie Mellon on dialogue management has similarly migrated from the notion of a script-based strategy to an agenda-based strategy, the basic distinction being that the agenda specifies a goal plan in terms of a hierarchy without the explicit temporal order inherent in a script, thus allowing greater flexibility [35.74, 75].

Another very powerful approach to dialogue management, so-called plan-based systems [35.76], is emerging out of the artificial intelligence (AI) community. An AI-style approach to dialogue management also characterizes the information state update (ISU) strategy described in [35.25]. Context information is collected into a central data structure containing, principally, a *dialogue move tree* capturing conversational threads and an *activity tree* representing the past and planned activities. It will be interesting to see if these systems can converge in design with the computer-science-based approaches adopted in the speech community.

35.2 Development Issues

Spoken dialogue systems require first and foremost the availability of high-performance human language technology components such as speech recognition and language understanding. However, the development of these systems also demands that we pay close attention to a host of other issues. While many of these issues may have little to do with human language technologies per se, they are nonetheless crucial to successful system development. In this section, we will outline two of these development issues.

35.2.1 Data Collection

Developing spoken dialogue systems is a classic chicken-and-egg problem. In order to develop the system capabilities, one needs to have a large corpus of data for system development, training, and evaluation. In order to collect data that reflect actual usage, one needs to have a system that users can speak to. Such a corpus is critical for techniques that exploit machine learning, but even when the grammar is manually developed, it is impossible for linguists to imagine the rich variety of user utterances that occur in natural usage.

Figure 35.2 illustrates a typical cycle of system development. For a new domain or language, one must first develop some limited natural language capabilities, thus enabling an *experimenter-in-the-loop*, or *wizard-of-oz*, data collection paradigm, in which an experimenter types the spoken sentences to the system, after removing spontaneous speech artifacts. This process has the advantage of eliminating potential recognition errors. The resulting data are then used for the development and training of the speech recognition and natural language components. As these components begin to mature, it becomes feasible to collect more data using the *system-in-the-loop,* or *wizardless*, paradigm, which is both more realistic and more cost effective. Performance evalua-

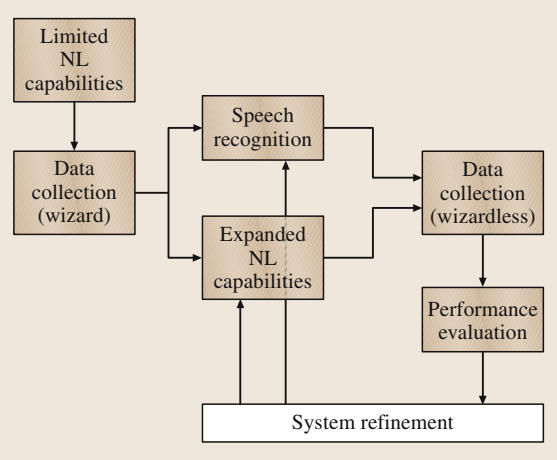


Fig. 35.2 Illustration of data collection procedures

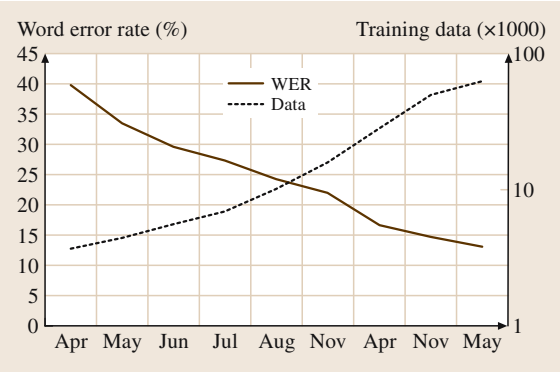


Fig. 35.3 Comparison of recognition performance and the number of utterances collected from real users over time in the MIT weather domain. Note that the x-axis has a non-linear time scale, reflecting the time when new versions of the recognizer were released. WER = word error rate.

tion using newly collected data will facilitate system refinement.

The means and scale of data collection for system development and evaluation has evolved considerably over the last decade. This is true for both the speech recognition and speech understanding communities, and can be seen in many of the systems in the recent automatic railway information systems for Europe (ARISE) project [35.77], and elsewhere. At MIT, for example, the VOYAGER urban navigation system was developed in 1989 by recruiting 100 subjects to come to our laboratory and ask a series of questions to an initial wizard-based system [35.7]. In contrast, the data collection procedure for the more-recent JUPITER weather information system consists of deploying a publicly available system, and recording the interactions [35.14]. There are large differences in the number of queries, the number of users, and the range of issues which the data provide. By using a system-in-the-loop form of data collection, system development and evaluation become iterative procedures. If unsupervised methods were used to augment the system ASR and NLU capabilities, system development could become continuous [35.78].

Figure 35.3 shows, over a two-year period, the cumulative amount of data collected from real users using the MIT JUPITER system and the corresponding word error rates (WER) of our recognizer. Before we made the system accessible through a toll-free number, the WER was about 10% for laboratory-collected data. The WER more than tripled during the first week of data collection. As more data were collected, we were able to build better lexical, language, and acoustic models. As a result, the WER continued to decrease over time. This negative correlation suggests that making the system available to real users is a crucial aspect of system development. If the system can provide real and useful information to users, they will continue to call, thus providing us with a constant supply of useful data. However, in order to get users to actually use the system, it needs to be providing *real* information to the user. Otherwise, there is little incentive for people to use the system other than to play around with it, or to solve toy problem scenarios that may or may not reflect problems of interest to real users.

35.2.2 Evaluation

One of the issues that faces developers of spoken dialogue systems is how to evaluate progress, in order to determine if they have created a usable system. Developers must decide what metrics to use to evaluate their systems to ensure that progress is being made. Metrics can include component evaluations, but should also assess the overall performance of their system.

For systems that conduct a transaction, it is possible to tell whether or not a user has completed a task. In these cases, it is also possible to measure accompanying statistics such as the length of time to complete the task, the number of turns, etc. It has been noted however, that such statistics may not be as important as user satisfaction [35.79, 80]. For example, a spoken dialogue interface may take longer than some alternative, yet users may prefer it due to other factors (less stressful, hands free, etc.). A better form of evaluation might be a measure of whether users liked the system, whether

they called to perform a real task (rather than browsing), and whether they would use it again, or recommend it to others. Evaluation frameworks such as [35.81] attempt to correlate system measurements with user satisfaction, in order to better quantify these effects [35.82].

Although there have been some recent efforts in evaluating language output technologies (e.g., TTS comparisons [35.56]), evaluation for ASR and NLU have been more common since they are more amenable to automatic methods where it is necessary to decide what a correct answer is. ASR evaluation has tended to be the most straightforward, although there are a range of phenomena that are not necessarily obvious how to evaluate (e.g., crosstalk, mumbling, partial words). NLU evaluation can also be performed by comparing some form of meaning representation with a reference. The problem with NLU understanding is that there is no common meaning representation among different research sites, so cross-site comparison becomes difficult. In the DARPA (Defense Advanced Research Projects Agency) spoken language systems (SLS) program for example, the participants ultimately could agree only on comparing to an answer coming from a common database. Unfortunately, this necessarily led to the creation of a large document defining principals of interpretation for all conceivable queries [35.83]. In order to keep the response across systems consistent, systems were restricted from taking the initiative, which is a major constraint on dialogue research.

One way to show progress for a particular system is to perform longitudinal evaluations for recognition and understanding. In the case of JUPITER, as shown in Fig. 35.3, we continually evaluate on standard test sets, which we can redefine periodically in order to keep from tuning to a particular data set [35.14, 84]. Since data continually arrive, it is not difficult to create new sets and re-evaluate older system releases on these new data.

Some systems make use of dialogue context to provide constraints for recognition, for example, favoring candidate hypotheses that mention a date after the system has just asked for a date. Thus, any reprocessing of utterances in order to assess improvements in recognition or understanding performance at a later time need to be able to take advantage of the *same* dialogue context as was present in the original dialogue with the user. To do this, the dialogue context must be recorded at the time of data collection, and reutilized in the subsequent offline processing, in order to avoid giving the original system an unwarranted advantage.

35.3 Historical Perspectives

35.3.1 Large-Scale Government Programs

Research and development of spoken dialogue systems flourished in the 1990s, largely due to government funded research programs in the US and Europe. In the late 1980s the DARPA spoken language systems (SLS) program was initiated in the US [35.83,85], while the Esprit speech understanding and dialog (SUNDIAL) program was underway in Europe [35.86]. The task domains for these two programs were remarkably similar in that both involved database access for travel planning, with the US program concentrating on air travel, and the European program on both air and train travel. All of the systems focused within a narrowly defined area of expertise, and vocabulary sizes were generally limited to several thousand words. By the end of the millennium, these systems could typically run in real time on standard workstations with no additional hardware.

The DARPA SLS program focused entirely on the input side – starting with spoken input and ending with the extraction of the entities from a flight database. By requiring that all system developers use the same database, it has been possible to compare the performance of various systems based on their ability to extract the correct information from the database, using a set of prescribed training and test data, and a set of interpretation guidelines. Unlike the SLS program, however, the European SUNDIAL project aimed at building systems that could be publicly deployed. For this reason, the SUNDIAL project designated dialogue modeling and spoken language generation as integral parts of the research program. As a result, this has led to some interesting advances in Europe in dialogue control mechanisms.

Since the end of the SUNDIAL and SLS programs in 1993, and 1995 respectively, there have been other sponsored programs in spoken dialogue systems. In the European automatic railway information systems for Europe (ARISE) project, participants developed train timetable information systems covering three different languages [35.77]. In the US, a DARPA communicator program [35.87, 88] emphasized dialogue-based interactions incorporating both

speech input and output technologies. One of the properties of this program was that participants used a common system architecture to encourage component sharing across sites [35.89]. Participants in this program developed both their own dialogue domain, and a common complex travel task [35.90].

35.3.2 Some Example Systems

In addition to the research sponsored by these larger programs, there have been many other independent initiatives as well. Some of these systems are research prototypes [35.64, 91, 92], whereas others are actual deployed systems [35.1, 29, 38, 93]. Please see *Zue* and *Glass* [35.94] for a more comprehensive, comparative description of some of them. In the remainder of this section, we will briefly describe a selected subset of the systems to allow readers to gain a sense of the breadth of systems being developed.

There is a strong predominance in the research community on systems supporting database access. Even within this space, the subtopics of air and train travel have dominated [35.6, 64, 93, 95–100], partially as a consequence of the funding opportunities, but also due to the fact that the level of complexity is well matched to research goals, in terms of supporting mixed-initiative dialogue and some degree of planning. The Waxholm project at KTH (the Royal Institute of Technology) in Sweden [35.92, 101] concerns *boat* traffic in the Stockholm archipelago, a slight departure from the main theme. Carnegie Mellon's *Let's Go!* application, providing information on bus schedules within metropolitan Pittsburgh [35.3, 4], is noteworthy because it has been deployed and made available to the general public. This system has a fairly large vocabulary, since there are nearly 15 000 unique bus stops along nearly 2500 routes. The system is accessible via telephone, and receives calls on a regular basis. A recent evaluation experiment showed a similar precipitous drop in performance once the system migrated from the research lab into the public domain as we observed with our Jupiter system. Despite the fact that the raw word error rate on transcripts was 60%, sentence understanding error rate was only 45%, and 44% of the dialogues were successfully completed.

Another system that illustrates migration from research into practical deployment is AT&T's *How May I Help You?* application [35.38]. This system launched a whole new approach to language understanding, where the goal was simply to map a user's first sentence after the *How may I help you?* prompt into one of several semantic classes, in order to route the call to the correct operator or automated subsystems, which can in turn be machine-initiated dialogue systems. A statistical approach to understanding seemed essential for this task, but it has led to widespread adoption of statistical methods for a broader space of applications.

Two other popular topics for dialogue systems are directory assistance [35.26, 29, 102] and map-based navigation [35.7, 103, 104]. While directory assistance is an easier task in terms of dialogue planning than flight scheduling, it can be associated with a very-large-vocabulary name-recognition problem, leading to research on error recovery subdialogues. City guide domains are typically multimodal and critically depend on map displays to provide visual feedback to the user.

35.4 New Directions

35.4.1 User Simulation

One of the most critical barriers to widespread deployment of spoken dialogue systems is the costly development cycle that is critically dependent upon interaction with human users to expose system weaknesses, guide grammar development, and provide domain-specific language model data. Most problematic is the initial stage when the system inadequacies are so apparent that potential users quickly become discouraged and lose interest. One promising strategy to accelerate system development at the early stage involves a user simulation paradigm in which a computer role plays the user side of a simulated conversation. At each turn, the simulated user interprets each system response and decides, using a simple stochastic model, what to say next. In the case of a scripted dialogue this can be pretty straightforward: when the system asks for a date, provide one. Flexible dialogue design poses a more-challenging task to the simulator, but the result is a more-interesting dialogue with greater potential to reveal inadequacies in the dialogue manager that can then be repaired by developers.

The simulated user response can be represented directly in terms of attribute–value pairs, thus bypassing speech recognition and natural language understanding.

Amazon EX1010 Page 727

Alternatively, it can be converted into a well-formed sentence, using either a formal language generation system or a template-based approach, associating each set of attribute–value pairs with a set of sentence patterns to choose from. The most complete user simulation model involves speech synthesis, producing a waveform that can then be directly processed through the speech recognizer to introduce recognition errors. Recognition errors can also be simulated stochastically.

One of the earliest attempts at user simulation was the work by *Levin* et al. [35.105, 106] where the simulator was used mostly to tune the parameters of a dialogue manager in the flight domain. Simulation experiments by *Lopez-Gozar* [35.107] allowed them to evaluate two different recognition front-ends and two different user confirmation strategies, within a fast-food domain. *Chung* [35.108] developed a user simulation in a flexible restaurant guide domain and showed how it could pinpoint design errors and allow developers to improve system behavior, for example in offering alternatives when the specified constraints yielded no matches from the database. *Chung* et al. [35.109] extended this work to include automatic generation of language-model data for the recognizer, harvested from simulation runs. *Filisko* and *Seneff* [35.110] exploited user simulation to model both compliant and noncompliant behavior in an error recovery subdialogue, for large-vocabulary city-name acquisition tasks. Recently, user simulation has played a critical role in research in Markov decision processes for dialogue modeling [35.111].

35.4.2 Machine Learning and Dialogue Management

Statistical methods and machine learning techniques have been extremely successful in a number of different technologies related to speech processing, e.g., speech recognition, language understanding, information extraction, and language translation. However, these techniques have been unusually slow to penetrate the dialogue management domain. The main reason is that such techniques depend critically on large corpora of manually annotated data, and, for dialogue systems, this translates into the need for detailed, annotated log files for thousands of user dialogues with a pre-existing system. Another roadblock has been uncertainty in how to formulate a tractable machine learning paradigm for the highly heuristic task of dialogue management.

These barriers are beginning to break down in recent years, however, and the community witnessed its first workshop devoted to *Statistical and Empirical Approaches for Spoken Dialogue Systems* in the summer of 2006 [35.112]. A powerful method for side-stepping the data collection issue is to collect synthetic data from user simulation runs, although one ultimately has to confirm that the results carry over to real users. The beauty of simulation is that the user's intended actions are known, so that no manual annotation is required. And the developer can simulate compliant or noncompliant behavior, known or unknown vocabulary choices, etc., in controlled experiments, generating enormous amounts of training data effortlessly.

Levin et al. [35.113] were pioneers in introducing machine learning techniques combined with user simulation, in experiments where they showed that a reasonable policy could be learned in the flight domain through trial and error. Bayesian reinforcement learning, Markov decision processes (MDPs), and Bayesian belief networks (BBNs) [35.114] have evolved into the more-complex partially observable Markov decision process (POMDP) model [35.115], which is beginning to catch on as a method to replace heuristic rules governing dialogue management decisions. However, it is as yet unclear whether these techniques can scale to realistic domains.

Machine learning techniques can in many cases be put to better use in solving a restricted part of the dialogue management problem rather than attempting to replace an existing dialogue manager completely. For example, user simulation data can be used to create a decision tree [35.116], and subsequently to produce a rule-based system for deciding whether to invoke implicit or explicit confirmation, or to seek a spoken spelling of a potentially unknown word [35.110]. *Bohus* and *Rudnicky* [35.27] have developed a data-driven approach that integrates information across multiple turns to aid in the decision process for implicit versus explicit confirmation. Straightforward machine learning techniques have also been successfully applied to the task of deciding whether to reject a user's utterance due to suspicions of gross recognition error [35.25]. *Johnston* and *Bangalore* [35.117] borrowed techniques from the statistical machine translation community to train a set of *edit rules*, leading to more-robust handling of multimodal inputs. Through an automated intent-mapping algorithm, *Tur* [35.118] has shown how annotated data from one domain can be used to build an initial speech understanding model for a new but related domain.

35.4.3 Portability

Creating a robust, mixed-initiative dialogue system can require a tremendous amount of effort on the part of researchers. In order for this technology to ultimately be successful, the process of porting existing technology to new domains and languages must be made easier.

Over the past few years, many research efforts have been directed towards making it easier for non-experts to create new domains. Systems which modularize their dialogue manager try to take advantage of the fact that a dialogue can often be broken down into a smaller set of subdialogues (e.g., dates, addresses), in order to make it easier to construct dialogue interaction for a new domain [35.1, 119, 120]. Researchers at Oregon Graduate Institute have introduced rapid-development toolkits for creating spoken dialogue systems [35.119], which have been used by students to create their own systems [35.121]. MIT's SpeechBuilder framework [35.122] allows novice developers to configure and compile a dialogue system using an intuitive Web-based interface. At Carnegie Mellon University, researchers are exploring ideas to formalize dialogue systems around the concept of *speech graffiti*, arguing that users would be willing to comply to a simplified sublanguage of English in order to gain the advantage of substantially improved recognition and understanding performance [35.123]. A grammar is automatically derived from an appliance specification script, making it relatively easy for developers to create systems based on the *speech graffiti* language syntax. To date, these approaches have been applied only to directed dialogue strategies, and their utility has not been demonstrated beyond toy systems. Much more research is needed in this area before complex dialogue strategies will be able to generalize to new domains with minimal effort and expertise.

Currently, the development of speech recognition and language understanding technologies has been domain and language specific, requiring a large amount of annotated training data. However, it may be costly, or even impossible, to collect a large amount of training data for certain applications or languages. Therefore, an important research topic is to produce a conversational system in a new domain and language given at most a small amount of domain-specific training data. To achieve this goal, the algorithmic aspects of the system must be cleanly separated from the application-specific aspects. Automatic or semiautomatic methods will greatly benefit the acquisition of the acoustic models, language models, grammars, semantic structures for language understanding, and dialogue models required by a new application. Real deployment of multilingual spoken language technology cannot take place without adequately addressing this issue.

35.4.4 Multimodal, Multidomain, and Multilingual Application Development

In this section, we mention several examples of multimodal, multilingual, and multidomain systems representative of the state of the art in these dimensions. Multimodality is an especially exciting topic recently due to the widespread adoption of Internet telephony, offering the potential for effortless interaction at a Web page via microphone input at the computer. Our own research on a multimodal restaurant-domain application has allowed us to explore a number of design issues related to both multimodal input (e.g., drawing a line along a street on a map while asking, *What restaurants are on this street?*), and multimedia output, interfacing with Google maps to provide richly informative visual feedback, which in turn reduces the importance of verbal summarization [35.11]. To provide an integrated search in multimodal understanding, *Johnston* and *Bangalore* [35.124] developed a novel strategy for tight coupling between speech and mouse clicks via a joint interpretation within a weighted finite-state transducer (FST) framework. Another example of a rather unique multimodal dialogue system is WITAS (Wallenberg Laboratory for Information Technology and Autonomous Systems) [35.125], which allows the user to interact with a simulated robotic helicopter, via speech and mouse clicks on a map. The user can instruct the virtual helicopter to fly to different locations, follow vehicles, and deliver goods. A system supporting E-mail applications, AthosMail, which is both multimodal (speech and touch tone) and multilingual (English, Finnish, and Swedish) is described in [35.126]. This system allows users to navigate a set of email messages by voice to read, mark, and delete selected messages.

A challenge in spoken dialogue systems is the notion of supporting multiple domains under a single access point. This introduces the requirement of navigation among the different domains. In our own research [35.127] we have configured systems that allow users to discuss weather, flights, flight status, traffic, and a city guide in a single phone call, although users must refer to each domain by name to explicitly request a domain switch. A system that supports interaction in Chinese in two domains, weather and stock quotes,

is described in [35.128], which addresses both dialogue portability issues and seamless domain switching. A domain-independent goal-oriented table-driven approach to dialogue management is accomplished by externalizing domain-dependent knowledge in a separate table. Another table determines which domain to invoke to answer each question, turn-by-turn, with high accuracy.

35.5 Concluding Remarks

In this paper, we have attempted to outline some of the important research challenges that must be addressed before spoken language technologies can be put to pervasive use. The timing for the development of human language technology is particularly opportune, since the world is mobilizing to develop the information highway that will be the backbone of future economic growth. Human language technology will play a central role in providing an interface that will drastically change the human–machine communication paradigm from *programming* to *conversation*. It will enable users to access, process, manipulate, and absorb a vast amount of information efficiently. While much work needs to be done, the progress made collectively by the community thus far gives us every reason to be optimistic about fielding such systems, albeit with limited capabilities, in the near future.

References

35.1 E. Barnard, A. Halberstadt, C. Kotelly, M. Phillips: A consistent approach to designing spoken-dialog systems, Proc. ASRU Workshop (ASRU, Keystone 1999)

35.2 S.J. Boyce: Natural spoken dialogue systems for telephony applications, Commun. ACM **43**(9), 29–34 (2000)

35.3 A. Raux, B. Langner, D. Bohus, A. Black, M. Eskenazi: Let's Go public! Taking a spoken dialog system to the real world, Proc. Interspeech (2005) pp. 885–888

35.4 A. Raux, B. Langner, A. Black, M. Eskenazi: Let's Go: Improving spoken dialog systems for the elderly and non-natives, Proc. Interspeech (2003) pp. 753–756

35.5 G. Flammia: Discourse Segmentation of Spoken Dialogue: An Empirical Approach. Ph.D. Thesis (MIT, Cambridge 1998)

35.6 S. Seneff, R. Lau, J. Glass, J. Polifroni: The mercury system for flight browsing and pricing, MIT Spoken Language System Group Annual Progress Report (1999) pp. 23–28

35.7 J. Glass, G. Flammia, D. Goodine, M. Phillips, J. Polifroni, S. Sakai, S. Seneff, V. Zue: Multilingual spoken-language understanding in the MIT voyager system, Speech Commun. **17**, 1–18 (1995)

35.8 R. Lau, G. Flammia, C. Pao, V. Zue: WebGalaxy – Integrating spoken language and hypertext navigation, Proc. Eurospeech (1997) pp. 883–886

35.9 H. Meng, S. Busayapongchai, J. Glass, D. Goddeau, L. Hetherington, E. Hurley, C. Pao, J. Polifroni, S. Seneff, V. Zue: A conversational system in the automobile classifieds domain, Proc. ICSLP (1996) pp. 542–545

35.10 S. Seneff, V. Zue, J. Polifroni, C. Pao, L. Hetherington, D. Goddeau, J. Glass: The preliminary development of a displayless Pegasus system, Proc. ARPA Spoken Language Technology Workshop (1995) pp. 212–217

35.11 A. Gruenstein, S. Seneff, C. Wang: Scalable and portable web-based multimodal dialogue interaction with geographical databases, Proc. Interspeech (2006)

35.12 W. Wang, J. Glass, H. Meng, J. Polifroni, S. Seneff, V. Zue: Yinhe: A Mandarin Chinese version of the galaxy system, Proc. Eurospeech (1997) pp. 351–354

35.13 V. Zue, S. Seneff, J. Polifroni, M. Phillips, C. Pao, D. Goddeau, J. Glass, E. Brill: Pegasus: A spoken language interface for on-line air travel planning, Speech Commun. **15**, 331–340 (1994)

35.14 V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. Hazen, L. Hetherington: JUPITER: A telephone-based conversational interface for weather information, IEEE Trans. Speech Audio. Process. **8**(1), 85–96 (2000)

35.15 P. Dalsgaard, L. Larsen, I. Thomsen (Eds.): *Proc. ESCA Tutorial and Research Workshop on Spoken Dialogue Systems: Theory and Application* (1995)

35.16 P. Cohen, M. Johnson, D. McGee, S. Oviatt, J. Clow, I. Smith: The efficiency of multimodal interaction: A case study, Proc. ICSLP (1998) pp. 249–252

35.17 S. Seneff, D. Goddeau, C. Pao, J. Polifroni: Multimodal discourse modelling in a multi-user

multi-domain environment, Proc. ICSLP (1996) pp. 188–191
35.18 D. Massaro: *Perceiving Talking Faces: From Speech Perception to a Behavioral Principle* (MIT Press, Cambridge 1997)
35.19 W. Ward: Modelling non-verbal sounds for speech recognition, Proc. DARPA Workshop on Speech and Natural Language (1989) pp. 47–50
35.20 J. Butzberger, H. Murveit, M. Weintraub: Spontaneous speech effects in large vocabulary speech recognition applications, Proc. ARPA Workshop on Speech and Natural Language (1992) pp. 339–344
35.21 L. Hetherington, V. Zue: New words: Implications for continuous speech recognition, Proc. Eurospeech (1991) pp. 475–931
35.22 A. Asadi, R. Schwartz, J. Makhoul: Automatic modelling for adding new words to a large vocabulary continuous speech recognition system, Proc. ICASSP (1991) pp. 305–308
35.23 G. Chung, S. Seneff, C. Wang: Automatic acquisition of names using speak and spell mode in spoken dialogue systems, Proc. HLT-NAACL (2003) pp. 197–200
35.24 C. Pao, P. Schmid, J. Glass: Confidence scoring for speech understanding systems, Proc. ICSLP (1998) pp. 815–818
35.25 M. Gabsdil, O. Lemon: Combining acoustic and pragmatic features to predict recognition performance in spoken dialogue systems, Proc. ACL (2004)
35.26 A. Kellner, B. Rueber, H. Schramm: Using combined decisions and confidence measures for name recognition in automatic directory assistance systems, Proc. ICSLP (1998) pp. 2859–2862
35.27 D. Bohus, A. Rudnicky: Constructing accurate beliefs in spoken dialog systems, Proc. ASRU (2005) pp. 272–277
35.28 V. Souvignier, A. Kellner, B. Rueber, H. Schramm, F. Seide: The thoughtful elephant: Strategies for spoken dialogue systems, IEEE T. Speech Audi. P. **8**(1), 51–62 (2000)
35.29 R. Billi, R. Canavesio, C. Rullent: Automation of Telecom Italia Directory Assistance Service: Field trial results, Proc. IVTTA (1998) pp. 11–16
35.30 R.J. Lippmann: Speech recognition by humans and machines, Speech Commun. **22**(1), 1–15 (1997)
35.31 R. Bobrow, R. Ingria, D. Stallard: Syntactic and semantic knowledge in the DELPHI unification grammar, Proc. DARPA Speech and Natural Language Workshop (1990) pp. 230–236
35.32 S. Seneff: TINA: A natural language system for spoken language applications, Comput. Linguist. **18**(1), 61–86 (1992)
35.33 J. Dowding, J. Gawron, D. Appelt, J. Bear, L. Cherny, R. Moore, D. Moran: Gemini: A natural language system for spoken language understanding, Proc. ARPA Workshop on Human Language Technology (1993) pp. 21–24
35.34 W. Ward: The CMU air travel information service: Understanding spontaneous speech, Proc. ARPA Workshop on Speech and Natural Language (1990) pp. 127–129
35.35 E. Jackson, D. Appelt, J. Bear, R. Moore, A. Podlozny: A template matcher for robust NL interpretation, Proc. DARPA Speech and Natural Language Workshop (1991) pp. 190–194
35.36 S. Seneff: Robust parsing for spoken language systems, Proc. ICASSP (1992) pp. 189–192
35.37 D. Stallard, R. Bobrow: Fragment processing in the DELPHI system, Proc. DARPA Speech and Natural Language Workshop (1992) pp. 305–310
35.38 A. Gorin, G. Riccardi, J. Wright: How may I help you?, Speech Commun. **23**, 113–127 (1997)
35.39 S. Miller, R. Schwartz, R. Bobrow, R. Ingria: Statistical language processing using hidden understanding models, Proc. ARPA Speech and Natural Language Workshop (1994) pp. 278–282
35.40 Y. Chow, R. Schwartz: The N-Best Algorithm: An efficient procedure for finding top N sentence hypotheses, Proc. ARPA Workshop on Speech and Natural Language (1989) pp. 199–202
35.41 L. Hetherington, M. Phillips, J. Glass, V. Zue: A word network search for continuous speech recognition, Proc. Eurospeech (1993) pp. 1533–1536
35.42 L. Mangu, E. Brill, A. Stolcke: Finding consensus among words: Lattice-based word error minimization, Proc. Eurospeech (1999)
35.43 D. Goodine, S. Seneff, L. Hirschman, M. Phillips: Full integration of speech and language understanding in the MIT spoken language system, Proc. Eurospeech (1991) pp. 845–848
35.44 D. Goddeau: Using probabilistic shift-reduce parsing in speech recognition systems, Proc. ICSLP (1992) pp. 321–324
35.45 W. Ward: Integrating semantic constraints into the SPHINX-II recognition search, Proc. ICASSP (1994) pp. 17–20
35.46 R. Moore, D. Appelt, J. Dowding, J. Gawron, D. Moran: Combining linguistic and statistical knowledge sources in natural-language processing for ATIS, Proc. ARPA Spoken Language Systems Workshop (1995) pp. 261–264
35.47 E. Reiter, R. Dale: *Building Natural Language Generation Systems* (Cambridge Univ. Press, Cambridge 2000)
35.48 D. McDonald, L. Bolc (Eds.): *Natural Language Generation Systems (Symbolic Computation Artificial Intelligence)* (Springer, Heidelberg, Berlin 1998)
35.49 J. Glass, J. Polifroni, S. Seneff: Multilingual language generation across multiple domains, Proc. ICSLP (1994)
35.50 A. Oh: Stochastic Natural Language Generation for Spoken Dialog Systems. M.S. Thesis (CMU, Mt. Pleasant 2000)

Amazon EX1010 Page 731

35.51 V. Demberg, J. Moore: Information presentation in spoken dialogue systems, Proc. EACL (2006)

35.52 D. Klatt: Review of text-to-speech conversion for English, J. Acoust. Soc. Am. **82**(3), 737–793 (1987)

35.53 Y. Sagisaka, N. Kaiki, N. Iwahashi, K. Mimura: ATR ν -talk speech synthesis system, Proc. ICSLP (1992) pp. 483–486

35.54 M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, A. Syrdal: The AT&T next-gen TTS system, Proc. ASA (ASA, Berlin 1999)

35.55 X. Huang, A. Acero, J. Adcock, H.W. Hon, J. Goldsmith, J. Liu, M. Plumpe: Whistler: A trainable text-to-speech system, Proc. ICSLP (1996) pp. 2387–2390

35.56 J. van Santen, L. Pols, M. Abe, D. Kahn, E. Keller, J. Vonwiller: Report on the 3rd ESCA TTS workshop evaluation procedure, Proc. 3rd ESCA Workshop on Speech Synthesis (1998) pp. 329–332

35.57 K. McKeown, S. Pan, J. Shaw, D. Jordan, B. Allen: Language generation for multimedia healthcare briefings, Proc. Applied Natural Language Proc (1997)

35.58 A. Black, K. Lenzo: Limited domain synthesis, Proc. ICSLP (2000)

35.59 R. Sproat, A. Hunt, M. Ostendorf, P. Taylor, A. Black, K. Lenzo, M. Edgington: Sable: A standard for TTS markup, Proc. ICSLP (1998) pp. 1719–1722

35.60 B. Grosz, C. Sidner: Plans for discourse, Intentions in Communication (MIT Press, 1990)

35.61 D. Thomson, J. Wisowaty: User confusion in natural language services, Proc. ESCA Workshop on Interactive Dialogue in Multi-Modal Systems (1999) pp. 189–196

35.62 D. Sadek: Design considerations on dialogue systems: From theory to technology – The case of Artimis, Proc. ESCA Workshop on Interactive Dialogue in Multi-Modal Systems (1999) pp. 173–188

35.63 L. Boves, E. den Os: Applications of speech technology: Designing for usability, Proc. IEEE Worshop on ASR and Understanding (1999) pp. 353–362

35.64 J. Allen, L. Schubert, G. Ferguson, P. Heeman, C. Hwang, T. Kato, W. Light, N. Martin, B. Miller, M. Poesio, D. Traum: The TRAINS project: A case study in defining a conversational planning agent, J. Exp. Theor. Artif. Intell. **7**, 7–48 (1995)

35.65 M. Denecke, A. Waibel: Dialogue strategies guiding users to their communicative goals, Proc. Eurospeech (1997) pp. 2227–2230

35.66 L. Devillers, H. Bonneau-Maynard: Evaluation of dialog strategies for a tourist information retrieval system, Proc. ICSLP (1998) pp. 1187–1190

35.67 S. Rosset, S. Bennacef, L. Lamel: Design strategies for spoken language dialog systems, Proc. Eurospeech (1999) pp. 1535–1538

35.68 A. Rudnicky, E. Thayer, P. Constantinides, C. Tchou, R. Shern, K. Lenzo, W. Xu, A. Oh: Creating natural dialogs in the Carnegie Mellon communicator system, Proc. Eurospeech (1999) pp. 1531–1534

35.69 E. Filisko, S. Seneff: Error detection and recovery in spoken dialogue systems, Proc. Workshop on Spoken Language Understanding for Conversational Systems and Higher Level Linguistic Knowledge for Speech Processing (2004)

35.70 G. Di Fabbrizio, C. Lewis: Florence: A dialogue manager framework for spoken dialogue systems, Proc. ICSLP (2004)

35.71 D. Goddeau, H. Meng, J. Polifroni, S. Seneff, S. Busayapongchai: A form-based dialogue manager for spoken language applications, Proc. ICSLP (1996) pp. 701–704

35.72 R. Pieraccini, E. Levin, W. Eckert: AMICA: The AT&T mixed initiative conversational architecture, Proc. Eurospeech (1997) pp. 1875–1879

35.73 S. Seneff: Response planning and generation in the MERCURY flight reservation system, Comput. Speech Lang., Vol. 16 (2002) pp. 283–312

35.74 P. Constantinides, S. Hansma, A. Rudnicky: A schema-based approach to dialog control, Proc. ICSLP (1998) pp. 409–412

35.75 X. Wei, A. Rudnicky: Task-based dialog management using an agenda, Proc. ANLP/NAACL Workshop on Conversational Systems (2000)

35.76 J. Allen, D. Byron, M. Dzikovska, G. Ferguson, L. Galescu, A. Stent: An architecture for a generic dialogue shell, Nat. Lang. Eng. **6**(3), 1–16 (2000)

35.77 E. den Os, L. Boves, L. Lamel, P. Baggia: Overview of the ARISE project, Proc. Eurospeech (1999) pp. 1527–1530

35.78 T. Kemp, A. Waibel: Unsupervised training of a speech recognizer: Recent experiments, Proc. Eurospeech (1999) pp. 2725–2728

35.79 A. Rudnicky, M. Sakamoto, J. Polifroni: Evaluating spoken language interaction, Proc. DARPA Speech and Natural Language Workshop (1989) pp. 150–159

35.80 L. Lamel, S. Bennacef, J.L. Gauvain, H. Dartigues, J. Temem: User evaluation of the mask kiosk, Proc. ICSLP (1998) pp. 2875–2878

35.81 M. Walker, D. Litman, C. Kamm, A. Abella: PARADISE: A general framework for evaluating spoken dialogue agents, Proc. ACL/EACL (1997) pp. 271–280

35.82 M. Walker, J. Boland, C. Kamm: The utility of elapsed time as a usability metric for spoken dialogue systems, Proc. ASRU Workshop (ASRU, Keystone 1999) pp. 1167–1170

35.83 L. Hirschman: Multi-site data collection for a spoken language corpus, Proc. DARPA Workshop on Speech and Natural Language (1992) pp. 7–14

35.84 J. Polifroni, S. Seneff, J. Glass, T. Hazen: Evaluation methodology for a telephone-based conversational system, Proc. Int. Conf. on Lang. Resources and Evaluation (1998) pp. 42–50

35.85 D. Pallett, J. Fiscus, W. Fisher, J. Garafolo, B. Lund, A. Martin, M. Pryzbocki: Benchmark tests for the ARPA spoken language program, Proc. ARPA Spo-

Amazon EX1010 Page 732

ken Language Systems Technology Workshop (1995) pp. 5–36
35.86 J. Peckham: A new generation of spoken dialogue systems: Results and lessons from the SUNDIAL project, Proc. Eurospeech (1993) pp. 33–40
35.87 M. Walker, J. Aberdeen, J. Boland, E. Bratt, J. Garofolo, L. Hirschman, A. Le, S. Lee, S. Narayanan, K. Papineni, B. Pellom, J. Polifroni, A. Potamianos, P. Prabhu, A. Rudnicky, G. Sanders, S. Seneff, D. Stallard, S. Whittaker: DARPA communicator dialog travel planning systems: The June 2000 data collection, Proc. Eurospeech (2001)
35.88 M. Walker, A. Rudnicky, R. Prasad, J. Aberdeen, E.O. Bratt, J. Garofolo, H. Hastie, A. Le, B. Pellom, A. Potamianos, R. Passonneau, S. Roukos, G. Sanders, S. Seneff, D. Stallard: DARPA communicator: Cross-system results for the 2001 evaluation, Proc. ICSLP (2002) pp. 273–276
35.89 S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, V. Zue: GALAXY-II: A reference architecture for conversational system development, Proc. ICSLP (1998) pp. 931–934
35.90 M. Eskenazi, A. Rudnicky, K. Gregory, P. Constantinides, R. Brennan, C. Bennett, J. Allen: Data collection and processing in the Carnegie Mellon communicator, Proc. Eurospeech (1999) pp. 2695–2698
35.91 D. Jurafsky, C. Wooters, G. Tajchman, J. Segal, A. Stolcke, E. Fosler, N. Morgan: The Berkeley restaurant project, Proc. ICSLP (1994) pp. 2139–2142
35.92 M. Blomberg, R. Carlson, K. Elenius, B. Granstrom, J. Gustafson, S. Hunnicutt, R. Lindell, L. Neovius: An experimental dialogue system: Waxholm, Proc. Eurospeech (1993) pp. 1867–1870
35.93 H. Aust, M. Oerder, F. Seide, V. Steinbiss: The Philips automatic train timetable information system, Speech Commun. **17**, 249–262 (1995)
35.94 V. Zue, J. Glass: Conversational interfaces: Advances and challenges, Proc. IEEE, Vol. 88 (2000), Special Issue on Spoken Language Processing
35.95 J. Kowtko, P. Price: Data collection and analysis in the air travel planning domain, Proc. DARPA Speech and Natural Language Workshop (1989)
35.96 G. Castagnieri, P. Baggia, M. Danieli: Field trials of the Italian ARISE train timetable system, Proc. IVTTA (1998) pp. 97–102
35.97 M. Cohen, Z. Rivlin, H. Bratt: Speech recognition in the ATIS domain using multiple knowledge sources, Proc. DARPA Spoken Language Systems Technology Workshop (1995) pp. 257–260
35.98 L. Lamel, S. Rosset, J.L. Gauvain, S. Bennacef, M. Garnier-Rizet, B. Prouts: The LIMSI ARISE system, Proc. IVTTA (1998) pp. 209–214
35.99 A. Sanderman, J. Sturm, E. den Os, L. Boves, A. Cremers: Evaluation of the Dutch train timetable information system developed in the ARISE project, Proc. IVTTA (1998) pp. 91–96
35.100 J. Sturm, E. den Os, L. Boves: Dialogue management in the Dutch ARISE train timetable information system, Proc. Eurospeech (1999) pp. 1419–1422
35.101 R. Carlson, S. Hunnicutt: Generic and domain- specific aspects of the Waxholm NLP and dialogue modules, Proc. ICSLP (1996) pp. 677–680
35.102 B. Buntschuh: VPQ: A spoken language interface to large scale directory information, Proc. ICSLP (1998) pp. 2863–2866
35.103 M. Johnston, S. Bangalore, G. Vasireddy, A. Stent, P. Ehlen, M. Walker, S. Whattaker, P. Maloor: MATCH: An architecture for multimodal dialogue systems, Proc. ACL (2002) pp. 376–383
35.104 S. Oviatt: Multimodal interfaces for dynamic interactive maps, Proc. CHI (1996) pp. 95–102
35.105 E. Levin, R. Pieraccini, W. Eckert: A stochastic model of human–machine interaction for learning dialogue strategies, IEEE T. Speech Audi. P. **8**(1), 11–23 (2000)
35.106 W. Eckert, E. Levin, R. Pieraccini: User modelling for spoken dialogue system evaluation, Proc. ASRU (1997) pp. 80–87
35.107 R. Lopez-Cozar, A. De la Torre, J.C. Segura, A.J. Rubio: Assessment of dialogue systems by means of a new simulation technique, Speech Commun. **40**, 387–407 (2003)
35.108 G. Chung: Developing a flexible spoken dialog system using simulation, Proc. ACL (2004) pp. 63–70
35.109 G. Chung, S. Seneff, C. Wang: Automatic induction of language model data for a spoken dialogue system, Proc. 6th SIGdial Workshop on Discourse and Dialogue Lisbon (2005)
35.110 E. Filisko, S. Seneff: Learning decision models in spoken dialogue systems via user simulation, Proc. AAAI Workshop: Statistical and Empirical Approaches for Spoken Dialog Systems (2006)
35.111 J. Schatzmann, K. Georgila, S. Young: Quantitative evaluation of user simulation techniques for spoken dialogue systems, Proc. Workshop on Discourse and Dialogue (2005)
35.112 P. Poupart, S. Seneff, J. Williams, S. Young: Cochairs. In: *Statistical and Empirical Approaches for Spoken Dialogue Systems*, ed. by AAAI (AAAI, Menlo Park 2006), Tech. Rep. WS-06-14
35.113 E. Levin, R. Pieraccini, W. Eckert: Using Markov decision process for learning dialogue strategies, Proc. ICASSP (1998) pp. 201–204
35.114 H. Meng, C. Wai, R. Pieraccini: The use of belief networks for mixed-initiative dialog modeling, IEEE T. Speech Audi. P. **11**(6), 757–773 (2003)
35.115 J. Williams, S. Young: Scaling POMDPs for dialog management with composite summary point-based value iteration (CSPBVI), Proc. AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems (2006) pp. 7–12

Amazon EX1010 Page 733

35.116 W. Cohen: Fast effective rule induction, Proc. 12th Int. Conf. Machine Learning (1995) pp. 115–123
35.117 M. Johnston, S. Bangalore: Learning edit machines for robust multimodal understanding, Proc. ICASSP (2006) pp. I617–I620
35.118 G. Tur: Multitask learning for spoken language understanding, Proc. ICASSP (2006) pp. I585–I588
35.119 S. Sutton: Universal speech tools: The CSLU toolkit, Proc. ICSLP (1998) pp. 3221–3224
35.120 J. Glass, S. Seneff: Flexible and personalizable mixed-initiative dialogue systems, HLT-NAACL 2003 Workshop on Research Directions in Dialogue Processing (2003)
35.121 S. Sutton, E. Kaiser, A. Cronk, R. Cole: Bringing spoken language systems to the classroom, Proc. Eurospeech (1997) pp. 709–712
35.122 J. Glass, E. Weinstein, S. Cyphers, J. Polifroni, G. Chung, M. Nakano: A framework for developing conversational user interfaces, Proc. CADUI (2004) pp. 354–365
35.123 T. Harris, R. Rosenfeld: A universal speech interface for appliances, Proc. ICSLP (2004)
35.124 M. Johnston, S. Bangalore: Finite-state multimodal parsing and understanding, Proc. 18th Intrenational Conference on Computational Linguistics (2000) pp. 369–375
35.125 O. Lemon, A. Gruenstein, S. Peters: Collaborative activities and multitasking in dialogue systems, Traitement Automatique des Langues **43**(2), 131–154 (2002)
35.126 M. Turunen, E.-P. Salonen, M. Hartikainen, J. Hakulinen, W. Black, A. Ramsay, A. Funk, A. Conroy, P. Thompson, M. Stairmand, L.K. Jokenen, J. Rissanen, K. Kanto, A. Kerminen, B. Gamback, M. Cheadle, F. Olsson, M. Sahlgren: AthosMail – A – multilingual adaptive spoken dialogue system for e-mail domain, Proc. COLING Workshop (2004)
35.127 S. Seneff, R. Lau, J. Polifroni: Organization, communication, and control in the GALAXY-II conversational system, Proc. Eurospeech (1999) pp. 1271–1274
35.128 W.-T. Hsu, H.-M. Wang, Y.-C. Lin: The design of a multi-domain Chinese dialogue system, Proc. ISCSLP (2002)

Amazon EX1010 Page 734

Part F Speaker Recognition

36 Overview of Speaker Recognition
A. E. Rosenberg, Piscataway, USA
F. Bimbot, Rennes, France
S. Parthasarathy, Sunnyvale, USA

37 Text-Dependent Speaker Recognition
M. Hébert, Montréal, Canada

38 Text-Independent Speaker Recognition
D. A. Reynolds, Lexington, USA
W. M. Campbell, Lexington, USA

Amazon EX1010 Page 735

36. Overview of Speaker Recognition

A. E. Rosenberg, F. Bimbot, S. Parthasarathy

An introduction to automatic speaker recognition is presented in this chapter. The identifying characteristics of a person's voice that make it possible to automatically identify a speaker are discussed. Subtasks such as speaker identification, verification, and detection are described. An overview of the techniques used to build speaker models as well as issues related to system performance are presented. Finally, a few selected applications of speaker recognition are introduced to demonstrate the wide range of applications of speaker recognition technologies. Details of text-dependent and text-independent speaker recognition and their applications are covered in the following two chapters.

36.1 **Speaker Recognition** 725
36.1.1 Personal Identity Characteristics.... 725
36.1.2 Speaker Recognition Definitions.... 726
36.1.3 Bases for Speaker Recognition 726
36.1.4 Extracting Speaker Characteristics from the Speech Signal 727
36.1.5 Applications 728

36.2 **Measuring Speaker Features** 729
36.2.1 Acoustic Measurements..... 729
36.2.2 Linguistic Measurements..... 730

36.3 **Constructing Speaker Models** 731
36.3.1 Nonparametric Approaches 731
36.3.2 Parametric Approaches 732

36.4 **Adaptation** 735

36.5 **Decision and Performance** 735
36.5.1 Decision Rules 735
36.5.2 Threshold Setting and Score Normalization 736
36.5.3 Errors and DET Curves..... 736

36.6 **Selected Applications for Automatic Speaker Recognition** 737
36.6.1 Indexing Multispeaker Data..... 737
36.6.2 Forensics..... 737
36.6.3 Customization: SCANmail..... 738

36.7 **Summary** 739

References 739

36.1 Speaker Recognition

36.1.1 Personal Identity Characteristics

Human beings have many characteristics that make it possible to distinguish one individual from another. Some individuating characteristics can be perceived very readily such as facial features and vocal qualities and behavior. Others, such as fingerprints, iris patterns, and DNA structure are not readily perceived and require measurements, often quite complex measurements, to capture distinguishing characteristics. In recent years biometrics has emerged as an applied scientific discipline with the objective of automatically capturing personal identifying characteristics and using the measurements for security, surveillance, and forensic applications [36.1]. Typical applications using biometrics secure transactions, information, and premises to authorized individuals. In surveillance applications, the goal is to detect and track a target individual among a set of nontarget individuals. In forensic applications a sample of biometric measurements is obtained from an unknown individual, the perpetrator. The task is to compare this sample with a database of similar measurements from known individuals to find a match.

Many personal identifying characteristics are based on physiological properties, others on behavior, and some combine physiological and behavioral properties. From the point of view of using personal identity characteristics as a biometric for security, physiological characteristics may offer more intrinsic security since they are not subject to the kinds of voluntary variations found in behavioral features. Voice is an example of a biometric that combines physiological and behav-

Amazon EX1010 Page 736

ioral characteristics. Voice is attractive as a biometric for many reasons. It can be captured non-intrusively and conveniently with simple transducers and recording devices. It is particularly useful for remote-access transactions over telecommunication networks. A drawback is that voice is subject to many sources of variability, including behavioral variability, both voluntary and involuntary. An example of involuntary variability is a speaker's inability to repeat utterances precisely the same way. Another example is the spectral changes that occur when speakers vary their vocal effort as background noise increases. Voluntary variability is an issue when speakers attempt to disguise their voices. Other sources of variability include physical voice variations due to respiratory infections and congestion. External sources of variability are especially problematic, including variations in background noise, and transmission and recording characteristics.

36.1.2 Speaker Recognition Definitions

Different tasks are defined under the general heading of speaker recognition. They differ mainly with respect to the kind of decision that is required for each task. In speaker identification a voice sample from an unknown speaker is compared with a set of labeled speaker models. When it is known that the set of speaker models includes all speakers of interest the task is referred to as *closed-set* identification. The label of the best matching speaker is taken to be the identified speaker. Most speaker identification applications are *open-set*, meaning that it is possible that the unknown speaker is not included in the set of speaker models. In this case, if no satisfactory match is obtained, a *no-match* decision is provided.

In a speaker verification trial an identity claim is provided or asserted along with the voice sample. In this case, the unknown voice sample is compared only with the speaker model whose label corresponds to the identity claim. If the quality of the comparison is satisfactory, the identity claim is accepted; otherwise the claim is rejected. Speaker verification is a special case of open-set speaker identification with a one-speaker target set. The speaker verification decision mode is intrinsic to most access control applications. In these applications, it is assumed that the claimant will respond to prompts cooperatively.

It can readily be seen that in the speaker identification task performance degrades as the number of speaker models and the number of comparisons increases. In a speaker verification trial only one comparison is required, so speaker verification performance is independent of the size of the speaker population.

A third speaker recognition task has been defined in recent years in National Institute of Standards and Technology (NIST) speaker recognition evaluations; it is generally referred to as speaker detection [36.2, 3]. The NIST task is an open-set identification decision associated exclusively with conversational speech. In this task an unknown voice sample is provided and the task is to determine whether or not one of a specified set of known speakers is present in the sample. A complicating factor for this task is that the unknown sample may contain speech from more than one speaker, such as in the summed two sides of a telephone conversation. In this case, an additional task called speaker tracking is defined, in which it is required to determine the intervals in the test sample during which the detected speaker is talking. In other applications where the speech samples are multispeaker, speaker tracking has also been referred to as speaker segmentation, speaker indexing, and speaker diarization [36.4–10]. It is possible to cast the speaker segmentation task as an acoustical change detection task without creating models. The time instants where a significant acoustic change occurs are assumed to be the boundaries between different speaker segments. In this case, in the absence of speaker models, speaker segmentation would not be considered a speaker recognition task. However, in most reported approaches to this task some sort of speaker modeling does take place. The task usually includes labeling the speaker segments. In this case the task falls unambiguously under the speaker recognition heading.

In addition to decision modes, speaker recognition tasks can be categorized by the kind of speech that is input. If the speaker is prompted or expected to provide a known text and if speaker models have been trained explicitly for this text, the input mode is said to be text dependent. If, on the other hand, the speaker cannot be expected to utter specified texts the input mode is text independent. In this case speaker models are not trained on explicit texts.

36.1.3 Bases for Speaker Recognition

The principal function associated with the transmission of a speech signal is to convey a message. However, along with the message, additional kinds of information are transmitted. These include information about the gender, identity, emotional state, health, etc. of the speaker. The source of all these kinds of information lie in both physiological and behavioral characteristics.

The physiological features are shown in Fig. 36.1 showing a cross-section of the human vocal tract. The shape of the vocal tract, determined by the position of articulators, the tongue, jaw, lips, teeth, and velum, creates a set of acoustic resonances in response to periodic puffs of air generated by the glottis for voiced sounds or aperiodic excitation caused by air passing through tight constrictions in the vocal tract. The spectral peaks associated with periodic resonances are referred to as speech formants. The locations in frequency and, to a lesser degree, the shapes of the resonances distinguish one speech sound from another. In addition, formant locations and bandwidths and spectral differences associated with the overall size of the vocal tract serve to distinguish the same sounds spoken by different speakers. The shape of the nasal tract, which determines the quality of nasal sounds, also varies significantly from speaker to speaker. The mass of the glottis is associated with the basic fundamental frequency for voiced speech sounds. The average basic fundamental frequency is approximately 100 Hz for adult males, 200 Hz for adult females, and 300 Hz for children. It also varies from individual to individual.

Speech signal events can be classified as segmental or suprasegmental. Generally, segmental refers to the features of individual sounds or segments, whereas suprasegmental refers to properties that extend over several speech sounds. Speaking behavior is associated with the individual's control of articulators for individual speech sounds or segments and also with suprasegmental characteristics governing how individual speech sounds are strung together to form words. Higher-level speaking behavior is associated with choices of words and syntactic units. Variations in fundamental frequency or pitch and rhythm are also higher-level features of the speech signal along with such qualities as breathiness, strength of vocal effort, etc. All of these vary significantly from speaker to speaker.

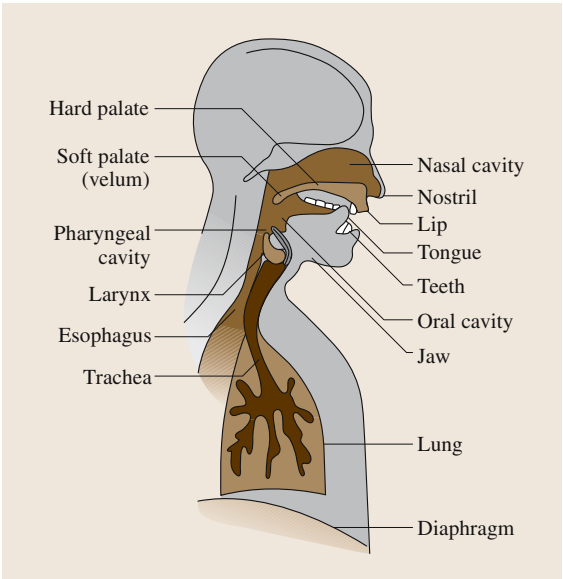


Fig. 36.1 Physiology of the human vocal tract (Reproduced with permission from L. H. *Jamieson* [36.11])

36.1.4 Extracting Speaker Characteristics from the Speech Signal

A perceptual view classifies speech as containing *low-level* and *high-level* kinds of information. Low-level features of speech are associated with the periphery in the brain's perception of speech and are relatively accessible from the speech signal. High-level features are associated with more-central locations in the perception mechanism. Generally speaking, low-level speaker features are easier to extract from the speech signal and model than high-level features. Many such features are associated with spectral correlates such as formant locations and bandwidths, pitch periodicity, and segmental timings. High-level features include the perception of words and their meaning, syntax, prosody, dialect, and idiolect.

It is not easy to extract stable and reliable formant features explicitly from the speech signal. In most instances it is easier to carry out short-term spectral amplitude measurements that capture low-level speaker characteristics implicitly. Short-term spectral measurements are typically carried out over 20–30 ms windows and advanced every 10 ms. Short speech sounds have durations less than 100 ms whereas stressed vowel sounds can last for 300 ms or more. Advancing the time window every 10 ms enables the temporal characteristics of individual speech sounds to be tracked and the 30 ms analysis window is usually sufficient to provide good spectral resolution of these sounds and at the same time short enough to resolve significant temporal characteristics. There are two principal methods of short-term spectral analysis, filter bank analysis and linear predictive coding (LPC) analysis. In filter bank analysis the speech signal is passed through a bank of bandpass filters covering a range of frequencies consistent with the transmission characteristics of the signal. The spacing of the filters can be uniform or, more likely, spaced nonuniformly, consistent with perceptual criteria such as the mel or bark scale [36.12], which provides a linear spacing in frequency below 1000 Hz

and logarithmic spacing above. The output of each filter is typically implemented as a windowed, short-term Fourier transform using fast Fourier transform (FFT) techniques. This output is subject to a nonlinearity and low-pass filter to provide an energy measurement. LPC-derived features almost always include regression measurements that capture the temporal evolution of these features from one speech segment to another. It is no accident that short-term spectral measurements are also the basis for speech recognizers. This is because an analysis that captures the differences between one speech sound and another can also capture the difference between the same speech sound uttered by different speakers, often with resolutions surpassing human perception.

Other measurements that are often carried out are correlated with prosody such as pitch and energy tracking. Pitch or periodicity measurements are relatively easy to make. However, periodicity measurement is meaningful only for voiced speech sounds so it is necessary also to have a detector that can discriminate voiced from unvoiced sounds. This complication often makes it difficult to obtain reliable pitch tracks over long-duration utterances.

Long-term average spectral and fundamental frequency measurements have been used in the past for speaker recognition, but since these measurements provide feature averages over long durations they are not capable of resolving detailed individual differences.

Although computational ease is an important consideration for selecting speaker-sensitive feature measurements, equally important considerations are the stability of the measurements, including whether they are subject to variability, noise, and distortions from one measurement of a speaker's utterances to another. One source of variability is the speaker himself. Features that are correlated with behavior such as pitch contours – pitch measured as a function of time over specified utterances – can be consciously varied from one token of an utterance to another. Conversely, cooperative speakers can control such variability. More difficult to deal with are the variability and distortion associated with recording environments, microphones, and transmission media. The most severe kinds of variability problems occur when utterances used to train models are recorded under one set of conditions and test utterances are recorded under another.

A block diagram of a speaker recognition is shown in Fig. 36.2, showing the basic elements discussed in this section. A sample of speech from an unknown speaker is input to the system. If the system is a speaker verification system, an identity claim or assertion is also input. The speech sample is recorded, digitized, and analyzed. The analysis is typically some sort of short-term spectral analysis that captures speaker-sensitive features as described earlier in this section. These features are compared with prototype features compiled into the models of known speakers. A matching process is invoked to compare the sample features and the model features. In the case of closed-set speaker identification, the match is assigned to the model with the best matching score. In the case of speaker verification, the matching score is compared with a predetermined threshold to decide whether to accept or reject the identity claim. For open-set identification, if the matching score for the best matching model does not pass a threshold test, a no-match decision is made.

36.1.5 Applications

As mentioned, the most widespread applications for automatic speaker recognition are for security. These are typically speaker verification applications intended to control access to privileged transactions or information remotely over a telecommunication network. These are usually configured in a text-dependent mode in which customers are prompted to speak personalized verification phrases such as personal identification numbers

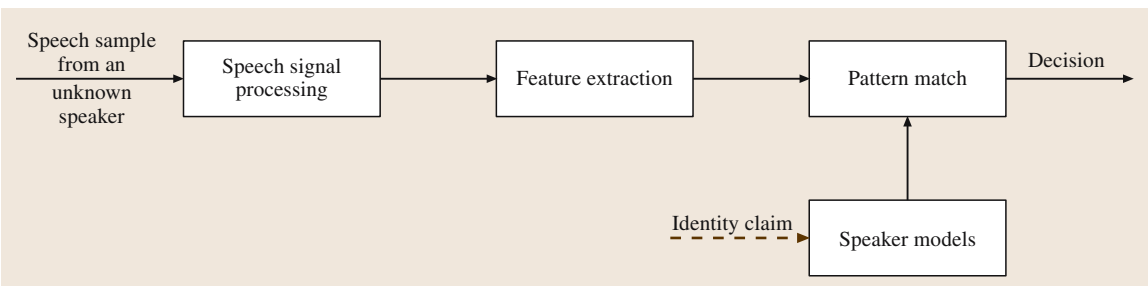


Fig. 36.2 Block diagram of a speaker recognition system

(PINs) spoken as a string of digits. Typically, PIN utterances are decoded using a speaker-independent speech recognizer to provide an identity claim. The utterances are then processed in a speaker recognition mode and compared with speaker models associated with the identity claim. Speaker models are trained by recording and processing prompted verification phrases in an enrollment session.

In addition to security applications, speaker verification may be used to offer personalized services to users. For example, once a speaker verification phrase is authenticated, the user may be given access to a personalized phone book for voice repertory dialing.

A forensic application is likely to be an open-set identification or verification task. A sample of speech exists from an unknown perpetrator. A suspect is required to speak utterances contained in the suspect speech sample in order to train a model. The suspect speech sample is compared both with the suspect and nonsuspect models to decide whether to accept or reject the hypothesis that the suspect and perpetrator voices are the same.

In surveillance applications the input speech mode is most likely to be text independent. Since the speaker may be unaware that his voice is being monitored, he cannot be expected to speak specified texts. The decision task is open-set identification or verification.

Large amounts of multimedia data, including speech, are being recorded and stored on digital media. The existence of such large amounts of data has created a need for efficient, versatile, and accurate data mining tools for extracting useful information content from the data. A typical need is to search or browse through the data, scanning for specified topics, words, phrase, or speakers. Most of this data is multispeaker data, collected from broadcasts, recorded meetings, telephone conversations, etc. The process of obtaining a list of speaker segments from such data is referred to as speaker indexing, segmentation, or diarization. A more-general task of annotating audio data from various audio sources including speakers has been referred to as audio diarization [36.10].

Still another speaker recognition application is to improve automatic speech recognition by adapting speaker-independent speech models to specified speakers. Many commercial speech recognizers do adapt their speech models to individual users, but this cannot be regarded as a speaker recognition application unless speaker models are constructed and speaker recognition is a part of the process. Speaker recognition can also be used to improve speech recognition for multispeaker data. In this situation speaker indexing can provide a table of speech segments assigned to individual speakers. The speech data in these segments can then be used to adapt speech models to each speaker. Speech recognition of multispeaker speech samples can be improved in another way. Errors and ambiguities in speech recognition transcripts can be corrected using the knowledge provided by speaker segmentation assigning the segments to the correct speakers.

36.2 Measuring Speaker Features

36.2.1 Acoustic Measurements

As mentioned in Sect. 36.1, low-level acoustic features such as short-time spectra are commonly used in speaker modeling. Such features are useful in authentication systems because speakers have less control over spectral details than higher-level features such as pitch.

Short-Time Spectrum

There are many ways of representing the short-time spectrum. A popular representation is the mel-frequency cepstral coefficients (MFCC), which were originally developed for speaker-independent speech recognition. The choice of center frequencies and bandwidths of the filter bank used in MFCC were motivated by the properties of the human auditory system. In particular, this representation provides limited spectral resolution above 2 kHz, which might be detrimental in speaker recognition. However, somewhat counterintuitively, MFCCs have been found to be quite effective in speaker recognition.

There are many minor variations in the definition of MFCC but the essential details are as follows. Let $\{S(k), 0 \le k < K\}$ be the discrete Fourier transform (DFT) coefficients of a windowed speech signal $\hat{s}(t)$. A set of triangular filters are defined such that

$$w_j(k) = \begin{cases} \frac{(k/K) f_s - f_{c_{j-1}}}{f_{c_j} - f_{c_{j-1}}}, & l_j \le k \le c_j, \\ f_{c_{j+1}} - \left(\frac{k}{K}\right) f_s / f_{c_{j+1}} - f_{c_j}, & c_j < k \le u_j, \\ 0, & \text{elsewhere}, \end{cases} \tag{36.1}$$

where $f_{c_{j-1}}$ and $f_{c_{j+1}}$ are the lower and upper limits of the pass band for filter j with $f_{c_0} = 0$ and $f_{c_j} < f_s/2$ for all j, and l_j, c_j and u_j are the DFT indices corresponding to the lower, center, and upper limits of the pass band for filter j. The log-energy at the outputs for the J filters are given by

$$e(j) = \ln\left[\frac{1}{\sum_{k=l_j}^{u_j} w_j(k)} \sum_{k=l_j}^{u_j} \|S(k)\|^2 w_j(k)\right] , \quad (36.2)$$

and the MFCC coefficients are the discrete cosine transform of the filter energies computed as

$$C(k) = \sum_{j=0}^{J} e(j) \cos\left[k\left(j - \frac{1}{2}\right)\frac{\pi}{J}\right] ,$$
$$k = 1, 2, \ldots, K . \quad (36.3)$$

The zeroth coefficient $C(0)$ is set to be the average log-energy of the windowed speech signal. Typical values of the various parameters involved in the MFCC computation are as follows. A cepstrum vector is calculated using a window length of 20 ms and updated every 10 ms. The center frequencies f_{c_j} are uniformly spaced from 0 to 1000 Hz and logarithmically spaced above 1000 Hz. The number of filter energies is typically 24 for telephone-band speech and the number of cepstrum coefficients used in modeling varies from 12 to 18 [36.13].

Cepstral coefficients based on short-time spectra estimated using linear predictive analysis and perceptual linear prediction are other popular representations [36.14].

Short-time spectral measurements are sensitive to channel and transducer variations. Cepstral mean subtraction (CMS) is a simple and effective method to compensate for convolutional distortions introduced by slowly varying channels. In this method, the cepstral vectors are transformed such that they have zero mean. The cepstral average over a sufficiently long speech signal approximates the estimate of a stationary channel [36.14]. Therefore, subtracting the mean from the original vectors is roughly equivalent to normalizing the effects of the channel, if we assume that the average of the clean speech signal is zero. Cepstral variance normalization, which results in feature vectors with unit variance, has also been shown to improve performance in text-independent speaker recognition when there is more than a minute of speech for enrollment. Other feature normalization methods, such as feature warping [36.15] and Gaussianization [36.16], map the observed feature distribution to a normal distribution over a sliding window, and have been shown to be useful in speaker recognition.

It has been long established that incorporating dynamic information is useful for speaker recognition and speech recognition [36.17]. The dynamic information is typically incorporated by extending the static cepstral vectors by their first and second derivatives computed as:

$$\Delta C_k = \frac{\sum_{t=-l}^{l} t c_{t+k}}{\sum_{t=-l}^{l} |t|} , \quad (36.4)$$

$$\Delta\Delta C_k = \frac{\sum_{t=-l}^{l} t^2 c_{t+k}}{\sum_{t=-l}^{l} t^2} . \quad (36.5)$$

Pitch

Voiced sounds are produced by a quasiperiodic opening and closing of the vocal folds in the larynx at a *fundamental frequency* that depends on the speaker. Pitch is a complex auditory attribute of sound that is closely related to this fundamental frequency. In this chapter, the term pitch is used simply to refer to the measure of periodicity observed in voiced speech.

Prosodic information represented by pitch and energy contours has been used successfully to improve the performance of speaker recognition systems [36.18]. There are a number of techniques for estimating pitch from the speech signal [36.19] and the performance of even simple pitch-estimation techniques is adequate for speaker recognition. The major failure modes occur during speech segments that are at the boundaries of voiced and unvoiced sounds and can be ignored for speaker recognition. A more-significant problem with using pitch information for speaker recognition is that speakers have a fair amount of control over it, which results in large intraspeaker variations and mismatch between enrollment and test utterances.

36.2.2 Linguistic Measurements

In traditional speaker authentication applications, the enrollment data is limited to a few repetitions of a password, and the same password is spoken to gain access to the system. In such cases, speaker models based on short-time spectra are very effective and it is difficult to

extract meaningful *high-level* or linguistic features. In applications such as indexing broadcasts by speaker and passive surveillance, a significant amount of enrollment data, perhaps several minutes, may be available. In such cases, the use of linguistic features has been shown to be beneficial [36.18].

Word Usage

Features such as vocabulary choices, function word frequencies, part-of-speech frequencies, etc., have been shown to be useful in speaker recognition [36.20]. In addition to words, spontaneous speech contains fillers and hesitations that can be characterized by statistical models and used for identifying speakers [36.20, 21]. There are a number of issues with speaker recognition systems based on lexical features: they are susceptible to errors introduced by large-vocabulary speech recognizers, a significant amount of enrollment data is needed to build robust models, and the speaker models are likely to characterize the topic of conversation as well as the speaker.

Phone Sequences and Lattices

Models of phone sequences output by speech recognizers using phonotactic grammars, typically phone unigrams, can be used to represent speaker characteristics [36.22]. It is assumed that these models capture speaker-specific pronunciations of frequently occurring words, choice of words, and also an implicit characterization of the acoustic space occupied by the speech signal from a given speaker. It turns out that there is an optimal tradeoff between the constraints used in the recognizer to produce the phone sequences and the robustness of the speaker models of phone sequences. For example, the use of lexical constraints in the automatic speech recognition (ASR) reproduces phone sequences found in a predetermined dictionary and prevents phone sequences that may be characteristic of a speaker but not represented in the dictionary.

The phone accuracy computed using one-best output phone strings generated by ASR systems without lexical constraints is typically not very high. On the other hand, the correct phone sequence can be found in a phone lattice output by an ASR with a high probability. It has been shown that it is advantageous to construct speaker models based on phone-lattice output rather than the one-best phone sequence [36.22]. Systems based on one-best phone sequences use the counts of a term such as a phone unigram or bigram in the decoded sequence. In the case of lattice outputs, these raw counts are replaced by the expected counts given by

$$\boldsymbol{E}[C(\tau|X)] = \sum_{Q} p(Q|X)C(\tau|Q)\,, \tag{36.6}$$

where Q is a path through the phone lattice for the utterance X with associated probability $p(Q|X)$, and $C(\tau|Q)$ is the count of the term τ in the path Q.

Other Linguistic Features

A number of other features that have been found to be useful for speaker modeling are (a) pronunciation modeling of carefully chosen words, and (b) prosodic statistics such as pitch and energy contours as well as durations of phones and pauses [36.23].

36.3 Constructing Speaker Models

A speaker recognition system provides the ability to construct a model λ_s for speaker s using enrollment utterances from that speaker, and a method for comparing the quality of match of a *test* utterance to the speaker model. The choice of models is determined by the application constraints. In applications in which the user is expected to say a fixed password each time, it is beneficial to develop models for words or phrases to capture the temporal characteristics of speech. In passive surveillance applications, the test utterance may contain phonemes or words not seen in the enrollment data. In such cases, less-detailed models that model the overall acoustic space of the user's utterances tend to be effective. A survey of general techniques that have been used in speaker modeling follows. The methods can be broadly classified as nonparametric or parametric. Nonparametric models make few structural assumptions and are effective when there is sufficient enrollment data that is matched to the test data. Parametric models allow a parsimonious representation of the structural constraints and can make effective use of the enrollment data if the constraints are appropriately chosen.

36.3.1 Nonparametric Approaches

Templates

This is the simplest form of speaker modeling and is appropriate for fixed-password speaker verification sys-

tems [36.24]. The enrollment data consists of a small number of repetitions of the password spoken by the target speaker. Each enrollment utterance $\boldsymbol{X}$ is a sequence of feature vectors $\{\boldsymbol{x}_t\}_{t=0}^{T-1}$ generated as described in Sect. 36.2, and serves as the *template* for the password as spoken by the target speaker. A test utterance $\boldsymbol{Y}$ consisting of vectors $\{\boldsymbol{y}_t\}_{t=0}^{T'-1}$, is compared to each of the enrollment utterances and the identity claim is accepted if the distance between the test and enrollment utterances is below a decision threshold. The comparison is done as follows. Associated with each pair of vectors, x_i and y_j, is a distance, $d(\boldsymbol{x}_i, \boldsymbol{y}_j)$. The feature vectors of $\boldsymbol{X}$ and $\boldsymbol{Y}$ are aligned using an algorithm referred to as *dynamic time warping* to minimize an overall distance defined as the average intervector distance $d(x_i, y_j)$ between the aligned vectors [36.12].

This approach is effective in simple fixed-password applications in which robustness to channel and transducer differences are not an issue. This technique is described here mostly for historical reasons and is rarely used in real applications today.

Nearest-Neighbor Modeling

Nearest-neighbor models have been popular in nonparametric classification [36.25]. This approach is often thought of as estimating the local density of each class by a Parzen estimate and assigning the test vector to the class with the maximum local density. The local density of a class (speaker) with enrollment data $\boldsymbol{X}$ at a test vector $\boldsymbol{y}$ is defined as

$$p_{\mathrm{nn}}(\boldsymbol{y}; \boldsymbol{X}) = \frac{1}{V[d_{\mathrm{nn}}(\boldsymbol{y}, \boldsymbol{X})]}\,, \tag{36.7}$$

where $d_{\mathrm{nn}}(\boldsymbol{y}, \boldsymbol{X}) = \min_{\boldsymbol{x}_j \in \boldsymbol{X}} \|\boldsymbol{y} - \boldsymbol{x}_j\|$ is the nearest-neighbor distance and $V(r)$ is the volume of a sphere of radius r in the D-dimensional feature space. Since $V(r)$ is proportional to r^D,

$$\ln[p_{\mathrm{nn}}(\boldsymbol{y}; \boldsymbol{X})] \approx -D \ln[d_{\mathrm{nn}}(\boldsymbol{y}, \boldsymbol{X})]\,. \tag{36.8}$$

The log-likelihood score of the test utterances $\boldsymbol{Y}$ with respect to a speaker specified by enrollment $\boldsymbol{X}$ is given by

$$s_{\mathrm{nn}}(\boldsymbol{Y}; \boldsymbol{X}) \approx -\sum_{\boldsymbol{y}_j \in \boldsymbol{Y}} \ln[d_{\mathrm{nn}}(\boldsymbol{y}, \boldsymbol{X})]\,, \tag{36.9}$$

and the speaker with the greatest $s(\boldsymbol{Y}; \boldsymbol{X})$ is identified.

A modified version of the nearest-neighbor model, motivated by the discussion above, has been successfully used in speaker identification [36.26]. It was found empirically that a score defined as

$$\begin{aligned} s'_{\mathrm{nn}}(\boldsymbol{Y}; \boldsymbol{X}) = {} & \frac{1}{N_y} \sum_{\boldsymbol{y}_j \in \boldsymbol{Y}} \min_{\boldsymbol{x}_i \in \boldsymbol{X}} \|\boldsymbol{y}_j - \boldsymbol{x}_i\|^2 \\ & + \frac{1}{N_x} \sum_{\boldsymbol{x}_j \in \boldsymbol{X}} \min_{\boldsymbol{y}_i \in \boldsymbol{Y}} \|\boldsymbol{y}_i - \boldsymbol{x}_j\|^2 \\ & - \frac{1}{N_y} \sum_{\boldsymbol{y}_j \in \boldsymbol{Y}} \min_{\boldsymbol{y}_i \in \boldsymbol{Y};\, j \neq i} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 \\ & - \frac{1}{N_x} \sum_{\boldsymbol{x}_j \in \boldsymbol{X}} \min_{\boldsymbol{x}_i \in \boldsymbol{X};\, j \neq i} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \end{aligned} \tag{36.10}$$

gives much better performance than $s_{\mathrm{nn}}(\boldsymbol{Y}; \boldsymbol{X})$.

36.3.2 Parametric Approaches

Vector Quantization Modeling

Vector quantization constructs a set of representative samples of the target speaker's enrollment utterances by clustering the feature vectors. Although a variety of clustering techniques exist, the most commonly used is k-means clustering [36.14]. This approach partitions N feature vectors into K disjoint subsets S_j to minimize an overall distance such as

$$D = \sum_{j=1}^{J} \sum_{\boldsymbol{x}_i \in S_j} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)\,, \tag{36.11}$$

where $\boldsymbol{\mu}_j = (1/N_j) \sum_{\boldsymbol{x}_i \in S_j} \boldsymbol{x}_i$ is the centroid of the N_j samples in the j-th cluster. The algorithm proceeds in two steps:

1. Compute the centroid of each cluster using an initial assignment of the feature vectors to the clusters.
2. Reassign $\boldsymbol{x}_i$ to that cluster whose centroid is closest to it.

These steps are iterated until successive steps do not reassign samples.

This algorithm assumes that there exists an initial clustering of the samples into K clusters. It is difficult to obtain a good initialization of K clusters in one step. In fact, it may not even be possible to reliably estimate K clusters because of data sparsity. The Linde–Buzo–Gray (LBG) algorithm [36.27] provides a good solution for this problem. Given m centroids, the LBG algorithm produces additional centroids by perturbing one or more of the centroids using a heuristic. One common heuristic is to choose the $\boldsymbol{\mu}$ for the cluster with the largest variance and produce two centroids $\boldsymbol{\mu}$ and $\boldsymbol{\mu} + \boldsymbol{\epsilon}$. The enrollment feature vectors are assigned to the resulting $m+1$ centroids. The k-means algorithm described previously can

then be applied to refine the centroid estimates. This process can be repeated until $m = M$ or the cluster sizes fall below a threshold. The LBG algorithm is usually initialized with $m = 1$ and computes the centroid of all the enrollment data. There are many variations of this algorithm that differ in the heuristic used for perturbing the centroids, the termination criteria, and similar details. In general, this algorithm for generating VQ models has been shown to be quite effective. The choice of K is a function of the size of enrollment data set, the application, and other system considerations such as limits on computation and memory.

Once the VQ models are established for a target speaker, scoring consists of evaluating D in (36.11) for feature vectors in the test utterance. This approach is general and can be used for text-dependent and text-independent speaker recognition, and has been shown to be quite effective [36.28]. Vector quantization models can also be constructed on sequences of feature vectors, which are effective at modeling the temporal structure of speech. If distance functions and centroids are suitably redefined, the algorithms described in this section continue to be applicable.

Although VQ models are still useful in some situations, they have been superseded by models such as the Gaussian mixture models and hidden Markov models, which are described in the following sections.

Gaussian Mixture Models

In the case of text-independent speaker recognition (the subject of Chap. 38) where the system has no prior knowledge of the text of the speaker's utterance, Gaussian mixture models (GMMs) have proven to be very effective. This can be thought of as a refinement of the VQ model. Feature vectors of the enrollment utterances $\boldsymbol{X}$ are assumed to be drawn from a probability density function that is a mixture of Gaussians given by

$$p(\boldsymbol{x}|\lambda) = \sum_{k=1}^{K} w_k p_k(\boldsymbol{x}|\lambda_k) \,, \tag{36.12}$$

where $0 \le w_k \le 1$ for $1 \le k \le K$, $\sum_{k=1}^{K} w_k = 1$, and

$$p_k(\boldsymbol{x}|\lambda_k) = \frac{1}{(2\pi)^{D/2}|\Sigma_k|^{1/2}} \times \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^{\mathrm{T}}\Sigma_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right) \,, \tag{36.13}$$

λ represents the parameters $(\boldsymbol{\mu}_i, \Sigma_i, w_i)_{i=1}^{K}$ of the distribution. Since the size of the training data is often small, it is difficult to estimate full covariance matrices reliably. In practice, $\{\Sigma_k\}_{k=1}^{K}$ are assumed to be diagonal.

Given the enrollment data $\boldsymbol{X}$, the maximum-likelihood estimates of the λ can be obtained using the *expectation-maximization* (EM) algorithm [36.12]. The K-means algorithm can be used to initialize the parameters of the component densities. The posterior probability that $\boldsymbol{x}_t$ is drawn from the component $p_m(\boldsymbol{x}_t|\lambda_m)$ can be written

$$P(m|\boldsymbol{x}_t, \lambda) = \frac{w_m p_m(\boldsymbol{x}_t|\lambda_m)}{p(\boldsymbol{x}_t|\lambda)} \,. \tag{36.14}$$

The maximum-likelihood estimates of the parameters of λ in terms of $P(m|\boldsymbol{x}_t, \lambda)$ are

$$\boldsymbol{\mu}_m = \frac{\sum_{t=1}^{\mathrm{T}} P(m|\boldsymbol{x}_t, \lambda)\boldsymbol{x}_t}{\sum_{t=1}^{\mathrm{T}} P(m|\boldsymbol{x}_t, \lambda)} \,, \tag{36.15}$$

$$\Sigma_m = \frac{\sum_{t=1}^{\mathrm{T}} P(m|\boldsymbol{x}_t, \lambda)\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}}{\sum_{t=1}^{\mathrm{T}} P(m|\boldsymbol{x}_t, \lambda)} - \boldsymbol{\mu}_m\boldsymbol{\mu}_m^{\mathrm{T}} \,, \tag{36.16}$$

$$w_m = \frac{1}{T}\sum_{t=1}^{\mathrm{T}} P(m|\boldsymbol{x}_t, \lambda) \,. \tag{36.17}$$

The two steps of the EM algorithm consist of computing $P(m|\boldsymbol{x}_t, \lambda)$ given the current model, and updating the model using the equations above. These two steps are iterated until a convergence criteria is satisfied.

Test utterance scores are obtained as the average log-likelihood given by

$$s(\boldsymbol{Y}|\lambda) = \frac{1}{T}\sum_{t=1}^{\mathrm{T}} \log[p(\boldsymbol{y}_t|\lambda)] \,. \tag{36.18}$$

Speaker verification is often based on a likelihood-ratio test statistic of the form $p(\boldsymbol{Y}|\lambda)/p(\boldsymbol{Y}|\lambda_{\mathrm{bg}})$ where λ is the speaker model and λ_{bg} represents a background model [36.29]. For such systems, speaker models can also be trained by adapting λ_{bg}, which is generally trained on a large independent speech database [36.30]. There are many motivations for this approach. Generating a speaker model by adapting a well-trained background GMM may yield models that are more robust to channel differences, and other kinds of mismatch between enrollment and test conditions than models estimated using only limited enrollment data. Details of this procedure can be found in Chap. 38.

Speaker modeling using GMMs is attractive for text-independent speaker recognition because it is simple to implement and computationally inexpensive. The fact that this model does not model temporal aspects of speech is a disadvantage. However, it has been difficult to exploit temporal structure to improve speaker recognition performance when the linguistic content of test utterances does not overlap significantly with the linguistic content of enrollment utterances.

Hidden Markov Models

In applications where the system has prior knowledge of the text and there is significant overlap of what was said during enrollment and testing, text-dependent statistical models are much more effective than GMMs. An example of such applications is access control to personal information or bank accounts using a voice password. Hidden Markov models (HMMs) [36.12] for phones, words, or phrases, have been shown to be very effective [36.31, 32]. Passwords consisting of word sequences drawn from specialized vocabularies such as digits are commonly used. Each word can be characterized by an HMM with a small number of states, in which each state is represented by a Gaussian mixture density. The maximum-likelihood estimates of the parameters of the model can be obtained using a generalization of the EM algorithm [36.12].

The ML training aims to approximate the underlying distribution of the enrollment data for a speaker. The estimates deviate from the *true* distribution due to lack of sufficient training data and incorrect modeling assumptions. This leads to a suboptimal classifier design. Some limitations of ML training can be overcome using discriminative training of speaker models in which an attempt is made to minimize an overall cost function that depends on misclassification or detection errors [36.33–35]. Discriminative training approaches require examples from competing speakers in addition to examples from the target speaker. In the case of closed-set speaker identification, it is possible to construct a misclassification measure to evaluate how likely a test sample, spoken by a target speaker, is misclassified as any of the others. One example of such a measure is the *minimum classification error* (MCE) defined as follows. Consider the set of S discriminant functions $\{g_k(\boldsymbol{x}; \Lambda_s), 1 \le s \le S\}$, where $g_k(\boldsymbol{x}; \Lambda_s)$ is the log-likelihood of observation $\boldsymbol{x}$ given the models Λ_s for speaker s. A set of misclassification measures for each speaker can be defined as

$$d_s(\boldsymbol{x}; \Lambda) = -g_s(\boldsymbol{x}; \Lambda_s) + G_s(\boldsymbol{x}; \Lambda), \tag{36.19}$$

where Λ is the set of all speaker models and $G_s(\boldsymbol{x}; \Lambda)$ is the antidiscriminant function for speaker s. $G_s(\boldsymbol{x}; \Lambda)$ is defined so that $d_s(\boldsymbol{x}; \Lambda)$ is positive only if $\boldsymbol{x}$ is incorrectly classified. In speech recognition problems, $G_s(\boldsymbol{x}; \Lambda)$ is usually defined as a collective representation of all competing classes. In the speaker identification task, it is often advantageous to construct pairwise misclassification measures such as

$$d_{ss'}(\boldsymbol{x}; \Lambda) = -g_s(\boldsymbol{x}; \Lambda_s) + g'_s\left(\boldsymbol{x}; \Lambda'_s\right), \tag{36.20}$$

with respect to a set of competing speakers s', a subset of the S speakers. Each misclassification measure is embedded into a smooth empirical *loss function*

$$l_{ss'}(\boldsymbol{x}; \Lambda) = \frac{1}{1 + \exp(-\alpha d_{ss'}(\boldsymbol{x}; \Lambda))}, \tag{36.21}$$

which approximates a loss directly related to the number of classification errors, and α is a smoothness parameter. The loss functions can then be combined into an overall loss given by

$$l(\boldsymbol{x}; \Lambda) = \sum_s \sum_{s' \in \mathcal{S}_c} l_{ss'}(\boldsymbol{x}; \Lambda) \delta_s(\boldsymbol{x}), \tag{36.22}$$

where $\delta_s(\boldsymbol{x})$ is an indicator function which is equal to 1 when $\boldsymbol{x}$ is uttered by speaker s and 0 otherwise, and $\mathcal{S}_c$ is the set of competing speakers. The total loss, defined as the sum of $l(\boldsymbol{x}; \Lambda)$ over all training data, can be optimized with respect to all the model parameters using a gradient-descent algorithm. A similar algorithm has been developed for speaker verification in which samples from a large number of speakers in a development set is used to compute a minimum verification measure [36.36].

The algorithm described above is only to illustrate the basic principles of discriminative training for speaker identification. Many other approaches that differ in their choice of the loss function or the optimization method have been developed and shown to be effective [36.35, 37].

The use of HMMs in text-dependent speaker verification is discussed in detail in Chap. 37.

Support Vector Modeling

Traditional discriminative training approaches such as those based on MCE have a tendency to overtrain on the training set. The complexity and generalization ability of the models are usually controlled by testing on

a held-out development set. Support vector machines (SVMs) [36.38] provide a way for training classifiers using discriminative criteria and in which the model complexity that provides good generalization to test data is determined automatically from the training data. SVMs have been found to be useful in many classification tasks including speaker identification [36.39].

The original formulation of SVMs was for two-class problems. This seems appropriate for speaker verification in which the positive samples consist of the enrollment data from a target user and the negative samples are drawn from a large set of imposter speakers. Many extensions of SVMs to multiclass classification have also been developed and are appropriate for speaker identification. There are many issues with SVM modeling for speaker recognition, including the appropriate choice of features and the kernel. The use of SVMs for text-independent speaker recognition is the subject of Chap. 38.

Other Approaches

Most state-of-the-art speaker recognition systems use some combination of the modeling methods described in the previous sections. Many other interesting models have been proposed and have been shown to be useful in limited scenarios. Eigenvoice modeling is an approach in which the speaker models are confined to a low-dimensional linear subspace obtained using independent training data from a large set of speakers. This method has been shown to be effective for speaker modeling and speaker adaptation when the enrollment data is too limited for the effective use of other text-independent approaches such as GMMs [36.40]. Artificial neural networks [36.41] have also been shown to be useful in some situations, perhaps in combination with GMMs. When sufficient enrollment data is available, a method for speaker detection that involves comparing the test segment directly to similar segments in enrollment data has been shown to be effective [36.42].

36.4 Adaptation

In most speaker recognition scenarios, the speech data available for enrollment is too limited for training models that adequately characterize the range of test conditions in which the system needs to operate. For example, in fixed-password speaker authentication systems used in telephony services, enrollment data is typically collected in a single call. The enrollment and test conditions may be mismatched in a number of ways: the telephone handset that is used, the location of the call, which determines the kinds of background noises, and the channel over which speech is transmitted such as cellular or landline networks. In text-independent modeling, there are likely to be additional problems because of mismatch in the linguistic content. A very effective way to mitigate the effects of mismatch is model adaptation.

Models can be adapted in an unsupervised way using data from authenticated utterances. This is common in fixed-password systems and can reduce the error rate significantly. It is also necessary to update the decision thresholds when the models are adapted. Since the selection of data for model adaptation is not supervised, there is the possibility that models are adapted on imposter utterances. This can be disastrous. The details of unsupervised model and threshold adaptation and the various issues involved are explained in detail in Chap. 37.

Speaker recognition is often incorporated into other applications that involve a dialog with the user. Feedback from the dialog system can be used to supervise model adaptation. In addition, meta-information available from a dialog system such as the history of interactions can be combined with speaker recognition to design a flexible and secure authentication system [36.43].

36.5 Decision and Performance

36.5.1 Decision Rules

Whether they are used for speaker identification or verification, the various models and approaches presented in Sect. 36.3 provide a score $s(Y|\lambda)$ measuring the match between a given test utterance Y and a speaker model λ. Identification systems yield a set of such scores corresponding to each speaker in a target list. Verification systems output only one score using the speaker model of the claimed speaker. An

accept or reject decision has to be made using this score.

Decision in closed-set identification consists of choosing the identified speaker $\hat{S}$ as the one that corresponds to the maximum score:

$$\hat{S} = \arg\max_j s(Y|\lambda_j) \,, \tag{36.23}$$

where the index j ranges over the whole set of target speakers.

Decision in verification is obtained by comparing the score computed using the model for the claimed speaker S_i given by $s(Y|\lambda_i)$ to a predefined threshold θ. The claim is accepted if $s(Y|\lambda_i) \geq \theta$, and rejected otherwise.

Open-set identification relies on a step of closed-set identification eliciting the most likely identity, followed by a verification step to determine whether the hypothesized identity match is good enough.

36.5.2 Threshold Setting and Score Normalization

Efficiency and robustness require that the score $s(Y|\lambda)$ be quite readily exploited in a practical application. In particular, the threshold θ should be as insensitive as possible across users and application context.

When the score is obtained in a probabilistic framework or can be interpreted as a (log) likelihood ratio (LLR), Bayesian decision theory [36.44] states that an optimal threshold for verification can be theoretically set once the desired false acceptance c_{fa} and false rejection c_{fr}, and the a priori probability p_{imp} of an impostor trying to enter the system, are specified. The optimal choice of the threshold is given by:

$$\theta^* = \frac{c_{\text{fa}}}{c_{\text{fr}}} \frac{p_{\text{imp}}}{1 - p_{\text{imp}}} \,. \tag{36.24}$$

In practice, however, the score $s(Y|\lambda)$ does not behave as theory would predict since the statistical models are not ideal. Various normalization procedures have been proposed to alleviate this problem. Initial work by *Li* and *Porter* [36.45] has inspired a number of score normalization techniques that intend to make the statistical distribution of $s(Y|\lambda)$ as independent as possible across speakers, acoustic conditions, linguistic content, etc. This has lead to a number of threshold normalization schemes, such as the Z-norm, H-Norm, and T-norm, which use side information, the distance between models, and speech material from a development set to determine the normalization parameters. These normalization procedures are discussed in more detail in Chaps. 37, 38 and [36.46]. Even so, the optimal threshold for a given operating condition is generally estimated experimentally from development data that is appropriate for a given scenario.

36.5.3 Errors and DET Curves

The performance of an identification system is related to the probability of misclassification, which corresponds to cases when the identified speaker is not the actual one.

Verification systems are evaluated based on two types of errors: false acceptance, when an impostor speaker succeeds in being verified with an erroneous claimed identity, and false rejection, when a target user claiming his/her genuine identity is rejected. The *a posteriori* estimates of the probabilities p_{fa} and p_{fr} of these two types of errors vary in the opposite way from each other when the decision threshold θ is varied. The tradeoff between p_{fa} and p_{fr} (sometimes mapped to the probability of detection p_{d}, defined as $1 - p_{\text{fr}}$) is often displayed in the form of a receiver operating characteristic (ROC), a term commonly used in detection theory [36.44]. In speaker recognition systems a different representation of the same data, referred to as the detection error tradeoff (DET) curve, has become popular.

The DET curve [36.47] is the standard way to depict the system behavior in terms of hypotheses separability by plotting p_{fa} as a function of p_{fr}. Rather than the probabilities themselves, the normal deviates corresponding to the probabilities are plotted. For a particular threshold value, the corresponding error rates p_{fa} and p_{fr} appear as a specific point on this DET curve. A popular point is the one where $p_{\text{fa}} = p_{\text{fr}}$, which is called the equal error rate (EER). Plotting DET curves is a good way to compare the potential of two methods in a laboratory but it is not suited for predicting accurately the performance of a system when deployed in real-life conditions.

The decision threshold θ is often chosen to optimize a cost that is a function of the probability of false acceptance and false rejection as well as the prior probability of an imposter attack. One such function is called the detection cost function (DCF), defined as [36.48]

$$C = p_{\text{imp}} c_{\text{fa}} p_{\text{fa}} + (1 - p_{\text{imp}}) c_{\text{fr}} p_{\text{fr}} \,. \tag{36.25}$$

The DCF is indeed a way to evaluate a system under a particular operating condition and to summarize into a single figure its estimated performance in a given application scenario. It has been used as the primary figure of merit for the evaluation of systems participating in the yearly NIST speaker recognition evaluations [36.48].

36.6 Selected Applications for Automatic Speaker Recognition

Text-dependent and text-independent speaker recognition technology and their applications are discussed in detail in the following two Chaps. 37 and 38. A few interesting, but perhaps not primary, applications of speaker recognition technology are described in this section. These applications were chosen to demonstrate the wide range of applications of speaker recognition.

36.6.1 Indexing Multispeaker Data

Speaker indexing can be approached as either a supervised or unsupervised task. Supervised means that prior speaker models exist for the speakers of interest included in the data. The data can then be scanned and processed to determine the segments associated with each of these speakers. Unsupervised means that prior speaker models do not exist. The type of approach taken depends on the type and amount of prior knowledge available for particular applications. There may be knowledge of the identities of the participating speakers and there may even be independent labeled speech data available for constructing models for these speakers, such as in the case of some broadcast news applications [36.6, 49, 50]. In this situation the task is supervised and the techniques for speaker segmentation or indexing are basically the same as used for speaker detection [36.9, 50, 51].

A more-challenging task is unsupervised segmentation. An example application is the segmentation of the speakers in a two-person telephone conversation [36.4, 9, 52, 53]. The speaker identities may or may not be known but independent labelled speech data for constructing speaker models is generally not available. Following is a possible approach to the unsupervised segmentation problem. The first task is to construct unlabeled single-speaker models from the current data. An initial segmentation of the data is carried out with an acoustic change detector using a criterion such as the generalized likelihood ratio (GLR) [36.4, 5] or Bayesian information criterion (BIC) [36.8, 54, 55]. The hypothesis underlying this process is that each of the resulting segments will be single-speaker segments. These segments are then clustered using an agglomerative clustering algorithm with a criterion for measuring the pairwise similarity between segments [36.56–58]. Since in the cited application the number of speakers is known to be two, the clustering terminates when two clusters are obtained. If the acoustic change criterion and the matching criterion for the clustering perform well the two clusters of segments will each contain segments mostly from one speaker or the other. These segment clusters can then be used to construct protospeaker models, typically GMMs. Each of these models is then used to resegment the data to provide an improved segmentation which, in turn, will provide improved speaker models. The process can be iterated until no further significant improvement is obtained. It then remains to apply speaker labels to the models and segmentations. Some independent knowledge is required to accomplish this. As mentioned earlier, the speakers in the telephone conversation may be known, but some additional information is required to assign labels to the correct models and segmentations.

36.6.2 Forensics

The perspective of being able to identify a person on the basis of his or her voice has received significant interest in the context of law enforcement. In many situations, a voice recording is a key element, and sometimes the only one available, for proceeding with an investigation, identifying or clearing a suspect, and even supporting an accusation or defense in a court of law.

The public perception is that voice identification is a straightforward task, and that there exists a reliable *voiceprint* in much the same way as there are fingerprints or genetic (DNA) prints. This is not true in general because the voice of an individual has a strong behavioral component, and is only partly based on anatomical properties. Moreover, the conditions under which the test utterance is recorded are generally not known or controlled. The test voice sample might be from an anonymous call, wiretapping, etc. For these reasons, the use of voice recognition in the context of forensic applications must be approached with caution [36.59].

The four procedures that are generally followed in the forensic context are described below.

Nonexpert Speaker Recognition by Lay Listener(s)

This procedure is used in the context of a voice lineup when a victim or a witness has had the opportunity of hearing a voice sample and is asked to say whether he or she recognizes this voice, or to determine if this voice sample matches one of a set of utterances. Since it is difficult to set up such a test in a controlled way and calibrate to the matching criteria an individual subject may use, such procedures can be used only to suggest a possible course of action during an investigation.

Expert Speaker Recognition

Expert study of a voice sample might include one or more of aural–perceptual approaches, linguistic analysis, and spectrogram examination. In this context, the expert takes into account several levels of speaker characterization such as pitch, timbre, diction, style, idiolect, and other idiosyncracies, as well as a number of physical measurements including fundamental frequencies, segment durations, formants, and jitter. Experts provide a decision on a seven-level scale specified by the International Association for Identification (IAI) standard [36.60] on whether two voice samples (the disputed recording and a voice sample of the suspect) are more or less likely to have been produced by a the same person. Subjective heterogeneous approaches coexist between forensic practitioners and, although the technical invalidity of some methods has been clearly established, they are still used by some. The expert-based approach is therefore generally used with extreme caution.

Semiautomatic Methods

This category refers to systems for which a supervised selection of speech segments is conducted prior to a computer-based analysis of the selected material. Whereas a calibrated metric can be used to evaluate the similarity of specific types of segments such as words or phrases, these systems tend to suffer from a lack of standardization.

Automatic Methods

Fully automated methods using state-of-the-art techniques offer an attractive paradigm for forensic speaker verification. In particular, these automatic approaches can be run without any (subjective) human intervention, they offer a reproducible procedure, and they lend themselves to large-scale evaluation. Technological improvements over the years, as well as progress in the presentation, reporting, and interpretation of the results, have made such methods attractive. However, levels of performance remain highly sensitive to a number of external factors ranging from the quality and similarity of recording conditions, the cooperativeness of speakers, and the potential use of technologies to fake or disguise a voice.

Thanks to a number of initiatives and workshops (in particular the series of ISCA and IEEE Odyssey workshops), the past decade has seen some convergence in terms of formalism, interpretation, and methodology between forensic science and engineering communities. In particular, the interpretation of voice forensic evidence in terms of Bayesian decision theory and the growing awareness of the need for systematic evaluation have constituted significant contributions to these exchanges.

36.6.3 Customization: SCANmail

Customization of services and applications to the user is another class of applications of speaker recognition technology. An example of a customized messaging system is one where members of a family share a voice mailbox. Once the family members are enrolled in a speaker recognition system, there is no need for them to identify themselves when accessing their voice mail. A command such as *Get my messages* spoken by a user can be used to identify and authenticate the user, and provide only those messages left for that user. There are many such applications of speaker recognition technology. An interesting and successful application of *caller identification* to a voicemail browser is described in this section.

SCANMail is a system developed for the purpose of providing useful tools for managing and searching through voicemail messages [36.61]. It employs ASR to provide text transcriptions, information retrieval on the transcriptions to provide a weighted set of search terms, information extraction to obtain key information such as telephone numbers from transcription, as well as automatic speaker recognition to carry out caller identification by processing the incoming messages. A graphical user interface enables the user to exercise the features of the system. The caller identification function is described in more detail below.

Two types of processing requests are handled by the caller identification system (CIS). The first type of request is to assign a speaker label to an incoming message. When a new message arrives, ASR is used to produce a transcription. The transcription as well as the speech signal is transmitted to the CIS for caller identification. The CIS compares the processed speech signal with the model of each caller in the recipient's address book. The recipient's address book is populated with speaker models when the user adds a caller to the address book by providing a label to a received message. A matching score is obtained for each of the caller models and compared to a caller-dependent rejection threshold. If the matching score exceeds the threshold, the received message is assigned a speaker label. Otherwise, CIS assigns an *unknown* label to the message.

The second type of request originates with the user action of adding a caller to an address book as mentioned earlier. In the course of reviewing a received message, the user has the capability to supply a caller label to the

message. The enrollment module in the CIS attempts to construct a speaker model for a new user using that message. The acoustic models are trained using text-independent speaker modeling. Acoustic models can be augmented with models based on meta-information, which may include personal information such as the caller's name or contact information left in the message, or the calling history.

36.7 Summary

Identifying speakers by voice was originally investigated for applications in speaker authentication. Over the last decade, the field of speaker recognition has become much more diverse and has found numerous applications. An overview of the technology and sample applications were presented in this chapter.

The modeling techniques that are applicable, and the nature of the problems, vary depending on the application scenario. An important dichotomy is based on whether the content (text) of the speech during training and testing overlaps significantly and is known to the system. These two important cases are the subject of the next two chapters.

References

36.1 J.S. Dunn, F. Podio: Biometrics Consortium website, http://www.biometrics.org (2007)

36.2 M.A. Przybocki, A.F. Martin: The 1999 NIST speaker recognition evaluation, using summed two-channel telephone data for speaker detection and speaker tracking, Eurospeech 1999 Proceedings (1999) pp. 2215–2218, http://www.nist.gov/speech/publications/index.htm

36.3 M.A. Przybocki, A.F. Martin: Nist speaker recognition evaluation chronicles, Odyssey Workshop 2004 Proc. (2004) pp. 15–22

36.4 H. Gish, M.-H. Siu, R. Rohlicek: Segregation of speakers for speech recognition and speaker identification, Proc. ICASSP (1991) pp. 873–876

36.5 L. Wilcox, F. Chen, D. Kimber, V. Balasubramanian: Segmentation of speech using speaker identification, Proc. ICASSP (1994) pp. 161–164

36.6 J.-L. Gauvain, L. Lamel, G. Adda: Partitioning and transcription of broadcast news data, Proc. of ICSLP (1998) pp. 1335–1338

36.7 S.E. Johnson: Who spoke when? – automatic segmentation and clustering for determining speaker turns, Proc. Eurospeech (1999) pp. 2211–2214

36.8 P. Delacourt, C.J. Wellekens: Distbic: A speaker-based segmentation for audio data indexing, Speech Commun. **32**, 111–126 (2000)

36.9 R.B. Dunn, D.A. Reynolds, T.F. Quatieri: Approaches to speaker detection and tracking in conversational speech, Digital Signal Process. **10**, 93–112 (2000)

36.10 S.E. Tranter, D.A. Reynolds: An overview of automatic speaker diarization systems, IEEE Trans. Speech Audio Process. **14**, 1557–1565 (2006)

36.11 L.H. Jamieson: Course notes for speech processing by computer, http://cobweb.ecn.purdue.edu ee649/notes/ (2007) Chap. 1

36.12 L.R. Rabiner, B.-H. Juang: *Fundamentals of Speech Recognition* (Prentice-Hall, Englewood Cliffs 1993)

36.13 S. Davis, P. Mermelstein: Comparison of parametric representation for monosyllable word recognition in continuously spoken sentences, IEEE Trans. Acoust. Speech Signal Process. **28**, 357–366 (1980)

36.14 X. Huang, A. Acero, H.-W. Hon: *Spoken Language Processing: A Guide to Theory, Algorithm and System Development* (Prentice-Hall, Englewood Cliffs 2001)

36.15 J. Pelecanos, S. Sridharan: Feature warping for robust speaker verification, Proc. ISCA Workshop on Speaker Recognition – 2001: A Speaker Odyssey (2001)

36.16 B. Xiang, U. Chaudhari, J. Navratil, G. Ramaswamy, R. Gopinath: Short-time Gaussianization for robust speaker verification, Proc. ICASSP, Vol. 1 (2002) pp. 681–684

36.17 S. Furui: Comparison of speaker recognition methods using static features and dynamic features, IEEE Trans. Acoust. Speech Signal Process. **29**, 342–350 (1981)

36.18 J.P. Campbell, D.A. Reynolds, R.B. Dunn: Fusing high- and log-level features for speaker recognition, Proc. Eurospeech, Vol. 1 (2003)

36.19 W. Hess: *Pitch Determination of Speech Signals* (Springer, Berlin, Heidelberg 1983)

36.20 G. Doddington: Speaker recognition based on idiolectal differences between speakers, Proc. Eurospeech (2001) pp. 2521–2524

36.21 W.D. Andrews, M.A. Kohler, J.P. Campbell, J.J. Godfrey: Phonetic, idiolectal, and acoustic speaker recognition, Proceedings of Odyssey Workshop (2001)

36.22 A. Hatch, B. Peskin, A. Stolcke: Improved phonetic speaker recognition using lattice decoding, Proc. ICASSP, Vol. 1 (2005)

36.23 D. Reynolds, W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, J. Godfrey, D. Jones, B. Xiang: The SuperSID project: Exploiting high-level information for high-accuracy speaker recognition, Proc. ICASSP (2003) pp. 784–787

36.24 A.E. Rosenberg: Automatic speaker verification: A review, Proc. IEEE **64**, 475–487 (1976)

36.25 K. Fukunaga: *Introduction to Statistical Pattern Recognition*, 2nd edn. (Elsevier, New York 1990)

36.26 A.L. Higgins, L.G. Bahler, J.E. Porter: Voice identification using nearest-neighbor distance measure, Proc. ICASSP (1993) pp. 375–378

36.27 Y. Linde, A. Buzo, R.M. Gray: An algorithm for vector quantization, IEEE Trans. Commun. **28**, 94–95 (1980)

36.28 F.K. Soong, A.E. Rosenberg, L.R. Rabiner, B.H. Juang: A vector quantization approach to speaker recognition, Proc. IEEE ICASSP (1985) pp. 387–390

36.29 D.A. Reynolds, R.C. Rose: Robust text independent speaker identification using Gaussian mixture speaker models, IEEE Trans. Speech Audio Process. **3**, 72–83 (1995)

36.30 D.A. Reynolds, T.F. Quatieri, R.B. Dunn: Speaker verification using adapted Gaussian mixture models, Digital Signal Process. **10**, 19–41 (2000)

36.31 A.E. Rosenberg, S. Parthasarathy: Speaker background models for connected digit password speaker verification, Proc. ICASSP (1996) pp. 81–84

36.32 S. Parthasarathy, A.E. Rosenberg: General phrase speaker verification using sub-word background models and likelihood-ratio scoring, Proc. Int. Conf. Spoken Language Processing (1996) pp. 2403–2406

36.33 O. Siohan, A.E. Rosenberg, S. Parthasarathy: Speaker identification using minimum classification error training, Proc. ICASSP (1998) pp. 109–112

36.34 A.E. Rosenberg, O. Siohan, S. Parthasarathy: Small group speaker identification with common password phrases, Speech Commun. **31**, 131–140 (2000)

36.35 L. Heck, Y. Konig: Discriminative training of minimum cost speaker verification systems, Proc. RLA2C – Speaker Recognition Workshop (1998) pp. 93–96

36.36 A. Rosenberg, O. Siohan, S. Parthasarathy: Speaker verification using minimum verification error training, Proc. ICASSP (1998) pp. 105–108

36.37 J. Navratil, G. Ramaswamy: Detac – a discriminative criterion for speaker verification, Proc. Int. Conf. Spoken Language Processing (2002)

36.38 V.N. Vapnik: *The Nature of Statistical Learning Theory* (Springer, New York 1995)

36.39 W.M. Campbell, D.A. Reynolds, J.P. Campbell: Fusing discriminative and generative methods for speaker recognition: experiments on switchboard and NFI/TNO field data, Proc. ODYSSEY 2004 – The Speaker and Language Recognition Workshop (2004) pp. 41–44

36.40 O. Thyes, R. Kuhn, P. Nguyen, J.-C. Junqua: Speaker identification and verification using eigenvoices, Proc. ICASSP (2000) pp. 242–245

36.41 K.R. Farrell, R. Mammone, K. Assaleh: Speaker recognition using neural networks and conventional classifiers, IEEE Trans. Speech Audio Process. **2**, 194–205 (1994)

36.42 D. Gillick, S. Stafford, B. Peskin: Speaker detection without models, Proc. ICASSP (2005)

36.43 G.N. Ramaswamy, R.D. Zilca, O. Alecksandrovich: A programmable policy manager for conversational biometrics, Proc. Eurospeech (2003)

36.44 H.V. Poor: *An Introduction to Signal Detection and Estimation* (Springer, Berlin, Heidelberg 1994)

36.45 K.P. Li, J.E. Porter: Normalizations and selection of speech segments for speaker recognition scoring, Proc. IEEE ICASSP (1988) pp. 595–598

36.46 F. Bimbot: A tutorial on text-independent speaker verification, EURASIP J. Appl. Signal Process. **4**, 430–451 (2004)

36.47 A. Martin, G. Doddington, T. Kamm, M. Ordowski, M. Przybocki: The det curve in assessment of detection task performance, Proc. Eurospeech (1997) pp. 1895–1898

36.48 A. Martin, M. Przybocki: The NIST 1999 speaker recognition evaluation – an overview, Digital Signal Process. **10**, 1–18 (2000)

36.49 M.A. Siegler, U. Jain, B. Raj, R.M. Stern: Automatic segmentation, classification, and clustering of broadcast news data, Proc. DARPA Speech Recognition Workshop (1997) pp. 97–99

36.50 A.E. Rosenberg, I. Magrin-Chagnolleau, S. Parthasarathy, Q. Huang: Speaker detection in broadcast news databases, Proc. Int. Conf. on Spoken Lang. Processing (1998) pp. 1339–1342

36.51 J.-F. Bonastre, P. Delacourt, C. Fredouille, T. Merlin, C. Wellekens: A speaker tracking system based on speaker turn detection for nist evaluation, Proc. ICASSP (2000) pp. 1177–1180

36.52 A.G. Adami, S.S. Kajarekar, H. Hermansky: A new speaker change detection method for two-speaker segmentation, Proc. ICASSP (2002) pp. 3908–3911

36.53 A.E. Rosenberg, A. Gorin, Z. Liu, S. Parthasarathy: Unsupervised segmentation of telephone conversations, Proc. Int. Conf. on Spoken Lang. Processing (2002) pp. 565–568

36.54 S.S. Chen, P.S. Gopalakrishnan: Speaker, environment and channel change detection and

Amazon EX1010 Page 751

clustering via the bayesian information criterion, Proc. DARPA Broadcast News Transcription and Understanding Workshop (1998), http://www.nist.gov/speech/publications/darpa98/index.htm

36.55 A. Tritschler, R. Gopinath: Improved speaker segmentation and segments clustering using the bayesian information criterion, Proc. Eurospeech (1999)

36.56 A.D. Gordon: *Classification: Methods for the Exploratory Analysis of Multivariate Data* (Chapman Hall, Englewood Cliffs 1981)

36.57 F. Kubala, H. Jin, R. Schwartz: Automatic speaker clustering, Proc. DARPA Speech Recognition Workshop (1997) pp. 108–111

36.58 D. Liu, F. Kubala: Online speaker clustering, Proc. ICASSP (2003) pp. 572–575

36.59 J.-F. Bonastre, F. Bimbot, L.-J. Boë, J. Campbell, D. Reynolds, I. Magrin-Chagnolleau: Person authentication by voice: a need for caution, Proc. Eurospeech (2003) pp. 33–36

36.60 Voice Identification and Acoustic Analysis Subcommittee of the International Association for Identification: Voice comparison standards, J. Forensic Identif. **41**, 373–392 (1991)

36.61 A.E. Rosenberg, S. Parthasarathy, J. Hirschberg, S. Whittaker: Foldering voicemail messages by caller using text independent speaker recognition, Proc. Int. Conf. on Spoken Language Processing (2000)

Amazon EX1010 Page 752

37. Text-Dependent Speaker Recognition

M. Hébert

Text-dependent speaker recognition characterizes a speaker recognition task, such as verification or identification, in which the set of words (or lexicon) used during the testing phase is a subset of the ones present during the enrollment phase. The restricted lexicon enables very short enrollment (or registration) and testing sessions to deliver an accurate solution but, at the same time, represents scientific and technical challenges. Because of the short enrollment and testing sessions, text-dependent speaker recognition technology is particularly well suited for deployment in large-scale commercial applications. These are the bases for presenting an overview of the state of the art in text-dependent speaker recognition as well as emerging research avenues. In this chapter, we will demonstrate the intrinsic dependence that the lexical content of the password phrase has on the accuracy. Several research results will be presented and analyzed to show key techniques used in text-dependent speaker recognition systems from different sites. Among these, we mention multichannel speaker model synthesis and continuous adaptation of speaker models with threshold tracking. Since text-dependent speaker recognition is the most widely used voice biometric in commercial deployments, several results drawn from realistic deployment scenarios are also included.

37.1 **Brief Overview** 743
37.1.1 Features 744
37.1.2 Acoustic Modeling 744
37.1.3 Likelihood Ratio Score 745
37.1.4 Speaker Model Training 746
37.1.5 Score Normalization and Fusion 746
37.1.6 Speaker Model Adaptation 747
37.2 **Text-Dependent Challenges** 747
37.2.1 Technological Challenges 747
37.2.2 Commercial Deployment Challenges 748
37.3 **Selected Results** 750
37.3.1 Feature Extraction 750
37.3.2 Accuracy Dependence on Lexicon 751
37.3.3 Background Model Design 752
37.3.4 T-Norm in the Context of Text-Dependent Speaker Recognition 753
37.3.5 Adaptation of Speaker Models 753
37.3.6 Protection Against Recordings 757
37.3.7 Automatic Impostor Trials Generation 759
37.4 **Concluding Remarks** 760
References 760

37.1 Brief Overview

There exists significant overlap and fundamental differences between text-dependent and text-independent speaker recognition. The underlying technology and algorithms are very often similar. Advances in one field, frequently text-independent speaker recognition because of the NIST evaluations [37.1], can be applied with success in the other field with only minor modifications. The main difference, as pointed out by the nomenclature, is the lexicon allowed by each. Although not restricted to a specific lexicon for enrollment, text-dependent speaker recognition assumes that the lexicon active during the testing is a subset of the enrollment lexicon. This limitation does not exist for text-independent speaker recognition where any word can be uttered during enrollment and testing. The known overlap between the enrollment and testing phase results in very good accuracy with a limited amount of enrollment material (typically less than 8 s of speech). In the case of unknown-text speaker recognition, much more enrollment material is required (typically more than 30 s) to achieve similar accuracy. The theme of *lexical content* of the enrollment and testing sessions is central to text-dependent speaker recognition and will be recurrent during this chapter.

Amazon EX1010 Page 753

Traditionally, text-independent speaker recognition was associated with speaker recognition on entire conversations. Lately, work from *Sturim* et al. [37.2] and others [37.3] has helped bridge the gap between text-dependent and text-independent speaker recognition by using the most frequent words in conversational speech and applying text-dependent speaker recognition techniques to these. They have shown the benefits of using text-dependent speaker recognition techniques on a text-independent speaker recognition task.

Table 37.1 illustrates the challenges encountered in text-dependent speaker recognition (adapted from [37.4]). It can be seen that the two main sources of degradation in the accuracy are channel and lexical mismatch. Channel mismatch is present in both text-dependent and text-independent speaker recognition, but mismatch in the lexical content of the enrollment and testing sessions is central to text-dependent speaker recognition.

Throughout this chapter, we will try to quantify accuracy based on application data (from trial data collections, comparative studies or live data). We will favor live data because of its richness and relevance. Special care will be taken to reference accuracy on publicly available data sources (some may be available for a fee), but in some other cases an explicit reference is impossible to preserve contractual agreements. Note that a comparative study of off-the-shelf commercial text-dependent speaker verification systems was presented at Odyssey 2006 [37.5].

This chapter is organized as follows. The rest of this section explains at a high-level the main components of a speaker recognition system with an emphasis on particularities of text-dependent speaker recognition. The reader is strongly encouraged, for the sake of completeness, to refer to the other chapters on speaker recognition. Section 37.2 presents the main technical and commercial deployment challenges. Section 37.3 is formed by a collection of selected results to illustrate the challenges of Sect. 37.2. Concluding remarks are found in Sect. 37.4.

Table 37.1 Effect of different mismatch types on the EER for a text-dependent speaker verification task (after [37.4]). The corpus is from a pilot with 120 participants (gender balanced) using a variety of handsets. Signal-to-noise ratio (SNR) mismatch is calculated using the difference between the SNR during enrollment and testing (verification). For the purposes of this table, an absolute value of this difference of more than 10 db was considered mismatched. Channel mismatch is encountered when the enrollment and testing sessions are not on the same channel. Finally, lexical mismatch is introduced when the lexicon used during the testing session is different from the enrollment lexicon. In this case, the password phrase was always a three-digit string. LD0 stands for a lexical match such that the enrolment and testing were performed on the same digit string. In LD2, only two digits are common between the enrollment and testing; in LD4 there is only one common digit. For LD6 (complete lexical mismatch), the enrollment lexicon is disjoint from the testing lexicon. Note that, when considering a given type of mismatch, the conditions are matched for the other types. At EERs around 8%, the 90% confidence interval on the measures is 0.8%

Type of mismatch	Accuracy (EER) (%)
No mismatch	7.02
SNR mismatch	7.47
Channel mismatch	9.76
Lexical mismatch (LD2)	8.23
Lexical mismatch (LD4)	13.4
Complete lexical mismatch (LD6)	36.3

37.1.1 Features

The first text-dependent speaker recognition system descriptions that incorporate the main features of the current state of the art date back to the early 1990s. In [37.6] and [37.7], systems have feature extraction, speaker models and score normalization using a likelihood ratio scheme. Since then, several groups have explored different avenues. The work cited below is not restricted to the text-dependent speaker recognition field, nor is it intended as an exhaustive list. Feature sets usually come in two flavors: MEL [37.8] or LPC (linear predictive coding) [37.6, 9] cepstra. Cepstral mean subtraction and feature warping have proved effective on cellular data [37.10] and are generally accepted as an effective noise robustness technique. The positive role of dynamic features in text-dependent speaker recognition has recently been reported in [37.11]. Finally, a feature mapping approach [37.12] has been proposed as an equivalent to speaker model synthesis [37.13]; this is an effective channel robustness technique.

37.1.2 Acoustic Modeling

Several modeling techniques and their associated scoring schemes have been investigated over the years. By far the most common modeling scheme across

speaker recognition systems is the hidden Markov model (HMM) [37.14]. The unit modeled by the HMM depends heavily on the type of application (Fig. 37.1). In an application where the enrollment and testing lexicon are identical and in the same order (*My voice is my password* as an example), a sentence-level HMM can be used. When the order in which the lexicon appears in the testing phase is not the same as the enrollment order, a word-level unit is used [37.9, 15]. The canonical application of word-level HMMs is present in digit-based speaker recognition dialogs. In these, all digits are collected during the enrollment phase and a random digit sequence is requested during the testing phase. Finally, phone-level HMMs have been proposed to refine the representation of the acoustic space [37.16–18]. The choice of HMMs in the context of text-dependent speaker recognition is motivated by the inclusion of inherent time constraints.

The topology of the HMM also depends on the type of application. In the above, standard left-to-right N-state HMM have been used. More recently, single-state HMMs [also called Gaussian mixture models (GMMs)] have been proposed to model phoneme-level acoustics in the context of text-dependent speaker recognition [37.19] and later applied to text-independent speaker recognition [37.20]. In this case, the temporal information represented by the sequence of phonemes is dictated by an external source (a speech recognition system) and not inscribed in the model's topology. Note that GMMs have been extensively studied, and proved very effective, in the context of text-independent speaker recognition.

In addition to the mainstream HMMs and GMMs, there exists several other modeling methods. Support vector machine (SVM) classifiers have been suggested for speaker recognition by *Schmidt* and *Gish* [37.21] and have become increasingly used in the text-independent speaker recognition field [37.22, 23]. To our knowledge, apart from [37.24, 25], there has been no thorough study of an SVM-based system on a text-dependent speaker recognition task. In this context, the key question is to assess the robustness of an SVM-based system to a restricted lexicon. Dynamic time warping (DTW) algorithms have also been investigated as the basis for text-dependent speaker recognition [37.26, 27]. Finally, neural networks (NNs) modeling methods also form the basis for text-dependent speaker recognition algorithms [37.28, 29]. Since the bulk of the literature and advances on speaker recognition are based on algorithms that are built on top of HMM or GMM, we will focus for the rest of this chapter on those. We believe, however, that the main conclusions and results herein apply largely to the entire field of text-dependent speaker recognition.

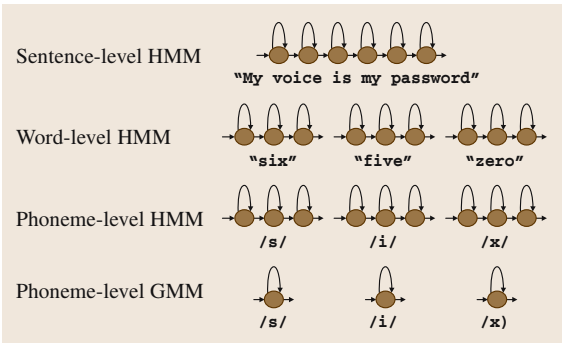


Fig. 37.1 Hidden Markov model (HMM) topologies

37.1.3 Likelihood Ratio Score

As mentioned in a previous section, speaker recognition can be split into speaker identification and verification. In the case of speaker identification, the score is simply the likelihood, template score (in the case of DTW), or posterior probability in the case of an NN. For speaker verification, the standard scoring scheme is based on the competition between two hypothesis [37.30].

- H_0: the test utterance is from the claimed speaker C, modeled by λ;
- H_1: the test utterance is from a speaker other than the claimed speaker C, modeled by $\overline{\lambda}$.

Mathematically, the likelihood ratio $[L(\boldsymbol{X}|\lambda)]$ detector score is expressed as

$$L(\boldsymbol{X}|\lambda) = \log p(\boldsymbol{X}|\lambda) - \log p(\boldsymbol{X}|\overline{\lambda}) \,, \tag{37.1}$$

where $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_\mathrm{T}\}$ is the set of feature vectors extracted from the utterance and $p(\boldsymbol{X}|\lambda)$ is the likelihood of observing $\boldsymbol{X}$ given model λ. H_0 is represented by a model λ of the claimed speaker C. As mentioned above, λ can be an HMM or a GMM that has been trained using features extracted from the utterances from the claimed speaker C during the enrollment phase. The representation of H_1 is much more subtle because it should, according to the above, model all potential speakers other than C. This is not tractable in a real system. Two main approaches have been studied to model $\overline{\lambda}$. The first consists of selecting N background or cohort speakers, to model individually $(\lambda_0, \lambda_1, \ldots, \lambda_{N-1})$ and to combine their likelihood score on the test utterance.

The other approach uses speech from a pool of speakers to train a single model, called a general, background or universal background model (UBM). A variant of the UBM, widely used for its channel robustness, is to train a set of models by selecting utterances based on some criteria such as gender, channel type, or microphone type [37.8]. This technique is similar in spirit to the one presented in [37.12]. Note that for the case of text-dependent speaker recognition, it is beneficial to train a UBM with data that lexically match the target application [37.19].

37.1.4 Speaker Model Training

In order to present a conceptual understanding of the text-dependent speaker recognition field, unless otherwise stated, we will assume only two types of underlying modeling: a single GMM for all acoustic events, which is similar to the standard modeling found in text-independent tasks. We will call this the *single-GMM* approach. The other modeling considered is represented as *phoneme-level GMM* on Fig. 37.1. This approach will be called phonetic-class-based verification (PCBV as per [37.19]). This choice is motivated by simplicity, availability of published results, as well as current trends to merge known and text-independent speaker recognition (Sect. 37.1).

For these types of modeling, training of the speaker model is performed using a form of Bayesian adaptation [37.30, 31], which alters the parameters of $\overline{\lambda}$ using the features extracted from the speech collected during the enrollment phase. As will be shown later, this form of training for the speaker models is well suited to allow adaptation coefficients that are different for means, variances, and mixture weights. This, in turn has an impact on the accuracy in the context of text-dependent speaker recognition.

37.1.5 Score Normalization and Fusion

Although the score coming from the likelihood ratio detector (37.1) discriminates genuine speakers from imposters well, it remains fragile. Several score normalization techniques have been proposed to improve robustness. We will discuss a few of those.

The first approach is called the H-norm, which stands for handset normalization, and is aimed at normalizing handset variability [37.32], especially cross-channel variability. A similar technique called the Z-norm has also been investigated in the context of text-dependent speaker recognition with adaptation [37.33]. Using a set ($\simeq 200$) of impostor test utterances with known handset and/or gender labels, a newly trained speaker model is challenged. The scores calculated using (37.1) are fitted using a Gaussian distribution to estimate their mean $[\mu_{\mathrm{H}}(\lambda)]$ and standard deviations $[\sigma_{\mathrm{H}}(\lambda)]$ for each label H. At test time, a handset and/or gender labeler [37.8,32] is used to identify the label H of the testing utterance. The normalized score is then

$$L_{\text{H-norm}}(\boldsymbol{X}|\lambda, H) = \frac{L(\boldsymbol{X}|\lambda) - \mu_{\mathrm{H}}(\lambda)}{\sigma_{\mathrm{H}}(\lambda)} \,. \tag{37.2}$$

This technique is computationally very efficient because $\mu_{\mathrm{H}}(\lambda)$ and $\sigma_{\mathrm{H}}(\lambda)$ are calculated once after the enrollment. It can, however, become inefficient when adaptation of the speaker model (Sect. 37.1.6) occurs because $\mu_{\mathrm{H}}(\lambda)$ and $\sigma_{\mathrm{H}}(\lambda)$ need to be recalculated.

Another score normalization technique widely used is called test normalization or T-norm [37.34]. This approach is applied to text-dependent speaker recognition in [37.35]. It can be viewed as the dual to H-norm in that, instead of challenging the target speaker model with a set imposter test utterances, a set of imposter speaker models (T) are challenged with the target test utterance. Assuming a Gaussian distribution of those scores, $\mu_{\mathrm{T}}(\boldsymbol{X})$ and $\sigma_{\mathrm{T}}(\boldsymbol{X})$ are calculated and applied using

$$L_{\text{T-Norm}}(\boldsymbol{X}|\lambda, T) = \frac{L(\boldsymbol{X}|\lambda) - \mu_{\mathrm{T}}(\boldsymbol{X})}{\sigma_{\mathrm{T}}(\boldsymbol{X})} \,. \tag{37.3}$$

By construction, this technique is computationally very expensive because the target test utterance has to be applied to the entire set of imposter speaker models.

Notwithstanding the computational cost, the H-norm and T-norm developed in the context of text-independent speaker recognition need to be adapted for the text-dependent speaker recognition. These techniques have been shown to be heavily dependent on the lexicon of the set of imposter utterances (H-norm [37.36]) and on the lexicon of the utterances used to train the imposter speaker models (T-norm [37.35]). The issue of lexical dependency or mismatch is not present in a text-independent speaker recognition task, but heavily influences text-dependent speaker recognition system designs [37.4]. We will come back to this question later (Sect. 37.3.2 and Sect. 37.3.4).

Finally, as in the text-independent speaker recognition systems, score fusion is present in text-dependent systems and related literature [37.29]. The goal of fusion is to combine classifiers that are assumed to make uncorrelated errors in order to build a better performing overall system.

37.1.6 Speaker Model Adaptation

Adaptation is the process of extending the enrollment session to the testing sessions. Common wisdom tells us that *the more speech you train with, the better the accuracy will be*. This has to be balanced with requirements from commercial deployments where a very long enrollment sessions is negatively received by end customers. A way to circumvent this is to fold back into the enrollment material any testing utterance that the system has a good confidence of having been spoken by the same person as the original speaker model. Several studies on unknown [37.37, 38] and text-dependent [37.39, 40] speaker recognition tasks have demonstrated the effectiveness of this technique. Speaker model adaptation comes in two flavors. Supervised adaptation, also known as retraining or manual adaptation, implies an external verification method to assess that the current speaker is genuine. That can be achieved using a secret piece of information or another biometric method. The second method is called unsupervised adaptation. In this case, the decision taken by the speaker recognition system (a verification system in this case) is used to decide on the application of adaptation of the speaker model with the current test utterance. Supervised adaptation outperforms its unsupervised counterpart in all studies. A way to understand this fact is to consider that unsupervised adaptation requires a good match between the target speaker model and the testing utterance to adapt the speaker model; hence this new utterances does not bring new variability representing the speaker, the transmission channel, the noise environment, etc. The supervised adaptation scheme, since it is not based on the current utterance, will bring these variabilities to the speaker model in a natural way. Under typical conditions, supervised adaptation can cut, on text-dependent speaker verification tasks, the error rates by a factor of five after 10–20 adaptation iterations.

37.2 Text-Dependent Challenges

The text-dependent speaker recognition field faces several challenges as it strives to become a mainstream biometric technique. We will segregate those into two categories: technological and deployment. The technology challenges are related to the core algorithms. Deployment challenges are faced when bringing the technology into an actual application, accepting live traffic. Several of these challenges will be touched on in subsequent sections where we discuss the current research landscape and a set of selected results (Sect. 37.3). This section is a superset of challenges found in a presentation by *Heck* at the Odyssey 2004 workshop [37.41]. Note that the points raised here can all give rise to new research avenues; some will in fact be discussed in following sections.

37.2.1 Technological Challenges

Limited Data and Constrained Lexicon

As mentioned in Sect. 37.1, text-dependent speaker recognition is characterized by short enrollment and testing session. Current commercial applications use enrollment sessions that typically consist of multiple repetitions (two or three) of the enrollment lexicon. The total speech collected is usually 4–8 s (utterances are longer than that, but silence is usually not taken into account). The testing session consists of a single (or sometimes two) repetitions of a subset of the enrollment lexicon, for a total speech input of 2–3 s. These requirements are driven by usability studies which show that shorter enrollment and testing sessions are best perceived by end customers.

The restricted nature of the lexicon (hence text-dependent speaker recognition), is a byproduct of the short enrollment sessions. To achieve deployable accuracies under the short enrollment and testing constraints, the lexicon has to be restricted tremendously. Table 37.2 lists several examples of enrollment lexicon present is deployed applications. Table 37.3 describes typical testing strategies given the enrollment lexicon. In most cases, the testing lexicon is chosen to match the enrollment lexicon exactly. Note that, for random (and pseudo random) testing schemes, a *2-by-4* approach is sometimes used: in order to reduce the cognitive load, a four-digit string repeated twice is requested from

Table 37.2 Examples of enrolment lexicon

Abreviation	Description
E	Counting from 1 to 9: *one two three* ...
T	10-digit telephone number
S	9-digit account number
N	First and last names
MVIMP	*My voice is my password*

Table 37.3 Examples of testing lexicon. Note that the abbreviations refer to Table 37.2 and each line gives depicts a potential testing lexicon given the enrolment lexicon

Abreviation	Description
E	Counting from 1 to 9: *one two three ...*
R	Random digit sequence *2 6 8 5 2 6 8 5*
pR	Pseudorandom digit sequence from E *2 3 6 7 2 3 6 7*
T	Same 10-digit telephone number as enrolment
	Random digit sequence selected from enrolment lexicon
	Pseudorandom digit sequence selected from enrolment lexicon
S	Similar to T but for a nine-digit account number
N	First and last names
MVIMP	*My voice is my password*

the user. This makes for a longer verification utterance without increasing the cognitive load: a totally random eight-digit string could hardly be remembered by a user. Table 37.4 shows a summary of the accuracy in different scenarios. We reserve discussion of these results for Sect. 37.3.2.

Channel Usage

It is not rare to see end customers in live deployments using a variety of handset types: landline phones, pay phones, cordless phones, cell phones, etc. This raises the issue of their impact on accuracy of channel usage. A cross-channel attempt is defined as a testing session originating from a different channel than the one used during the enrollment session. It is not rare to see the proportion of cross-channel calls reach 25–50% of all genuine calls in certain applications. The effect on the accuracy is very important ranging from doubling the EER [37.4, 42] to quadrupling [37.42] the EER on some commercial deployments. This is a significant area where algorithms must be improved. We will come back to this later.

Aging of Speaker Models

It has been measured in some commercial trials [37.42] and in data collections [37.15] that the accuracy of a text-dependent speaker recognition system degrades slowly over time. In the case of [37.15], the error rate increased by 50% over a period two months. There exists several sources of speaker model aging, the main ones being

Table 37.4 Speaker verification results (EERs) for different lexicon. Refer to Tables 37.2 and 37.3 for explanations of the acronyms. Empty cells represent the fact that pseudorandom strings (pR) do not apply to S since the pseudorandom string is extracted from an E utterance. Italicized results depict conditions that are not strictly text-dependent speaker verification experiments. At EERs of 5–10%, the 90% confidence interval on the measures is 0.3–0.4%

Verify	E	S	R	pR	N
Enroll					
E	6.16%	10.2%	13.2%	10.4%	*36.2%*
S	11.6%	5.05%	14.2%		*39.3%*
R	10.7%	9.43%	11.5%	10.0%	*36.4%*
pR	10.0%		11.3%	8.05%	*35.6%*
N	*38.9%*	*39.7%*	*39.3%*	*39.1%*	10.6%

natural aging, channel usage, and behavioral changes. Natural aging is related to the physiological changes that occur to the phonatory apparatus over long periods of time. Channel usage changes over time can cause the speaker model to become outdated with respect to the current channel usage. Finally, behavioral changes occur when users get more exposure to the voice interface and thus alter the way in which they interact with it. As an example, first-time users of a speech application (usually the enrollment session) tend to cooperate with the system by speaking slowly under friendly conditions. As these users get more exposure to the application, they will alter the way that they interact with it and use it in adverse conditions (different channels, for example). All of these factors affect the speaker models and scoring, and thus are reflected in the accuracy. The common way to mitigate this effect is to use speaker model adaptation (Sect. 37.1.6 and Sect. 37.3.5).

37.2.2 Commercial Deployment Challenges

Dialog Design

One of the main pitfalls in deploying a speech-based security layer using text-dependent speaker recognition is poor dialog design choices. Unfortunately, these decisions are made very early in the life cycle of an application and have a great impact on the entire life of the application. Examples [37.41] are

1. small amount of speech collected during enrollment and/or verification
2. speech recognition difficulty of the claim of identity (such as a first and last names in a long list)

3. poor prompting and error recovery
4. lexicon mismatch between enrollment and verification

One of the challenges in deploying a system is certainly protection against recordings since the lexicon is very restricted. As an example, in the case where the enrollment and verification lexicon is *My voice is my password* or a telephone number, once a fraudster has gained access to a recording from the genuine speaker, the probability that they can gain access has greatly increased. This can be addressed by several techniques (one can also think about combining them). The first technique consists of explicitly asking for a randomized subset of the lexicon. This does not lengthen the enrollment session and is best carried out if the enrollment lexicon consists of digits. The second is to perform the verification process across the entire dialog even if the lexical mismatch will be high (Sect. 37.3.2 and Sect. 37.3.6), while maintaining a short enrollment session. A third technique is to keep a database of *trusted* telephone numbers for each user (home, mobile, and work) and to use this external source of knowledge to improve security and ease of use [37.43]. Finally, a challenge by a secret *knowledge* question drawn from a set of questions can also be considered. These usually require extra steps during the enrollment session. It is illusory to think that a perfect system (no errors) can be designed, the goal is simply to raise the bar of

1. the amount of information needed and
2. the sophistication required by a fraudster to gain access.

There are two other considerations that come into play in the design of an application. The first is related to the choice of the token for the identity claim in the case of speaker verification. The identity claim can be combined with the verification processing in systems that have both speaker and speech recognition. In this case, an account number or a name can be used. As can be seen from Table 37.4, verification using text (first and last names) is challenging, mainly due to the short length of speech. For a very large-scale deployment, recognition can also be very challenging. *Heck* and *Genoud* have suggested combining verification and recognition scores to re-sort the N-best list output from the recognizer and achieve significant recognition accuracy gains [37.44]. Other means of claiming an identity over the telephone include caller identification (ID) and keypad input. In these cases, the verification utterance can be anything, including a lexicon common to all users. The second consideration is the flexibility that the enrollment lexicon provides to dynamically select a subset of the lexicon with which to challenge the user in order to protect against recordings (see above). This is the main reason why digit strings (telephone and account number, for example) are appealing for a relatively short enrollment session. A good speaker model can be built to deliver good accuracy even with a random subset of the enrollment lexicon as the testing lexicon (Sect. 37.3.2).

Cost of Deployment

The cost of deploying a speaker recognition system into production is also a challenge. Aside from dialog design and providing the system with a central processing unit (CPU), storage, and bandwidth, setting the operating point (the security level or the target false-acceptance rate) has a major impact on cost. As can be seen from the discussion above, there are a wide variety of dialogs that can be implemented, and all of these require their own set of thresholds depending on the level of security required. This is a very complex task that is usually solved by collecting a large number of utterances and hiring professional services from the vendor to recommend those thresholds. This can be very costly for the application developer. Recently, there has been an effort to build off-the-shelf security settings into products [37.36]. This technique does not require any data and is accurate enough for small- to medium-scale systems or initial security settings for a trial. Most application developers, however, want to have a more-accurate picture of the accuracy of their security layer and want a measurement on actual data of the standard false accept (FA), false reject (FR), and reprompt rates (RR, the proportion of genuine speakers that are reprompted after the first utterance). To this end a data collection is set up. The most expensive portion of data collection is to gather enough impostor attempts to set the decision threshold to achieve the desired FA rate with a high level of confidence. Collecting genuine speaker attempts is fairly inexpensive by comparison. An algorithm aimed at setting the FA rate without specifically collecting impostor attempts has been presented [37.45]. See Sect. 37.3.7 for more details.

Forward Compatibility

Another challenge from a deployment perspective, but that has ramifications into the technology side, is forward compatibility. The main point here is that the database of enrollee (those that have an existing speaker model) should be forward compatible to revision of: (a) the

application, and (b) the software and its underlying algorithms. Indeed, an application that has been released using a security layer based on a first name and last name lexicon is confined to using this lexicon. This is very restrictive. Also, in commercial systems, the enrollment utterances are not typically saved: the speaker model is the unit saved. This speaker model is a parameterized version of the enrollment utterances. The first step that goes into this parameterization is the execution of the front-end feature extractor (Sect. 37.1.1). The definition of these features is an integral part of the speaker model and any change to this will have a negative impact on accuracy. This also restricts what research can contribute to an existing application.

37.3 Selected Results

In this section, we will present several results that either support claims and assertions made earlier or illustrate current challenges in text-dependent speaker recognition. It is our belief that most if not all of these represent potential areas for future advances.

37.3.1 Feature Extraction

Some of the results presented below are extracted from studies done on text-independent speaker recognition tasks. We believe that the algorithms presented should also be beneficial to text-dependent tasks, and thus could constitute the basis for future work.

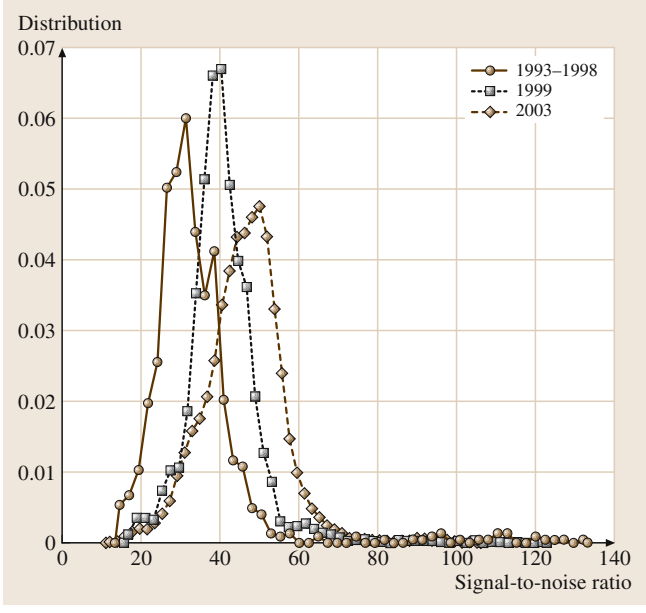


Fig. 37.2 Signal-to-noise ratio distribution from cellular waveforms for three different periods. The data are from a mix of in-service data, pilot data, and data collection

Impact of Codecs on Accuracy

The increasing penetration of cellular phones in society has motivated researchers to investigate the impact of different codecs on speaker recognition accuracy. In 1999, a study of the impact of different codecs was presented [37.46]. Speech from an established corpora was passed through different codecs (GSM, G.729 and G723.1), and resynthesized. The main conclusion of this exercise was that the accuracy drops as the bit rate is reduced. In that study, speaker recognition from the codec parameters themselves was also presented.

Figure 37.2 presents the distribution of the signal-to-noise ratio (SNR) from different internal corpora (trials and data collections) for cellular data only. We have organized them by time periods. A complete specification of those corpora is not available (codecs used, environmental noise conditions, analog versus digital usage, etc.). Nevertheless, it is obvious that speech from cellular phones is cleaner in the 2003 corpora than ever before. This is likely due to more-sophisticated codecs and better digital coverage. It would be interesting to see the effect on speaker recognition and channel identification of recent codecs like CDMA (code division multiple access) in a study similar to [37.46]. This is particularly important for commercial deployments of (text-dependent) speaker recognition, which are faced with the most up-to-date wireless technologies.

Feature Mapping

Feature mapping was introduced by *Reynolds* [37.12] to improve channel robustness on a text-independent speaker recognition task. Figure 37.3a describes the offline training procedure for the background models. The root GMM is usually trained on a collection of utterances from several speakers and channels using k-means and EM (expectation maximization) algorithms. MAP (maximum a posteriori) adaptation [37.31] is used to adapt the root GMM with utterances coming from single channels to produce GMMs for each channel. Because of

Amazon EX1010 Page 760

the MAP adaptation structure, there exists a one-to-one correspondence between the Gaussians from the root and channel GMMs, and transforms between Gaussians of these GMMs can be calculated [37.12]. The transforms from the channel GMMs Gaussians to the root GMM Gaussians can be used to map features from the those channels onto the root GMM. The online procedure is represented on Fig. 37.3b. For an incoming utterance, the channel is first selected by picking the most likely over the entire utterance based on log $p(\boldsymbol{X}|\overline{\lambda})$ from (37.1). The features are then mapped from the identified channel onto the root GMM. At this point, during training of the speaker model, mapped features are used to adapt the root GMM. Conversely, during testing, the mapped features are used to score the root and speaker model GMMs to perform likelihood-ratio scoring (Sect. 37.1.3).

Feature mapping has proved its effectiveness for channel robustness (see [37.12] for more details). It is of interest for text-dependent speaker recognition because it is intimately related to speaker model synthesis (SMS) [37.13], which has demonstrated its effectiveness for such tasks [37.40]. To our knowledge, feature mapping has never been implemented and tested on a text-dependent speaker recognition task.

Speaker and Speech Recognition Front Ends

The most common feature extraction algorithms for speech recognition are mel-filter cepstral coefficients (MFCCs) and linear predictive cepstral coefficients (LPCCs). These algorithms have been developed with the objective of classifying phonemes or words (lexicon) in a speaker-independent fashion. The most common feature extraction algorithms for speaker recognition are, surprisingly, MFCC or LPCC. This is surprising because of the fact that speaker recognition objective is the classification of speakers, with no particular emphasis on lexical content. A likely, but still to be proven, explanation for this apparent dichotomy is that MFCC and LPCC are very effective at representing a speech signal in general. We believe that other approaches are worth investigating.

Several studies have tried to change the speaker recognition paradigm for feature extraction (see [37.47, 48], to name a few). In [37.47], a neural net with five layers is discriminatively trained to maximize speaker discrimination. Then the last two layers are discarded and the resulting final layer constitutes the feature extractor. Authors report a 28% relative improvement over MFCCs in a text-independent speaker recognition task.

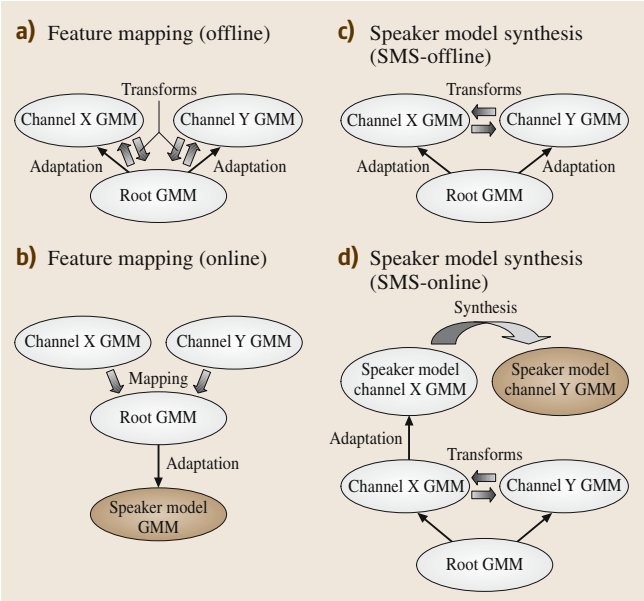


Fig. 37.3a–d Feature mapping and speaker model synthesis (SMS). GMMs with oblique lines were constructed using synthesized data

Although developed with channel robustness in mind, we believe that this technique holds a lot of potential. In [37.48], wavelet packet transforms are used to analyze the speech time series instead of the standard Fourier analysis. Authors report a 15–27% error rate reduction on an text-independent speaker recognition task. Despite the improvements reported, these algorithms have not reached mainstream adoption to replace MFCCs or LPCCs.

37.3.2 Accuracy Dependence on Lexicon

As mentioned in the Chap. 36, the theme of the lexical content of the password phrase is central in text-dependent speaker recognition. A study by *Kato* and *Shimizu* [37.15] has demonstrated the importance of preserving the sequence of digits to improve accuracy. The authors report a relative improvement of more than 50% when the digit sequence in the testing phase preserves the order found during enrollment.

Table 37.4 presents a similar trend as well as additional conditions. The data for these experiments was collected in September of 2003 from 142 unique speakers (70 males and 72 females). Each caller was requested to complete at least four calls from a variety of handsets (landline, mobile, etc.) in realistic noise conditions. In each call, participants were requested to read a sheet

Amazon EX1010 Page 761

with three repetitions of the phrases E, S, R, pR, and N (refer to Tables 37.2 and 37.3 for explanations of the acronyms). There were only eight unique S digit strings in the database in order to use round-robin imposter attempts, and a given speaker was assigned only one S string. The interesting fact about this data set is that we can perform controlled experiments: for every call, we can substitute E for S and vice versa, or any other types of utterances. This allows the experimental conditions to preserve:

1. callers
2. calls (and thus noise and channel conditions) and vary lexical content only

The experiments in Table 37.4 are for speaker verification and the results presented are the equal error rates (EERs). All results are on 20 k genuine speakers attempts and 20 k imposter attempts. The enrollment session consists of three repetitions of the enrollment token while the testing sessions has two repetitions of the testing token. Let us define and use the following notation to describe an experiment's lexical content for the enrollment and verification: eXXX_vYY, which defines the enrollment as three repetitions of X, and the testing attempts as two repetitions of Y. For example, the EER for eEEE_vRR is 13.2%.

The main conclusion of *Kato* and *Shimizu* [37.15] are echoed in Table 37.4: sequence-preserving digit strings improves accuracy. Compare the EERs for eEEE_vRR with eEEE_vpRpR. Also, eEEE_vEE, eSSS_vSS, epRpRpR_vpRpR, and eNNN_vNN all perform better than eRRR_vRR. This illustrates the capture by the speaker model of coarticulation: E and R utterances have exactly the lexicon (1 to 9) but in a different order. Note that the accuracy of the first and last names is significantly worse than E or S on the diagonal of Table 37.4. This is due to the average length of the password phrase: an E utterance has on average 3.97 s of speech while an N utterance has only 0.98. Finally, we have included cross-lexicon results, which are more relevant to text-independent speaker recognition (for example eEEE_vNN). This illustrates the fact that, with very short enrollment and verification sessions, lexically mismatched attempts impact accuracy significantly. In [37.4], the effect of lexical mismatch is compared with the effect of SNR mismatch and channel mismatch. It is reported that a moderate lexical mismatch can degrade the accuracy more than SNR and is comparable to channel mismatch (Table 37.1). Finally, *Heck* [37.41] noted that 'advances in robustness to linguistic mismatches will form a very fruitful bridge between text-independent and dependent tasks.' We share this view and add that solving this problem would open avenues to perform accurate and non-intrusive protection against recordings by verifying the identity of a caller across an entire call even with a very short enrollment session. We will explore this more in Sect. 37.3.6.

37.3.3 Background Model Design

The design of background models is crucial to the resulting accuracy of a speaker recognition system. The effect of the lexicon can also be seen in this context. As an example, in a text-dependent speaker recognition task based on *My voice is my password* (MVIMP) as the password phrase, adapting a standard background model with utterances of the exact target lexicon can have a significant positive impact. Without going into the details of the data set, the EER drops from 16.3% to 11.8% when 5 k utterance of MVIMP were used to adapted the background model. This is consistent with one of the results from [37.19]. In [37.49], an algorithm for the selection of background speakers for a target user is presented as well as results on a text-dependent task. The algorithm is based on similarity between two users' enrollment sessions. Lexical content was not the focus of that study, but it would be interesting to see if the lexical content of each enrollment sessions had an influence on the selection of competitive background speakers, i. e., whether similar speakers have significant lexical overlap.

From the point of view of commercial deployments, the use of specialized background models for each password phrase, or on a per-target user basis, is unrealistic. New languages also require investments to develop language-specific background models. The technique in [37.50] does not require offline training of the background model. The enrollment utterances are used to train the 25-state HMM speaker model and a lower-complexity background model. The reasoning behind this is that the reduced complexity model will smear the speaker characteristics that are captured by the higher-complexity model (speaker model). Unfortunately, this technique has never been compared to a state-of-the-art speaker recognition system.

37.3.4 T-Norm in the Context of Text-Dependent Speaker Recognition

As mentioned in Sect. 37.1.5, the T-norm is sensitive to the lexicon of the utterances used to train the imposter speaker models composing the cohort [37.45]. In that study, the data used is a different organization of the data set described in Sect. 37.3.2 that allows a separate set of speakers to form the cohort needed by the T-norm. The notation introduced in Sect. 37.3.2 can also be adapted to describe the lexicon used for the cohort: eXXX_vYY_cZZZ describes an experiment for which the speaker models in the cohort are enrolled with three repetitions of Z. The baseline system used for the experiments in that study is described in *Teunen* et al. [37.13]. It uses gender- and handset-dependent background models with speaker model synthesis (SMS). The cohort speaker models are also tagged with gender and handset; the cohorts are constructed on a per-gender and per-handset basis. During an experiment, the selection of the cohort can be made after the enrollment session based on the detected handset and gender from the enrollment session. It can also be made at test time using the handset and gender detected from the testing utterance. We denote the set of cohorts selected at testing by C_t. In the results below, we consider only the experiments eEEE_vEE or eSSS_vSS with lexically rich (cSSS) or lexically poor cohorts (cEEE). A note on *lexically rich and poor* is in order: the richness comes from the variety of contexts in which each digit is found. This lexical richness in the cohort builds robustness with respect to the variety of digits strings that can be encountered in testing.

Table 37.5 shows the accuracy using test-time cohort selection C_t in a speaker verification experiment. It is interesting to note that the use of a lexically poor cohort (cEEE) in the context of an eSSS_vSS experiment significantly degrades accuracy. In all other cases in Table 37.5, the T-norm improves the accuracy. A smoothing scheme was introduced to increase robustness to the lexical poorness of the cEEE cohort. It is suggested that this smoothing scheme increases the robustness to lexical mismatch for the T-norm. The smoothing scheme is based on the structure of (37.1), which can be rewritten in a form similar to (37.3) using $\mu(\boldsymbol{X}) = \log p(\boldsymbol{X}|\overline{\lambda})$ and $\sigma(\boldsymbol{X}) = 1$. The smoothing is then an interpolation of the normalizing statistics between standard T-norm [$\mu_{\mathrm{T}}(\boldsymbol{X})$ and $\sigma_{\mathrm{T}}(\boldsymbol{X})$] and background model normalization [$\log p(\boldsymbol{X}|\overline{\lambda})$ and 1]. Figure 37.4 shows DET (detecttion error trade-off) curves for the eSSS_vSS experiment with different cohorts. It is shown that the T-norm with a cEEE cohort degrades the accuracy compared to the baseline (no T-norm) as mentioned above. Smoothed T-norm achieves the best accuracy irrespective of the cohort's lexical richness (a 28% relative improvement of FR at fixed FA).

Table 37.5 The FR rates at FA = 1% for various configurations [37.35]. Based on the lower number of trials (the impostor in our case), the 90% confidence interval on the measures is 0.6%. (© 2005 IEEE)

Experimental set-up	Baseline (no T-norm)	T-norm C^t cEEE	T-norm C^t cSSS
eEEE_vEE	17.10%	14.96%	14.74%
eSSS_vSS	14.44%	16.39%	10.42%

37.3.5 Adaptation of Speaker Models

Online adaptation of speaker models [37.39,40] is a central component of any successful speaker recognition application, especially text-dependent tasks because of the short enrollment sessions. The results presented in this section all follow the same protocol [37.40]. Unless otherwise stated, the data comes from a Japanese digit data collection. There were 40 speakers (gender balanced) making at least six calls: half from landlines

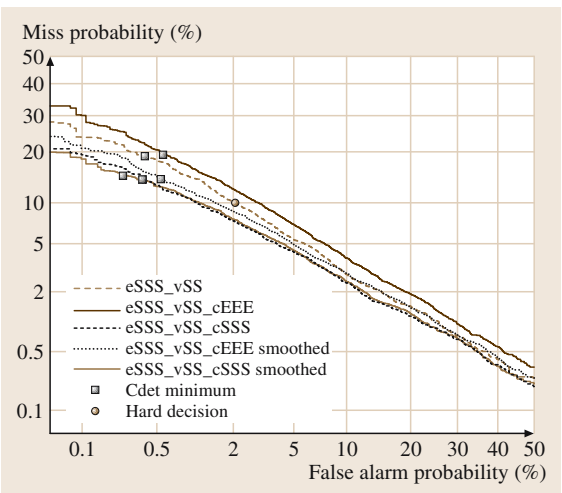


Fig. 37.4 DET curve showing the T-norm and its smoothed variant in the case of eSSS_vSS with the cohort selected at testing time (C^t). The interested reader should refer to the source paper for additional details. For clarity, the order in the legend is the same as the order of the curves at false alarm probability equal to 0.1% (after [37.35], ©2005 IEEE)

and have from cellular phones. The data was heavily recycled to increase the number of attempts by enrolling several speaker models for a given speaker and varying the enrollment lexicon (130–150 on average). For any given speaker model, the data was divided into three disjoint sets: an enrollment set to build the speaker model, an adaptation set, and a test set. The adaptation set was composed of one imposter attempt for every eight genuine attempts (randomly distributed). The experiments were designed as follows. First all of the speaker models were trained and the accuracy was measured right after the enrollment using the test set. Then, one adaptation utterance was presented to each of the speaker models. At this point a decision to adapt or not was made (see below). After this first iteration of adaptation, the accuracy was measured using the test set (without the possibility of adaptation on the testing data). The adaptation and testing steps were repeated for each adaptation iterations in the adaptation set. This protocol was designed to control with great precision all the factors related to the adaptation process: the accuracy was measured after each adaptation iteration using the same test set and they are therefore directly comparable.

Two different types of adaptation experiments can be designed based on how the decision to update the speaker models is made: supervised and unsupervised [37.39]. Both types give insight into the adaptation process and its effectiveness, and both have potential applicability in commercial deployments. Supervised adaptation experiments use the truth about the source of the adaptation utterances: an utterance is used for updating a speaker model only when it is from the target speaker. This allows the update process of the speaker models to be optimal for two reasons. The first is that there is no possibility of corruption of a speaker model by using utterances from an imposter. The second comes from the fact that all adaptation utterances from the target speaker are used to update speaker model. This allows more data to update the speaker model, but more importantly it allows poorly scoring utterances to update the speaker model. Because these utterances score poorly, they have the most impact on accuracy because they bring new and unseen information (noise conditions, channel types, etc.) into the speaker model. This has a significant impact on the cross-channel accuracy, as we will show below. Supervised adaptation can find its applicability in commercial deployments in a scenario where two-factor authentication is used, where one of the factors is acoustic speaker recognition. As an example, in a dialog where acoustic speaker recognition and authentication using a secret challenge question are used, supervised adaptation can be used if the answer to the secret question is correct.

In unsupervised adaptation, there is no certainty about the source of the adaptation utterance and usually the score on the adaptation utterance using the non-updated speaker model is used to make the decision to adapt or not [37.33, 36, 39, 40]. A disadvantage of this approach is the possibility that the speaker models may become adapted on imposter utterances that score high. This approach also reduces the number of utterances that are used to update the speaker model. More importantly, it reduces the amount of new and unseen information that is used for adaptation because this new and unseen information will likely score low on the existing speaker model and thus not be selected for adaptation.

Variable Rate Smoothing

Variable rate smoothing (VRS) was introduced in [37.30] for text-independent speaker recognition. The main idea is to allow means, variances, and mixture weights to be adapted at different rates. It is well known that the first moment of a distribution takes fewer samples to estimate than the second moment. This should be reflected in the update equations for speaker model adaptation by allowing the smoothing coefficient to be different for means, variances, and mixture weights. The authors reported little or no gains on their task. However, VRS should be useful for text-dependent speaker recognition tasks due to the short enrollment sessions. Please refer to [37.30] for the details. In [37.51], VRS was

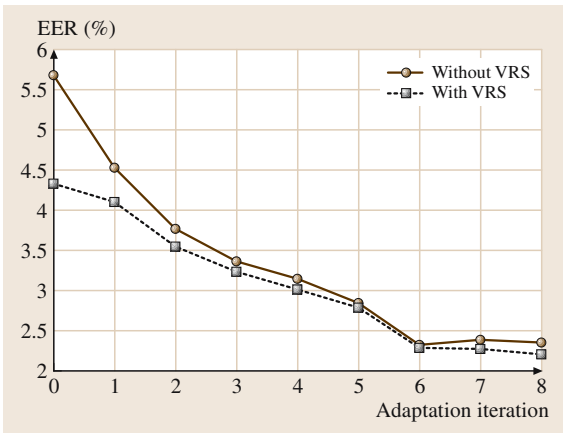


Fig. 37.5 The effect of unsupervised adaptation on the EER (percentage) with and without variable rate smoothing. Adaptation iteration 0 is the enrollment session. Based on the lower number of trials (genuine in our case), the 90% confidence interval on the measures is 0.3%. (After [37.51])

applied to text-dependent speaker recognition; Fig. 37.5 was adapted from that publication. It can be seen that, after the enrollment (iteration 0 on the graph), VRS is most effective because so little data has been used to train the speaker model: smoothing of the variances and mixture weights is not as aggressive as for means because the system does not have adequate estimates. As adaptation occurs, the two curves (with and without VRS) converge: at this point the estimates for the first and second moment in the distributions are accurate, the number of samples is high, and the presence of different smoothing coefficients becomes irrelevant.

Random Digit Strings

We now illustrate the effect of speaker model adaptation on contextual lexical mismatch for a digit-based speaker verification task. The experimental set-up is from a different organization of the data from Sect. 37.3.2 to follow the aforementioned adaptation protocol. Figure 37.6 illustrates the results. The testing is performed on a pseudorandom digit string (see Table 37.3 for details). Enrollment is either performed on a fixed digit string (eEEE) or on a series on pseudorandom digit strings (epRpRpR). Before adaptation occurs, the accuracy of epRpRpR is better than eEEE because the enrollment lexical conditions are matched to testing. However, as adaptation occurs and more pseudorandom utterances are added to the eEEE speaker model, the two curves converge. This shows the power of adaptation to reduce lexical mismatch and to alter the *enrollment* lexicon: in this context, the concept of *enrollment* lexicon becomes fuzzy as adaptation broadens the lexicon that was used to train the speaker model.

Speaker Model Synthesis and Cross-Channel Attempts

Speaker model synthesis (SMS) [37.13] is an extension of handset-dependent background modeling [37.8]. As mentioned before, SMS and feature mapping are dual to each another. Figure 37.3c presents the offline component of SMS. It is very similar to the offline component of feature mapping except that the transforms for means, variances, and mixture weights are derived to transform sufficient statistics from one channel GMM to another rather than from a channel GMM to the root GMM. During online operation, in enrollment, a set of utterances are tagged as a whole to a specific channel (the likeliest channel GMM – the enrollment channel). Then speaker model training (Sect. 37.1.4) uses adaptation with variable rate smoothing [37.30,51] of the enrollment channel GMM. The transforms that have been derived offline are then used at test time to synthesize the enrolled channel GMM across *all* supported channels (Fig. 37.3d). The test utterance is tagged using the same process as enrollment by picking the likeliest channel GMM (the testing channel). The speaker model GMM for the testing channel and the testing channel GMM are then used in the likelihood ratio scoring scheme described in Sect. 37.1.3 and (37.1).

The power of speaker model adaptation (Sect. 37.1.6) when combined with SMS is its ability to synthesize

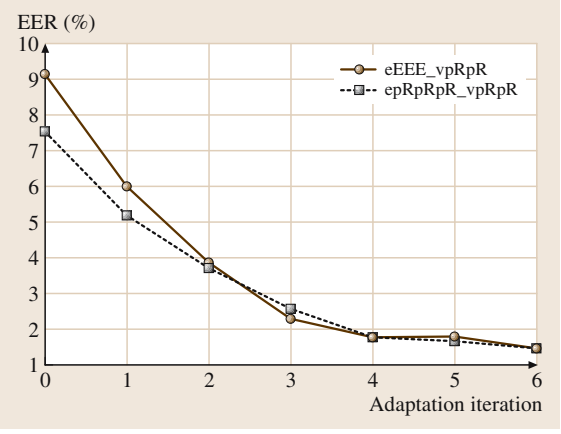


Fig. 37.6 The effect of unsupervised adaptation on reducing the contextual lexical mismatch as depicted by a reduction of the EER. Adaptation iteration 0 is the enrollment session. Based on the lower number of trials (genuine in our case), the 90% confidence interval on the measures is 0.3%

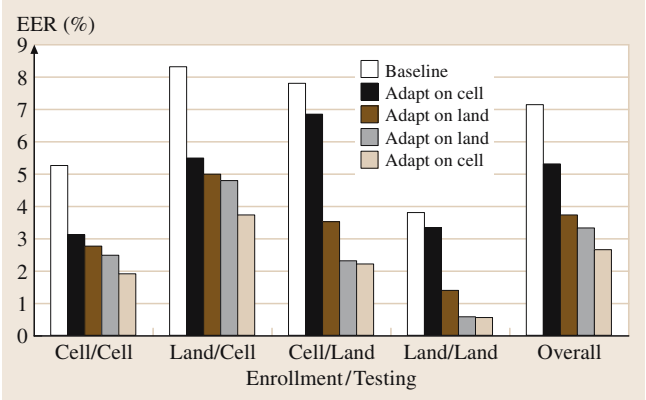


Fig. 37.7 The effect of speaker model adaptation and SMS on the cross-channel accuracy (EER). The interested reader should refer to this paper for additional details. The baseline is the enrollment session. Based on the lower number of trials (genuine in our case), the 90% confidence interval on the measures is 0.6%. (After [37.40])

sufficient statistics across all supported channels. For example, assume that a speaker is enrolled on channel X and a test utterance is tagged as belonging to channel Y. Then, if the test utterance is to be used for adaptation of the speaker model, the sufficient statistics from that utterance is gathered. The transform from Y → X is used to synthesize sufficient statistics from channel Y to channel X before adaptation of the speaker model (on channel X) occurs. Concretely, this would allow adaptation utterances from channel Y to improve the accuracy on all other channels.

Figure 37.7 illustrates the effect of speaker model adaptation with SMS. Results are grouped in *enrollment/testing* conditions: within a group, the enrollment and testing channels are fixed, the only variable is the adaptation material. For each group, the first bar is the accuracy after enrollment. The second bar is the accuracy after one iteration of adaptation on cellular data. The third bar shows the accuracy after the iteration of adaptation on cellular data followed by an iteration on a landline data, and so on. Note that these results are for supervised adaptation and thus an iteration of adaptation on a given speaker model necessarily means an actual adaptation of the speaker model. There are two interesting facts about this figure. The first important feature is that the biggest relative gain in accuracy is when the channel for the adaptation data is matched with the previously unseen testing utterance channel (see the relative improvements between the first and second bars in the *cell/cell* and *land/cell* or between the second and third bars in the *cell/land* and *land/land* results). This is expected since the new data is matched to the (previously unseen) channel of the testing utterance. The other important feature illustrates that the SMS (resynthesis of sufficient statistics) has the ability to improve accuracy even when adaptation has been performed on a different channel than the testing utterance. As an example, in the first block of Fig. 37.7, there is an improvement in accuracy between the second and third bars. The difference between the second and third bars is an extra adaptation iteration on *land* (landline data), but note that the testing is performed on *cell*. This proves that the sufficient statistics accumulated on the *land* channel have been properly resynthesized into the *cell* channel.

Setting and Tracking the Operating Point

Commercial deployments are very much concerned with the overall accuracy of a system but also the operating point, which is usually a specific false-acceptance rate. As mentioned earlier, setting the operating point for a very secure large-scale deployed system is a costly exercise, but for internal trials and low-security solutions, an approximate of the ideal operating point is acceptable. In [37.36] and later in [37.52], a simple algorithm to achieve this has been presented: frame-count-dependent thresholding (FCDT). The idea is simple: parameterize the threshold to achieve a target FA rate as a function of

1. the length of the password phrase
2. the maturity of the speaker model (how well it is trained)

At test time, depending on the desired FA rate, an offset is applied to the score (37.1). Note that the applied offset is speaker dependent because it depends on the length of the password phrase and the maturity of the speaker model.

This parameterization has been done on a large Japanese corpora. The evaluation was conducted on 12 test sets from different languages composed of data collection, trial data and in-service data [37.36]. The operating point for the system was set up at a target FA rate of 0.525% using the above algorithm. The average of the actual FA rates measured was 0.855% with a variance of 0.671%; this new algorithm outperformed previous algorithms [37.33].

In the context of adaptation of the speaker model, the problem of setting an operating point is transformed into a problem of maintaining a constant operating point for all speakers at all times [37.37]. Note that a similar problem arises in the estimation of confidence in speech recognition when adaptation of the acoustic

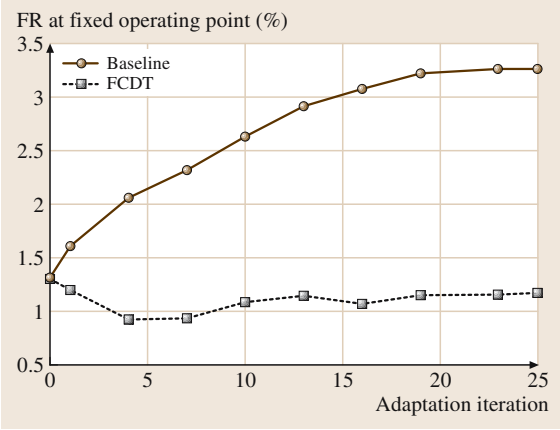


Fig. 37.8 The effect of speaker model adaptation on the FA rate with and without frame-count-dependent thresholding (FCDT). Adaptation iteration 0 is the enrollment session. The 90% confidence interval on the measures is 0.3%. (After [37.36])

Amazon EX1010 Page 766

models is performed [37.53]. FCDT, as well as other algorithms [37.33], can perform this task. Figure 37.8 presents the false-acceptance rate at a fixed operating point as a function of unsupervised adaptation iterations for an English digits task. After enrollment, both systems are calibrated to operate at FA = 1.3%. Then adaptation is performed. We can very easily see that the scores of the imposter attempts drift towards higher values, and hence the FA rate does not stay constant: the FA rate has doubled after 10 iterations. For commercial deployments, this problem is a crucial one: adaptation of the speaker models is very effective to increase the overall accuracy, but it must not be at the expense of the stability of the operating point. FCDT accomplishes this task: the FA rate stays roughly constant across the adaptation cycles. This leads us to think that FCDT is an effective algorithm to normalize scores against the levels of maturity of the speaker models.

Note that the imposter score drift towards higher values during speaker model adaptation in text-dependent tasks is the opposite behavior from the case of text-independent tasks [37.38, Fig. 3]. This supports the assertion that the existence of a restricted lexicon for text-dependent models has a significant impact on the behavior of speaker recognition systems: both text-dependent [37.36] and text-independent [37.38] systems being GMM-based. During the enrollment and adaptation sessions, several characteristics of the speech signal are captured in the speaker model: the speaker's intrinsic voice characteristics, the acoustic conditions (channels and noise), and the lexicon. In text-dependent speaker recognition, because of the restricted lexicon, the speaker model becomes a lexicon recognizer (the *mini-recognizer* effect). This effect increases the imposter scores because they use the target lexicon.

The FCDT algorithm can be implemented at the phone level in order to account for cases where the enrollment session (and/or speaker model adaptation) does not have a consistent lexicon. In [37.36], all experiments were carried out with enrollment and testing sessions that used exactly the same lexicon for a given user; this might seem restrictive. In the case of phone-level FCDT, the FCDT algorithm would be normalizing maturities of phone-level speaker models.

In the literature on T-norm (for text-dependent or text-independent systems; see Sect. 37.3.4), the speaker models composing the cohorts were all trained with roughly the same amount of speech. In light of the aforementioned results, this choice has the virtue of normalizing against different maturities of speaker models. We believe that the FCDT algorithm can also be used in the context of the T-norm to achieve this normalization.

37.3.6 Protection Against Recordings

As mentioned, protection against recordings is important for text-dependent speaker recognition systems. If the system is *purely* text dependent (that is the enrollment and testing utterances have the same lexicon sequence), once a fraudster has gained access to a recording, it can become relatively easy to break into an account [37.42]. This, however, must be put in perspective. A high-quality recording of the target speaker's voice is required as well as digital equipment to perform the playback. Furthermore, for any type of biometric, once a *recording* and playback mechanism are available the system becomes vulnerable. The advantage that voice authentication has over any other biometrics is that it is natural to prompt for a different sequence of the enrollment sequence: this is impossible for iris scans, fingerprints, etc. Finally, any nonbiometric security layer can be broken into almost 100% of the time once a *recording* of the secure token is available (for example, somebody who steals a badge can easily access restricted areas).

Several studies that assess the vulnerability of speaker recognition systems to altered imposter voices have been published. The general paradigm is that a fraudster gains access to recordings of a target user. Then using different technique the imposter's voice is altered to sound like the target speaker for any password phrase. An extreme case is a well-trained text-to-speech (TTS) system. This scenario is unrealistic because the amount of training material required for a good-quality TTS voice is on the order of hours of high-quality, phonetically balanced recorded speech. Studies along these lines, but using a smaller amount of data, can be found in [37.54, 55]. Even if these studies report the relative weakness of GMM-based speaker recognition systems, these techniques require sophisticated signal processing software and expertise to perform experimentation, along with high-quality recordings. A more-recent study [37.56] has also demonstrated the effect of speech transformation on imposter acceptance. This technique, again, requires technical expertise and complete knowledge of the speaker recognition system (feature extraction, modeling method, UBM, target speaker model, and algorithms). This is clearly beyond the grasp of fraudsters because implementations of security systems are usually kept secret, as are the internals algorithms of commercial speaker recognition systems.

Amazon EX1010 Page 767

Speaker Recognition Across Entire Calls

Protection against recordings can be improved by performing speaker recognition (in this case verification) across entire calls. The results presented here illustrate a technique to implement accurate speaker recognition across entire calls with a short enrollment session (joint unpublished work with Nikki Mirghafori). It relies heavily on speaker model adaptation (Sect. 37.1.6) and PCBV (Sect. 37.1.4). The verification layer is designed around a password phrase such as an account number. The enrollment session is made up of three repetitions of the password phrase only, while the testing sessions are composed of one repetition of the password phrase followed by non-password phrases. This is to simulate a dialog with a speech application after initial authentication has been performed. Adaptation is used to learn new lexical items that were not seen during enrollment and thus improve the accuracy when non-password phrases are used. The choice for this set-up is motivated by several factors. This represents a possible *upgrade* for currently deployed password-based verification application. It is also seamless to the end user and does not require re-enrollment: the non-password phrases are learnt using speaker model adaptation during the verification calls. Finally it is believed that this technique represents a very compelling solution for protection against recordings.

Note that this type of experiment is at the boundary between text-dependent and text-independent speaker recognition because the testing session is cross-lexicon for certain components. It is hard to categorize this type of experimental set-up because the enrollment session is very short and lexically constrained compared to its text-independent counterpart. Also, the fact that some testing is made cross-lexicon means that it does not clearly belong to the text-dependent speaker recognition field.

In order to benchmark this scenario, Japanese and Canadian French test sets were set up with eight-digit strings (account number) as the *password phrase*. The initial enrollment used three repetitions of the password phrase. We benchmark accuracy on the password and on non-password phrases. In these experiments, the non-password phrases were composed of general text such as first/last names, dates, and addresses. For adaptation, we used the same protocol as in Sect. 37.3.5 with a held-out set composed of non-password phrases (supervised adaptation). Section 37.3.5 has already demonstrated the effectiveness of adaptation on password phrases; these results show the impact, on both password and non-password phrases, of adapting on non-password phrases. Figure 37.9 presents the EER as a function of adaptation iteration, when adapting on non-password phrases for a single GMM or PCBV solution and testing on password phrases. It can be seen that the GMM solution is very sensitive to adaptation on non-password phrases,

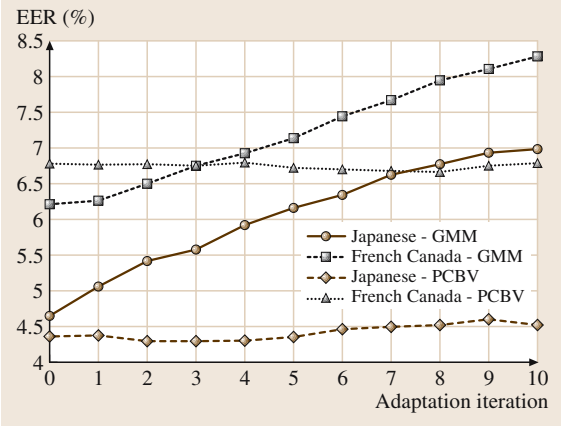


Fig. 37.9 The effect of speaker model adaptation with non-password phrases on the accuracy of password phrases (EER). Adaptation iteration 0 is the enrollment session. The experiments were carried out on over 24 k attempts from genuine speaker and imposters. Based on the lower number of trials (genuine in our case), the 90% confidence interval on the measures is 0.3%

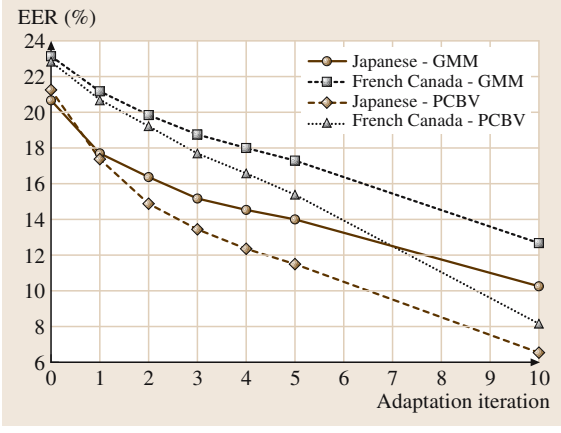


Fig. 37.10 The effect of speaker model adaptation with non-password phrases on the accuracy of non-password phrases (EER). Adaptation iteration 0 is the enrollment session. The experiments were carried out on over 14 k attempts from genuine speaker and imposters. Based on the lower number of trials (genuine in our case), the 90% confidence interval on the measures is 0.5%

Amazon EX1010 Page 768

Table 37.6 The measured FA rate using an automatic impostor trial generation algorithm for different conditions and data sets. Note that the target FA rate was 1.0%

Experimental setup	English_US_1	English_US_2	English_UK_1	Average
[37.45] algorithm without offset	2.10%	1.15%	2.39%	1.88%
[37.45] algorithm with 0.15 offset	1.13%	0.81%	1.35%	1.10%
New binning without offset	0.86%	0.75%	1.25%	0.95%

whereas the PCBV is not. This is due to the fact that PCBV uses alignments from a speech recognition engine to segregate frames into different modeling units while the GMM does not: this leads to smearing of the speaker model in the case of the GMM solution. Figure 37.10 shows the improvements in the accuracy on non-password phrases in the same context. Note that iterations 1–5 do not have overlapping phrases with the testing lexicon: iteration 10 has some overlap, which is not unrealistic from a speech application point of view. As expected, the accuracy of the non-password phrase is improved by the adaptation process for both GMM and PCBV, with a much greater improvement for PCBV. After 10 adaptation iterations, the accuracy is 6–8% EER (and has not yet reached a plateau), which makes this solution a viable solution. It can also be noted that PCBV with adaptation on non-password phrases improves the accuracy faster than its single-GMM counterpart, taking half the adaptation iterations to achieve a similar EER (Fig. 37.10). In summary, speaker model adaptation and PCBV form the basis for delivering stable accuracy on password phrases while dramatically improving the accuracy for non-password phrases. This set of results is another illustration of the power of speaker model adaptation and represents one possible implementation for protection against recordings. Any improvement in this area is important for the text-dependent speaker recognition field as well as commercial applications.

37.3.7 Automatic Impostor Trials Generation

As mentioned above, application developers usually want to know how secure their speech application is. Usually, the design of the security layer is based on the choice of the password phrase, the choice of the enrollment and verification dialogs, and the security level (essentially the FA rate). From these decisions follow the FR and RR rates. Using off-the-shelf threshold settings will usually only give a range of target FA rates, but will rarely give any hint on the FR and RR for the current designed dialog [37.36]. Often application developers want a realistic picture of the accuracy of their system (FA, FR, and RR) based on their data. Since the FA rate is important, this has to be measured with a high degree of confidence. To do this, one requires a tremendous amount of data. As an example, to measure an FA of $1\% \pm 0.3\%$ nine times out of ten, 3000 imposter trials are required [37.1]. For a higher degree of precision such as $\pm 0.1\%$, more than 30 000 imposter trials are needed. Collecting data for imposter trials results in a lot of issues; it is costly, requires data management and tagging, cannot really be done on production systems if adaptation of speaker models is enabled, etc. However, collecting genuine speaker attempts can be done simply by archiving utterances and the associated claimed identity; depending on the traffic, a lot of data can be gathered quickly. Note that some manual tagging may be required to flag true imposter attempts – usually low-scoring genuine speaker attempts. The data gathered is also valuable because it can come from the production system.

For password phrases that are common to all users of a system, generating imposter attempts is easy once the data has been collected and tagged: it can be done using a round-robin. However, if the password phrase is unique for each genuine speaker, a round-robin cannot be used. In this case, the lexical content of the imposter attempts will be mismatched to the target speaker models, the resulting attempt will be grossly unchallenging and will lead to underestimation of the actual FA rate. In [37.45], an algorithm to estimate the FA rate accurately using only genuine speaker attempts was presented. The essence of the idea is to use a round-robin for imposter trial generation, but to quantify the amount of lexical mismatch between the attempt and target speaker model. Each imposter attempt will have a lexical mismatch value associated with it. This can be thought of as a lexical distance (mismatch) between two strings. Intuitively, we want the following order for the lexical mismatch value with respect to the target string $1234 : 1234 < 1256 < 1526 < 5678$. Note that [37.45] and the following are based on digit strings, but can easily be applied to general text by using phonemes as the atom instead of digits. A variant of the Levenstein distance was used to bin imposter attempts. For each bin,

the threshold to achieve the target FA rate was calculated. A regression between the Levenstein distance and threshold for the target FA is used to extrapolate the operational threshold for the target FA rate. For the development of this algorithm, three test sets from data collections and trials were used. These had a set of *real* impostor attempts that we used to assess the accuracy of the algorithm. The first line of Table 37.6 shows the *real* FA rate measured at the operational threshold as calculated by the algorithm above. In [37.45], to achieve good accuracy, an offset of 0.15 needed to be introduced (the second line in the table). The algorithm had one free parameter. It was later noticed that, within a bin with a given Levenstein distance, some attempts were more competitive than others. For example, the target/attempt pairs 97 526/97 156 and 97 526/97 756 had the same Levenstein distance. However, the second pair is more competitive because all of the digits in the attempt are present in the target and hence have been seen during the enrollment. A revised binning was performed and is presented as the last line in Table 37.6. The average measured FA rate is much closer to the target FA rate and this revised algorithm does not require any free parameters.

Once the threshold for the desired FA rate has been calculated, it is simple to extract the FR and RR rates from the same data. Reducing the cost of deployment is critical for making speaker recognition a mainstream biometric technique. Any advances in this direction is thus important.

37.4 Concluding Remarks

This chapter on text-dependent speaker recognition has been designed to illustrate the current technical challenges of the field. The main challenges are robustness to channel and lexical mismatches. Several results were presented to illustrate these two key challenges under a number of conditions. Adaptation of the speaker models yields advantages to address these challenges but this needs to be properly engineered to be deployable on a large scale while maintaining a stable operating point. Several new research avenues were reviewed.

When relevant, parallels between the text-dependent and text-independent speaker recognition fields were drawn. The distinctions between the two fields becomes thin when considering the work by *Sturim* et al. [37.2] and text-dependent speaker recognition with heavy lexical mismatch, as described in Sect. 37.3.6. This research area should provide a very fertile ground for future advances in the speaker recognition field.

Finally, special care was taken to illustrate, using relevant (live or trial) data, the specific challenges facing text-dependent speaker recognition in actual deployment situations.

References

37.1 A. Martin, M. Przybocki, G. Doddington, D.A. Reynolds: The NIST speaker recognition evaluation – Overview, methodology, systems, results, perspectives, Speech Commun. **31**, 225–254 (2000)

37.2 D.E. Sturim, D.A. Reynolds, R.B. Dunnk, T.F. Quatieri: Speaker verification using text-constrained gaussian mixture models, Proc. IEEE ICASSP **2002**(1), 677–680 (2002)

37.3 K. Boakye, B. Peskin: Text-constrained speaker recognition on a text-independent task, Proc. Odyssey Speaker Recognition Workshop, Vol. 2004 (2004)

37.4 D. Boies, M. Hébert, L.P. Heck: Study of the effect of lexical mismatch in text-dependent speaker verification, Proc. Odyssey Speaker Recognition Workshop, Vol. 2004 (2004)

37.5 M. Wagner, C. Summerfield, T. Dunstone, R. Summerfield, J. Moss: An evaluation of commercial off-the-shelf speaker verification systems, Proc. Odyssey Speaker Recognition Workshop, Vol. 2006 (2006)

37.6 A. Higgins, L. Bahler, J. Porter: Speaker verification using randomized phrase prompting, Digit. Signal Process. **1**, 89–106 (1991)

37.7 M.J. Carey, E.S. Parris, J.S. Briddle: A speaker verification system using alpha-nets, Proc. IEEE ICASSP, Vol. 1981 (1981) pp. 397–400

37.8 L.P. Heck, M. Weintraub: Handset dependent background models for robust text-independent

Amazon EX1010 Page 770

speaker recognition, Proc. IEEE ICASSP **1997**(2), 1037–1040 (1997)
37.9 A.E. Rosenberg, S. Parthasarathy: The use of cohort normalized scores for speaker recognition, Proc. IEEE ICASSP **1996**(1), 81–84 (1996)
37.10 C. Barras, J.-L. Gauvain: Feature and score normalization for speaker verification of cellular data, Proc. IEEE ICASSP **2003**(2), 49–52 (2003)
37.11 Y. Liu, M. Russell, M. Carey: The role of dynamic features in text-dependent and -independent speaker verification, Proc. IEEE ICASSP **2006**(1), 669–672 (2006)
37.12 D. Reynolds: Channel robust speaker verification via feature mapping, Proc. IEEE ICASSP **2003**(2), 53–56 (2003)
37.13 R. Teunen, B. Shahshahani, L.P. Heck: A model-based transformational approach to robust speaker recognition, Proc. ICSLP **2000**(2), 495–498 (2000)
37.14 R.O. Duda, P.E. Hart, D.G. Stork: *Pattern Classification*, 2nd edn. (Wiley, New York 2001)
37.15 T. Kato, T. Shimizu: Improved speaker verification over the cellular phone network using phoneme-balanced and digit-sequence-preserving connected digit patterns, Proc. IEEE ICASSP **2003**(2), 57–60 (2003)
37.16 T. Matsui, S. Furui: Concatenated phoneme models for text-variable speaker recognition, Proc. IEEE ICASSP **1993**(2), 391–394 (1993)
37.17 S. Parthasarathy, A.E. Rosenberg: General phrase speaker verification using sub-word background models and likelihood ratio scoring, Proc. ICSLP **1996**(4), 2403–2406 (1996)
37.18 C.W. Che, Q. Lin, D.S. Yuk: An HMM approach to text-prompted speaker verification, Proc. IEEE ICASSP **1996**(2), 673–676 (1996)
37.19 M. Hébert, L.P. Heck: Phonetic class-based speaker verification, Proc. Eurospeech, Vol. 2003 (2003) pp. 1665–1668
37.20 E.G. Hansen, R.E. Slygh, T.R. Anderson: Speaker recognition using phoneme-specific GMMs, Proc. Odyssey Speaker Recognition Workshop, Vol. 2004 (2004)
37.21 M. Schmidt, H. Gish: Speaker identification via support vector classifiers, Proc. IEEE ICASSP **1996**(1), 105–108 (1996)
37.22 W.M. Campbell, D.E. Sturim, D.A. Reynolds, A. Solomonoff: SVM based speaker verification using a GMM supervector kernel and NAP variability compensation, Proc. IEEE ICASSP **2006**(1), 97–100 (2006)
37.23 N. Krause, R. Gazit: SVM-based speaker classification in the GMM model space, Proc. Odyssey Speaker Recognition Workshop, Vol. 2006 (2006)
37.24 S. Fine, J. Navratil, R.A. Gopinath: A hybrid GMM/SVM approach to speaker identification, Proc. IEEE ICASSP **2001**(1), 417–420 (2001)
37.25 W.M. Campbell: A SVM/HMM system for speaker recognition, Proc. IEEE ICASSP **2003**(2), 209–212 (2003)
37.26 S. Furui: Cepstral analysis techniques for automatic speaker verification, IEEE Trans. Acoust. Speech **29**, 254–272 (1981)
37.27 V. Ramasubramanian, A. Das, V.P. Kumar: Text-dependent speaker recognition using one-pass dynamic programming algorithm, Proc. IEEE ICASSP **2006**(2), 901–904 (2006)
37.28 A. Sankar, R.J. Mammone: Growing and pruning neural tree networks, IEEE Trans. Comput. **42**, 272–299 (1993)
37.29 K.R. Farrell: Speaker verification with data fusion and model adaptation, Proc. ICSLP **2002**(2), 585–588 (2002)
37.30 D.A. Reynolds, T.F. Quatieri, R. B.Dunn: Speaker verification using adapted gaussian mixture models, Digit. Signal Process. **10**, 19–41 (2000)
37.31 J.-L. Gauvain, C.-H. Lee: Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains, IEEE T. Speech Audi. Process. **2**, 291–298 (1994)
37.32 D.A. Reynolds: Comparison of background normalization methods for text-independent speaker verification, Proc. EuroSpeech **1997**(2), 963–966 (1997)
37.33 N. Mirghafori, L.P. Heck: An adaptive speaker verification system with speaker dependent a priori decision thresholds, Proc. ICSLP **2002**(2), 589–592 (2002)
37.34 R. Auckenthaler, M.J. Carey, H. Lloyd-Thomas: Score normalization for text-independent speaker verification systems, Digit. Signal Process. **10**, 42–54 (2000)
37.35 M. Hébert, D. Boies: T-Norm for text-dependent commercial speaker verification applications: effect of lexical mismatch, Proc. IEEE ICASSP **2005**(1), 729–732 (2005)
37.36 N. Mirghafori, M. Hébert: Parametrization of the score threshold for a text-dependent adaptive speaker verification system, Proc. IEEE ICASSP **2004**(1), 361–364 (2004)
37.37 T. Matsui, T. Nishitani, S. Furui: Robust methods for updating model and a priori threshold in speaker verificaiton, Proc. IEEE ICASSP, Vol. 1996 (1996) pp. 97–100
37.38 C. Barras, S. Meignier, J.-L. Gauvain: Unsupervised online adaptation for speaker verification over the telephone, Proc. Odyssey Speaker Recognition Workshop, Vol. 2004 (2004)
37.39 C. Fredouille, J. Mariéthoz, C. Jaboulet, J. Hennebert, J.-F. Bonastre, C. Mokbel, F. Bimbot:

Amazon EX1010 Page 771

Behavior of a bayesian adaptation method for incremental enrollment in speaker verification, Proc. IEEE ICASSP, Vol. 2000 (2000)
37.40 L.P. Heck, N. Mirghafori: Online unsupervised adaptation in speaker verification, Proc. ICSLP, Vol. 2000 (2000)
37.41 L.P. Heck: On the deployment of speaker recognition for commercial applications, Proc. Odyssey Speaker Recognition Workshop, Vol. 2004 (2004), keynote speech
37.42 K. Wadhwa: Voice verification: technology overview anf accuracy testing results, Proc. Biometrics Conference, Vol. 2004 (2004)
37.43 M.J. Carey, R. Auckenthaler: User validation for mobile telephones, Proc. IEEE ICASSP, Vol. 2000 (2000)
37.44 L.P. Heck, D. Genoud: Integrating speaker and speech recognizers: automatic identity claim capture for speaker verification, Proc. Odyssey Speaker Recognition Workshop, Vol. 2001 (2001)
37.45 M. Hébert, N. Mirghafori: Desperately seeking impostors: data-mining for competitive impostor testing in a text-dependent speaker verification system, Proc. IEEE ICASSP **2004**(2), 365–368 (2004)
37.46 T.F. Quatieri, E. Singer, R.B. Dunn, D.A. Reynolds, J.P. Campbell: Speaker and language recognition using speech codec partameters, Proc. EuroSpeech, Vol. 1999 (1999) pp. 787–790
37.47 L.P. Heck, Y. Konig, M.K. Sönmez, M. Weintraub: Robustness to telephone handset distortion in speaker recognition by discriminative feature design, Speech Commun. **31**, 181–192 (2000)
37.48 M. Siafarikas, T. Ganchev, N. Fakotakis, G. Kokkinakis: Overlapping wavelet packet features for speaker verification, Proc. EuroSpeech, Vol. 2005 (2005)
37.49 D. Reynolds: Speaker identification and verification using Gaussian mixture speaker models, Speech Commun. **17**, 91–108 (1995)
37.50 O. Siohan, C.-H. Lee, A.C. Surendran, Q. Li: Background model design for flexible and portable speaker verification systems, Proc. IEEE ICASSP **1999**(2), 825–829 (1999)
37.51 L.P. Heck, N. Mirghafori: Unsupervised on-line adaptation in speaker verification: confidence-based updates and improved parameter estimation, Proc. Adaptation in Speech Recognition, Vol. 2001 (2001)
37.52 D. Hernando, J.R. Saeta, J. Hernando: Threshold estimation with continuously trained models in speaker verification, Proc. Odyssey Speaker Recognition Workshop, Vol. 2006 (2006)
37.53 A. Sankar, A. Kannan: Automatic confidence score mapping for adapted speech recognition systems, Proc. IEEE ICASSP **2002**(1), 213–216 (2002)
37.54 D. Genoud, G. Chollet: Deliberate imposture: a challenge for automatic speaker verification systems, Proc. EuroSpeech, Vol. 1999 (1999) pp. 1971–1974
37.55 B.L. Pellom, J.H.L. Hansen: An experimental study of speaker verification sensitivity to computer voice-altered imposters, Proc. IEEE ICASSP **1999**(2), 837–840 (1999)
37.56 D. Matrouf, J.-F. Bonastre, C. Fredouille: Effect of speech transformation on impostor acceptance, Proc. IEEE ICASSP **2006**(2), 933–936 (2006)

Amazon EX1010 Page 772

38. Text-Independent Speaker Recognition

D. A. Reynolds, W. M. Campbell

In this chapter, we focus on the area of text-independent speaker verification, with an emphasis on unconstrained telephone conversational speech. We begin by providing a general likelihood ratio detection task framework to describe the various components in modern text-independent speaker verification systems. We next describe the general hierarchy of speaker information conveyed in the speech signal and the issues involved in reliably exploiting these levels of information for practical speaker verification systems. We then describe specific implementations of state-of-the-art text-independent speaker verification systems utilizing low-level spectral information and high-level token sequence information with generative and discriminative modeling techniques. Finally, we provide a performance assessment of these systems using the National Institute of Standards and Technology (NIST) speaker recognition evaluation telephone corpora.

38.1 **Introduction** 763

38.2 **Likelihood Ratio Detector** 764

38.3 **Features** 766
38.3.1 Spectral Features 766
38.3.2 High-Level Features 766

38.4 **Classifiers** 767
38.4.1 Adapted Gaussian Mixture Models 767
38.4.2 Support Vector Machines 771
38.4.3 High-Level Feature Classifiers 774
38.4.4 System Fusion 775

38.5 **Performance Assessment** 776
38.5.1 Task and Corpus 776
38.5.2 Systems 777
38.5.3 Results 777
38.5.4 Computational Considerations 778

38.6 **Summary** 778

References 779

38.1 Introduction

With the merging of telephony and computer networks, the growing use of speech as a modality in man–machine communication, and the need to manage ever increasing amounts of recorded speech in audio archives and multimedia applications, the utility of recognizing a person from his or her voice is increasing. While the area of speech recognition is concerned with extracting the linguistic message underlying a spoken utterance, speaker recognition is concerned with extracting the identity of the person speaking the utterance. Applications of speaker recognition are wide ranging, including: facility or computer access control [38.1,2], telephone voice authentication for long-distance calling or banking access [38.3], intelligent answering machines with personalized caller greetings [38.4], and automatic speaker labeling of recorded meetings for speaker-dependent audio indexing (speech skimming) [38.5, 6].

Depending upon the application, the general area of speaker recognition is divided into two specific tasks: identification and verification. In speaker identification, the goal is to determine which one of a group of known voices best matches the input voice sample. This is also referred to as *closed-set* speaker identification. Applications of pure closed-set identification are limited to cases where only enrolled speakers will be encountered, but it is a useful means of examining the separability of speakers' voices or finding similar sounding speakers, which has applications in speaker-adaptive speech recognition. In verification, the goal is to determine from a voice sample if a person is who he or she claims to be. This is sometimes referred to as the *open-set* problem, because this task requires distinguishing a claimed speaker's voice known to the system from a potentially large group of voices unknown to the system (i. e., impostor speakers). Verification is the basis

for most speaker recognition applications and the most commercially viable task. The merger of the closed-set identification and open-set verification tasks, called open-set identification, performs like closed-set identification for known speakers but must also be able to classify speakers unknown to the system into a *none of the above* category.

These tasks are further distinguished by the constraints placed on the speech used to train and test the system and the environment in which the speech is collected [38.7]. In a text-dependent system, the speech used to train and test the system is constrained to be the same word or phrase. In a text-independent system, the training and testing speech are completely unconstrained. Between text dependence and text independence, a vocabulary-dependent system constrains the speech to come from a limited vocabulary, such as the digits, from which test words or phrases (e.g., digit strings) are selected. Furthermore, depending upon the amount of control allowed by the application, the speech may be collected from a noise-free environment using a wide-band microphone or from a noisy, narrow-band telephone channel.

In this chapter, we focus on the area of text-independent speaker verification, with an emphasis on unconstrained telephone conversational speech. While many of the underlying algorithms employed in text-independent and text-dependent speaker verification are similar, text-independent applications have the additional challenge of operating unobtrusively to the user with little to no control over the user's behavior (i. e., the user is speaking for some other purpose, not to be verified, so will not cooperate to speak more clearly, use a limited vocabulary or repeat phrases). Further, the ability to apply text-independent verification to unconstrained speech encourages the use of audio recorded from a wide variety of sources (e.g., speaker indexing of broadcast audio or forensic matching of law-enforcement microphone recordings), emphasizing the need for compensation techniques to handle variable acoustic environments and recording channels.

This chapter is organized as follows. We begin by providing a general likelihood ratio detection task framework to describe the various components in modern text-independent speaker verification systems. We next describe the general hierarchy of speaker information conveyed in the speech signal and the issues involved in reliably exploiting these levels of information for practical speaker verification systems. We then describe specific implementations of state-of-the-art text-independent speaker verification systems utilizing low-level spectral information and high-level token sequence information with generative and discriminative modeling techniques. Finally we provide a performance assessment of these systems using the NIST speaker recognition evaluation telephone corpora.

38.2 Likelihood Ratio Detector

Given a segment of speech, Y, and a hypothesized speaker, S, the task of speaker detection, also referred to as verification, is to determine if Y was spoken by S. An implicit assumption often used is that Y contains speech from only one speaker. Thus, the task is better termed single-speaker detection. If there is no prior information that Y contains speech from a single speaker, the task becomes multispeaker detection. In this paper we will focus on the core single-speaker detection task. Discussion of systems that handle the multispeaker detection task can be found in [38.8].

The single-speaker detection task can be restated as a basic hypothesis test between

H_0 : Y is from the hypothesized speaker S

H_1 : Y is *not* from the hypothesized speaker S .

From statistical detection theory, the optimum test to decide between these two hypotheses is a likelihood ratio test given by

$$\frac{p(Y \mid H_0)}{p(Y \mid H_1)} \begin{cases} \geq \theta & \text{accept } H_0 \\ < \theta & \text{reject } H_0 \end{cases}, \tag{38.1}$$

where $p(Y \mid H_i), i = 0, 1$, is the probability density function for the hypothesis H_i evaluated for the observed speech segment Y, also referred to as the *likelihood* of the hypothesis H_i given the speech segment ($p(A \mid B)$ is referred to as a likelihood when B is considered the independent variable in the function). Strictly speaking, the likelihood ratio test is only optimal when the likelihood functions are known exactly, which is rarely the case. The decision threshold for accepting or rejecting H_0 is θ. Thus, the basic goal of a speaker detection system is to determine techniques to compute the likelihood ratio between the two likelihoods, $p(Y \mid H_0)$ and $p(Y \mid H_1)$. Depending upon the techniques used, these likelihoods

are explicitly or implicitly modeled and applied in the speaker detection systems.

Figure 38.1 shows the basic components found in speaker detection systems. The role of the front-end processing is to extract features from the speech signal that convey speaker-dependent information. In addition, techniques to minimize non-speaker-dependent effects from these features, such as linear filtering or additive noise, are also employed in the front-end processing. The output of this stage is a sequence of feature vectors representing the test segment, $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$, where $\boldsymbol{x}_t$ is a feature vector indexed at discrete time $t \in [1, 2, \ldots, T]$. Features can be continuous or discrete with multiple dimensions, may be extracted at asynchronous time instances and can cover variable temporal durations. For example, the sequence of words spoken as estimated by a speech recognizer (series of one-dimensional discrete, asynchronous, variable temporal duration values), or the overall average pitch and energy in an utterance (single two-dimensional continuous-value vector) could be used as features. Thus, features can be extracted that capture short- and long-span speaker characteristics. Common features used in text-independent speaker detection systems are described in Sect. 38.3.

These feature vectors are then used to compute a score to evaluate the likelihood ratio test between H_0 and H_1. The exact way in which the hypothesis likelihood functions are represented and scored against the feature vectors depends on the classifier employed. The parameters of the classifier are obtained from the speaker enrollment or training phase, which is discussed in Sect. 38.4.

In generative modeling approaches, the likelihoods are explicitly represented by probability density models. Mathematically, H_0 is represented by a model denoted λ_{hyp}, which characterizes the hypothesized speaker S in the feature space of $\boldsymbol{x}$. For example, one could assume that the feature vectors for H_0 were generated from a Gaussian distribution so that λ_{hyp} would denote the mean vector and covariance matrix parameters of a Gaussian distribution. The alternative hypothesis, H_1, is similarly represented by the model $\lambda_{\overline{\text{hyp}}}$. The likelihood ratio statistic (or score) is then $p(X \mid \lambda_{\text{hyp}})/p(X \mid \lambda_{\overline{\text{hyp}}})$. Often, the logarithm of this statistic is used giving the log-likelihood ratio (LLR) score for the utterance,

$$\Lambda(X) = \log p(X \mid \lambda_{\text{hyp}}) - \log p(X \mid \lambda_{\overline{\text{hyp}}}) \,. \qquad (38.2)$$

In discriminative modeling approaches such as support vector machines (SVMs), the two likelihoods are not explicitly modeled separately, but a detection score similar to the log likelihood ratio score is computed (see Sect. 38.4.2). Although not precise mathematically, it is useful to use this likelihood ratio framework for both generative and discriminative approaches since it emphasizes the important roles of the speaker and alternative hypotheses in the modeling and detection process.

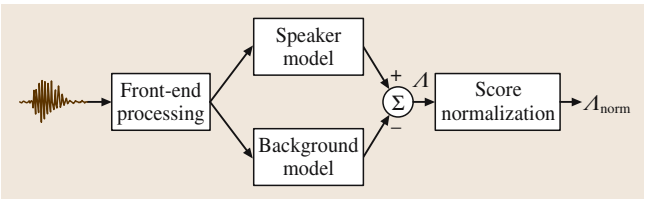


Fig. 38.1 Likelihood-ratio-based speaker detection system

After a likelihood ratio score is produced for the test speech and hypothesized speaker, it is often further processed by some form of score normalization and/or calibration. Despite mitigation steps taken during feature extraction (see Sect. 38.3.1) and model parameter estimation (see Sect. 38.4), the LLR score produced will be influenced by several nonspeaker factors, such as the spoken text in the test speech, differences in the microphones used and acoustic environment of the enrollment and test speech, and inaccuracies and assumptions in the models applied. Score normalization aims to compensate for these variabilities by removing score biases and scale factors estimated during enrollment. Several approaches have been proposed for score normalization, for example, Z-norm [38.9], H-norm [38.10], and T-norm [38.11], which compensate the LLR score as

$$\Lambda^{\text{norm}}(X) = \frac{\Lambda(X) - \mu(X, S)}{\sigma(X, S)} \,. \qquad (38.3)$$

The main difference in these score normalization techniques is how the biases, $\mu(X, S)$, and scale factors, $\sigma(X, S)$, are estimated. For example, in the H-norm, handset-microphone-dependent values are estimated, while in the T-norm scores from other speaker models are used to estimate these bias and scale factors.

A further step applied after or as part of score normalization is that of score calibration or confidence estimation. The aim in this process is to convert the LLR score values into *human-interpretable* values, such as posterior probabilities, with a defined range of, for example, (0–1). Confidence estimation techniques range from simple formula-based approaches that merely convert the LLR score to a posterior probability using a given prior probability, to more-sophisticated ap-

Amazon EX1010 Page 775

proaches that integrate in external signal measurement information, such as signal-to-noise measures, with the LLR score using artificial neural networks [38.12].

With this framework, we next describe specific examples of these components for modern text-independent speaker detection systems.

38.3 Features

One of the fundamental components in any speaker detection system is the features extracted from the speech signal that convey information about the speaker's identity. Unfortunately there is no single attribute of speech that conveys identity, rather it is conveyed through several different types or levels of information in the speech signal. These levels can roughly be categorized into a hierarchy running from low-level information, such as the sound of a person's voice, related to physical traits of the vocal apparatus, to high-level information, such as particular word usage (idiolect), related to learned habits, dialect, and style.

Features related to low-level information based on short-term spectral analysis are the dominant type used in text-independent speaker detection systems. This is primarily due to the computational ease of extracting these features and their proven effectiveness in terms of detection performance.

In recent years, there has been increased interest in exploiting high-level features for text-independent speaker detection systems. This has been spurred on by the continual advancement of phone and speech recognition systems used to extract features for high-level characterization. Work done at the 2002 SuperSID workshop [38.13] and subsequent related efforts demonstrated how systems using high- and low-level features could be successfully fused to improve overall accuracy. While high-level features are a promising new area of research in text-independent speaker recognition, they often have substantial computational costs (e.g., running a large-vocabulary speech recognizer), which need to be balanced against relatively modest accuracy gains.

38.3.1 Spectral Features

Several processing steps occur in the front-end analysis to extract spectral-based features. First, the speech is segmented into frames by a 20 ms window progressing at a 10 ms frame rate. A speech activity detector is then used to discard silence/noise frames. Typically, the speech detector discards 20–25% of the signal from conversational telephone speech.

Next, mel-scale cepstral feature vectors are extracted from the speech frames. The mel-scale cepstrum is the discrete cosine transform of the log-spectral energies of the speech segment Y. The spectral energies are calculated over logarithmically spaced filters with increasing bandwidths (*mel-filters*). A detailed description of the feature extraction steps can be found in [38.14]. For bandlimited telephone speech, cepstral analysis is usually performed only over the mel-filters in the telephone pass band (300–3400 Hz). All cepstral coefficients except the zeroth value (the DC level of the log-spectral energies) are retained in the processing. Lastly, delta-cepstra values are computed using a first-order derivative of an orthogonal polynomial temporal fit over ± 2 feature vectors (two to the left and two to the right over time) from the current vector [38.15].

Finally, the feature vectors are channel-normalized to remove linear channel convolutional effects. With cepstral features, convolutional effects appear as additive biases. Both cepstral mean subtraction (CMS) and RASTA filtering [38.16] have been used successfully and, in general, both methods have comparable performance for single-speaker detection tasks. When training and recognition speech are collected from different microphones or channels (e.g., different telephone handsets and/or lines), this is a crucial step for achieving good recognition accuracy. However, this linear compensation does not completely eliminate the performance loss under mismatched microphone conditions, so more-sophisticated compensation techniques such as feature mapping [38.17], where transformations are trained to map features coming from particular channels (e.g., microphones) into channel independent features, are also applied.

There are many variants of these cepstral features, such as using linear prediction coding (LPC) spectral analysis instead of fast Fourier transform (FFT)-based spectral analysis, but the basic steps are similar and performance does not vary significantly.

38.3.2 High-Level Features

For high-level feature extraction, input speech is converted into a series of tokens, $T = \{t_i\}$. The tokens are time-ordered discrete symbols and represent linguistically significant interpretations of the input signal.

Amazon EX1010 Page 776

Examples of token types are words, phones, and pitch gestures.

High-level feature extraction involves many tradeoffs. First, designing token extraction systems is, in general, a difficult process. If the high-level features are specific to the language of interest, then significant effort may be required to implement the system. For example, speech-to-text (STT) systems such as the BBN Byblos system [38.18] require training using large corpora in the language of interest and significant engineering effort to produce high-accuracy output. A second consideration in token extraction is how it will be used with other speaker recognition recognition systems. Some token systems, such as STT, have been shown to provide significant fusion gains when combined with standard acoustic systems [38.19], but have low speaker recognition accuracy alone. Other token systems, such as phone token [38.20], have excellent standalone accuracy, but do not provide significant complimentary information to standard spectral methods. A third consideration for token system selection is length of enrollment and verification. A token system that produces tokens at a low rate, e.g., words, requires multi-conversation enrollment to produce good results. Other higher rate tokens, e.g., phones, require less enrollment for equivalent accuracy.

Modeling of the token stream is usually accomplished by computing probabilities of n-grams of token contexts [38.21] and then applying a classifier. We discuss two methods based on SVMs and standard likelihood ratio techniques in Sect. 38.4.3.

38.4 Classifiers

In this section, we describe popular and successful generative and discriminative classifiers used for both high- and low-level features. First we present spectral-feature-based classifiers using adapted Gaussian mixture models and sequence-based support vector machines. This is followed by high-level (token sequence) feature based n-gram and SVM classifiers.

38.4.1 Adapted Gaussian Mixture Models

Gaussian Mixture Models

An important step in the implementation of the likelihood ratio detector of (38.2) is selection of the actual likelihood function, $p(X|\lambda)$. The choice of this function is largely dependent on the features being used as well as specifics of the application. For text-independent speaker recognition using continuous features, where there is no prior knowledge of what the speaker will say, the most successful likelihood function has been Gaussian mixture models (GMMs). In text-dependent applications, where there is strong prior knowledge of the spoken text, additional temporal knowledge can be incorporated by using hidden Markov models (HMMs) as the basis for the likelihood function. To date, however, use of more-complicated likelihood functions, such as those based on HMMs, have shown no advantage over GMMs for text-independent speaker detection tasks. As more-accurate knowledge of the spoken text is gained via speech recognition, lessening the distinction between text-dependent and text-independent tasks, this may change.

For a D-dimensional feature vector, $\boldsymbol{x}$, the mixture density used for the likelihood function is defined as

$$p(\boldsymbol{x}|\lambda) = \sum_{i=1}^{M} w_i \; p_i(\boldsymbol{x}) \; . \tag{38.4}$$

The density is a weighted linear combination of M unimodal Gaussian densities, $p_i(\boldsymbol{x})$, each parameterized by a mean $D \times 1$ vector, $\boldsymbol{\mu}_i$, and a $D \times D$ covariance matrix, Σ_i;

$$p_i(\boldsymbol{x}) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \times \exp\left[-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu}_i)^{\mathrm{T}} \; \Sigma_i^{-1} \; (\boldsymbol{x} - \boldsymbol{\mu}_i) \right] . \tag{38.5}$$

The mixture weights, w_i, furthermore satisfy the constraint $\sum_{i=1}^{M} w_i = 1$. Collectively, the parameters of the density model are denoted by $\lambda = \{w_i, \boldsymbol{\mu}_i, \Sigma_i\}$, where $i = 1, \ldots, M$.

While the general model form supports full covariance matrices, i. e., a covariance matrix with all its elements, usually only diagonal covariance matrices are used. This is done for three reasons. First, the density modeling of an Mth-order full covariance GMM can equally well be achieved using a larger-order diagonal covariance GMM. GMMs with $M > 1$ using diagonal covariance matrices can model distributions of feature vectors with correlated elements. Only in the degenerate case of $M = 1$ is the use of a diagonal covariance matrix incorrect for feature vectors with correlated elements. Second, diagonal-matrix

GMMs are more computationally efficient than full-covariance GMMs for training since repeated inversions of a $D \times D$ matrix are not required. Third, empirically we have observed that diagonal-matrix GMMs outperform full-matrix GMMs.

Given a collection of training vectors, maximum-likelihood model parameters are estimated using the iterative expectation-maximization (EM) algorithm [38.22]. The EM algorithm iteratively refines the GMM parameters to increase the likelihood of the estimated model for the observed feature vectors monotonically, i. e., for iterations k and $k+1$, $p(X|\lambda^{(k+1)}) > p(X|\lambda^{(k)})$. Generally, five iterations are sufficient for parameter convergence. The EM equations for training a GMM can be found in [38.23].

The feature vectors of X are assumed independent, so the log-likelihood of a model λ for a sequence of feature vectors, $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T\}$, is computed as

$$\log p(X \mid \lambda) = \sum_{t=1}^{\mathrm{T}} \log p(\boldsymbol{x}_t \mid \lambda) \,, \tag{38.6}$$

where $p(\boldsymbol{x}_t \mid \lambda)$ is computed as in (38.4). Often, the average log-likelihood value is used, by dividing $\log p(X \mid \lambda)$ by T. This is done to normalize out duration effects from the log-likelihood value. Since the incorrect assumption of independence is underestimating the actual likelihood value with dependencies, this scaling factor can also be considered a rough compensation factor to the likelihood value in (38.6).

The GMM can be viewed as a hybrid between a parametric and nonparametric density model. Like a parametric model it has structure and parameters that control the behavior of the density in known ways, but without constraints that the data must be of a specific distribution type, such as Gaussian or Laplacian. Like a nonparametric model, the GMM has many degrees of freedom to allow arbitrary density modeling, without undue computation and storage demands. It can also be thought of as a single-state HMM with a Gaussian mixture observation density, or an ergodic Gaussian observation HMM with fixed, equal transition probabilities. Here, the Gaussian components can be considered to be modeling the underlying broad phonetic sounds that characterize a person's voice. A more-detailed discussion of how GMMs apply to speaker modeling can be found in [38.24].

The advantages of using a GMM as the likelihood function are that it is computationally inexpensive, is based on a well-understood statistical model, and, for text-independent tasks, is insensitive to the temporal aspects of the speech, modeling only the underlying distribution of acoustic observations from a speaker. The latter is also a disadvantage in that higher-levels of information about the speaker conveyed in the temporal speech signal are not used.

Universal Background Model

We next describe how the GMM described above is used to build the likelihood ratio detector in (38.2). While the model for H_0, λ_{hyp}, is well defined and is estimated using training speech from S, the model for the alternative hypothesis, $\lambda_{\overline{\mathrm{hyp}}}$, is less well defined since it potentially must represent the entire space of possible alternatives to the hypothesized speaker. Two main approaches have been taken for this alternative hypothesis modeling. The first approach is to use a set of other speaker models to cover the space of the alternative hypothesis. In various contexts, this set of other speakers has been called likelihood ratio sets [38.2], cohorts [38.25], and background speakers [38.23]. Given a set of N background speaker models $\{\lambda_1, \ldots, \lambda_N\}$, the alternative hypothesis model is represented by

$$p(X \mid \lambda_{\overline{\mathrm{hyp}}}) = \mathcal{F}[p(X \mid \lambda_1), \ldots, p(X \mid \lambda_N)], \tag{38.7}$$

where $\mathcal{F}()$ is some function, such as the average or maximum, of the likelihood values from the background speaker set. The selection, size, and combination of the background speakers has been the subject of much research. In general, it has been found that to obtain the best performance with this approach requires the use of speaker-specific background speaker sets. This can be a drawback in an applications using a large number of hypothesized speakers, each requiring their own background speaker set.

The second major approach to alternative hypothesis modeling is to pool speech from several speakers and train a single model. Various terms for this single model are a general model [38.26], a world model, and a universal background model [38.27]. Given a collection of speech samples from a large number of speakers representative of the population of speakers expected during recognition, a single model λ_{bkg} is trained to represent the alternative hypothesis. Research on this approach has focused on selection and composition of the speakers and speech used to train the single model. The main advantage of this approach is that a single speaker-independent model can be trained once for a particular task and then used for all hypothesized speakers in that task. It is also possible to use multiple background models tailored to specific sets of speakers. Overall,

the use of a single background model [which we will refer to as the universal background model (UBM)], is the dominant approach used in text-independent speaker detection systems.

The UBM is a large GMM trained to represent the speaker-independent distribution of features. Specifically, we want to select speech that is reflective of the expected alternative speech to be encountered during recognition. This applies to both the type and quality of speech, as well as the composition of speakers. For example, in the NIST speaker recognition evaluation (SRE) single-speaker detection tests, it is known a priori that the speech comes from local and long-distance telephone calls, and that male hypothesized speakers will only be tested against male speech. In this case, we would train the UBM used for male tests using only male telephone speech. In the case where there is no prior knowledge of the gender composition of the alternative speakers, we would train using gender-independent speech.

Other than these general guidelines and experimentation, there is no objective measure to determine the right number of speakers or amount of speech to use in training a UBM. Empirically, from the NIST SRE we have observed no performance loss using a UBM trained with one hour of speech compared to one trained using six hours of speech. In both cases, the training speech was extracted from the same speaker population.

Adaptation of Speaker Model

In the GMM UBM system, the speaker model is derived by adapting the parameters of the UBM using the speaker's training speech and a form of Bayesian adaptation [38.28]. This is also known as Bayesian learning or *maximum a posteriori* (MAP) estimation. We use the term Bayesian adaptation since, as applied to the speaker-independent UBM to estimate the speaker-dependent model, the operation closely resembles speaker adaptation used in speech recognition applications. Unlike the standard approach of maximum-likelihood training of a model for the speaker independently of the UBM, the basic idea in the adaptation approach is to derive the speaker's model by updating the well-trained parameters in the UBM via adaptation. This provides a tighter coupling between the speaker's model and UBM, which not only produces better performance than decoupled models, but as discussed later in this section, also allows for a fast scoring technique. Like the EM algorithm, the adaption is a two-step estimation process. The first step is identical to the *expectation* step of the EM algorithm, where estimates of the sufficient statistics of the speaker's training data are computed for each mixture in the UBM. For a GMM mixture, these are the count, and the first and second moments required to compute the mixture weight, mean and variance. Unlike the second step of the EM algorithm, for adaptation these *new* sufficient statistic estimates are then combined with the *old* sufficient statistics from the UBM mixture parameters using a data-dependent mixing coefficient. The data-dependent mixing coefficient is designed so that mixtures with high counts of data from the speaker rely more on the new sufficient statistics for final parameter estimation and mixtures with low counts of data from the speaker rely more on the old sufficient statistics for final parameter estimation.

While it is possible to adapt all the parameters of the GMM during speaker model training, it has been widely shown that speaker detection performance is best when only the mean parameters are adapted [38.11, 29, 30]. Here, we will focus only on mean adaptation but details of adapting all parameters can be found in [38.30].

Given a UBM and training vectors from the hypothesized speaker, $X = \{\boldsymbol{x}_1 \ldots, \boldsymbol{x}_T\}$, we first determine the probabilistic alignment of the training vectors into the UBM mixture components (Fig. 38.2a). That is, for the mixture i in the UBM, we compute

$$\Pr(i|\boldsymbol{x}_t) = \frac{w_i p_i(\boldsymbol{x}_t)}{\sum_{j=1}^{M} w_j p_j(\boldsymbol{x}_t)} \,. \tag{38.8}$$

We then use $\Pr(i|\boldsymbol{x}_t)$ and $\boldsymbol{x}_t$ to compute the sufficient statistics for the mean parameter,

$$n_i = \sum_{t=1}^{\mathrm{T}} \Pr(i|\boldsymbol{x}_t), \tag{38.9}$$

$$E_i(\boldsymbol{x}) = \frac{1}{n_i} \sum_{t=1}^{\mathrm{T}} \Pr(i|\boldsymbol{x}_t)\boldsymbol{x}_t \,. \tag{38.10}$$

This is the same as the *expectation* step in the EM algorithm.

Lastly, these new sufficient statistics from the training data are used to update the old UBM sufficient statistics for the mixture i to create the adapted mean parameter for the mixture i (Fig. 38.2b):

$$\hat{\boldsymbol{\mu}}_i = \alpha_i E_i(\boldsymbol{x}) + (1-\alpha_i)\boldsymbol{\mu}_i \,. \tag{38.11}$$

The data-dependent adaptation coefficients controlling the balance between old and new estimates per mixture are $\{\alpha_i\}$, defined as

$$\alpha_i = \frac{n_i}{n_i + r} \,. \tag{38.12}$$

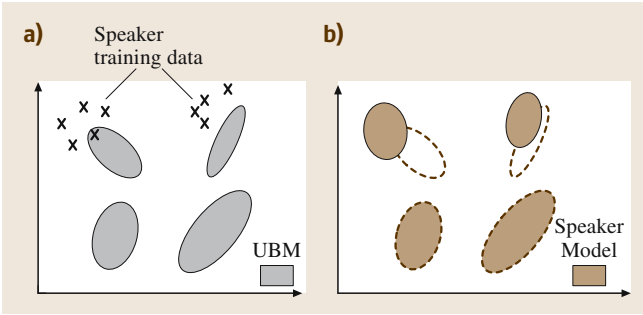


Fig. 38.2a,b Pictorial example of two steps in adapting a hypothesized speaker model. **(a)** The training vectors (*crosses*) are probabilistically mapped into the UBM mixtures. **(b)** The adapted mixture parameters are derived using the statistics of the new data and the UBM mixture parameters. The adaptation is data dependent, so UBM mixture parameters are adapted by different amounts

Here r is a fixed *relevance* factor that controls the amount of adaption from the UBM (e.g., higher values of r require more speaker data in a mixture component to adapt from the prior UBM mean).

The parameter updating as described in (38.11–38.12) can be derived from the general MAP estimation equations for a GMM using constraints on the prior distribution described in [38.28].

Using a data-dependent adaptation coefficient allows mixture-dependent adaptation of parameters. If a mixture component has a low probabilistic count, n_i, of new data, then $\alpha_i \to 0$ causing the de-emphasis of the new (potentially undertrained) parameters and the emphasis of the old (better-trained) parameters. For mixture components with high probabilistic counts, $\alpha_i \to 1$, causing the use of the new speaker-dependent parameters. The relevance factor is a way of controlling how much new data should be observed in a mixture before the new parameters begin replacing the old parameters. Values of r in the range 8–20 have been shown to work well for several speaker recognition tasks.

Comparison of published results strongly indicate that the adaptation approach provides superior performance over a decoupled system where the speaker model is trained independently of the UBM. One possible explanation for the improved performance is that the use of adapted models in the likelihood ratio is not affected by *unseen* acoustic events in recognition speech. Loosely speaking, if one considers the UBM as covering the space of speaker-independent, broad acoustic classes of speech sounds, then adaptation is the speaker-dependent *tuning* of those acoustic classes observed in the speaker's training speech. Mixture parameters for those acoustic classes not observed in the training speech are merely copied from the UBM. This means that, during recognition, data from acoustic classes unseen in the speaker's training speech produce approximately zero log-likelihood ratio values that contribute evidence neither toward nor against the hypothesized speaker. Speaker models trained using only the speaker's training speech will have low likelihood values for data from classes not observed in the training data, thus producing low likelihood ratio values. While this is appropriate for speech not from the speaker, it clearly can cause incorrect values when the unseen data occurs in test speech from the speaker.

Log-Likelihood Ratio Computation

The log-likelihood ratio for a test sequence of feature vectors X is computed as $\Lambda(X) = \log p(X|\lambda_{\text{hyp}}) - \log p(X|\lambda_{\text{ubm}})$. The fact that the hypothesized speaker model was adapted from the UBM, however, allows a faster scoring method than merely evaluating the two GMMs as in (38.6). This fast scoring approach is based on two observed effects. The first is that, when a large GMM is evaluated for a feature vector, only a few of the mixtures contribute significantly to the likelihood value. This is because the GMM represents a distribution over a large space but a single vector will be near only a few components of the GMM. Thus, likelihood values can be approximated very well using only the top-C best scoring mixture components.

The second observed effect, is that the components of the adapted GMM retain a correspondence with the mixtures of the UBM, so that vectors close to a particular mixture in the UBM will also be close to the corresponding mixture in the speaker model. Using these two effects, a fast scoring procedure operates as follows: for each feature vector, determine the top-C scoring mixtures in the UBM and compute the UBM likelihood using only these top-C mixtures. Next, score the vector against only the corresponding C components in the adapted speaker model to evaluate the speaker's likelihood. For a UBM with M mixtures, this requires only $M + C$ Gaussian computations per feature vector compared to $2M$ Gaussian computations for normal likelihood ratio evaluation. When there are multiple hypothesized speaker models for each test segment, the savings become even greater. Typical values of C are in the range 1–5.

GMM Compensation

In addition to compensation techniques applied in the feature and score domains, there are also many effec-

Amazon EX1010 Page 780

tive compensation techniques that can be applied in the model domain. For GMM-based classifiers, techniques that treat the undesired variability as a bias to the mean vectors have been shown to be very promising in improving the robustness of speaker recognition system. If we stack the means from a GMM into a *supervector* this can be written as

$$\boldsymbol{m}_j(s) = \boldsymbol{m}(s) + \boldsymbol{c}(s), \tag{38.13}$$

where $\boldsymbol{m}_j(s)$ is the supervector from speaker s's j-th enrollment session, $\boldsymbol{m}(s)$ is the desired compensated supervector for speaker s and $\boldsymbol{c}(s)$ is the undesired variability supervector. The main difference in the compensation techniques is in how they estimate and remove the variability vector $\boldsymbol{c}(s)$.

In speaker model synthesis (SMS) [38.31], the difference between bias vectors from a set of predefined channel types is used to synthetically generate a library of channel-dependent speaker models so as to allow matched-channel likelihood ratio scoring during recognition. Feature mapping [38.17], which is applied in the feature domain, is similar to SMS, but instead of synthetically creating unseen channel-dependent speaker models, acts to subtract off the channel-dependent biases in training and testing from the feature vectors directly.

More-recent latent factor analysis (LFA)-based techniques [38.32, 33], model the supervector bias as a low-dimensional normal bias,

$$\boldsymbol{c}(s) = U\boldsymbol{n}(s)\,, \tag{38.14}$$

where U is the low-rank session loading matrix. The LFA techniques are aimed specifically at compensation of session variability and do not require prior channel detectors or parameters. During training, the session bias is removed from each enrollment session and the session-independent supervectors are combined to produce the final speaker model. During recognition, the session bias for the test utterance is estimated and added to the speaker model to allow for matched session likelihood ratio scoring. A feature-space dual of LFA has also been developed and used successfully [38.34].

38.4.2 Support Vector Machines

A support vector machine (SVM) is a powerful classifier that has gained considerable popularity in recent years. An SVM is discriminative – it models the boundary between a speaker and a set of impostors. This approach contrasts to traditional methods for speaker recognition which separately model the probability distributions of the speaker and the general population. SVMs are based upon the structural risk minimization principal of *Vapnik* [38.35].

Two broad approaches using support vector machines have been proposed in the literature for speech applications. The first set of methods attempts to model emission probabilities for hidden Markov models [38.36–38]. This approach has been moderately successful in reducing error rates, but suffers from several problems. First, large training sets result in long training times for support vector methods. Second, the emission probabilities must be approximated [38.39], since the output of the support vector machine is not a probability. This approximation is needed to combine probabilities using the standard frame-independence assumption used in HMM speaker recognition.

A second set of methods is based upon comparing speech utterances using sequence kernels. Rather than model features from individual frames of speech, these methods instead model the entire sequence of feature vectors. Approaches include the generalized linear discriminant sequence kernel [38.40], Fisher kernel methods [38.41, 42], n-gram kernels [38.20], maximum-likelihood linear regression (MLLR) transform kernels [38.43], and GMM supervector kernels [38.44]. We focus on these latter methods, which have been successful in speaker recognition because of their accuracy and simplicity of implementation.

Basic SVM Theory

An SVM [38.35] models two classes using sums of a kernel function $K(\cdot,\cdot)$,

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i t_i K(\boldsymbol{x}, \boldsymbol{x}_i) + d, \tag{38.15}$$

where the t_i are the ideal outputs, $\sum_{i=1}^{N} \alpha_i t_i = 0$, and $\alpha_i > 0$. The vectors $\boldsymbol{x}_i$ are support vectors and obtained from the training set by an optimization process [38.45]. The ideal outputs are either 1 or -1, depending upon whether the corresponding support vector is in class 0 or class 1, respectively. For classification, a class decision is based upon whether the value, $f(\boldsymbol{x})$, is above or below a threshold.

The kernel $K(\cdot,\cdot)$ is constrained to have certain properties (the Mercer condition), so that $K(\cdot,\cdot)$ can be expressed as

$$K(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{b}(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{b}(\boldsymbol{y}), \tag{38.16}$$

where $\boldsymbol{b}(\boldsymbol{x})$ is a mapping from the input space (where $\boldsymbol{x}$ lives) to a possibly infinite-dimensional *expansion*

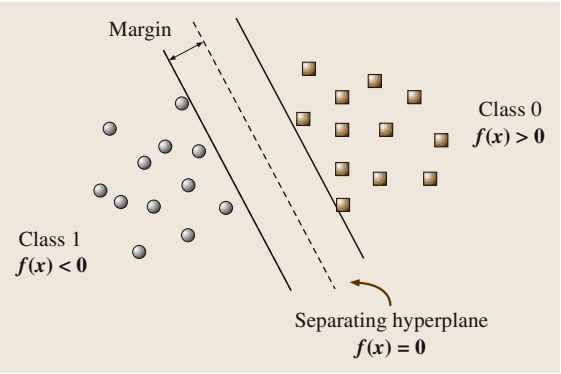


Fig. 38.3 Support vector machine concept

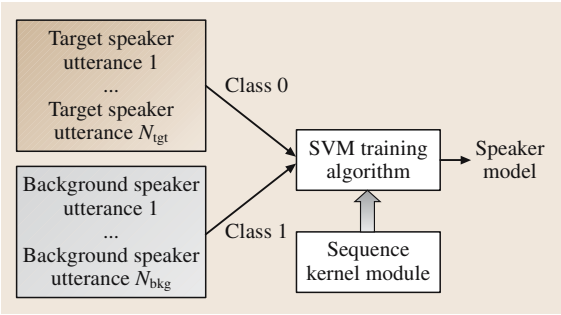


Fig. 38.4 SVM training strategy

space. The kernel is required to be positive semidefinite. The Mercer condition ensures that the margin concept is valid, and the optimization of the SVM is bounded.

The optimization condition relies upon a maximum margin concept (Fig. 38.3). For a separable data set, the system places a hyperplane in a high dimensional space so that the hyperplane has maximum margin. The data points from the training set lying on the boundaries (as indicated by solid lines in the figure) are the support vectors in (38.15). The focus, then, of the SVM training process is to model the boundary, as opposed to a traditional GMM UBM, which would model the probability distributions of the two classes.

Application of Support Vector Machines to Speaker Recognition

Discriminative training of an SVM for speaker recognition is straightforward, but several basic issues must be addressed: handling multiclass data, *world* modeling, and sequence comparison. The first two topics are handled in this section.

Since the SVM is a two-class classifier, we must recast speaker verification and identification in this structure. For speaker verification, a target model is trained for the speaker against a set of example speakers that have characteristics of the impostor population (Fig. 38.4). The set of example impostors is called a background set. In the figure, class 0 consists of all of the target speaker's utterances, and class 1 consists of the example impostors in the background set. The background speaker set is kept the same as we enroll different target speakers.

Figure 38.4 indicates the basic training strategy for SVMs using sequence kernels. Each utterance from a target or background speaker becomes a point in the SVM expansion space, see (Fig. 38.3). A sequence kernel module for comparing two utterances and producing a kernel value is implemented. The kernel module is connected into a standard SVM training tool that then produces a speaker model.

For speaker identification, a *one-versus-all* strategy is used. For a given target speaker, we pool all of the remaining nontarget speakers into class 1 and then train a speaker model. This operation is performed for all models to obtain our set of speaker identification models.

For speaker verification, the support vectors have an interesting interpretation. If $f(\boldsymbol{x})$ is an SVM for a target speaker, then we can write

$$f(\boldsymbol{x}) = \sum_{i\in\{i|t_i=1\}} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) - \sum_{i\in\{i|t_i=-1\}} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) + d. \quad (38.17)$$

The first sum can be interpreted as a per-utterance-weighted target score. The second sum has many of the characteristics of a cohort score [38.25] with some subtle differences. First, utterances rather than speakers are used as cohorts. Second, the weighting on these *cohort* utterances is not equal.

The interpretation of the SVM score as a cohort-normalized score suggests how the *world* of impostors should be modeled. Our background should have a rich speaker set, so that we can always find speakers close to the target speaker. The speakers in the background should have similar attributes – language, accent, channel type, etc. – for best performance. Note that this interpretation distinguishes the SVM approach from a universal background model method [38.30], which tries to model the impostor set with one model. Other methods for GMMs including cohort normalization [38.25] and T-norm [38.11] are closer to the SVM method; although, the latter method (T-norm) typ-

ically uses a fixed set of cohorts rather than picking out individual speakers.

Sequence Kernels for Speaker Recognition – General Structure

To apply an SVM, $f(\boldsymbol{x})$, to a speaker recognition application, we need a method for calculating kernel values from speech inputs. For recognition, a way is needed for taking a sequence of input feature vectors from an utterance, $\{\boldsymbol{x}_i\}$, and computing the SVM output, $f(\{\boldsymbol{x}_i\})$. Typically, each vector $\boldsymbol{x}_i$ would be the cepstral coefficients and deltas for a given frame of speech. One way of handling this situation is to assume that the kernel, $K(\cdot,\cdot)$, in the SVM (38.15) takes sequences as inputs, i. e., we can calculate $K(\{\boldsymbol{x}_i\},\{\boldsymbol{y}_j\})$ for two input sequences $\{\boldsymbol{x}_i\}$ and $\{\boldsymbol{y}_j\}$. This is called a sequence kernel method.

A challenge in applying the sequence kernel method is deriving a function for comparing sequences. A function is needed that, given two utterances, produces a measure of similarity of the speakers or languages. Also, a method that is efficient computationally is needed, since we will be performing many kernel inner products during training and scoring. Finally, the kernel must satisfy the Mercer condition.

One idea for constructing a sequence kernel is illustrated in Fig. 38.5. The basic approach is to compare two speech utterances, *utt 1* and *utt 2* by training a model on one utterance and then scoring the resulting model on another utterance. This process produces a value that measures the similarity between the two utterances. Two questions that follow from this approach are:

1. Can the train/test process be computed efficiently?
2. Is the resulting comparison a kernel (i. e., does it satisfy the Mercer condition)?

The answer to both of these questions is yes if the appropriate classifier is chosen – see next section.

Another idea for constructing sequence kernels is shown in Fig. 38.6. In this setup, a base model is adapted to obtain probability distributions which represent the utterances. A model comparison algorithm is then applied to get a measure of similarity. This approach has the useful property that it is naturally symmetric as long as the comparison calculation is symmetric.

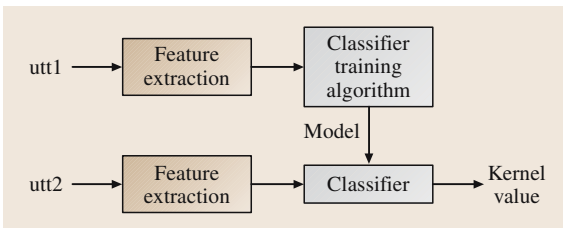


Fig. 38.5 General train/test sequence kernel

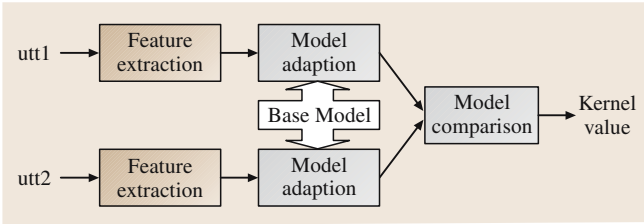


Fig. 38.6 Generative model sequence kernel

Sequence Kernels for Speaker Recognition – Specific Examples

For the train/test kernel shown in Fig. 38.5, a typical approach is the generalized linear discriminant sequence (GLDS) kernel [38.40]. In this method, the classifier is taken to be a polynomial discriminant function consisting of the monomials up to a certain degree; e.g., for two features, x_1 and x_2, and degree 2 the monomials are,

$$\boldsymbol{b}(\boldsymbol{x}) = \left(1, x_1, x_2, x_1^2, x_1x_2, x_2^2\right)^{\mathrm{T}}. \tag{38.18}$$

Suppose we have two sequences of feature vectors, $\{\boldsymbol{x}_i\}$ and $\{\boldsymbol{y}_j\}$. If a polynomial discriminant is trained using mean-squared error, then the resulting kernel is given by

$$K(\{\boldsymbol{x}_i\},\{\boldsymbol{y}_j\}) = \bar{b}_x^{\mathrm{T}}\bar{R}^{-1}\bar{b}_y. \tag{38.19}$$

In (38.19),

$$\bar{b}_x = \frac{1}{N_x}\sum_i \boldsymbol{b}(\boldsymbol{x}_i)\,, \tag{38.20}$$

i. e., $\bar{b}_x$ is the average expansion over all frames. The quantity $\bar{b}_y$ is defined similarly for the sequence, $\{\boldsymbol{y}_j\}$. The matrix, $\bar{R}$ is the correlation matrix of a background data set; typically, it is approximated with only diagonal terms. For details on the derivation of these equations, refer to [38.40].

An interesting generalization of the GLDS kernel is to replace the polynomial expansion by a general kernel using the kernel trick [38.46]. This idea leads to a method of transforming frame-level kernels into sequence kernels.

The GLDS kernel in (38.19) has the properties expected for a sequence kernel. The kernel is a single value from a sequence of vectors, symmetric, and positive definite. An interesting property of the kernel is that the kernel itself is a classifier. In a certain sense, the

SVM method acts as a method of strengthening a weaker classifier.

For the generative model sequence kernel, several methods have been proposed. Current methods are based upon adapting from a GMM or HMM base model. In [38.47], GMMs are adapted to the utterances, and an exponentiated Kullbach Leibler (KL) divergence is used to compare the utterances. In [38.43], adaptation of an HMM from a speech-to-text system is performed using maximum-likelihood linear regression (MLLR). The MLLR adaptation parameters are then compared with a weighted linear inner product. In [38.44], the adaptation is performed via MAP adaptation of a GMM. In this GMM supervector (GSV) system, the GMMs are compared using either an approximation to the KL divergence or an integral inner product. Finally [38.42] also fits loosely into this structure. A GMM model is trained from all utterance data. Then, the comparison is performed using both the base model and the target model via a modified Fisher kernel method [38.42, 48].

Nuisance Attribute Projection (NAP)

As with the GMM, compensations with SVM classifiers can also be applied directly in the model domain. The SVM nuisance attribute projection (NAP) method [38.49] works by removing subspaces that cause variability in the kernel. NAP constructs a new kernel,

$$\begin{aligned} K(\{\boldsymbol{x}_i\},\{\boldsymbol{y}_j\}) &= \left[P\bar{b}_x\right]^{\mathrm{T}}\left[P\bar{b}_y\right] \\ &= \bar{b}_x^{\mathrm{T}} P \bar{b}_y \\ &= \bar{b}_x^{\mathrm{T}}(I-\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}})\bar{b}_y, \end{aligned} \tag{38.21}$$

where P is a projection ($P^2 = P$), $\boldsymbol{v}$ is the direction being removed from the SVM expansion space, $\boldsymbol{b}(\cdot)$ is the SVM expansion, and $\|\boldsymbol{v}\|_2 = 1$. The design criterion for P and correspondingly $\boldsymbol{v}$ is

$$\boldsymbol{v}^* = \underset{\boldsymbol{v},\|\boldsymbol{v}\|_2=1}{\operatorname{argmin}} \sum_{i,j} W_{i,j} \left\| P\bar{b}_{z,i} - P\bar{b}_{z,j} \right\|_2^2 , \tag{38.22}$$

where $\bar{b}_{z,i}$ is the average expansion (38.20) of the i-th sequence in a background data set. Here, $W_{i,j}$ can be selected in several different ways. If we have channel nuisance variables (e.g., electret, carbon button, cell) and a labeled background set, then we can pick $W_{i,j} = 0$ when the channels of $\bar{b}_{z,i}$ and $\bar{b}_{z,j}$ are the same, and $W_{i,j} = 1$ otherwise. Another criterion is to design the projection based on session variability. In this case, we pick $W_{i,j} = 1$ if $\bar{b}_{z,i}$ and $\bar{b}_{z,j}$ correspond to the same speaker, and $W_{i,j} = 0$ otherwise. The idea in both cases is to reduce variability in the SVM kernel distance with respect to nuisances – channel or session. See [38.44] for more details and the relationship to LFA compensation for GMMs.

Note that, while NAP is described above in the context of spectral based features, it can also be applied when using high-level token features described in the next section.

38.4.3 High-Level Feature Classifiers

In this section, we consider methods for modeling a token stream as shown in Fig. 38.7. Two standard approaches have emerged – log-likelihood ratio and support vector machine methods.

Log-Likelihood Ratio Classifier

A token sequence can be modeled using a standard *bag of n-grams* approach [38.21, 50]. For a token sequence, n-grams are produced by the standard transformation of the token stream; e.g., for bigrams, the sequence of symbols from a conversation $t_1, t_2, \ldots, t_n$ is transformed to the sequence of bigrams of symbols $t_1\_t_2, t_2\_t_3 \ldots t_{n-1}\_t_n$. Probabilities of n-grams are then found with n fixed. That is, suppose unigrams and bigrams of symbols are considered, and the unique unigrams and bigrams of the corpus are designated $d_1 \ldots d_M$ and $d_1\_d_1, \ldots, d_M\_d_M$, respectively; then calculate probabilities

$$\begin{aligned} p(d_i) &= \frac{\#(t_k = d_i)}{\sum_l \#(t_k = d_l)}, \\ p(d_i\_d_j) &= \frac{\#(t_k\_t_{k+1} = d_i\_d_j)}{\sum_{l,m} \#(t_k\_t_{k+1} = d_l\_d_m)}, \end{aligned} \tag{38.23}$$

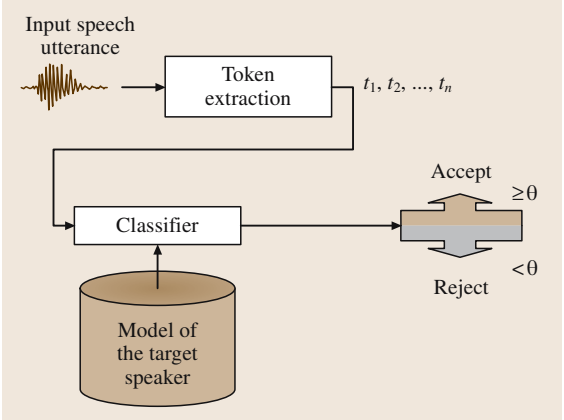


Fig. 38.7 High-level speaker recognition: basic setup

where $\#(t_k = d_i)$ indicates the number of symbols in the conversation equal to d_i, and an analogous definition is used for bigrams.

For simplicity, suppose that we have two utterances from two speakers, spk_1 and spk_2. Further suppose that the sequence of n-grams (for fixed n) in each conversation is $t_1, t_2 \ldots t_n$ and $u_1, u_2 \ldots u_m$ respectively. Denote the unique set of n-grams in the corpus as $d_1, \ldots, d_M$. A model can be built based upon the conversations for each speaker consisting of the probability of n-grams, $p(d_i|\text{spk}_j)$. The average log-likelihood ratio of the first conversation is computed as,

$$\begin{aligned}\text{score} &= \frac{1}{n}\sum_{i=1}^{n}\log\left(\frac{p(t_i|\text{spk}_2)}{p(t_i|\text{bkg})}\right)\\ &= \sum_{j=1}^{M}\frac{\#(t_i = d_j)}{n}\log\left(\frac{p(d_j|\text{spk}_2)}{p(d_j|\text{bkg})}\right)\\ &= \sum_{j=1}^{M} p(d_j|\text{spk}_1)\log\left(\frac{p(d_j|\text{spk}_2)}{p(d_j|\text{bkg})}\right), \end{aligned} \quad (38.24)$$

where $p(d_j|\text{bkg})$ is the probability of an n-gram in some large background data set. This scoring (38.24) can be easily generalized to multiple utterances per speaker; we just estimate the probabilities $p(d_j|\text{spk}_i)$ from a larger data set. Further details on this scoring method can be found in [38.21, 50].

SVM-Based Scoring

The score (38.24) can be transformed into a sequence kernel via a simple linearization; using $\log(x) \approx x - 1$ (the Taylor series expansion around $x = 1$), gives

$$\begin{aligned}\text{score} &\approx \sum_{j=1}^{M} p(d_j|\text{spk}_1)\frac{p(d_j|\text{spk}_2)}{p(d_j|\text{bkg})} - 1\\ &= \sum_{j=1}^{M}\frac{p(d_j|\text{spk}_1)}{\sqrt{p(d_j|\text{bkg})}}\frac{p(d_j|\text{spk}_2)}{\sqrt{p(d_j|\text{bkg})}} - 1. \end{aligned} \quad (38.25)$$

Once we have a sequence kernel, an SVM is trained in the same manner as the spectral-based SVMs in Fig. 38.4.

In analogy with the information retrieval literature [38.51], the result in (38.25) can be expressed in vector form. First, a vector $\boldsymbol{v}$ is constructed describing the conversation

$$\boldsymbol{v}_i = \begin{bmatrix} p(d_1|\text{spk}_i) \\ \vdots \\ p(d_M|\text{spk}_i) \end{bmatrix}. \quad (38.26)$$

In general, the vector $\boldsymbol{v}$ will be sparse since the conversation will not contain all potential unigrams, bigrams, etc. The entries of $\boldsymbol{v}$ are then weighted with a diagonal matrix, $\mathbf{D}$, with entries $D_{j,j} = 1/\sqrt{p(d_j|\text{bkg})}$. The final kernel is an inner product,

$$K(t_1^n, u_1^m) = (\mathbf{D}\boldsymbol{v}_1)^{\mathrm{T}}(\mathbf{D}\boldsymbol{v}_2)^{\mathrm{T}}. \quad (38.27)$$

The kernel (38.25) is referred to as the term frequency log-likelihood ratio (TFLLR) kernel [38.20].

The kernel in (38.25) can be generalized in a straightforward way by analyzing its components. In (38.25), the background weighting term, $1/p(d_j|\text{bkg})$, acts as a *commonality normalization*. That is, the resulting ratios in the vector (38.26) are approximately on the same scale. This normalization can be excessive if the probability estimate $p(d_j|\text{bkg})$ is small because n-gram is infrequent. To trade off between these extremes, it is natural to use a diagonal weight of the form

$$D_{j,j} = \min\left[C_j, g_j\left(\frac{1}{p(d_j|\text{bkg})}\right)\right], \quad (38.28)$$

where $g_j(\cdot)$ is a function which shrinks the dynamic range.

The general token sequence kernel obtained by combining (38.27) and (38.28) encompasses several useful weighting types. Setting $g_j(x) = \sqrt{x}$, TFLLR kernel is obtained. Second, similar to term frequency inverse document frequency (TFIDF) in information retrieval [38.51], we can use $g_j(x) = \log(x) + 1$; we refer to this weighting as a term frequency logarithmic (TFLOG) kernel. Third, an extreme case for (38.28) is to set $C_j = 1$. This results in weighting all terms equally, $D_{j,j} = 1$ for all j. Fourth, we note that both C_j and the function g_j are dependent on j. This variable weighting may make sense if we have some idea of the confidence of our estimates in the probabilities of n-grams in the background data set.

38.4.4 System Fusion

The fusion of outputs from systems modeling low- and high-level speaker information [38.19] as well as generative- and discriminative-based classifiers [38.52] have demonstrated large accuracy improvements over the constituent input systems. If the recognizers do not behave uniformly (i. e., their errors are not completely correlated) then combining the scores may make it possible to exploit complementary information that exists in the scores.

Fusion is yet another classifier that is trained on feature vectors consisting of scores from the input systems.

Ideally, a fusion classifier would be trained for each speaker, but in practice there is often limited enrollment speech per speaker that is best utilized for training the speaker model. So speaker recognition fusion systems are typically trained in a speaker-independent fashion to optimize classification of input scores as coming from the detector hypothesis H_0 (true trial) or H_1 (false trial). Vectors of system scores are computed on a development data set consisting of true and false trials from a large population of development speaker models, aggregated, and used to train a fusion classifier.

Many different fusion classifiers have been used. At the simplest level, the fusion consists of a linear combination of the system scores. The weights used in the combination can be equal, if it is assumed that each system contributes equally to overall performance and have common numeric bias and scale ranges. The weights are usually empirically adjusted on the development data set to minimize the detection error rate.

Another common and successful fusion classifier is a simple logistic regression or single-layer perceptron. This classifier is similar to the linear combination, but has a sigmoid compression function applied to the linear combination. The weights are trained using a minimum mean-squared error criteria for the target values of 0 and 1 via a form of gradient descent [38.53] on the development data. A benefit of this type of fusion classifier is that the fusion scores are often reasonable posterior probabilities estimates that can be used as confidence values.

Fusion classifiers can also use information in addition to the system scores. For example, indicators of training and/or testing data conditions, such as speech durations, spoken language or signal-to-noise measures, can be used for explicit or implicit fusion guidance. Explicit guidance is when the indicator information is used to train and apply conditional fusion classifiers, such as fusion classifiers trained on scores from speaker models trained with specific durations of enrollment speech or for speech coming from particular languages. Conditional fusion classifiers can use different system scores, for example high-level system scores depending on an English speech recognizer may be excluded when the train and/or test speech is known to not be spoken in English. Implicit guidance is when the external information is used as another element with the system scores in the fusion vector. This approach relies on the fusion classifier to learn correlations of external measures with system scores to optimize accuracy. Implicit guidance is often inferior to explicit guidance since the fusion classifiers do not have enough complexity or enough training data to learn and represent the desired conditions.

38.5 Performance Assessment

The speaker recognition approaches described in this chapter have been subject to extensive development, and this section describes their evaluation under conditions that follow protocols established by the US National Institute of Standards and Technology (NIST). Since 1995, NIST has coordinated several evaluations with the goal of assessing the existing state of the art in speaker recognition technology. The most recent speaker recognition evaluation (SRE 2006) protocol will form the basis of the performance results presented in this section. A description of the 2006 NIST speaker recognition evaluation plan can be found in [38.54].

38.5.1 Task and Corpus

The task in the NIST SRE is the basic speaker detection task described in Sect. 38.2: determine if the speaker in the training speech is the same as the speaker in the test speech. This task is evaluated for various conditions of amount of training speech, amount of test speech, recording source (telephone or auxiliary microphones), and presence of other speakers in the speech. In all there were 15 conditions in the 2006 SRE. More details of these conditions can be found in [38.55].

The data comes from the Mixer telephone speech corpus collected by the Linguistic Data Consortium (LDC) [38.56] which consists of thousands of telephone conversations between hundreds of speakers within the US. The speakers cover a distribution of age, gender, location, and native languages. The participants in a telephone call are given general topics to discuss, but the conversations are unscripted and about five minutes in duration. Participants were encouraged to make many calls to the system over several weeks and to use varied telephone instruments and locations to provide large session and handset variability in the data. Participants

Amazon EX1010 Page 786

that have a common native non-English language are also instructed to conduct conversations in their native language.

For results presented here we focus on the one and eight conversation-side train and one conversation-side test telephone speech conditions. Each conversation-side provides nominally 2.5 minutes of speech. To induce mismatch, training data for a speaker comes from a single telephone number and the test speech comes from a different telephone number. Results from other conditions can be found in [38.55, 57].

Performance is shown in terms of equal error rate (EER), minimum decision cost function (minDCF) [38.54] and detection error tradeoff (DET) curves [38.58]. The minDCF is defined as

$$\text{minDCF} = \min_{\theta} 0.1\,P_{\text{miss}}(\theta) + 0.99\,P_{\text{fa}}(\theta), \qquad (38.29)$$

where $P_{\text{miss}}(\theta)$ and $P_{\text{fa}}(\theta)$ are the probability of miss and false alarm, respectively, at the detection threshold θ.

38.5.2 Systems

Below are some speaker detection systems evaluated on the 2006 SRE that exemplify the features and classifiers detailed in this chapter. More details and other system used in the 2006 SRE can be found in [38.59].

GMM UBM
Features. Mel cepstra plus delta cepstra
Classifier. 2048 mixture GMM UBM
Compensations. RASTA and mean/variance normalization in feature domain, LFA in model domain, Z-norm, and T-norm in score domain

SVM-GLDS
Features. Mel cepstra and LPC cepstra plus delta cepstra
Classifier. SVM with GLDS kernel
Compensations. RASTA and mean/variance normalization in feature domain, NAP in model domain, T-norm in score domain

SVM-GSV
Features. Mel cepstra plus delta cepstra
Classifier. SVM with generative model sequence kernel using GMM UBM mean supervectors
Compensations. NAP in model domain, T-norm in score domain

LLR-WORD
Features. Speech recognition system word sequences from word lattices
Classifier. n-gram log-likelihood ratio
Compensations. Z-norm and T-norm in score domain

SVM-WORD
Features. Speech recognition system word sequences from word lattices
Classifier. SVM TFLOG kernel using n-gram vectors
Compensations. None

38.5.3 Results

In Fig. 38.8, we show performance for the one conversation train and one conversation test condition in terms of minDCF and EER for the above individual systems, fusion of all systems and fusion of the spectral-based systems only (GMM-UBM, SVM-GLDS, and SVM-GSV). Figure 38.9 shows the same for the eight conversation train and one conversation test condition. Results for both figures use all trials from the train/test condition (e.g., both English and non-English conversations).

These results demonstrate some general observations about the various speaker recognition systems. First, we see that the spectral based systems have comparable performance to each other and significantly better performance than systems using word-based tokens. We also see that increased training data improves performance for all systems. For word-token-based systems this gain is attributable to increased instances of sparse speaker-dependent idiolect events. The improvement in

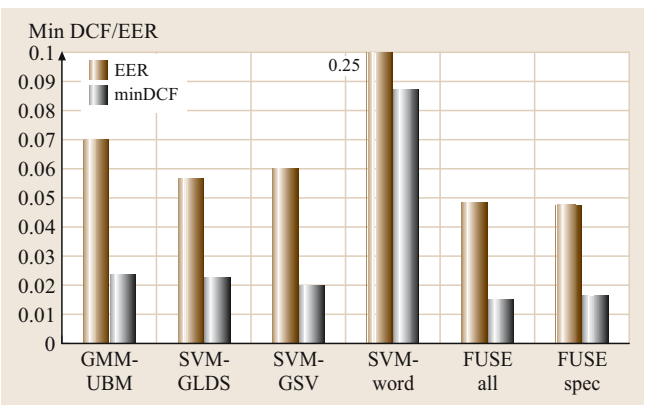


Fig. 38.8 Performance in terms of minDCF and EER for one conversation train and one conversation test for low-level, high-level, and fusion systems

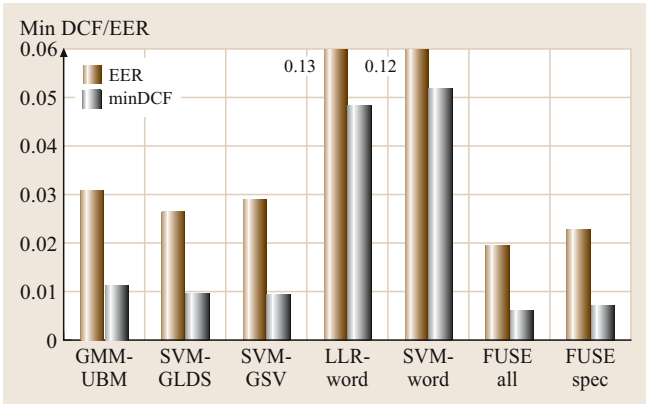


Fig. 38.9 Performance in terms of minDCF and EER for eight conversation trains and one conversation test for low-level, high-level, and fusion systems

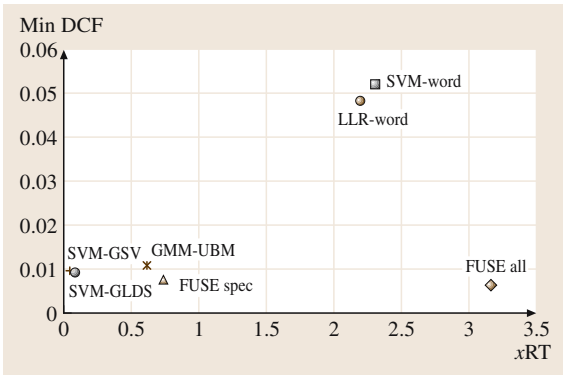


Fig. 38.10 MinDCF versus real-time factor for eight conversation trains and one conversation test for low-level, high-level, and fusion systems

spectral based systems is primarily due to observing more session variability of the speaker's speech. Lastly we observe that system fusion provides further gains in performance. This gain is seen when fusing only spectral based systems as well as when fusing spectral and word-token-based system. The extra gain from fusing in the word-token-based systems is virtually zero for the one conversation train condition and about 16% relative at EER for the eight conversation train condition.

While not shown, the results when restricting the trials to English-only conversations are only slightly better than using all trials. On average the EER is decreased by about 0.75%. An analysis of cross-lingual train/test condition indicates that spectral-based speaker recognition systems are relatively robust to the spoken language. High-level system that depend on matching the input language to the language of the speech recognizer are much more susceptible to degradations under cross-lingual conditions.

38.5.4 Computational Considerations

Although the focus of the discussion of speaker recognition systems has been performance, any practical implementation of such a system needs to account for computational complexity. A tradeoff plot of minDCF versus real-time processing factor for the above systems is shown in Fig. 38.10. The real-time processing factor is the multiple of input speech duration a system requires to produce a speaker detection score. (e.g., 2xRT means a 1 minute of speech requires 2 minutes of processing time. Real-time factors were computed using common input and CPU.) As exemplified here, since high-level systems often rely on the output of a speech recognition system, there is a significant computational cost for the incremental gains in fusing high-level systems with low-level systems. However, in applications where a speech recognition system is already being used for other purposes, reusing the output for high-level speaker recognition systems may be practical.

38.6 Summary

This chapter has presented a variety of methods for implementing automatic text-independent speaker verification systems. The speaker verification problem was cast as a general likelihood ratio detection task where features conveying speaker information from low-level spectral information to high-level word sequences are extracted and compared to models of speaker-dependent and general speaker-independent characteristics. Two of the most widely used classifiers, the generative GMM-UBM and the discriminative SVM, were presented as methods for performing the model generation and likelihood ratio comparison. To handle degradations due to channel and session variability compensation techniques in the feature, model, and score domains were discussed. Finally results from the 2006 NIST speaker recognition evaluation showed the relative performance of systems built using low- and high-level features with generative and discriminative classifiers and well as

Amazon EX1010 Page 788

fusions of these systems. Overall performance trends indicate that text-independent speaker verification on conversational telephone speech has shown significant improvement over years of research with state-of-the-art systems producing EERs below 2%. For the latest advances in the field of automatic language recognition, the reader is referred to the most recent proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), the Odyssey Speaker and Language Recognition Workshop, and the International Conference on Spoken Language Processing (Interspeech – ICSLP).

References

38.1 J. Naik, G. Doddington: Evaluation of a high performance speaker verification system for access Control, Proc. ICASSP (1987) pp. 2392–2395

38.2 A. Higgins, L. Bahler, J. Porter: Speaker verification using randomized phrase prompting, Digital Signal Process. **1**, 89–106 (1991)

38.3 J. Naik, L. Netsch, G. Doddington: Speaker verification over long distance telephone lines, Proc. ICASSP (1989) pp. 524–527

38.4 C. Schmandt, B. Arons: A conversational telephone messaging system, IEEE Trans. Consumer Electron. **30**(3), xxi–xxiv (1984)

38.5 L. Wilcox, F. Chen, D. Kimber, V. Balasubramanian: Segmentation of speech using speaker identification, Proc. ICASSP (1994) pp. I.161–I.164

38.6 B.M. Arons: *Interactively Skimming Recorded Speech*, Ph.D. Thesis (MIT Press, Cambridge 1994)

38.7 G. Doddington: Speaker recognition – identifying people by their voices, Proc. IEEE **73**(11), 1651–1664 (1985)

38.8 R.B. Dunn, D.A. Reynolds, T.F. Quatieri: Approaches to speaker detection and tracking in multi-speaker audio, Digital Signal Process. **10**(1), 93–112 (2000)

38.9 K. Li, J. Porter: Normalizations and selection of speech segments for speaker recognition scoring, Proc. ICASSP (1988) pp. 595–598

38.10 D.A. Reynolds, T.F. Quatieri, R.B. Dunn: Speaker verification using adapted Gaussian mixture models, Digital Signal Process. **10**(1), 19–41 (2000)

38.11 R. Auckenthaler, M. Carey, H. Lloyd-Thomas: Score Normalization for Text-Independent Speaker Verification Systems, Digital Signal Process. **10**, 42–54 (2000)

38.12 W.M. Campbell, D.A. Reynolds, J.P. Campbell, K. Brady: Estimating and evaluating confidence for forensic speaker recognition, Proc. ICASSP (2005) pp. 18–23

38.13 D. Reynolds, W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, J. Godfrey, D. Jones, B. Xiang: The SuperSID project: Exploiting high-level information for high-accuracy speaker recognition, Proc. ICASSP (2003) pp. 784–787

38.14 F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, D.A. Reynolds: A tutorial on text-independent speaker verification, EURASIP J. Appl. Signal Process. **4**, 430–451 (2004)

38.15 F. Soong, A. Rosenberg: On the use of instantaneous and transitional spectral information in speaker recognition, Proc. ICASSP (1986) pp. 877–880

38.16 H. Hermansky, N. Morgan, A. Bayya, P. Kohn: RASTA-PLP speech analysis technique, Proc. ICASSP (1992) pp. I.121–I.124

38.17 D.A. Reynolds: Channel robust speaker verification via feature mapping, Proc. ICASSP (2003) pp. I.53–I.56

38.18 L. Nguyen, S. Matsoukas, J. Davenport, F. Kubala, R. Schwartz, J. Makhoul: Progress in transcription of broadcast news using Byblos, Speech Commun. **38**(1-2), 213–230 (2002)

38.19 J.P. Campbell, D.A. Reynolds, R.B. Dunn: Fusing high- and low-level features for speaker recognition, Proc. European Conf. Speech Commun. Technol. (2003) pp. 2665–2668

38.20 W.M. Campbell, J.P. Campbell, D.A. Reynolds, D.A. Jones, T.R. Leek: High-level speaker verification with support vector machines, Proc. ICASSP (2004) pp. I.73–I.76

38.21 G. Doddington: Speaker recognition based on idiolectal differences between speakers, Proc. European Conf. Speech Commun. Technol. (2001)

38.22 A. Dempster, N. Laird, D. Rubin: Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. **39**, 1–38 (1977)

38.23 D.A. Reynolds: Speaker identification and verification using gaussian mixture speaker models, Speech Commun. **17**(1-2), 91–108 (1995)

38.24 D.A. Reynolds: Automatic speaker recognition using Gaussian mixture speaker models, Lincoln Lab. J. **8**(2), 173–192 (1995)

38.25 A.E. Rosenberg, J. DeLong, C.H. Lee, B.H. Juang, F.K. Soong: The use of cohort normalized scores for speaker verification, Int. Conf. Speech Lang. Process. (1992) pp. 599–602

38.26 M. Carey, E. Parris, J. Bridle: A speaker verification system using alphanets, Proc. ICASSP (1991) pp. 397–400

38.27 D.A. Reynolds: Comparison of background normalization methods for text-independent speaker verification, Proc. European Conf. Speech Commun. Technol. (1997) pp. 963–967

Amazon EX1010 Page 789

38.28 J.L. Gauvain, C.-H. Lee: Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains, IEEE Trans. Speech Audio Process. **2**(2), 291–298 (1994)

38.29 S. Vuuren: *Speaker Verification in a Time-Feature Space*, Ph.D. Thesis (OGI, Beaverton 1999)

38.30 D.A. Reynolds, T.F. Quatieri, R. Dunn: Speaker verification using adapted Gaussian mixture models, Digital Signal Process. **10**(1–3), 19–41 (2000)

38.31 R. Teunen, B. Shahshahani, L. Heck: A model-based transformational approach to robust speaker recognition, Proc. Int. Conf. Spoken Lang. Process. (2000)

38.32 P. Kenny, G. Boulianne, P. Dumouchel: Eigenvoice modeling with sparse training data, IEEE Trans. Speech Audio Process. **13**(3), 345–354 (2005)

38.33 R. Vogt, B. Baker, S. Sriharan: Modelling session variability in text-independent speaker verification, Proc. Interspeech (2005) pp. 3117–3120

38.34 C. Vair, D. Colibro, F. Castaldo, E. Dalmasso, P. Laface: Channel factors compensation in model and feature domain for speaker recognition, Proc. Odyssey Speaker and Language Workshop (2006)

38.35 N. Cristianini, J. Shawe-Taylor: *Support Vector Machines* (Cambridge Univ. Press, Cambridge 2000)

38.36 M. Schmidt, H. Gish: Speaker identification and verification using Gaussian mixture speaker models, Proc. ICASSP (1996) pp. 105–108

38.37 V. Wan, W.M. Campbell: Support vector machines for verification and identification, neural networks for signal processing X, Proc. 2000 IEEE Signal Process. Workshop (2000) pp. 775–784

38.38 A. Ganapathiraju, J. Picone: Hybrid SVM/HMM architectures for speech recognition, Speech Transcription Workshop (2000)

38.39 J.C. Platt: Probabilities for SV machines. In: *Advances in Large Margin Classifiers*, ed. by A.J. Smola, P.L. Bartlett, B. Schölkopf, D. Schurmans (MIT Press, Cambridge 2000) pp. 61–74

38.40 W.M. Campbell: Generalized linear discriminant sequence kernels for speaker recognition, Proc. ICASSP (2002) pp. 161–164

38.41 S. Fine, J. Navrátil, R.A. Gopinath: A hybrid GMM/SVM approach to speaker recognition, Proc. ICASSP (2001)

38.42 V. Wan, S. Renals: SVMSVM: support vector machine speaker verification methodology, Proc. ICASSP (2003) pp. 221–224

38.43 A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, A. Venkataraman: MLLR transforms as features in speaker recognition, Proc. European Conf. Speech Commun. Technol. (2005) pp. 2425–2428

38.44 W.M. Campbell, D.E. Sturim, D.A. Reynolds: Support vector machines using GMM supervectors for speaker verification, IEEE Sign. Process. Lett. **13**(5), 308–311 (2006)

38.45 R. Collobert, S. Bengio: SVMTorch: Support vector machines for large-scale regression problems, J. Machine Learn. Res. **1**, 143–160 (2001)

38.46 J. Louradour, K. Daoudi, F. Bach: SVM speaker verification using an incomplete Cholesky decomposition sequence kernel, IEEE 2006 Odyssey: The Speaker and Language Recognition Workshop (2006)

38.47 P.J. Moreno, P. Ho, N. Vasconcelos: A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. In: *Advances in Neural Information Processing 16*, ed. by S. Thrun, L.K. Saul, B. Schölkopf (MIT Press, Cambridge 2004)

38.48 T.S. Jaakkola, D. Haussler: Exploiting generative models in discriminative classifiers. In: *Advances in Neural Information Processing 11*, ed. by M.S. Kearns, S.A. Solla, D.A. Cohn (MIT Press, Cambridge 1998) pp. 487–493

38.49 A. Solomonoff, W.M. Campbell, I. Boardman: Advances in channel compensation for SVM speaker recognition, Proc. ICASSP (2005)

38.50 W.D. Andrews, M.A. Kohler, J.P. Campbell, J.J. Godfrey, J. Hernández-Cordero: Gender-dependent phonetic refraction for speaker recognition, Proc. ICASSP (2002) pp. I.149–I.153

38.51 T. Joachims: *Learning to Classify Text Using Support Vector Machines* (Kluwer Academic, Dordrecht 2002)

38.52 W.M. Campbell, D.A. Reynolds, J.P. Campbell: Fusing discriminative and generative methods for speaker recogntion: experiments on Switchboard and NFI/TNO field data, Proc. Odyssey Speaker and Language Workshop (2004) pp. 41–44

38.53 L. Kukolich, R. Lippman: *LNKnet User's Guide* Manual and software available online at http://www.ll.mit.edu/IST/lnknet (2004)

38.54 *The 2006 NIST Speaker Recognition Evaluation Plan*http://www.nist.gov/speech/tests/spk/2006/sre-06_evalplan-v9.pdf (2006)

38.55 M.A. Pryzbocki, A.F. Martin, A.N. Le: NIST speaker recognition evaluation chronicles part 2, Proc. Odyssey Speaker and Language Workshop (2006)

38.56 J.P. Campbell, H. Nakasone, C. Cieri, D. Miller, K. Walker, A.F. Martin, M.A. Pryzbocki: The MMSR bilingual and crosschannel corpora for speaker recognition research and evaluation, Proc. Odyssey Speaker and Language Workshop (2004) pp. 29–32

38.57 D.E. Sturim, W.M. Campbell, D.A. Reynolds, R.B. Dunn, T.F. Quatieri: Robust speaker recognition with cross-channel data: MIT-LL results on the 2006 NIST SRE auxiliary microphone task, Proc. ICASSP (2007)

Amazon EX1010 Page 790

38.58 A. Martin, G. Doddington, T. Kamm, M. Ordowski, M. Pryzbocki: The DET curve in assessment of detection task performance, Proc. European Conf. Speech Commun. Technol. (1997) pp. 1895–1898

38.59 W.M. Campbell, D.E. Sturim, W. Shen, D.A. Reynolds, J. Navratil: The MIT-LL/IBM 2006 speaker recognition system: High-performance reduced-complexity recognition, Proc. ICASSP (2007)

Amazon EX1010 Page 791

Part G Language

Part G Language Recognition

Ed. by C.-H. Lee

39 Principles of Spoken Language Recognition
C.-H. Lee, Atlanta, USA

40 Spoken Language Characterization
M. P. Harper, College Park, USA
M. Maxwell, College Park, USA

41 Automatic Language Recognition Via Spectral and Token Based Approaches
D. A. Reynolds, Lexington, USA
W. M. Campbell, Lexington, USA
W. Shen, Lexington, USA
E. Singer, Lexington, USA

42 Vector-Based Spoken Language Classification
H. Li, Singapore
B. Ma, Singapore
C.-H. Lee, Atlanta, USA

Amazon EX1010 Page 792

39. Principles of Spoken Language Recognition

C.-H. Lee

In this introductory chapter to Part G of this Handbook on spoken language recognition, we provide a brief overview of the principles of state-of-the-art language recognition approaches, and a general discriminative training framework to improve the performance and robustness of language recognition systems. It is followed by three chapters. The first of these addresses issues related to spoken language characterization in which knowledge sources can be utilized to distinguish one language from another. The second chapter deals with language identification based on phone recognition followed by language modeling using either spectral or token-based approaches. The third chapter presents vector-space characterization approaches to converting speech utterances into spoken document vectors for modeling and classification. With recent progress in speech processing, machine learning, and text categorization, we expect significant technology advances in spoken language recognition in the years to come.

This chapter is organized as follows. Section 39.2 briefly describes the principle of spoken language recognition. Sections 39.3 and 39.4 formulate the popular parallel phone recognition followed by language modeling (P-PRLM) and vector-space characterization (VSC) approaches to spoken language identification. In Sect. 39.5 we extend these formulations to spoken language verification. Finally a general discriminative training framework for non-support vector machine (non-SVM) classifiers is presented in Sect. 39.6, followed by a brief summary in Sect. 39.7.

39.1 **Spoken Language** 785
39.2 **Language Recognition Principles** 786
39.3 **Phone Recognition Followed by Language Modeling (PRLM)** 788
39.4 **Vector-Space Characterization (VSC)** 789
39.5 **Spoken Language Verification** 790
39.6 **Discriminative Classifier Design** 791
39.7 **Summary** 793
References 793

39.1 Spoken Language

A spoken language can be identified using information from a number of sources. When human beings are constantly exposed to a language without being given any linguistic knowledge, they learn to discriminate subtle differences by perceiving speech cues in the specific languages. Nonetheless lexical knowledge is usually the main source of information for human listeners, especially when the languages to be distinguished are similar. For automatic spoken language recognition by machines, state-of-the-art systems often consider unknown utterances as a concatenation of signal patterns, and probabilistic modeling and classification techniques are then adopted to characterize these patterns and their corresponding language identities. Although lexical models are not explicitly utilized, models of phones in a subset of languages are often used to decode speech for further processing. Most human listeners are usually constrained by their surroundings from effectively learning a large number of languages. On the other hand machines are equipped to take in virtually an unlimited amount of speech and text. It is interesting to note that advanced machines therefore have the potential to surpass human performance in spoken language recognition.

With a coordinated effort between government funding agencies and technical communities for data collection and benchmark performance evaluation, the current availability of large collections of speech examples from a selected set of languages, easy access to fast and affordable computing equipment, and recent

Amazon EX1010 Page 793

advances in speech modeling and machine learning, many new algorithms have recently been explored in this area. We are now witnessing rapid progress in the area of spoken language modeling and recognition technologies.

In this introductory chapter to Part G of this Handbook on spoken language recognition, we provide a brief discussion of the principles of state-of-the-art language recognition approaches. It is then followed by three more chapters. The first addresses issues of spoken language characterization, in which knowledge sources that can be utilized to distinguish one language from another are presented. Contrasts between human and machine language recognition are made to highlight that language cues perceived by human listeners can be incorporated into current technologies to improve machine capabilities. It is also noted that there are a few orders of magnitude more languages and dialects in existence than current systems are equipped to handle. Many of these languages are rare and extensive data collection to construct recognizers for these languages can be prohibitive. Some of today's prevailing technologies may have to be modified to alleviate these limitations. New paradigms may also have to be established to examine the language recognition problem from a completely different perspective.

The two other chapters in this part of the Handbook describe prevailing trends in current language recognition system design. The first attempts to divide language recognition approaches into two broad categories. The first method is spectral-based methods, in which each spoken utterance or segment is represented by a sequence of feature vectors that are often short-time spectral representation of speech frames. This is similar to what is typically done in the front-end feature-extraction module of an automatic speech recognition (ASR) [39.1] system. These spectral vectors are assumed to be generated from different source languages, and therefore have different characteristics. A collection of models, one for each language to be considered, can then be built with labeled spectral vector examples collected from all the languages. These models can be probabilistic, such as a Gaussian mixture model (GMM) [39.2], or vector based, such as a support vector machine (SVM) [39.3]. It can be seen that this spectral-based framework is purely acoustic, while no linguistic information, such as phones or words, is used. The second method is called the token-based approach, in which an intermediate set of speech units, also referred to as tokens, is used to represent speech. An utterance is first decoded and segmented into a sequence of such tokens. These token streams are then used to extract features, designing classifiers, and performing spoken language classification.

The last chapter in this part of the Handbook represents a new vector-space characterization (VSC) [39.4] approach to language recognition in which an utterance is considered as a spoken document. A term-document matrix representation of all utterances in a training set can then be established for indexing and retrieval purposes. The terms here can refer to acoustic words (AWs), which are sequences of acoustic letters (alphabets), or fundamental sound units, and the vector elements are often co-occurrence statistics describing the frequencies of AWs in each spoken utterance. Language recognition with an unknown spoken query can then be treated as a document-retrieval problem similar to that which is commonly faced in the information retrieval (IR) [39.5] community. It can also be cast as a text categorization (TC) [39.6, 7] problem in which all training documents are used to build text categorizers, or topic classifiers, one for each language to be considered. In so doing language recognition can be accomplished by comparing the unknown spoken query vector with each of the language classifiers to make a recognition decision. This vector-based modeling framework allows one to represent speech utterances with a very high-dimension document vector, sometime in the tens of thousands, so that many of the feature extraction and classifier learning algorithms currently available in the IR and TC literatures can be easily adopted for spoken language recognition. New language cues can also be exploited and incorporated into the spoken document vectors, along the same lines as a recently proposed automatic speech attribute transcription (ASAT) [39.8] paradigm for ASR, to improve system performance.

39.2 Language Recognition Principles

Automatic spoken language identification (LID) is a process of determining the language spoken in a speech sample. LID technology is needed in many applications such as multilingual conversational systems [39.9], spoken language translation [39.10], multilingual speech recognition [39.11], and spoken document re-

trieval [39.12]. It is also a topic of great importance in the areas of intelligence and security, where the language identities of recorded messages and archived materials need to be established before any information can be extracted from them. In the past few decades, researchers have explored many speech and language knowledge sources, including articulatory parameters [39.13], acoustic features [39.14], prosody [39.15,16], phonotactic [39.17,18], and lexical knowledge [39.19]. Recently, investigators have reported promising results using *shifted-delta-cepstral* acoustic features [39.20]. From the perspective of human language recognition, humans constantly exposed to a language without being given any linguistic knowledge learn to determine the identity of the language by detecting some distinct speech cues in the specific language. For example, an English-speaking listener can often appreciate the tonal nature of Mandarin Chinese. It is also noted in human perceptual experiments that listeners with a multilingual background often perform better than monolingual listeners in identifying unfamiliar languages [39.21]. These reasons motivate us to explore useful speech attributes in languages, along the same lines as a recently proposed ASAT paradigm for automatic speech recognition (ASR) [39.8]. Other useful speech and language features, such as prosodic and syllabic content, can also be incorporated into this feature vector.

In a typical pattern recognition setting, one must evaluate the a posterior probability, $P(\lambda_l|X)$, of the l-th language to be considered with a model λ_l given an unknown utterance X. To make a decision, the language that yields the maximum $P(\lambda_l|X)$ is usually identified as the target language. Many algorithms developed in automatic speech and speaker recognition [39.22, 23] can be adopted and extended to language recognition. Recent advances in acoustic and language modeling [39.24] have also contributed to progress in language recognition.

Taking advantage of recent advances in continuous speech recognition with hidden Markov models HMMs [39.25], probabilistic approaches [39.26–29] have been developed by exploiting techniques in acoustic phone modeling and n-gram language modeling. These characterize a spoken language using probability distributions of spectral features in the form of linguistically defined symbols, such as phones and syllable-like units [39.17, 30], where phone models are used to decode speech utterances into sequences of such fundamental symbols. An interpolated phone-based n-gram language model is then constructed for each language, and to derive phonotactic scores. Such an example is parallel phone recognition followed by language modeling (P-PRLM) [39.18], which uses multiple single-language phone recognizers as a front end and language-dependent language models as the back end. The phonotactic approach has been shown to provide superior performance on NIST language recognition evaluation (LRE) tasks [39.27]. It is generally agreed that the fusion of multiple phonotactic features improves performance. For example, the P-PRLM approach derives multiple sets of phonotactic features. Others have found that multiresolution phonotactic analysis, such as phone unigram, bigram, and trigram approaches, complement each other [39.17, 31, 32].

A key question here is whether a phone definition is needed to represent fundamental speech units. If we can *tokenize* speech with a manageable set of *spoken letters* and develop models to decode spoken utterances into sequences of these acoustic units, it is clear that the statistics of these spoken letters and their co-occurrences can be used to discriminate one spoken language from another. Although common sounds are shared considerably across spoken languages, the statistics of these sounds, such as phone n-grams, can differ considerably from one language to another. An interesting generalization through acoustic units is to represent any spoken utterance (also referred to as a spoken document when presented in the form of *spoken letters*) with a high-dimensional feature vector, where each element carries sound co-occurrence statistics. This is similar to a latent semantic indexing (LSI) vector representation [39.33] of text documents, which is commonly used in information retrieval (IR) systems [39.5]. Such statistics are considered to be salient features for indexing and retrieving documents. We call this paradigm vector-space characterization of speech [39.4].

To develop a universal set of fundamental speech units to cover the acoustic characterization of all spoken languages we relax the notion of language-specific phonetic definitions. To relate this universal unit set to language discrimination, it is well known that the entropy of English can be effectively reduced when high-order statistics of letters are computed [39.34]. For example, of the 26 English letters plus the space character, a few of them, such as n, s, and t, occur more often in text than others, such as x and q. By incorporating this first-order statistics, or unigram, the entropy of English text can easily be reduced from 4.76 to 4.03 bits. When letter bigrams and trigrams are added, the entropy is further reduced. This set of statistics can be used to discriminate among languages already decoded, even

if no extra dictionary information is explicitly utilized. The same notion can be extended to spoken language identification. This is sometimes referred to as a bag-of-sound representation [39.26], in analogous to the bag-of-words representation of text documents used in IR. Just as in P-PRLM multiple bags of sound can also be used [39.35].

A model corresponding to the acoustic letters approach mentioned above is called an acoustic segment model (ASMs) [39.36] and can be used to decode spoken utterances into strings of such units. HMMs [39.25] are often used to model the collection of ASMs, which can be established in a bottom-up unsupervised manner. ASMs have been used to construct an acoustic lexicon for isolated word recognition with high accuracy [39.36]. Acoustic words (AWs) can now be formed by grouping adjacent acoustic letters, and serve as a basis to build feature vectors for spoken documents and build language classifiers. Therefore, automatic language classification can be formulated as a text categorization (TC) [39.6] problem based on LSI-derived feature vectors and discriminative classifier design [39.37]. Many existing feature representation and machine learning techniques widely available in the IR and TC literature can now be adopted.

39.3 Phone Recognition Followed by Language Modeling (PRLM)

Consider a speech utterance X to be represented by a sequence vectors of length τ, $O = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_i, \ldots, \boldsymbol{o}_\tau\}$, in which $\boldsymbol{o}_i$ is a feature vector extracted from X at time i. We can express the a posteriori probability of language l using Bayes theorem, as follows:

$$P(l|O) = P(O|l)\frac{P(l)}{P(O)}\,, \tag{39.1}$$

where $P(l)$ and $P(O)$ are prior probabilities of observing language l and vector sequence O, respectively. Without loss of generality we can assume $P(l)$ to be equal for all languages. The language-independent term, $P(O)$, usually does not affect our decision rules. They will be dropped hereafter in this chapter. Now we can apply the *maximum a posteriori* (MAP) decision rule for LID as:

$$\hat{l} = \arg\max_l P(l|O) = \arg\max_l P(O|\lambda_l)\,, \tag{39.2}$$

where λ_l is an model for the l-th language. Here the model λ_l can be any reasonable characterization of the feature vectors. A straightforward choice is to consider all such vectors to be generated from a language-specific density, and a GMM can then be used to model the source. This is exactly the same formulation as in GMM-based text-independent speaker identification [39.38]. This approach is purely frame based, and no segmental information is used. SVMs [39.39] have been used to design both high-performance GMM-based speaker and language recognition systems [39.2, 3, 40].

If we fold in some fundamental speech units, and associate a given utterance with a sequence of such units then more-detailed models can be incorporated. For example, if we have a set of phone HMMs λ_l^{AM} and a set of phone language models λ_l^{LM} trained with speech data with implied phone sequence labels from language l, then we have

$$\hat{l} = \arg\max_l \sum_{\forall q} P\left(O|q, \lambda_l^{\mathrm{AM}}\right) P\left(q|\lambda_l^{\mathrm{LM}}\right)\,, \tag{39.3}$$

where q is a candidate phone sequence. In many cases the sum in (39.3) is approximated by finding the most dominant phone sequence $\hat{q}_l$ with the l-th phone model λ_l^{AM} using Viterbi decoding,

$$\hat{q}_l = \arg\max_{\forall q} P\left(O|q, \lambda_l^{\mathrm{AM}}\right)\,, \tag{39.4}$$

and solving for the following:

$$\hat{l} \approx \arg\max_l \left[P\left(O|\hat{q}_l, \lambda_l^{\mathrm{AM}}\right) P\left(\hat{q}_l|\lambda_l^{\mathrm{LM}}\right)\right]. \tag{39.5}$$

This is known as the phone recognition followed by language modeling (PRLM) approach to LID. Since obtaining high-accuracy phone and language models for each individual language under consideration is not always possible, one can consider training a single set of phone models to cover all languages and use it to decode all spoken utterances. Although the decoding performance for sounds not properly modeled by this model set is usually not good, we can still get reasonable performance by dropping the dependency on l for the acoustic score term and the phone sequence. Now (39.5) is solved by:

$$\hat{l} \approx \arg\max_l P\left(\hat{q}|\lambda_l^{\mathrm{LM}}\right)\,. \tag{39.6}$$

It is interesting to note that phone models are used only for decoding, and not for scoring when deter-

mining the identified language. We can also expand the phone model collection to include multiple sets of models, each trained by speech examples from a subset of languages. Now the set of F phone models, $\{\lambda_1^{\mathrm{AM}}, \ldots, \lambda_f^{\mathrm{AM}}, \ldots, \lambda_F^{\mathrm{AM}}\}$, are all used to decode a given spoken utterance, resulting in a set of F phone sequences, $Q = \{q_1, \ldots, q_f, \ldots, q_F\}$. Furthermore for the l-th language, we can train F sets of phone language models, $\{\lambda_{l,1}^{\mathrm{LM}}, \ldots, \lambda_{l,f}^{\mathrm{LM}}, \ldots, \lambda_{l,F}^{\mathrm{LM}}\}$, to compute F language-specific scores, $\{P(q_f|\lambda_{l,f}^{\mathrm{LM}}), f = 1, \ldots, F\}$. These scores can then be combined to make language classification decisions. This the basic idea behind parallel PRLM (P-PRLM) [39.18], which is by far the most successful approach to LID.

39.4 Vector-Space Characterization (VSC)

Vector-space modeling has become a standard tool in IR systems since its introduction several decades ago [39.5]. It uses a vector to represent a text document or a query. It has also been applied to text categorization [39.41] in which training vectors are used to design a collection of topic classifiers. A vector-based TC approach to LID has been proposed recently [39.37].

Suppose that an utterance X, represented by a sequence of speech feature vectors O, is decoded or tokenized, into a *spoken document*, $d(X)$, consisting of a series of I acoustic units, $d(X) = \{t_1 \ldots, t_i \ldots, t_I\}$, where each unit is drawn from a universal inventory, $U = \{u_1, \ldots, u_j, \ldots u_J\}$, of J acoustic letters shared by all the spoken languages to be considered, such that $t_i \in U$. We are then able to establish a collection of acoustic words by grouping units occurring consecutively to obtain a vocabulary of M distinct words, $W = \{w_1, \ldots, w_m, \ldots w_M\}$, such that each w_m can be a single-letter word like (u_j), a double-letter word like $(u_j u_k)$, a triple-letter word like $(u_j u_k u_l)$, and so on. Usually the vocabulary size, M, is equal to the total number of n-gram patterns needed to form words, e.g., $M = J + J \times J + J \times J \times J$ if acoustic words up to three tokens in length are considered valid. Next we can use some form of function $f(w_m)$, such as LSI [39.33], to evaluate the significance of having the word w_m in the document, $d(X)$. We are now ready to establish an M-dimension feature vector, $\boldsymbol{v} = [f(w_1), \ldots, f(w_m), \ldots f(w_M)]^{\mathrm{T}}$ in which $\boldsymbol{x}^{\mathrm{T}}$ denotes the transpose of the vector $\boldsymbol{x}$ for each spoken document.

It is clear that we need a number of fundamental units sufficient to cover the acoustic variation in the sound space. However a large J will result in a feature vector with a very high dimension if we would like to cover as many unit combinations when forming acoustic words. For example, with a moderate value of $J = 256$, we have $M = 65\,792$ even when we only consider words less than or equal to two letters. This is already a very large dimensionality not commonly utilized in speech and language processing algorithms. Finally, a vector-based classifier evaluates a goodness of fit, or score function $S_l(\boldsymbol{v}) = S(\boldsymbol{v}|\lambda_l)$ between a given vector $\boldsymbol{v}$ and a model of the l-th spoken language λ_l to make a decision. Any vector-based classifier can be used to design spoken language identification and verification systems. A goodness of fit, such as an inner product using a linear discriminant function (LDF) [39.42], can be used to evaluate vector-based scores:

$$S(\boldsymbol{v}|\lambda_l) \propto \gamma_l^{\mathrm{T}} \cdot \boldsymbol{v}\,, \tag{39.7}$$

where γ_l is a language-dependent weight vector of equal dimensionality to $\boldsymbol{v}$, with each attribute representing the contribution of its individual n-gram probability to the overall language score. The spoken document vector, $\boldsymbol{v}$ in (39.7), is high dimensional in nature when patterns up to triplets are included. When multiple inventories of models, like in P-PRLM, are used to produce a collection of F document vectors, $\{\boldsymbol{v}_f, f = 1, \ldots, F\}$, a large composite document vector, or supervector, $\boldsymbol{v} = [\boldsymbol{v}_1^{\mathrm{T}}, \ldots, \boldsymbol{v}_f^{\mathrm{T}}, \ldots, \boldsymbol{v}_F^{\mathrm{T}}]^{\mathrm{T}}$ can be established by stacking these F vectors. It has been shown that such supervectors have more discrimination power than single document vectors for language recognition [39.35].

Term weighting [39.43] is widely used to render the value of the attributes in a document vector by taking into account the frequency of occurrence of each attribute. It is interesting to note that attribute patterns that occur often in a few documents but not as often in others give high indexing power to these documents. On the other hand, patterns that occur very often in all documents possess little indexing power. This desirable property leads to a number of term weighting schemes, such as *tf-idf* (term-frequency-inverse document frequency) [39.44] and LSI [39.33], which are common for information retrieval [39.5], natural language call routing [39.45], and text categorization [39.41]. VSC is motivated by the same ideas, which aim to derive weights that dis-

criminate between languages using high-dimensional salient-feature vectors.

After representing a spoken document as a vector of statistics of AWs, LID becomes a vector-based classification problem. Many classifier designs in the machine learning literature for high-dimensional vector classification are readily available. For example conventional techniques, such as SVM [39.39], classification and regression tree (CART) [39.46] and artificial neural networks (ANN) [39.47], can be used to train vector-based classifiers, such that LID can be solved in the form:

$$\hat{l} = \arg\max_{l} S(\boldsymbol{v}|\lambda_l) \,. \tag{39.8}$$

We can also consider an indirect vectorization mechanism based on the scores computed for all the classes of interest, obtained from an ensemble of classifiers. Since these scores characterize the distribution of the outputs of a variety of classifiers, they do provider significant discriminatory power among classes, and can be grouped together to form *score supervectors* describing the overall behavior of a score distribution over different classifiers for all competing classes. One example of such a supervector is to concatenate HMM state scores from all the competing models to form an overall score vector. This approach has been shown to have good discriminatory power in isolated letter recognition [39.48, 49]. Since these score supervectors are usually obtained from a finite set of ensemble classifiers, their dimensionality is often much lower than that of the spoken document vectors discussed above. These supervectors are more amendable to treatment using conventional probabilistic modeling frameworks.

39.5 Spoken Language Verification

So far we have only discussed spoken language identification. Another recognition problem of interest is spoken language verification, which can be cast as a statistical hypothesis testing problem, i. e., testing a null hypothesis H_0 that an utterance O is from a claimed language against an alternative hypothesis H_1 that O is not from that language. Many issues we will discuss in this section have similarities to those commonly addressed in speaker and utterance verification [39.50]. According to the Neyman–Pearson lemma [39.51], an optimal test can be formulated as: given a test speech O, accept H_0 if

$$\frac{P(O|H_0)}{P(O|H_1)} > r_{\text{th}} \,, \tag{39.9}$$

where $P(O|H_0)$ and $P(O|H_1)$ are the probability distributions for the null and alternative hypotheses, respectively. The constant r_{th} is a verification threshold. The test in (39.9) is known as a probability ratio test. The ratio $P(O|H_0)/P(O|H_1)$ is called a *probability ratio statistic*. The threshold r_{th} is referred to as the *critical value*; the region $A = \{O : [P(O|H_0)/P(O|H_1)] > r_{\text{th}}\}$ is called the acceptance region, and the region $\overline{A}$, containing points not in A, is called the critical region of the test.

There are two verification decisions to be made, namely rejection and acceptance of the null hypothesis. Correspondingly there are two types of errors, namely false rejection (type I) and false acceptance (type II). The power of a test is defined as the probability of correct rejection, i. e., rejecting H_0 when it should be rejected, which is one minus the maximum of the probability of a type II error. The level of significance of a test is defined as the maximum of the probability of type I errors. The level of significance is computed based on the distribution of the likelihood test statistic and the choice of the threshold.

To test some simple hypotheses under a few regularity conditions, a generalization of the Neyman–Pearson lemma shows that a likelihood-ratio test (LRT) is the *most powerful test* (smallest type II error test) for a given level of significance (type I error) if $P(O|H_0)$ and $P(O|H_1)$ are known exactly. For testing more-complex composite hypotheses, an optimal test is usually hard to come by. Furthermore in most practical verification applications we do not have access to complete knowledge of H_0 and H_1, and therefore optimal tests cannot be designed. In these cases we can attempt to evaluate $P(O|H_0)$ and $P(O|H_1)$ based on some assumed forms for their distributions, such as those discussed in Sect. 39.2, in which case a probability ratio test can still be used.

Even in cases where a probabilistic characterization is not easy to determine, we can still compute scores such as $S(O|H_0)$ and $S(O|H_1)$, so that the VSC models presented in Sect. 39.3 are equally applicable to the design of spoken language verification systems. Even with probabilistic modeling we can also evaluate scores of the form $S(O|H_i) = \log P(O|H_i)$ so we can use any score to compute a generalized log likelihood ratio (GLLR) test

statistic, which can be considered as a distance measure:

$$d(O, \Lambda) = -S(O|\lambda_{H_0}) + S(O|\lambda_{H_1}) , \quad (39.10)$$

where $\Lambda = (\lambda_{H_0}, \lambda_{H_1})$ is the collection of all models. We then accept H_0 if $d(O, \Lambda) < \tau_{\text{th}}$. Here τ_{th} is another form of the verification threshold. This formulation can be applied directly to two-class verification problems. However the alternative hypothesis H_1 is often difficult to model. For example if H_0 is defined for a claimed language l, H_1 is often a composite hypothesis representing all other competing languages other than l. There are often more negative and diverse samples to train the *impostor* model λ_{H_1} than positive and consistent samples to train the target model λ_{H_0}.

One way to alleviate this difficulty is to train one model for each of the languages of interest to obtain a collection of models $\Lambda = (\lambda_1, \ldots, \lambda_L)$, one for each language. Now, to verify whether an unknown utterance O is generated from a claimed language l, the target score can be computed as $S_l(O|H_0) = S(O|\lambda_l)$. Furthermore we can assume that $S_l(O|H_1)$ is a function of all the competing language scores other than $S(O|\lambda_l)$. One approach is to assume that $S_l(O|H_1)$ is evaluated as an antidiscriminant function commonly used in discriminative training (DT) in ASR [39.52,53] to compute a geometric average of all competing scores as

$$S_l(O|H_1) = \log \left\{ \frac{1}{L-1} \sum_{i \neq l} \exp[\eta S(O|\lambda_i)] \right\}^{\frac{1}{\eta}} , \quad (39.11)$$

where η is a positive constant. It is noted that the right-hand side of (39.11) is the L_η norm in real analysis, and that it converges to $\max_i S(O|\lambda_i)$ as $\eta \to \infty$. This score has been used in multicategory (MC) text categorization [39.37], and it was shown that MC TC outperforms binary TC, especially in cases when there are very few positive examples, sometimes only one sample, to train a topic classifier. The same idea was also applied to the design of language verification systems in the 2005 NIST language recognition evaluation [39.42]. Part of the successes in [39.37] and [39.42] can be attributed to discriminative classifier learning, which will be discussed in Sect. 39.6.

By now it is clear that the modeling and classifier design techniques discussed in Sects. 39.3 and 39.4 for language identification can be extended to language verification as well. One remaining key issue is the selection of verification thresholds, which is critical in designing high-performance spoken language verification systems for real-world applications. They are often determined empirically according to the specific application requirements. Since the distance measure in (39.10) also characterizes a separation between the target and competing models [39.54], we can plot histograms of $d_l(O, \Lambda)$ over positive and negative training samples, respectively. The size of the overlap region of the two curves is often a good indicator for predicting type I and type II errors. The verification thresholds can then be determined by selecting a value in the overlap region to balance the false-rejection and false-acceptance errors. One good illustration was demonstrated in the recent 2005 NIST LRE [39.42].

39.6 Discriminative Classifier Design

We have now briefly addressed most of the research issues in designing language recognition systems. More detail can be found in the last two chapters of this Part G. Next we will discuss an important methodology for discriminative training that has created a lot of enthusiasm in the fields of speech recognition [39.52,53], utterance verification [39.55], and text categorization [39.41], but not yet been fully utilized in the language recognition community. So far SVM is still the dominating discriminative classifier design approach to LID. However the techniques used in SVM training are not easily extended to other popular classifiers, such as GMM, HMM, ANN, LDF, and CART. In the following we describe a general framework to formulate a broad family of DT algorithms for any classifier based on any performance metric.

Consider a set of training patterns $\boldsymbol{X} = \{X_i, 1 \leq i \leq N\}$, with N being the total number of training tokens, and O_i, an observation vector sequence associated with X_i. Each token belongs to one of L classes, C_l, $1 \leq l \leq L$. The goal of pattern classification is to use the labeled set X to design a decision rule based on a set of parameters $\Lambda = \{\lambda_l, l = 1, ..., L\}$ such that the classification error is minimized. The optimal classification approach is the Bayes decision rule if the a priori probabilities $P(C_l)$ and the class-conditional probability $P(O|C_l)$ are known. Unfortunately, in pattern recognition applications we rarely have complete knowledge of the probabilistic structure of the problem. We only have some vague knowledge about the distribution and are given a finite number of training samples to design the

classifiers. A conventional approach to this problem is to assume a parametric form for the conditional densities and obtain estimates of the parameters from the given finite set of labeled training patterns. However, any assumed parametric distributions will create some mismatch between the true and estimated conditional distributions. The limited availability of training data is another problem for effective characterization of the conditional densities.

An alternative approach that alleviates these problems is to use a set of class discriminant functions (similar to the score functions discussed above) $\{g_l(O, \Lambda) : l = 1, \ldots, L\}$ to perform classification and hypothesis testing, and to use discriminative training to estimate the parameters Λ. The form of the probability distributions is not needed in this case. Each $g_l(O; \Lambda)$ evaluates the similarity between a given test utterance O and the class C_l. To combine current speech pattern classification techniques and the discriminative training approach we can use the conditional class distributions as the class discriminant functions and obtain the classifier parameters accordingly based on minimum error classification (MCE) [39.52,53] and minimum verification error (MVE) [39.55] training. Recently maximal figure-of-merit (MFoM) [39.41] learning for vector-based classifiers has also been shown to give superior and robust performance compared with SVM-trained classifiers in text categorization.

Three additional functions are required to formulate discriminative training, namely:

1. the class antidiscriminant function, $G_l(O; \Lambda)$ [similar to the quantity in (39.11)]
2. the class misclassification measure, $d_l(O; \Lambda)$, which is usually defined as the difference between $G_l(O; \Lambda)$ and $g_l(O; \Lambda)$ [similar to the measure in (39.10)]
3. the class loss function, $l_l(O; \Lambda) = l[d_l(O; \Lambda)]$, which measures the loss or cost incurred for a given value of $d_k(O; \Lambda)$

The loss is often a smooth 0–1 function such as a sigmoid to approximate error counts based on its distance from the decision boundary, $d_l(O; \Lambda) = \beta_l$:

$$l_l(O; \Lambda) = \frac{1}{1 + \exp[-\alpha_l(d_l - \beta_l)]} , \tag{39.12}$$

where α_l and β_l are parameters for the sigmoid function characterizing the slope near, and the location of, the decision boundary, respectively.

Given the four sets of functions the optimization objective functions can be defined differently, depending on the performance metrics to be optimized in each application. For language identification, we are interested in minimizing the overall approximate empirical classification error rate for all training data, $\boldsymbol{O} = \{O_i, 1 \le i \le N\}$:

$$L(\boldsymbol{O}, \Lambda) = \frac{1}{N} \sum_{i=1}^{N} \sum_{l=1}^{L} l(d_l) \mathbf{1}(O_i \in C_l) , \tag{39.13}$$

where the function $\mathbf{1}(S)$ is the indicator function for a logical variable S. For language verification we can approximate the empirical false-rejection (type I) and false-acceptance (type II) error rates for the l-th language as

$$L_{l1}(\boldsymbol{O}, \Lambda) = \frac{1}{N_l} \sum_{i=1}^{N_l} l(d_l) \mathbf{1}(O_i \in C_l) \tag{39.14}$$

and

$$L_{l2}(\boldsymbol{O}, \Lambda) = \frac{1}{\overline{N}_l} \sum_{i=1}^{\overline{N}_l} l(d_l) \mathbf{1}(O_i \in \overline{C}_l) , \tag{39.15}$$

where N_l and $\overline{N}_l$ are the numbers of tokens in and not in class C_l in the training set, respectively, and $\overline{C}_l$ denotes the set of all tokens not belonging to the class C_l. According to the application requirements we can specify ω_{l1} and ω_{l2}, for the l-th class, as the costs of making type I and II errors. Now the overall empirical average cost can be defined as:

$$L(\boldsymbol{O}, \Lambda) = \frac{1}{L} \sum_{l=1}^{L} \omega_{l1} L_{ll}(\boldsymbol{O}, \Lambda) + \omega_{l2} L_{l2}(\boldsymbol{O}, \Lambda) . \tag{39.16}$$

In IR and TC applications precision, *recall*, and $F1$ measures are commonly adopted as performance metrics. They are defined for the l-th class as:

$$\mathrm{Pr}_l(\boldsymbol{O}, \Lambda) = \frac{(1 - L_{l1}) \times N_l}{(1 - L_{l1}) \times N_l + L_{l2} \times \overline{N}_l} , \tag{39.17}$$

$$\mathrm{Re}_l(\boldsymbol{O}, \Lambda) = 1 - L_{l1} , \tag{39.18}$$

and

$$\mathrm{F1}_l(\boldsymbol{O}, \Lambda) = \frac{2 \times \mathrm{Pr}_l \times \mathrm{Re}_l}{\mathrm{Pr}_l + \mathrm{Re}_l} . \tag{39.19}$$

Now we can define the overall loss objective as the negative of the average F1 value over all L classes:

$$L(\boldsymbol{O}, \Lambda) = -\frac{1}{L} \sum_{l=1}^{L} \mathrm{F1}_l(\boldsymbol{O}, \Lambda) . \tag{39.20}$$

The three empirical loss functions in (39.13), (39.16), and (39.20) are the objectives to be minimized in MCE, MVE, and MFoM discriminative learning, respectively. Generalized probabilistic descent (GPD) algorithms [39.53] are often used to solve these optimization problems such that Λ is adjusted from one iteration to the next according to

$$\Lambda_{q+1} = \Lambda_q + \Delta\Lambda_q \tag{39.21}$$

with Λ_q being the parameter set at the q-th iteration. The correction term $\Delta\Lambda_q$, in a batch mode, is:

$$\Delta\Lambda_q = -\epsilon_q V_q \nabla L(\Lambda_q)\,, \tag{39.22}$$

where V_q is a positive-definite learning matrix and ϵ_q is a small positive real number representing the learning step size. We usually set V_q to be a diagonal matrix and define $\epsilon_q = (1 - q/Q_c)$, with Q_c denoting a prescribed maximum number of iterations to meet the convergence conditions, as required by the stochastic gradient search algorithm. GPD algorithms usually converge slowly to local minima. To speed up their convergence, QuickProp algorithms can also be applied to adjust the learning rate dynamically [39.56]. Globally optimal algorithms are also being actively pursued by the machine learning community.

These mentioned discriminative training algorithms are flexible enough to deal with any performance metric for any given classifier as long as these metrics can be expressed as functions of the classifier parameters. They can also be considered as a decision feedback mechanism in that every training token will go through a classification process first and its contributions to parameter adjustment depend on how well the decision is made. Therefore they usually work very well on the training data. On the other hand, SVM-based learning algorithms are designed to provide a *margin* to serve as a tolerance region around the decision boundaries implied by the classifiers. In so doing they usually provide better generalization capabilities than other learning algorithms. Since the test risk is often expressed as a function of the empirical risk and a regularization penalty term as a function of the VC dimension [39.39], we can design a family of margin-based discriminative training algorithm to enhance both the accuracy and robustness of the classifiers. This is a promising new direction, and some attempts in this direction have recently been proposed for estimating HMM parameters (e.g., [39.57]).

39.7 Summary

In this chapter, we have provided an overview of the three chapters to be presented later in this Part G. We have also highlighted two currently popular approaches to spoken language identification, namely phone recognition followed by language modeling and vector-space characterization. We demonstrated that the techniques developed for LID can be directly extended to spoken language verification if we can properly evaluate scores for the competing null and alternative hypotheses. Finally, we briefly presented a general framework for discriminative classifier design of non-SVM classifiers. The field of spoken language classification has witnessed rapid technological progress in recent years. We expect this trend to continue as novel paradigms are explored and high-performance systems are developed.

References

39.1 L.R. Rabiner, B.-H. Juang: *Fundamentals of Speech Recognition* (Prentice Hall, Englewood Cliffs 1993)

39.2 W.M. Campbell, D.E. Sturim, D.A. Reynolds: Support vector machines using GMM supervectors for speaker recognition, IEEE Signal Process. Lett. **13**(5), 308–311 (2006)

39.3 W.M. Campbell, J.P. Campbell, D.A. Reynolds, E. Singer, P.A. Torres-Carrasquillo: Support vector machines for speaker and language recognition, Comput. Speech Lang. **20**(2-3), 210–229 (2005)

39.4 H. Li, B. Ma, C.-H. Lee: A vector space modeling approach to spoken language identification, IEEE Trans. Audio Speech Lang. Process. **15**(1), 271–284 (2007)

39.5 G. Salton: *The SMART Retrieval System* (Prentice-Hall, Englewood Cliffs 1971)

39.6 F. Sebastiani: Machine learning in automated text categorization, ACM Comput. Surv. **34**(1), 1–47 (2002)

39.7 T. Joachims: *Learning to Classify Text Using Support Vector Machines* (Kluwer Academic, Dordrecht 2002)

Amazon EX1010 Page 801

39.8 C.-H. Lee: From knowledge-ignorant to knowledge-rich modeling: a new speech research paradigm for next generation automatic speech recognition, Proc. ICSLP (2004) pp. 109–112

39.9 V.W. Zue, J.R. Glass: Conversational interfaces: advances and challenges, Proc. IEEE **88**(8), 1166–1180 (2000)

39.10 A. Waibel, P. Geutner, L.M. Tomokiyo, T. Schultz, M. Woszczyna: Multilinguality in speech and spoken language systems, Proc. IEEE **88**(8), 1181–1190 (2000)

39.11 B. Ma, C. Guan, H. Li, C.-H. Lee: Multilingual speech recognition with language identification, Proc. ICSLP (2002) pp. 505–508

39.12 P. Dai, U. Iurgel, G. Rigoll: A novel feature combination approach for spoken document classification with support vector machines, Proc. Multimedia Information Retrieval Workshop (2003) pp. 1–5

39.13 K. Kirchhoff, S. Parandekar, J. Bilmes: Mixed memory Markov models for automatic language identification, Proc. ICASSP (2002) pp. 761–764

39.14 M. Sugiyama: Automatic language recognition using acoustic features, Proc. ICASSP (1991) pp. 813–816

39.15 A.G. Adami, H. Hermansky: Segmentation of speech for speaker and language recognition, Proc. Eurospeech (2003) pp. 841–844

39.16 M. Adda-Decker, F. Antoine, P.B. de Mareuil, I. Vasilescu, L. Lamel, J. Vaissiere, E. Geoffrois, J.-S. Liénard: Phonetic knowledge, phonotactics and perceptual validation for automatic language identification, Proc. ICPhS (2003) pp. 747–750

39.17 T.J. Hazen: *Automatic Language Identification Using a Segment-Based Approach*, M.Sc. Thesis (MIT, New York 1993)

39.18 M.A. Zissman: Comparison of four approaches to automatic language identification of telephone speech, IEEE Trans. Speech Audio Process. **4**(1), 31–44 (1996)

39.19 D. Matrouf, M. Adda-Decker, L.F. Lamel, J.-L. Gauvain: Language identification incorporating lexical information, Proc. ICSLP (1998) pp. 181–184

39.20 P.A. Torres-Carassquilo, E. Singer, M.A. Kohler, R.J. Greene, D.A. Reynolds, J.R. Deller Jr.: Approaches to language identification using Gaussian mixture models and shifted delta cepstral features, Proc. ICSLP (2002) pp. 89–92

39.21 Y.K. Muthusamy, N. Jain, R.A. Cole: Perceptual benchmarks for automatic language identification, Proc. ICASSP (1994) pp. 333–336

39.22 C.-H. Lee, F.K. Soong, K.K. Paliwal (Eds.): *Automatic Speech and Speaker Recognition: Advanced Topics* (Kluwer Academic, Dordrecht 1996)

39.23 C.-H. Lee, Q. Huo: On adaptive decision rules and decision parameter adaptation for automatic speech recognition, Proc. IEEE **88**(8), 1241–1269 (2000)

39.24 J.L. Gauvain, L. Lamel: Large-vocabulary continuous speech recognition: advances and applications, Proc. IEEE **88**(8), 1181–1200 (2000)

39.25 L.R. Rabiner: A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE **77**(2), 257–286 (1989)

39.26 H. Li, B. Ma: A phonotactic language model for spoken language identification, Proc. ACL (2005) pp. 515–522

39.27 E. Singer, P.A. Torres-Carrasquillo, T.P. Gleason, W.M. Campbell, D.A. Reynolds: Acoustic, phonetic and discriminative approaches to automatic language recognition, Proc. Eurospeech (2003) pp. 1345–1348

39.28 Y. Yan, E. Barnard: An approach to automatic language identification based on language dependent phone recognition, Proc. ICASSP (1995) pp. 3511–3514

39.29 K.M. Berkling, E. Barnard: Language identification of six languages based on a common set of broad phonemes, Proc. ICSLP (1994) pp. 1891–1894

39.30 T. Nagarajan, H.A. Murthy: Language identification using parallel syllable-like unit recognition, Proc. ICASSP (2004) pp. 401–404

39.31 K.M. Berkling, E. Barnard: Analysis of phoneme-based features for language identification, Proc. ICASSP (1994) pp. 289–292

39.32 P.A. Torres-Carrasquillo, D.A. Reynolds, R.J. Deller Jr.: Language identification using Gaussian mixture model tokenization, Proc. ICASSP (2002) pp. 757–760

39.33 J.R. Bellegarda: Exploiting latent semantic information in statistical language modeling, Proc. IEEE **88**(8), 1279–1296 (2000)

39.34 C.E. Shannon: Prediction the Entropy of Printed English, Bell Syst. Tech. J. **30**, 50–64 (1951)

39.35 H. Li, B. Ma, R. Tong: Vector-based spoken language recognition using output coding, Proc. Interspeech (2006)

39.36 C.-H. Lee, F.K. Soong, B.-H. Juang: A segment model based approach to speech recognition, Proc. ICASSP (1988) pp. 501–504

39.37 S. Gao, B. Ma, H. Li, C.-H. Lee: A text-categorization approach to spoken language identification, Proc. Interspeech (2005) pp. 2837–2840

39.38 D.A. Reynolds, R.C. Rose: Robust text-independent speaker identification using Gaussian mixture speaker models, IEEE Trans. Speech Audio Process. **3**(1), 72–83 (1995)

39.39 V. Vapnik: *The Nature of Statistical Learning Theory* (Springer, Berlin, Heidelberg 1995)

39.40 W.M. Campbell, T. Gleason, J. Navratil, D. Reynolds, W. Shen, E. Singer, P.A. Torres-Carrasquillo: Advanced language recognition using cepstra and phonotactics: MITLL system performance on the NIST 2005 language recongnition evaluation, Proc. IEEE Odyssey Speaker and Language Recognition Workshop (2006)

Amazon EX1010 Page 802

39.41 S. Gao, W. Wu, C.-H. Lee, T.-S. Chua: A MFoM learning approach to robust multiclass multi-label text categorization, Proc. ICML (2004) pp. 42–49

39.42 J. Li, S. Yaman, C.-H. Lee, B. Ma, R. Tong, D. Zhu, H. Li: Language recognition based on score distribution feature vectors and discriminative classfier fusion, Proc. IEEE Odyssey Speaker and Language Reognition Workshop (2006)

39.43 K.S. Jones: A statistical interpretation of term specificity and its application in retrieval, J. Docum. **28**, 11–20 (1972)

39.44 J. Chu-Carroll, B. Carpenter: Vector-based natural languagecall routing, Computat. Linguist. **25**(3), 361–388 (1999)

39.45 H.K.J. Kuo, C.-H. Lee: Discriminative training of natural language call routers, IEEE Trans. Speech Audio Process. **11**(1), 24–35 (2003)

39.46 L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone: *Classification and Regression Trees* (Chapman Hall, New York 1984)

39.47 S. Haykin: *Neural Networks: A Comprehensive Foundation* (McMillan, Englewood 1994)

39.48 S. Katagiri, C.-H. Lee: A new hybrid algorithm for speech recognition based on HMM segmentation and discriminative classification, IEEE Trans. Speech Audio Process. **1**(4), 421–430 (1993)

39.49 K.-Y. Su, C.-H. Lee: Speech recognition using weighted HMM and subspace projection approaches, IEEE Trans. Speech Audio Process. **2**(1), 69–79 (1994)

39.50 C.-H. Lee: A unified statistical hypothesis testing approach to speaker verification and verbal information verification, Proc. COST Workshop on Speech Technology in the Public Telephone Network: Where are we today? (1997) pp. 62–73

39.51 E.L. Lehmann: *Testing Statistical Hypotheses* (Wiley, New York 1959)

39.52 B.-H. Juang, W. Chou, C.-H. Lee: Discriminative methods for speech recognition, IEEE Trans. Speech Audio Process. **5**(3), 257–265 (1997)

39.53 S. Katagiri, B.-H. Juang, C.-H. Lee: Pattern recognition using a generalized probabilistic descent method, Proc. IEEE **86**(11), 2345–2373 (1998)

39.54 Y. Tsao, J. Li, C.-H. Lee: A study on separation between acoustic models and its applications, Proc. InterSpeech (2005)

39.55 M. Rahim, C.-H. Lee: String-based minimum verification error (SB-MVE) training for speech recognition, Comput. Speech Lang. **11**(2), 147–160 (1997)

39.56 S.E. Fahlman: An empirical study of learning speed in back-propagation networks, CMU CS Tech. Rep. **CMU-CS-88-162** (1998)

39.57 J. Li, M. Yuan, C.-H. Lee: Soft margin estimation of hidden Markov model parameters, Proc. InterSpeech (2006)

Amazon EX1010 Page 803

40. Spoken Language Characterization

M. P. Harper, M. Maxwell

This chapter describes the types of information that can be used to characterize spoken languages. Automatic spoken language identification (LID) systems, which are tasked with determining the identity of the language of speech samples, can utilize a variety of information sources in order to distinguish among languages. In this chapter, we first define what we mean by a language (as opposed to a dialect). We then describe some of the language collections that have been used to investigate spoken language identification, followed by discussion of the types of features that have been or could be utilized by automatic systems and people. In general, approaches used by people and machines differ, perhaps sufficiently to suggest building a partnership between human and machine. We finish with a discussion of the conditions under which textual materials could be used to augment our ability to characterize a spoken language.

40.1 Language versus Dialect 798
40.2 Spoken Language Collections 800
40.3 Spoken Language Characteristics 800
40.4 Human Language Identification 804
40.5 Text as a Source of Information on Spoken Languages 806
40.6 Summary 807
References 807

As we move into an increasingly globalized society, we are faced with an ever-growing need to cope with a variety of languages in computer-encoded text documents, documents in print, hand-written (block and cursive) documents, speech recordings of various qualities, and video recordings potentially containing both speech and textual components. A first step in coping with these language artifacts is to identify their language (or languages).

The scope of the problem is daunting given that there are around 7000 languages spoken across the world, as classified by the Ethnologue [40.1]. Indeed it can be difficult to collect materials for all of these languages, let alone develop approaches capable of discriminating among them. Yet the applications that would be supported by the ability to effectively and efficiently identify the language of a text or speech input are compelling: document (speech and text) retrieval, automated routing to machine translation or speech recognition systems, spoken dialog systems (e.g., for making travel arrangements), data mining systems, and systems to route emergency calls to an appropriate language expert.

In practice, the number of languages that one might need to identify is for most purposes much less than 7000. According to the Ethnologue, only about 330 languages have more than a million speakers. Thus, one might argue that in practice, there is a need for language identification (ID) of a set of languages that number perhaps in the hundreds. For example, Language Line Services [40.2] provides interpreter services to public and private clients for 156 languages, which they claim represents around 98.6% of all their customer requests for language services. Of course, for some purposes, the set of languages of interest could be far smaller.

This chapter discusses the knowledge sources that could be utilized to characterize a spoken language in order to distinguish it automatically from other languages. As a first step, we define what we mean by a language (as opposed to a dialect). We then describe some of the language collections that have been used to investigate spoken language identification, followed by a discussion of the types of linguistic features that have been or could be used by a spoken language identification (LID) system to determine the identity of the language of a speech sample. We also describe the cues that humans can use for language identification. In general, the approaches used by people and machines differ, perhaps sufficiently to point the way towards building a part-

nership between human and machine. We finish with a discussion of the conditions under which textual materials could be used to augment our ability to characterize a spoken language.

40.1 Language versus Dialect

When talking about language identification, it is important to define what we mean by a language. At first glance, the definition of *language* would seem to be simple, but it is not. Traditionally, linguists distinguish between languages and dialects by saying that dialects are mutually intelligible, unlike distinct languages. But in practice, this distinction is often unclear: mutual intelligibility is relative, depending on the speaker's and listener's desire to communicate, the topic of communication (with common concepts being typically easier to comprehend than technical concepts or concepts which happen to be foreign to one or the other participant's culture), familiarity of the hearer with the speaker's language (with time, other *accents* become more intelligible), degree of bilingualism, etc.

Political factors can further blur the distinction between dialect and language, particularly when two peoples want – or do not want – to be considered distinct. The questionable status of Serbo-Croatian is an obvious example; this was until recently considered to be more or less a single unified language. But with the breakup of Yugoslavia, the Serbian and Croatian languages (and often Bosnian) have been distinguished by many observers, despite the fact that they are largely mutually intelligible. The status of the various spoken varieties of Arabic is an example of the opposite trend. While many of these varieties are clearly distinct languages from the standpoint of mutual intelligibility, the desire for Arab unity has made some claim that they are merely dialects [40.3, 4].

Writing systems may also cause confusion for the distinction between languages and dialects. This is the case for Hindi and Urdu, for example, which in their spoken form are for the most part mutually intelligible (differing slightly in vocabulary). But they use radically different writing systems (Devanagari for Hindi, and a Perso-Arabic script for Urdu), in addition to being used in different countries, and so are generally treated as two different languages [40.5]. Hence, the distinction between language and dialect is often unclear, and so in general it is better viewed as a continuum rather than a dichotomy.

At a higher level, one can characterize a language by its language family, which is a phylogenetic unit such that all members are descended from a common ancestor (languages that cannot be reliably classified into a family are called language isolates). For example, Romance languages (e.g., French, Spanish), Germanic languages (e.g., German, Norwegian), Indo-Aryan languages (e.g., Hindi, Bengali), Slavic languages (e.g., Russian, Czech), and Celtic languages (e.g., Irish and Scots Gaelic) among others are believed to be descended from a common ancestor language some thousands of years ago, and hence are grouped into the Indo-European language family. Language families are often subdivided into branches, although the term family is not restricted to one level of a language tree (e.g., the Germanic branch of Indo-European language family is often called the Germanic family). As a result of their common descent, the languages of a language family or branch generally share characteristics of phonology, vocabulary, and grammar, although these shared resemblances may be obscured by changes in a particular language, including borrowings from languages of other language families. Thus, while English is a Germanic language, its grammar is substantially different from that of other Germanic languages, and a large portion of its vocabulary is derived from non-Germanic languages, particularly French [40.6].

The issues of language, dialect, and language family have repercussions for systems that try to assign a language tag to a piece of text or a sample of speech, and in particular for the International Organization for Standardization (ISO) standard language codes. The original standard, ISO 639-1, listed only 136 codes to distinguish languages. This allowed for the identification of most *major* languages, but not minor languages. In fact some of the codes did not represent languages at all. The code for *Quechua*, for example, corresponds to an entire language family consisting of a number of mutually unintelligible languages; arguably worse, the code for *North American Indian* refers to a number of language families, most of which include several distinct languages. For an analysis of some of the problems with ISO 639-1 and its revision, ISO 639-2, see [40.7].

At the other end of the spectrum of language classification is the Ethnologue [40.1], a listing of nearly 7000 languages. This work takes an explicitly linguistic

Amazon EX1010 Page 805

view of language classification, i. e., it attempts to assign distinct names to all and only mutually unintelligible varieties. Many observers have accused the Ethnologue of being a *splitter*, i. e., of claiming too many distinctions. This is a debatable point, but it does serve to highlight a fundamental problem of classifying a language sample as belonging to this language or that: it is hard to classify a language artifact if we cannot agree ahead of time what the possibilities are. And indeed, for some purposes, the finer-grained classification like the Ethnologue may be superfluous, while for others it may be crucial. One example of the need for fine-grained classification would be Arabic *dialect* identification, where it may be desirable to determine which variety of Arabic someone is speaking.

To some extent, the problem of conflicting classification criteria (ISO versus Ethnologue) has been resolved by unifying the two systems as ISO 639-3 [40.8]. In addition to the set of 7000 languages already listed in the Ethnologue, ISO 639-3 adds various other languages, including extinct languages and artificial languages (such as Klingon) and so-called *macro languages*, which are really language families (such as Arabic). However, even given that the standard can now be agreed on – the set of languages in ISO 639-3 – there does remain a difficulty for language identification, namely the level of granularity for language classification. The reader should remember that the problem exists, and that identifying a text or speech recording as *Arabic* may be adequate for some purposes, but insufficient for others.

The continuum from language to dialect has another implication for language identification: the existence of significant differences among dialects of a single language can make identification of that language more difficult, particularly in its spoken form (written forms of languages tend to be more standardized [40.9]). Dialectal differences can occur in all linguistic aspects: lexicon, grammar (syntax and morphology), and phonology; but it will probably be the phonology that causes the most problems for language identification in speech, since this is the level of representation that is commonly used in existing systems.

In English, for example, the differences between rhotic and non-rhotic dialects are well known [40.10]. The Scots dialects of English demonstrate even greater differences from other dialects of English in their vowel systems, where they have largely lost the distinction between so-called *long* and *short* vowels; and among the consonants, Scots English has retained the voiceless velar fricative and the voiceless labiovelar, both of which have merged with other phonemes in other dialects of English [40.11]. See the SCOTS project [40.12] for some recorded speech examples. Likewise, spoken Mandarin is strongly influenced by the native dialect spoken in a region, as well as other factors such as age [40.13].

Dialects that differ primarily in their phonology are often called *accents*, although this term is also used to refer to pronunciation by a non-native speaker of a language. In this regard, while foreign accents might be dismissed as irrelevant to language identification for some purposes, in the case of major languages, there may be significant communities of non-native speakers who use the major language as a trade language. For example, in India there are 21 (currently) official languages, but English is defined by the Constitution of India, as well as by later laws, as one of the languages of communication for the federal government (the other being Hindi). The result of this and other factors is that English is spoken non-natively by a large portion of the Indian population, and it has acquired a distinctly Indian pronunciation as a consequence [40.14]. Indian English lacks voiceless aspirated stops; what would be alveolar consonants in other varieties of English are often retroflexed; and the stress patterns are altered with the effect that vowels that would be reduced in other Englishes to schwa appear instead in their nonreduced forms.

English is used as a trade language or lingua franca in many other parts of the world as well, and these local versions of English are often significantly different from the English spoken in countries where it is a native language (see [40.15] for descriptions of some of these varieties). The same is true of French, Hausa, and many other languages that are used as trade languages (cf. [40.16] and the Ethnologue [40.1]). Such lingua franca varieties can differ significantly from *standard* varieties in ways which would likely impact language identification.

Additionally, there is a substantial amount of variability in the spoken realization of a particular language [40.17–19] due to a variety of factors, including:

- mispronunciation
- individual speaking style
- genre (e.g., conversations versus formal presentations)
- variations in speaking rate
- the speaker's psychological state
- the speaker's language repertoire
- the social and economic background of the speaker
- the speaker's first language (where the speaker is speaking in a second language)
- channel characteristics

This variability creates a challenge when identifying the language of a speech sample. A greater understanding of language, dialect, and accent is an important first step in developing an approach for language identification.

40.2 Spoken Language Collections

Engineering research in spoken language identification has been based on a very small sampling of languages from the 7000 identified by the Ethnologue. The first multilingual speech collection targeted for language identification system evaluation, the Oregon Graduate Institute (OGI) multilanguage corpus, was released in 1994 and consisted of spoken responses to prompts recorded over telephone lines by speakers of 12 languages (see [40.20–22]). The set was selected to contain both unrelated languages (e.g., German, Vietnamese, and Tamil), as well as more closely related languages (e.g., English and German). It covers languages with various types of prosodic phenomena that occur in spoken languages, such as tone (e.g., Mandarin and Vietnamese) and pitch accents (Japanese), as well as languages with various levels of complexity at the syllable level. See appendix A in [40.23] for a family language tree for the languages in this collection and appendix B for a table comparing the phone inventories for these languages. Perhaps one of the most important requirements for the languages selected was the ability to access sufficient numbers of speakers of the language in the United States, a factor that has been important in collections used for evaluation.

A second collection, the Linguistic Data Consortium (LDC) CallFriend corpus (see the list under the LID heading at [40.24]), was released in 1996 and contains telephone conversations among speakers of 14 languages or dialects, most of which appeared in the OGI multilanguage corpus. This collection was used in National Institute of Standards and Technology (NIST) evaluations of LID systems in 1996 and 2003. The CallHome collection, which was released between 1996 and 1997, is an additional multilingual resource containing six languages that were collected for large-vocabulary speech recognition; however, it does not expand the number of languages beyond what was available from CallFriend. Given the limited sampling of languages in these collections, it is not surprising that researchers developing spoken LID systems have not investigated whether languages naturally cluster into groups that parallel the classification of languages into families and branches.

Additional speech collections have been developed with an increased number of languages. The Center for Spoken Language Understanding (CSLU) 22 languages corpus, which was initially collected from 1994 through 1997, contains spoken utterances in 21 languages. It has gone through several versions, resulting in an increased number of transcribed utterances, and was released through LDC in 2005 (see [40.25]). LDC is also currently collecting the MIXER corpus which will contain telephone calls over 24 languages or dialects [40.26]. The 2005 NIST evaluation data contained speech from the Mixer and CallFriend corpora, as well as some data collected at the Oregon Health and Science University. The data in all of these collections will hopefully stimulate new methods for constructing speech-based LID systems. However, even 20 or so languages is far from the number of languages that would need to be identified to support Language Line Services.

The collection of significant quantities of comparable telephone speech in multiple languages has become more challenging in recent years. There is an implicit assumption that the training and development data are collected under conditions that are comparable to the evaluation materials. Data must also be collected in such a way that it is impossible to identify language based on speaker, gender distribution, domain, channel characteristics, etc. Finally, due to the falling cost of long-distance telephone calls, incentives such as free long-distance calls are less attractive to potential participants. This difficulty in collecting comparable speech corpora for a large number of languages has obvious implications for speech-based LID; we return to this issue later.

40.3 Spoken Language Characteristics

A speech-based LID model for a particular language is trained to represent the corresponding language and to differentiate it from others. Hence an important challenge to speech-based LID systems is the effective incorporation of discriminative knowledge sources into their models. Some systems use only the digitized

speech utterances and the corresponding true identities of the languages being spoken for training, whereas others require additional information such as phonetic and/or orthographic transcriptions, which can be expensive to produce. During the language recognition phase, a new audio sample is compared to each of the language-dependent models, and the language of the closest matching model (e.g., using maximum likelihood) is selected. We will discuss various levels of knowledge that can be used to identify a language in this section, touching upon research that utilizes the representation where appropriate.

Automatically Derived Features from the Speech Signal. Languages can be identified based on features that are automatically derived from the speech signal itself. Systems utilizing this type of information are motivated by the observation that different languages are made up of a variety of different sounds; hence, feature vectors automatically extracted from the speech signal over short time frames (segments) can be used to discriminate among the languages. Such systems typically use a multistep process (including modules to remove silence from the samples, to reduce channel effects, etc.) to convert the digitized speech signal into a feature vector representation. Given feature extraction and knowledge of which training samples correspond to each language, a classifier is constructed for each language based on language-dependent patterns of feature vectors. Methods include approaches that model only the static distribution of acoustic features given the language (e.g., [40.27]) and approaches that also utilize the patterns of change of these feature vectors over time (e.g., [40.28]). A variety of computational models have been investigated, including Gaussian mixture models [40.29], Hidden Markov models [40.29,30], artificial neural networks [40.22], and support vector machines [40.31]. Although these acoustic-based systems do not require training data that is labeled with explicit linguistic units such as phonemes or words, they also do not perform as accurately as systems that use linguistic knowledge; however, improved accuracy can be achieved by integrating acoustic- and linguistic-based knowledge sources [40.23, 32].

Phonological Information. Phonology is a branch of linguistics that studies sound systems of human languages [40.33]. In a given language, a phoneme is a symbolic unit at a particular level of representation; the phoneme can be conceived of as representing a family of related phones that speakers of a language think of as being categorically the same. While the notion of phonemes has been controversial among linguists (see [40.34] for some history), it has proven a useful abstraction for speech processing. *Ladefoged* [40.35] speculates that 'there are probably about 600 different consonants' (p. 194) across the languages of the world; vowels and suprasegmental distinctions (such as tone) are not quite as numerous.

Phonetic symbols provide a way to transcribe the sounds of spoken languages. There are several phonetic alphabets; one commonly used by linguists is the international phonetic alphabet (IPA). This alphabet, including a set of diacritic modifiers, was established as a standard by the International Phonetic Association (IPA) in order to provide an accurate representational system for transcribing the speech sounds of all languages (the IPA website is at [40.36]). The goal for the IPA is to provide a representation for all of the phonemes expressed in all human languages, such that separate symbols are used for two sounds only if there exists a language for which these two sounds are distinguished phonemically. For the most recent update of the IPA, see [40.37].

Given the availability of a phonetic or phonemic representation for language sounds, there are various ways in which this information can help to differentiate among languages, including:

1. *Phonemic inventory:* It is possible to distinguish some languages from one another based on the presence of a phoneme or phone that appears in one language but not the other. Phoneme inventories range from a low of 11 (for Rotokas, a Papuan language; see [40.38]) to a high of a hundred or more (in certain Khoisan languages of southern Africa [40.39]). It is likely that no two languages share exactly the same phoneme inventory. Furthermore, even if two languages were to share a common set of phonemes, the phonemes themselves will likely differ in their relative frequency patterns [40.22].
2. *Broad class inventory:* Patterns of broad phonetic categories (e.g., vowels, fricatives, plosives, nasals, and liquids) have also been utilized to distinguish among languages in an attempt to avoid the need for fine phonetic recognition. For example, *Muthusamy* [40.22] evaluated the use of seven broad phonetic categories (vowel, fricative, stop, closure or silence, prevocalic sonorant, intervocalic sonorant, and postvocalic sonorant). However, *Hazen* [40.23] found that, even though the broad class phone

Amazon EX1010 Page 808

recognizers tend to be more accurate than the more-traditional finer-grained phone recognizers, using broad class inventories leads to a lower-accuracy language identification system.

3. *Phonotactics:* Phonotactics refers to the arrangements of phones or phonemes within words. Even if two languages were to share a common phoneme inventory, it is likely that they would differ in phonotactics. The first proponents of using phonotactic features were *House* and *Neuburg* [40.40], who believed accurate language identification could be achieved by making use of the statistics of the linguistic events in an utterance, in particular, language-specific phonetic sequence constraints. (They suggested using a sequence of broad phonetic classes as a way of obtaining more-reliable feature extraction across languages, although this has been found to result in less accurate language identification systems [40.23].) An implementation using this approach would typically involve several steps, where the first is to map an utterance to a sequence of phonetic labels (i. e., tokenization into phones), which would then be used to identify the language based on the observed n-grams.
   There are several variants of this approach to audio based language identification. Phone-based LID uses a single-language phone recognizer trained for some arbitrary language with sufficient resources (not necessarily one of the target languages) to tokenize the speech input into *phones* for that language, followed by the use of n-gram probabilistic language models (one for each target language) to calculate the likelihood that the symbol sequence was produced in each of the target languages, with the highest likelihood language being selected. The parallel phone recognition followed by language model (PPRLM) is similar except that it uses phone recognizers from several languages, together with some method to normalize and combine the results from the parallel streams [40.41].
   A third variant would be to train a phone recognizer on a broad-coverage phonetic database (such as that in [40.35]). To reduce the time and cost to develop speech systems in a new language, researchers have been investigating the development and use of a multilingual phone set that represents sounds across the languages to be modeled. *Schultz* and *Waibel* [40.42] in their research on multilingual speech recognition defined a phone set covering 12 languages. They assume that *the articulatory representations of phonemes are so similar across languages, that phonemes can be considered as units which are independent from the underlying language* (p. 1) [40.43].
   Hazen and *Zue* [40.23, 32] developed a LID system that was based on 87 language-independent phone units that were obtained by hand clustering approximately 900 phone labels found in training transcriptions. Researchers at the computer sciences laboratory for mechanics and engineering sciences (LIMSI) have also been investigating the use of a *universal* phone set [40.44, 45] and have attempted to identify objective acoustic criteria for clustering the language-dependent phones. *Corredor-Ardoy* et al. [40.46], found that their LID system using a language-independent set (based on clustering) performed as well as their best methods using language-dependent phones. *Ma* and *Li* [40.47] evaluated the use of a universal sound recognizer to transcribe utterances into a sequence of sound symbols that act as a common phone set for all of the languages to be identified. They then used statistics related to the large-span co-occurrence of the sound patterns, which they dubbed the bag-of-sounds approach, to identify the language of the utterance.
4. *Articulatory features:* Speech can be characterized by parallel streams of articulatory features which are used in concert to produce a sequence of phonemes. These features could be exploited to differentiate one language from another. For example, the phoneme /t/ can be realized either with or without aspiration, with a dental or alveolar closure, and with lips rounded or not [40.23]. *Kirchhoff* and *Parandekar* [40.48] utilized a set of pseudoarticulatory classes that were designed to capture characteristics of the speech production process, including: manner of articulation, consonantal place of articulation, vocalic place of articulation, lip rounding, front–back tongue position, voicing, and nasality. They developed an alternative approach to audio-based language identification that was based on the use of parallel streams of these subphonemic events together with modeling of some of the statistical dependencies between the streams.

Syllable Structure. A syllable is a unit of pronunciation that is larger than a single sound, composed of a peak of sonority (usually a vowel, but sometimes a sonorous consonant), bordered by troughs of sonority (typically consonants) [40.33]. Languages can be characterized by common syllable types, typically defined in terms of sequences of consonants (C) and vowels (V). However, it should be noted that breaking sequences of Cs and

Vs into syllables – the process of syllabification – is often difficult, and even controversial [40.49, 50]. However, since languages generally allow or disallow certain types of syllable structures (e.g., Slavic languages often have complex consonant clusters in contrast to Asian languages), this representation may help in discriminating among languages. For example CCCCVC is a valid syllable type in Russian, but not in most other languages.

Zhu et al. [40.51] developed a LID system whose acoustic decoder produces syllable streams, which they then used in syllable-based (rather than phone-based) *n*-gram language models. Accents of foreign speakers of English manifest themselves differently given their position within the syllable, a fact that has been used to improve accent identification [40.52].

Prosodic Information. The duration, pitch, and stress patterns in one language often differ from another. For example, different languages have distinct intonation patterns. In stress languages, pitch is often one correlate of stress used to mark syllable prominence in words (and in pitch accent languages, such as Swedish, the primary correlate); whereas, in tone languages, a change in the meaning of a word is signalled by the tone on the syllables or other tone-bearing units (e.g., Mandarin Chinese or Thai). For stress languages, patterns of stress can provide an important cue for discriminating between two languages. Some of the stress patterns are initial stress (e.g., Hungarian), penultimate stress (e.g., Polish or Spanish), final stress (e.g., French or Turkish), and mixed stress (e.g., Russian or Greek). Prosodic cues of duration can also be potentially useful. For example, some languages (such as Finnish) distinguish long and short vowels and/or consonants.

One ostensibly useful typology investigated by both linguists and psychologists involves the rhythm of a language, where a distinction is made between stress-timed languages (in which stressed syllables are longer than unstressed syllables, all else being equal, e.g., English), syllable-timed (each syllable has comparable time duration, e.g., French), and mora-timed (each mora has essentially constant duration, e.g., Japanese) languages [40.53]. Although this classification remains controversial [40.54], rhythmic modeling has been investigated by *Rouas* et al. [40.55, 56] for language identification. Their rhythm model was able to discriminate fairly accurately between languages on a read speech corpus, but less well on a spontaneous speech corpus [40.55]. Indeed, extracting reliable prosodic cues from spontaneous speech is a challenge due to its variability.

In language identification systems that utilize prosodic features, these features are typically combined with other knowledge sources to achieve reasonable accuracy. *Muthusamy* [40.22] was able to incorporate pitch variation, duration, and syllable rate features into his LID model. *Hazen* and *Zue* [40.23, 32] integrated duration and pitch information into their LID model, with the duration model being more accurate on its own than the pitch model. *Tong* et al. [40.57] successfully integrated prosodic features (i. e., duration and pitch) with spectrum, phonotactic, and bag-of-sounds features, where pitch variation and phoneme duration were especially useful for short speech segments.

Lexical Information. Each language has its own vocabulary, which should help in identifying a language more reliably. For speech inputs, this would require the availability of a speech recognition system for each of the candidate languages, along with the requisite training, tuning, and evaluation materials needed to ensure the speech model is adequate. *Schultz* et al. [40.58, 59] developed a LID system for four languages based on large-vocabulary continuous speech recognition (LVCSR). They found that word-based systems with trigram word language modeling significantly outperform phone-based systems with trigram phone modeling on the four-language task, suggesting that the lexical level provides language discrimination ability, even when word error rates are fairly high. *Matrouf* et al. [40.60] found that incorporating lexical information with a phone-based approach yielded relative error reductions of 15–30%, and that increasing lexical coverage for a language had a positive effect on system performance. *Hieronymus* and *Kadambe* [40.61] constructed a LID system based on LVCSR for five languages (English, German, Japanese, Mandarin Chinese, and Spanish), obtaining 81% and 88% correct identification given 10 and 50 second utterances, respectively, without using confidence measures and 93% and 98% correct with confidence measures. Although each language clearly has its own vocabulary that enables language identification systems to discriminate among a candidate set of languages more effectively, for spoken languages, this information is generally quite expensive to obtain and use.

Morphology. Morphology is a branch of grammar that investigates the structure of words [40.33]. The field of morphology is divided into two subfields: inflectional morphology, which investigates affixes that signal grammatical relationships that do not change the grammatical

Amazon EX1010 Page 810

class of a word (e.g., affixes marking tense, number, and case) and derivational morphology, which focuses on word formation involving affixes, such as *-ment*, that can be used to create a new word form with a possibly different grammar class, such as the noun *amendment* derived from the verb *amend*. Since languages form words in a variety of different ways, morphology could provide an excellent cue for automatic language identification. For example, one could use common suffixes to discriminate among some of the Romance languages (e.g., *-ment* in French, *-miento* in Spanish, *-mento* in Portugese, and *-mente* in Italian). Although in speech the morphology of a word is covered in part by phonotactics, with morphological knowledge of its candidate languages, a LID system could focus on specific portions of words when discriminating between two languages.

Syntax. Languages and dialects also differ in the ways that words are arranged to create a sentence. They differ in the presence or absence of words with different parts of speech, as well as in the ways that words are marked for various types of roles in a sentence. In conjunction with other kinds of information (e.g., accents), errors in grammatical usage could provide a helpful cue for identifying the first language of someone speaking a second language. For example, someone who learned Mandarin as a first language would tend to make determiner (deletion, substitution, and insertion) and agreement errors in English or German.

Languages also often differ from each other in the word order of a sentence's subject (S), verb (V), and object (O). For example, English is considered to be an SVO language because the subject typically appears before the verb, which occurs before the object; whereas, Japanese is an SOV language. Even if two languages have the same word form, it is likely that the word would appear in very different word contexts across languages. Although words may be sufficient to distinguish among languages, syntax could play an especially important role for discriminating among dialects of a language.

Language and dialect identification systems that are based on LVCSR would utilize an acoustic model, a dictionary, and a language model for each language or dialect in the candidate set. The language models utilize word co-occurrence statistics that capture some aspects of the candidate language's syntactic structure. These systems could be expanded to utilize syntax more directly by using structured language models (e.g., [40.62, 63]).

Other Information. Higher-level knowledge sources such as semantics and pragmatics are rarely used by audio-based LID systems, although this type of knowledge could potentially help. In addition, information about the source of the speech data (e.g., country of origin of a broadcast news show) could be used to narrow down the language choices.

Research on automatic language identification suggests that the more knowledge that goes into a decision about which language corresponds to an audio sample, the greater the accuracy; hence, knowledge integration is important. However, there is a trade-off between accuracy and efficiency, and furthermore, there is a need for resources to support the knowledge brought into the automatic system. Much of the work has struck an engineering balance in addressing this problem; they use the resources that can be obtained simply and reliably for the set of languages to be discriminated among.

It is important to note that the length of an audio sample (the amount of speech available) will impact the knowledge sources that can be reliably used in determining its language. The smaller the samples used, the less likely that a key piece of higher-level knowledge will be available to discriminate a particular language from the others.

40.4 Human Language Identification

A human who knows the language being spoken is capable of positively identifying short samples of speech quickly and accurately. Even if they do not know the language, people can make sound decisions about the identity of a language given some exposure to the language, and with some training about cues that differentiate a set of candidate languages, this capability can be improved and expanded.

There have been several experiments reported in the literature that consider human ability in a scenario where the decision is based on a combination of their prior knowledge about certain languages (a variable that is difficult to control) and a limited amount of online training for the languages being identified. *Muthusamy* et al. [40.22, 64] had human subjects listen to short samples of 10 different languages and guess the language

of the sample. In general, he found that familiarity with a language was an important factor affecting accuracy, as was the length of the speech sample (with longer samples leading to greater accuracy). Subjects were able to improve their LID accuracy only slightly over time with feedback, but they were quite aware of the cues they used for language discrimination (e.g., phonemic inventory, word spotting, and prosody).

Maddieson and *Vasilescu* [40.65] examined the effect of exposure to academic linguistics on language identification accuracy of five language. Subjects with more than *passing* familiarity with the language were excluded from the study. They found that prior casual exposure to a language and linguistic education level (ranging from no linguistic training to a PhD) were not effective predictors of performance on a five-language identification task; however, they found that linguistic training did predict improved performance on a language discrimination task (in which subjects were asked to decide if a sample was one of the five target languages, similar to one of those languages, or unlike them).

Several human studies have been conducted in an attempt to determine what types of information people can effectively utilize when making decisions about the identity of a language. These experiments involve the presentation of speech stimuli that were obtained by modifying the speech samples presented to the subjects. For example, *Mori* et al. [40.66] found that their subjects were able to identify two languages (Japanese and English) fairly reliably even when segmental information was reduced using signal editing techniques. They argue that other cues such as intensity and pitch are being used to make these judgments. *Navratil* [40.67] evaluated the importance of various types of knowledge, including lexical, phonotactic, and prosodic, by humans asked to identify the language (Chinese, English, French, German, or Japanese) of a speech sample. Subjects were presented unaltered speech samples, samples containing randomly ordered syllables from speech samples, and samples for which the spectral shape was flattened and vocal-tract information removed (leaving F_0 and amplitude). Navratil found that humans on six second samples were far more accurate at identifying unaltered speech samples (96%) than samples with shuffled syllables (73.9%), and were more accurate with the shuffled syllable samples than samples with only prosodic cues remaining (49.4%). Based on these experiments, it appears that the lexical and phonotactic information provides discriminative information that is used more reliably by humans.

None of the subjects in these listening experiments were explicitly trained to identify cues to discriminate one language from another, hence, these experiments did not explore the full range of human capability that could be achieved with training on a particular set of languages to be identified. People can identify the language of an audio input fairly reliably when they do not speak/understand the language by being taught to use a variety of cues that are discriminative for a language or language family. Some combination of the following sorts of clues can be used to identify a particular language or to narrow the possibilities down to a smaller set of languages:

- general impression ('gestalt'), i. e., what a given language sounds like, which may help narrow the language down to a geographic area or a language family
- stress patterns, where these may be most reliably discerned at pause boundaries or on assimilated loan words
- vowel and/or consonant durations
- the presence or absence of nasalization on (some) vowels
- the presence or absence of lexical tone
- syllable structure, particularly the presence or absence of consonant clusters
- the presence of unusual sounds, such as front rounded vowels, glottalized consonants, clicks, or retroflexed consonants
- reduplication (particularly full-word reduplication)
- the presence of common words, particularly short high-frequency words that are easily recognizable (e.g., determiners, prepositions)

Some of these features would obviously be difficult for computers to use, including the 'gestalt' of the language or detecting reduplication. Other features are similar to what programs doing spoken language identification already utilize, e.g., the use of consonant clusters (phonotactics). Still others of these features suggest possible directions for future work in automated language ID, for example recognizing unusual sounds.

There is also a more-general difference in the methodology used by humans and speech-based LID algorithms. Most speech-based LID systems do not utilize the high-level knowledge that people tend to use. Also, most of the automatic LID algorithms tend to process and combine evidence from the entire speech stream when making a decision about the identity of a language; whereas, humans rely on very specific cues taken from small portions of the sample to refine their hypotheses.

For example, two of the cues used by human experts for LID on texts are the pairings of diacritics with base characters and the presence of short (but common) words. Diacritics are especially difficult for image-based language identification systems, which can have difficulty differentiating them from noise in the image. In contrast, the primary cue used by computers for computer encoded text or in speech is the statistical frequency character or phone *n*-grams. *n*-grams would be hard for people to learn, with the exception of unusual single letters or sounds and morphemes which are cognate with English morphemes; anything more probably requires the user to refer to a *cheat sheet*. Even harder for people is computing the statistics of *n*-grams; judgments with a finer granularity than *frequent* or *rare* would be difficult for people to make.

In summary, while it is possible for computers to make use of more of the characteristics of languages than humans use to identify them, it is probably not practical to teach people to use the methodology used by automatic LID systems to do language identification.

40.5 Text as a Source of Information on Spoken Languages

Given the difficulty in building comparable speech corpora for a significant number of languages, and in particular for rare languages, another source of information that might be mined to learn more about the spoken form of a language is its written form. (While most unwritten languages have small speaker populations, and the total number of speakers of unwritten languages is much smaller than the number of speakers of written languages, there are still significant numbers of languages of the world – perhaps more than half – which are unwritten.) It is comparatively easy to build a text corpus for a written language; however, there are several issues that affect whether the textual data will be useful in characterizing the spoken language. An important issue is how close the written language is to its typical spoken form. There are many aspects to this question, but two of the most significant issues are diglossia and complex orthographies.

A diglossic language situation exists when two (or more) forms of a language coexist, and the forms diverge to a significant degree; typically one form is perceived as *high* (correct), and the other as *low* (vulgar). For our purposes, diglossia is relevant when the high and low varieties correspond to the written and the spoken languages respectively, where they differ significantly in style and vocabulary. Tamil is a typical example; the written form of the language is considered classical, and the spoken forms (there is more than one dialect) are considered low. In such a situation, written corpora may not be representative of the spoken form of the language.

Arabic is another important example of a diglossic situation: the written form, known as modern standard Arabic (MSA), is taught in schools; whereas, the spoken varieties – of which the Ethnologue [40.1] lists nearly 40 – are strikingly different from MSA in vocabulary, morphology, and phonology (the latter with reference to how MSA is generally read out loud, for instance on news broadcasts).

As for the complex orthography issue, for our purposes a complex orthography is one in which the written forms of words do not have a direct mapping to the spoken forms. English is a notorious example of this, and so to a lesser extent is French. A related issue is orthographies which undermark phonemic distinctions in the spoken language. Written Arabic is an example, since the short vowels are not normally written, resulting in significant ambiguity: multiple morphological analyses, each corresponding to different pronunciations, are possible for a large percentage of the words in running Arabic (MSA) text.

Many orthographies that are complex today are so only because the spoken language has changed faster than the written language. Orthographies that have been developed in the recent past tend to be less complex, i. e., they usually map more or less directly to the phonemes of the spoken language (or to a standard dialect of the language), and can therefore be said to be *phonemic*. However, some otherwise phonemic modern orthographies fail to discriminate a subset of the phonemic contrasts of the language, whether in practice or in principle. For example, while Yoruba is supposed to be written with tone marks for the high and low tones (and optionally for the mid tone), in practice these are often omitted, as are the dots under the Yoruba letters 'e' and 'o', intended to indicate a more-open vowel than the same letters without the dot. Finally, some orthographies omit certain phonemic distinctions (e.g., tone) on the principle that the distinction in question is not important enough in that language.

Assuming that a language's orthography is close to phonemic, it is also necessary to know the mapping from individual characters (or sequences of characters) to a phonemic representation. This is because orthographies do not always use the letters in a standard way. In Hungarian, for example, the letter 's' represents an alveopalatal (like the digraph 'sh' in English), and the digraph 'sz' represents an alveolar fricative (like the English letter 's' in most words); Polish represents these sounds in exactly the opposite way. Fortunately, this information about alphabets can generally be obtained.

In summary, then, it would be possible at least in principle to expand the number of spoken languages that can be investigated and characterized linguistically by using more readily obtainable text corpora in place of speech corpora. Whether these corpora can be used to enhance speech-based LID systems is an open question.

40.6 Summary

In this chapter, we have discussed a variety of knowledge sources that could be utilized to characterize a spoken language in order to distinguish it automatically from other languages. We discussed differences between a language and a dialect, and also described some of the variability in a language that could potentially challenge LID algorithms. Currently available corpora for evaluating language identification systems have only *scratched the surface* of the possible space of languages that could be investigated. Most state-of-the-art systems are able to detect tens of languages as opposed to the 100 or more that would be required by Language Line Services. It remains to be seen whether acoustic- and phonotactic-based systems will effectively scale up to handle these one hundred plus languages. As the number of languages and dialects increase, it is likely that systems will need to utilize more linguistic insight to achieve accuracies comparable to those obtained over a smaller set of languages. These larger systems could utilize more lexical information to achieve target accuracies; however, this knowledge source comes with a high cost for system development.

We also discussed several experiments reported in the literature that investigated a person's ability to accurately identify a spoken language. None of these studies involved a situation where participants were trained to accurately identify a language based on salient language-specific high-frequency cues. We enumerated some cues that have been commonly used, some of which overlap with features used by automatic LID systems. However, since some of the cues that humans use would be difficult to incorporate into an automatic LID system (e.g., a general impression of the language), it is interesting to contemplate whether there would be some way to build a partnership between trained humans and LID systems.

We ended this chapter by discussing conditions under which textual materials could potentially augment our knowledge of a spoken language, in particular, a rare language. There are a number of factors that impact the correspondence between spoken and written language forms; however, if there is a good correspondence, the written form could be used to gain a deeper understanding of the kinds of features that would help discriminate the language from others.

References

40.1 R.G. Gordon Jr. (ed): *Ethnologue: Languages of the World*, 15th edn. (SIL International, Dallas 2005), online version: http://www.ethnologue.com

40.2 Language Line Services, http://www.languageline.com

40.3 A.G. Chejne: *The Arabic Language Its Role in History* (University of Minnesota Press, Minneapolis 1969)

40.4 N. Haeri: *Sacred Language Ordinary People: Dilemmas of Culture and Politics in Egypt* (Palgrave MacMillan, New York 2003)

40.5 C.P. Masica: *The Indo-Aryan Languages* (Cambridge Univ. Press, Cambridge 1991)

40.6 B.A. Fennel: *A History Of English: a Sociolinguistic Approach*, Blackwell Textbooks in linguistics, Vol. 17 (Blackwell, Malden 2001)

40.7 P. Constable, G. Simons: An Analysis of ISO 639: Preparing the way for advancements in language identification standards, 20th International Unicode Conference (2002), available as http://www.ethnologue.com/14/iso639/An_analysis_of_ISO_639.pdf

40.8 ISO/DIS 639-3, http://www.sil.org/iso639-3/default.asp

40.9 D. Barton: *Literacy: An Introduction to the Ecology of Language* (Blackwell, Malden 1994)

40.10 C.W. Kreidler: *Describing Spoken English: An Introduction* (Routledge, London 1997)

40.11 A. McMahon: *Lexical Phonology and the History of English*, Cambridge Studies in Linguistics 91 (Cambridge Univ. Press, Cambridge 2000)

40.12 The SCOTS Project, http://www.scottishcorpus.ac.uk

40.13 R. Sproat, T.F. Zheng, L. Gu, D. Jurafsky, I. Shanfran, J. Li, Y. Zheng, H. Zhou, Y. Su, S. Tsakalidis, P. Bramsen, D. Kirsch: *Dialectical Chinese Speech Recognition: Final Technical Report* (Johns Hopkins University, Baltimore 2004), CLSP

40.14 J.C. Wells: *Accents of English*, Vol. 3 (Cambridge Univ. Press, Cambridge 1982)

40.15 P. Trudgill, J. Hannah: *International English: A Guide to Varieties of Standard English* (Oxford Univ. Press, New York 2002)

40.16 A. Sakaguchi: Towards a clarification of the function and status of international planned languages. In: *Status and Function of Language and Language Varieties*, ed. by U. Ammon (Walter de Gruyter, Berlin 1989) p. 399

40.17 W. Byrne, M. Finke, S. Khudanpur, J. McDonough, H.J. Nock, M. Riley, M. Saralar, C. Wooters, G. Zavaliagkos: Pronunciation modelling using a handlabelled corpus for conversational speech recognition, Proc. ICASSP (1998) pp. 313–316

40.18 E. Fosler-Lussier, N. Morgan: Effects of speaking rate and word frequency on conversational pronunciations, Speech Commun. **29**, 137–158 (1999)

40.19 S. Greenberg: Speaking in shorthand – A syllablecentric perspective for understanding pronunciation variation, Speech Commun. **29**, 159–176 (1999)

40.20 Multi-Language Telephone Speech Corpus Distribution, http://www.ldc.upenn.edu/Catalog/readme_files/ogi readme.html, January 1994.

40.21 OGI Multilanguage Corpus, http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC94S17

40.22 Y.K. Muthusamy: *A Segmental Approach to Automatic Language Identification* (Oregon Graduate Institute of Science and Technology, Beaverton 1993), Ph.D. Thesis

40.23 T.J. Hazen: *Automatic Language Identification Using a Segment-Based Approach* (MIT, Cambridge 1993), Masters Thesis

40.24 LDC Projects, http://www.ldc.upenn.edu/Catalog/project_index.jsp

40.25 CLSU: 22 Languages Corpus, http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2005S26

40.26 MIXER Telephone Study, http://mixer.ldc.upenn.edu

40.27 D. Cimarusti, R.B. Ives: Development of an automatic identification system of spoken languages: Phase 1, Proc. ICASSP (1982) pp. 1661–1663

40.28 P.A. Torres-Carrasquillo, D.A. Reynolds, J.R. Deller Jr.: Language identification using Gaussian mixture model tokenization, Proc. ICASSP (2002) pp. 757–760

40.29 M.A. Zissman: Automatic Language Identification Using Gaussian Mixture and Hidden Markov Models, Proc. ICASSP (1993) pp. 399–402

40.30 K.K. Wong, M. Siu: Automatic language identification using discrete hidden Markov model, Interspeech (2004) pp. 1633–1636

40.31 W.M. Campbell, E. Singer, P.A. Torres-Carrasquillo, D.A. Reynolds: Language recognition with support vector machines, Proceedings of Odyssey: The Speaker and Language Recognition Workshop (2004) pp. 41–44

40.32 T.J. Hazen, V. Zue: Segment-based automatic language identification, J. Acoust. Soc. Am. **101**(4), 2323–2331 (1997)

40.33 D. Crystal: *A Dictionary of Linguistics and Phonetics*, 2nd edn. (Basil Blackwell, Oxford: 1985)

40.34 M. Kenstowicz: Generative Phonology. In: *Encyclopedia of Language and Linguistics*, ed. by K. Brown (Elsevier, Amsterdam 2005), 2nd edn.

40.35 P. Ladefoged: *Vowels and Consonants: An Introduction to the Sounds of Languages* (Blackwell, Oxford 2005)

40.36 The International Phonetic Association, http://www.arts.gla.ac.uk/IPA/index.html

40.37 Reproduction of the International Phonetic Alphabet, http://www.arts.gla.ac.uk/IPA/ipachart.html , 2005.

40.38 I. Maddieson: *Sound Patterns of Language* (Cambridge Univ. Press, Cambridge 1984)

40.39 A. Traill: *Phonetic and Phonological Studies of !Xóõ bushman* (Helmut Buske, Hamburg 1985)

40.40 A.S. House, E.P. Neuberg: Toward automatic identification of the languages of an utterance: preliminary methodological considerations, J. Acoust. Soc. Am. **62**(3), 708–713 (1977)

40.41 M.A. Zissman: Comparison of four approaches to automatic language identification of telephone speech, IEEE Trans. Speech Audio Process. **4**(1), 31–44 (1996)

40.42 T. Schultz, A. Waibel: Language independent and language adaptive acoustic modeling for speech recognition, Speech Commun. **35**(1-2), 31–51 (2001)

40.43 A. Black, T. Schultz: Speaker clustering for multilingual synthesis, Proceedings of the ISCA Tutorial and Research Workshop on Multilingual Speech and Language Processing (2006)

40.44 M. Adda-Decker, F. Antoine, P.B. de Mareuil, I. Vasilescu, L. Lamel, J. Vaissiere, E. Geoffrois, J.-S. Liénard: Phonetic knowledge, phonotactics and perceptual validation for automatic language identification, International Congress of Phonetic Sciences (2003)

40.45 P.B. de Mareüil, C. Corredor-Ardoy, M. Adda-Decker: Multi-lingual automatic phoneme clustering, Int. Congress Phonetic Sci. (1999) pp. 1209–1213

Amazon EX1010 Page 815

40.46 C. Corredor-Ardoy, J.L. Gauvain, M. Adda-Decker, L. Lamel: Language Identification with Language-independent Acoustic Models, Proc. Eurospeech (1997) pp. 355–358

40.47 B. Ma, H. Li: Spoken language identification using bag-of-sounds, International Conference on Chinese Computing (2005)

40.48 K. Kirchhoff, S. Parandekar: Multi-stream statistical language modeling with application to automatic language identification, Proceedings of the 7th European Conference on Speech Communication and Technology Proceedings of Eurospeech (2001) pp. 803–806

40.49 J. Blevins: The syllable in phonological theory. In: *The Handbook of Phonological Theory*, Blackwell Handbooks in Linguistics, Vol. 1, ed. by J.A. Goldsmith (Blackwell, Oxford 1995) pp. 206–244

40.50 M. Kenstowicz: *Phonology in Generative Grammar*, Blackwell Textbooks in Linguistics, Vol. 7 (Blackwell, Oxford 1994)

40.51 D. Zhu, M. Adda-Decker, F. Antoine: Different size multilingual phone inventories and context-dependent acoustic models for language identification, Interspeech (2005) pp. 2833–2836

40.52 K. Berkling, M. Zissman, J. Vonwiller, C. Cleirigh: Improving accent identification through knowledge of English syllable structure, Proceedings of the 5th International Conference on Spoken Language Processing (1998) pp. 89–92

40.53 E. Grabe, E.L. Low: Durational variability in speech and the rhythm class hypothesis. In: *Laboratory Phonology*, ed. by C. Gussenhoven, N. Warner (Mouton de Gruyter, Berlin 2002) pp. 515–546

40.54 R.M. Dauer: Stress-timing and syllable-timing reanalysed, J. Phonet. **11**, 51–62 (1983)

40.55 J. Rouas, J. Farinas, F. Pellegrino, R. Andre-Obrecht: Modeling prosody for language identification on read and spontaneous speech, Proc. ICASSP (2003) pp. 40–43, vol. 6

40.56 J. Rouas, J. Farinas, F. Pellegrino, R. André-Obrecht: Rhythmic unit extraction and modelling for automatic language identification, Speech Commun. **47**(4), 436–456 (2005)

40.57 R. Tong, B. Ma, D. Zhu, H. Li, E.S. Chang: Integrating acoustic, prosodic and phonotactic features for spoken language identification, Proc. ICASSP (2006) pp. 205–208

40.58 T. Schultz, I. Rogina, A. Waibel: Experiments with LVCSR based language identification, Proceedings of the Speech Symposium SRS XV (1995)

40.59 T. Schultz, I. Rogina, A. Waibel: LVCSR-based language identification, Proc. ICASSP (1996) pp. 781–784

40.60 D. Matrouf, M. Adda-Decker, L. Lamel, J. Gauvain: Language identification incorporating lexical information, Proceedings of the 5th International Conference on Spoken Language Processing (1998) pp. 181–184

40.61 J. Hieronymus, S. Kadambe: Robust spoken language identification using large vocabulary speech recognition, Proc. ICASSP (1997) pp. 779–782

40.62 C. Chelba: *Exploiting Syntactic Structure for Natural Language Modeling* (Johns Hopkins University, Baltimore 2000), Ph.D. thesis

40.63 W. Wang, M. Harper: The SuperARV language model: Investigating the effectiveness of tightly integrating multiple knowledge sources, Proceedings of Conference of Empirical Methods in Natural Language Processing (2002) pp. 238–247

40.64 Y.K. Muthusamy, E. Barnard, R.A. Cole: Reviewing automatic language identification, IEEE Signal Process. Mag. **11**(4), 33–41 (1994)

40.65 I. Maddieson, I. Vasilescu: Factors in human language identification, Proceedings of the 5th International Conference on Spoken Language Processing (2002) pp. 85–88

40.66 K. Mori, N. Toba, T. Harada, T. Arai, M. Komatsu, M. Aoyagi, Y. Murahara: Human language identification with reduced spectral information, Proceedings of the 6th European Conference on Speech Communication and Technology (1999) pp. 391–394

40.67 J. Navratil: Spoken language recognition – A step towards multilinguality in speech processing, IEEE Trans. Speech Audio Process. **9**(6), 678–685 (2001)

Amazon EX1010 Page 816

41. Automatic Language Recognition Via Spectral and Token Based Approaches

D. A. Reynolds, W. M. Campbell, W. Shen, E. Singer

Automatic language recognition from speech consists of algorithms and techniques that model and classify the language being spoken. Current state-of-the-art language recognition systems fall into two broad categories: spectral- and token-sequence-based approaches. In this chapter, we describe algorithms for extracting features and models representing these types of language cues and systems for making recognition decisions using one or more of these language cues. A performance assessment of these systems is also provided, in terms of both accuracy and computation considerations, using the National Institute of Science and Technology (NIST) language recognition evaluation benchmarks.

41.1 **Automatic Language Recognition** 811
41.2 **Spectral Based Methods** 812
41.2.1 Shifted Delta Cepstral Features 812
41.2.2 Classifiers 813
41.3 **Token-Based Methods** 815
41.3.1 Tokens 815
41.3.2 Classifiers 816
41.4 **System Fusion** 818
41.4.1 Methods 819
41.4.2 Output Scores 820
41.5 **Performance Assessment** 820
41.5.1 Task and Corpus 820
41.5.2 Systems 821
41.5.3 Results 822
41.5.4 Computational Considerations 822
41.6 **Summary** 823
References 823

41.1 Automatic Language Recognition

Automatic language recognition from speech consists of algorithms and techniques that model and classify the language being spoken. The term recognition can refer to either identification of the spoken language from a set of languages known to the system, or detection of the presence of a particular language out of a larger set of languages unknown to the system. As with other information conveyed by the speech signal, cues about the language being spoken occur at several levels. At the top level the words being spoken to communicate a coherent message are the basis of what defines a language. However, relying on automatic extraction of the words presupposes the availability of a reliable speech recognizer in the languages of interest, which is often not the case in many practical applications requiring language recognition. Further, language recognition is often used as a low computation pre-processor to determine which higher computation speech recognizer should be run, if any. Fortunately, cues about the language are also transmitted at lower levels in the speech signal, such as in the phoneme inventory and co-occurrences (phonotactics), the rhythm and timing (prosodics) and the acoustic sounds (spectral characteristics), that are more amenable to automatic extraction and modeling.

Current state-of-the-art language recognition systems are based on processing these low-level cues and fall into two broad categories: spectral based and token sequence based. This chapter describes algorithms for extracting features and models representing these types of language cues and systems for making recognition decisions using one or more of these language cues. Additionally, a performance assessment of these systems is provided, in terms of both accuracy and computation considerations, using the National Institute of Science and Technology (NIST) language recognition evaluation benchmarks.

Amazon EX1010 Page 817

41.2 Spectral Based Methods

Spectral-based methods for language recognition operate by extracting measurements of the short-term speech spectrum over fixed analysis frames and then modeling characteristics of these measurements, or features, for each language to be recognized. Classification techniques that have proved successful for language recognition are generative, via Gaussian mixture models (GMMs), and discriminative, via support vector machines (SVMs). This section first describes the spectral feature extraction process and then the GMM and SVM classifiers used for recognition.

41.2.1 Shifted Delta Cepstral Features

The features used for language recognition systems are based on the cepstral coefficients derived from a mel-scale filterbank analysis typically used in other speech processing tasks such as speech and speaker recognition. A block diagram of the filterbank feature extraction system is shown in Fig. 41.1. The feature extraction consists of the following steps. Every 10 ms the speech signal is multiplied by a Hamming window with a duration of 20 ms to produce a short-time speech segment for analysis. The discrete Fourier spectrum is obtained via a fast Fourier transform (FFT) from which the magnitude squared spectrum is computed. The magnitude spectrum is multiplied by a pre-emphasis filter to emphasize the high-frequency portion of the spectrum and the result is put through a bank of triangular filters. The filterbank used is similar to that in [41.1] and simulates critical band filtering with a set of triangular bandpass filters that operate directly on the magnitude spectrum. The critical band warping is done by approximating the mel-frequency scale, which is linear up to 1000 Hz and logarithmic above 1000 Hz. The center frequencies of the triangular filters follow a uniform 100 Hz mel-scale spacing, and the bandwidths are set so the lower and upper passband frequencies of a filter lie on the center frequencies of the adjacent filters, giving equal bandwidths on the mel-scale but increasing bandwidths on the linear frequency scale. The number of filters is selected to cover the signal bandwidth $[0, f_s/2]$ Hz, where f_s is the sampling frequency. For 8 kHz sampled telephone speech, there are 24 filters.

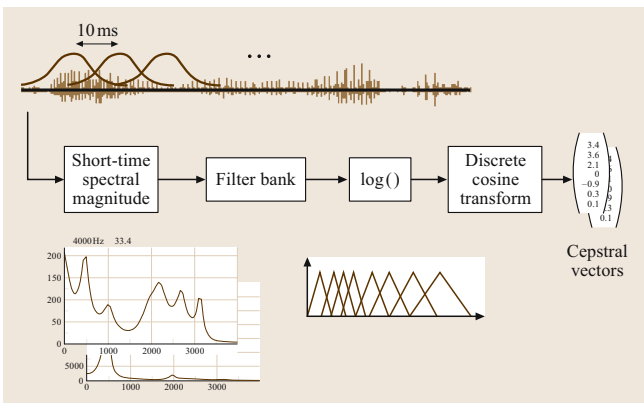


Fig. 41.1 Mel-scale filterbank cepstral feature extraction

The log energy output of each filter is then used as an element of a filterbank feature vector and the short-term mel-scale cepstral feature vector is obtained from the inverse cosine transform of the filterbank vector. To model short-term speech dynamics, the static cepstral vector is augmented by difference (delta) and acceleration (delta–delta) cepstra computed across several frames.

Since the speech used for training and testing language recognition systems can come from a variety of sources (involving different microphones and channels), it is important to apply some form of channel compensation to the features. Typical channel compensation techniques include blind deconvolution via RASTA (relative spectral) filtering [41.2] and per-utterance feature normalization to zero mean and unit variance. More-sophisticated compensation techniques, such as feature mapping [41.3], that explicitly model channel effects are also successfully used.

One of the significant advances in performing language recognition using GMMs was the discovery of a better feature set for language recognition [41.4]. The improved feature set, the shifted delta cepstral (SDC) coefficients, are an extension of delta-cepstral coefficients.

SDC coefficients are calculated as shown in Fig. 41.2. SDC coefficients are specified by four parameters, conventionally written as N-d-P-k. For a frame of data at time t, a set of MFCCs are calculated, i. e.,

$$c_0(t), c_1(t), \ldots, c_{N-1}(t) \,. \tag{41.1}$$

Note that the coefficient $c_0(t)$ is used. The parameter d determines the spread across which deltas are calculated, and the parameter P determines the gaps between successive delta computations. For a given time t,

$$\Delta c(t, i) = c(t + iP + d) - c(t + iP - d) \tag{41.2}$$

represents an intermediate calculation, where $i = 0, 1, \ldots, k$. The SDC coefficients are then k stacked

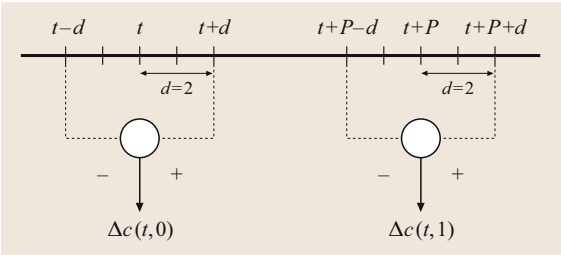


Fig. 41.2 Shifted delta cepstral coefficients

versions of (41.2),

$$\mathrm{SDC}(t) = \begin{pmatrix} \Delta c(t,0) \\ \Delta c(t,1) \\ \vdots \\ \Delta c(t,k-1) \end{pmatrix} . \tag{41.3}$$

Prior to the use of SDC coefficients, GMM-based language recognition was less accurate than alternate approaches [41.5]. SDC coefficients capture variation over many frames of data; e.g., the systems described in this chapter use 21 consecutive frames of cepstral coefficients. This long-term analysis might explain the effectiveness of the SDC features in capturing language-specific information.

41.2.2 Classifiers

Current state-of-the-art language recognition systems use either or both generative GMM-based classifiers and discriminative SVM-based classifiers.

Gaussian Mixture Models

The approach for a GMM classifier is to model the distribution of cepstral features for each language to be recognized. A language-specific GMM is given by

$$p(\boldsymbol{x}|\lambda) = \sum_{i=1}^{M} \pi_i \mathcal{N}_i(\boldsymbol{x}) , \tag{41.4}$$

where $\boldsymbol{x}$ is a D-dimensional feature vector, $\mathcal{N}_i(\boldsymbol{x})$ are the component densities, and π_i are the mixture weights. Each component density is a D-variate Gaussian function of the form

$$\mathcal{N}_i(\boldsymbol{x}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}_i|^{1/2}} \times \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\, \boldsymbol{\Sigma}_i^{-1}\, (\boldsymbol{x}-\boldsymbol{\mu}_i)\right\} , \tag{41.5}$$

with mean vector $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$. The mixture weights satisfy the constraint

$$\sum_{i=1}^{M} \pi_i = 1 , \tag{41.6}$$

which ensures that the mixture is a true probability density function. The complete language-specific Gaussian mixture density is parameterized by the mean vectors, covariance matrices, and mixture weights from all component densities and is represented by the notation

$$\lambda = \{\pi_i,\ \boldsymbol{\mu}_i,\ \boldsymbol{\Sigma}_i\} \quad i = 1, \ldots , M . \tag{41.7}$$

For language recognition using SDC features, diagonal covariances and a mixture order of $M = 2048$ are used. It was empirically determined [41.4] that higher-order GMMs provided substantial performance improvements when using SDC features.

The GMM parameters are estimated via the expectation-maximization (EM) algorithm [41.6] or by Bayesian adaptation from a universal background model (UBM) [41.7], as is done in speaker recognition (see Part F of this Handbook). Adapting from a background model has computational advantages in that it speeds up both model training and likelihood computations. For language recognition, the UBM is typically constructed by training a GMM using an aggregation of the training data from all available languages. Recently, discriminative training of GMM parameters based on maximum mutual information optimization has yielded very promising results [41.8].

For a general language recognition task, a GMM, λ_l, is trained for each of the desired languages, $l = 1, \ldots , L$, and the likelihood of a sequence of feature vectors, $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots , \boldsymbol{x}_T)$, extracted from a test utterance is computed as

$$p(\boldsymbol{X}|\lambda_l) = \prod_{t=1}^{T} p(\boldsymbol{x}_t|\lambda_l) . \tag{41.8}$$

The model likelihood is computed assuming independence between the feature vectors which means the temporal ordering of the feature vectors is unimportant. However, the SDC feature vectors $\boldsymbol{x}_t$ were themselves extracted over a multiframe time span and thereby encode local temporal sequence information.

The likelihood scores from the set of language models are then used by the back-end fusion/decision system to compute likelihood ratios or to fuse with other system scores (Sect. 41.4).

Support Vector Machines

Support vector machines (SVMs) are flexible classifiers that have recently shown promise in language recognition. SVMs rely on separating data in high-dimensional spaces using the maximum margin concept [41.9]. An SVM, $f(\boldsymbol{Z})$, is constructed from sums of a kernel function $K(\cdot,\cdot)$,

$$f(\boldsymbol{Z}) = \sum_i \alpha_i K(\boldsymbol{Z}, \boldsymbol{Z}_i) + d \,, \tag{41.9}$$

where $\boldsymbol{Z}$ is the input feature vector, $\sum_i \alpha_i = 0$, and $\alpha_i \neq 0$. The $\boldsymbol{Z}_i$ are support vectors and are obtained from the training set by an optimization process [41.10]. The ideal classifier values are either 1 or -1, depending on whether the corresponding support vector is in class 0 or class 1. For classification, a class decision is based on whether the value $f(\boldsymbol{Z})$ is above or below a threshold.

SVM kernels provide a method for comparing sequences of feature vectors (e.g., shifted delta cepstral coefficients). Given sequences of feature vectors from two utterances, $\boldsymbol{X} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{T_x})$ and $\boldsymbol{Y} = (\boldsymbol{y}_1, \cdots, \boldsymbol{y}_{T_y})$, the kernel produces a comparison that provides discrimination between the languages of the utterances. If the utterances are from the same language, a large positive value is desired; otherwise, the kernel should produce a large negative value.

The general setup for training a language recognition system using support vector machines is a *one-versus-all* strategy as shown in Fig. 41.3. For the example of English in the figure, class 1 contains only target (English) language data and class 0 contains all of the nontarget (non-English) language data pooled together. Training proceeds using a standard SVM training tool with a sequence kernel module (e.g., [41.10]). The resulting process produces a model for recognizing the target language.

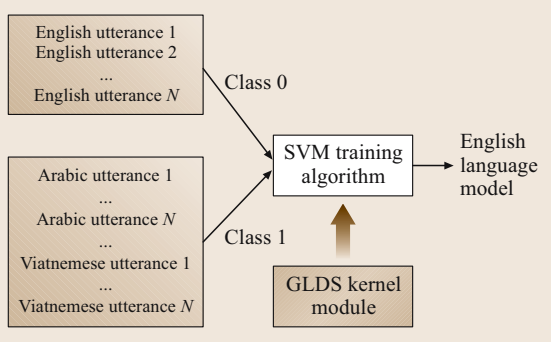


Fig. 41.3 Training a support vector machine for language recognition

A useful kernel for language recognition is the generalized linear discriminant sequence (GLDS) kernel [41.11]. The GLDS kernel is simply an inner product between average high-dimension expansions of a set of basis functions $b_j(\cdot)$,

$$\boldsymbol{b}(\boldsymbol{x}) = (b_1(\boldsymbol{x}) \;\cdots\; b_K(\boldsymbol{x}))^{\mathrm{T}} \,, \tag{41.10}$$

where K is the number of basis functions. Typically, monomials up to a given degree are used as basis functions. An example basis is

$$\boldsymbol{b}(\boldsymbol{x}) = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \\ x_1^2 \\ \vdots \\ x_1^{i_1} x_2^{i_2} \ldots x_n^{i_n} \\ \vdots \end{pmatrix} . \tag{41.11}$$

The formula for the kernel is

$$K_{\mathrm{GLDS}}(\boldsymbol{X}, \boldsymbol{Y}) = \bar{\boldsymbol{b}}_x^{\mathrm{T}} \bar{\boldsymbol{R}}^{-1} \bar{\boldsymbol{b}}_y \,. \tag{41.12}$$

The mapping $\boldsymbol{X} \to \bar{\boldsymbol{b}}_x$ is defined as

$$\boldsymbol{X} \to \frac{1}{T_x} \sum_{t=1}^{T_x} \boldsymbol{b}(\boldsymbol{x}_t) \,. \tag{41.13}$$

The vector $\bar{\boldsymbol{b}}_y$ is defined in an analogous manner to (41.13). $\bar{\boldsymbol{R}}$ is a correlation matrix derived from a large set of data from multiple languages and many speakers and is usually diagonal.

In a general language recognition task, a set of SVM language models $\{f_l\}$ are trained to represent the target languages $l = 1, \cdots, L$. The sequence of vectors $\boldsymbol{X}$ is extracted from a test utterance, mapped to an average expansion using (41.13), and converted to a set of scores $s_l = f_l(\bar{\boldsymbol{b}}_x)$. The SVM scores s_l are usually transformed to s'_l using a likelihood-ratio-type calculation

$$s'_l = s_l - \log\left(\frac{1}{M-1} \sum_{j \neq l} e^{-s_j}\right) . \tag{41.14}$$

The transformation (41.14) assumes that SVM scores behave like log-likelihood ratios (see [41.12, 13] for related discussions).

Several computational simplifications are available for implementing an SVM system for language recognition. For instance, language recognition scoring can be reduced to a single inner product. Also, language model training can be simplified to a linear kernel in Fig. 41.3. The reader is referred to [41.13] for additional details.

41.3 Token-Based Methods

Unlike acoustic approaches in which fixed-duration windows of the speech signal are used to extract features for classification, token-based approaches segment the input speech signal into logical units or tokens. These units could be based on a phonetic segmentation (e.g., phones), or a data-driven segmentation (e.g., GMM mixture component tokens [41.14]). Features are then extracted from the token stream, and classification is performed using a language model. Figure 41.4 illustrates the process.

This basic architecture allows for a variety of token-based approaches, but the phoneme recognition followed by language modeling (PRLM) approach originally described in [41.15, 16] has been shown to work well for language recognition. The remainder of this section will focus on PRLM and related techniques that model the phonotactic properties (phone stream dependencies) of languages. Other token-based techniques that should be noted include the work of [41.14] in which abstract tokens were used in place of phonetic units to achieve near state-of-the-art language recognition performance.

41.3.1 Tokens

In the PRLM framework, the tokenizer is a phone recognizer (PR) that converts an input speech signal into a sequence of phonetic units (phones). Each phone and its surrounding contexts are then combined to form phone n-grams, the features used for scoring. These features are scored against n-gram language models that correspond to specific target languages to make language decisions. Intuitively, these features represent the phonotactic dependencies (the rules governing the distribution of phonetic sounds within a language) between individual phonetic units. In an early work [41.17], *House* and *Neuberg* used transcribed data to demonstrate the feasibility of exploiting phonotactics to perform automatic language identification. Figure 41.5 illustrates the process.

Phone Recognition

Phone recognition is typically accomplished using hidden Markov models (HMMs) to represent phone units. Figure 41.6 shows a standard topology for different phonetic units. During recognition, these models are combined to allow any phone to transition to any other phone with equal probability. This configuration is often referred to as an *open phone loop* or *null* grammar, as shown in Fig. 41.7.

The phone decoding process is based on a sequence of observations $\boldsymbol{X} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T)$, where T is the number of speech frames. The observation $\boldsymbol{x}_t$ is a vector of acoustic features, usually mel-frequency cepstrum coefficients (MFCC) [41.1] or perceptual linear predic-

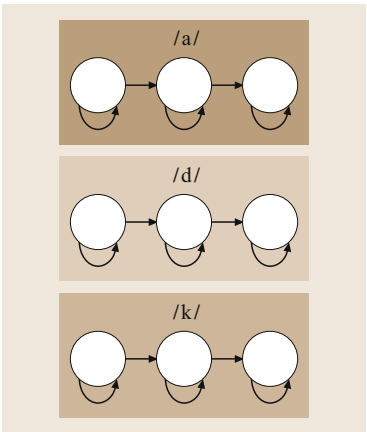


Fig. 41.6 HMM: typical HMM topology for phonetic units

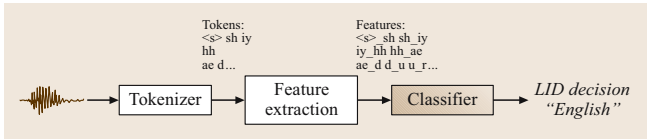


Fig. 41.4 Token-based language recognition

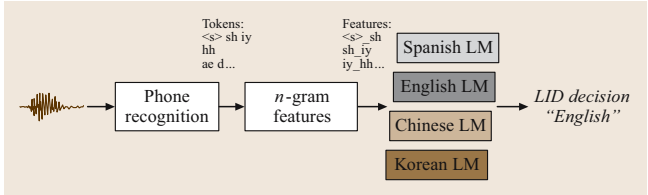


Fig. 41.5 PRLM: phone recognition followed by language modeling

Amazon EX1010 Page 821

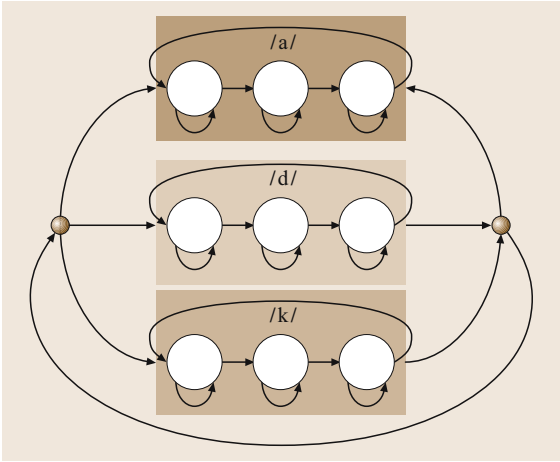


Fig. 41.7 Open phone loop (*null* grammar) with phone HMMs

tion (PLP) coefficients [41.18]). These vectors may also include dynamic features (i. e., first and second differences). Using standard assumptions (see Part E of this Handbook), the probability of this sequence with respect to the language model λ is

$$p(\boldsymbol{X}|\lambda) = \sum_{\forall \boldsymbol{s}} \prod_{t=1}^{T} p(\boldsymbol{x}_t|s_t, \lambda) \cdot P(s_t|s_{t-1}, \lambda) \,, \quad (41.15)$$

where $\boldsymbol{s} = s_1, \cdots, s_t, \cdots, s_T$ is a hypothesized sequence of states at times $t = 1, \cdots, T$. Each state s_t of the model has an observation probability $p(\boldsymbol{x}_t|s_t, \lambda)$ and a series of transition probabilities between all states and s_t. For simplicity, a bigram model $P(s_t|s_{t-1}, \lambda)$ has been selected for the state transition model. With a null grammar, the probabilities $P(s_t|s_{t-1}, \lambda)$ are uniform for fixed s_{t-1}. Typically, the observation probability $p(\boldsymbol{x}_t|s_t, \lambda)$ is modeled as a mixture of Gaussians resulting in a continuous-density HMM.

Given a sequence of observation vectors $\boldsymbol{X}$, the phone recognizer attempts to decode the maximum likelihood sequence of HMM states corresponding to $\boldsymbol{X}$:

$$\operatorname*{argmax}_{\boldsymbol{s}} P(\boldsymbol{s}|\boldsymbol{X}) \,. \quad (41.16)$$

Applying Bayes' rule and recognizing that the maximization is independent of $p(\boldsymbol{X})$:

$$\operatorname*{argmax}_{\boldsymbol{s}} \frac{p(\boldsymbol{X}|\boldsymbol{s})p(\boldsymbol{s})}{p(\boldsymbol{X})} = \operatorname*{argmax}_{\boldsymbol{s}} p(\boldsymbol{X}|\boldsymbol{s})P(\boldsymbol{s}) \,. \quad (41.17)$$

Equation (41.17) is the Viterbi approximation of (41.15); the goal is to maximize the following probability

$$p(\boldsymbol{X}|\boldsymbol{s}, \lambda)P(\boldsymbol{s}|\lambda) = \prod_{t=1}^{T} p(\boldsymbol{x}_t|s_t, \lambda) \cdot P(s_t|s_{t-1}, \lambda) \quad (41.18)$$

by selection of the state sequence $\boldsymbol{s}$.

The Viterbi algorithm is generally used to search the space of possible state sequences and a variety of pruning strategies can be applied to keep the search problem tractable. For language recognition, context independent phonetic units are typically used for decoding (see [41.19] as an exception). For a more-detailed description of HMM models of speech refer to Part E of this Handbook.

Phone Recognizer Training

A vital consideration in the training of phone recognizers is the availability of phonetically or orthographically labeled speech for multiple languages. These phone transcripts can be obtained from manual labeling by a trained listener or automatic labeling from a word transcript and pronunciation dictionary. As discussed later, the languages of the phone recognizers need not be those of the target languages. Phonetically transcribed corpora are relatively uncommon, and for a number of years were available only as part of the Oregon Graduate Institute multilanguage telephone speech (OGI-TS) corpus [41.20], which contained phonetically hand-labeled speech for six languages (English, German, Hindi, Japanese, Mandarin, and Spanish) collected over telephone channels. More recently, investigators have employed other corpora for training phone recognizers, including Switchboard (English) [41.21] (e.g., [41.22, 23]), CallHome (Arabic, Japanese, Mandarin, and Spanish) [41.21] (e.g., [41.22, 23]), and SpeechDat-East (Czech, Hungarian, Polish, Russian, and Slovak)d [41.24] (e.g., [41.8]).

41.3.2 Classifiers

Using the sequence of phone tokens, a number of different classification techniques can be applied to make a language recognition decision. Two popular techniques are described here in detail: PRLM, a generative model of phonetic n-gram sequences, and PR-SVM-LATT (phone recognition followed by lattice-based support vector machines), a discriminative approach to language recognition. Both approaches assume that languages differ in their phonotactic characteristics and that these differences will cause a phonetic recognition system to produce different distributions of token sequences.

Amazon EX1010 Page 822

PRLM

The basic PRLM decision rule is described by

$$L^* = \underset{L}{\operatorname{argmax}}\, P_L(W|\boldsymbol{X})\,, \tag{41.19}$$

where $\boldsymbol{X}$ are the acoustic observations, W is the hypothesized token sequence $W = w_1, \ldots, w_N$, and L^* is the optimal language decision. The term $P_L(W|\boldsymbol{X})$ can be decomposed using Bayes' rule:

$$P_L(W|\boldsymbol{X}) = \frac{\overbrace{p(\boldsymbol{X}|W)}^{\text{same } \forall L} \cdot P_L(W)}{\underbrace{p(\boldsymbol{X})}_{\text{same } \forall L}}\,. \tag{41.20}$$

For a given tokenizer, the terms $p(\boldsymbol{X}|W)$ and $p(\boldsymbol{X})$ are the same across language models, since $\boldsymbol{X}$ is known and W is determined from the phone recognition decoding process described above. As such, the following simplified PRLM decision rule can be applied:

$$L^* = \underset{L}{\operatorname{argmax}}\, P_L(W)\,. \tag{41.21}$$

As (41.21) suggests, only language model scores are used to make a language recognition decision. Phone recognition acts solely to discretize the input speech signal.

In real implementations $P_L(W)$ is approximated by an n-gram language model of fixed order,

$$P_L(W) \approx \prod_{i=1}^{N} \overbrace{P_L(w_i|w_{i-1}\ldots w_{i-(n-1)})}^{\text{language model}}\,. \tag{41.22}$$

Here, the probability $P_L(w_i|w_{i-1}\ldots w_{i-(n-1)}) = P_L(\hat{w}_i)$, is a look up of the frequency of occurrence of n-gram $\hat{w}_i = w_{i-n+1}w_{i-n+2}\cdots w_i$ in language L's training data. n-grams are a simple yet effective way of capturing the context preceding a token. For the language recognition problem, n-grams of order two and three (i. e., bigrams and trigrams) are commonly employed. Increasing the n-gram order should theoretically increase the ability of language recognition modeling techniques to incorporate longer term dependencies, but in practice these models are difficult to estimate given the paucity of training data. As the order n increases, the number of possible unique n-grams increases exponentially as $|\mathcal{W}|^n$, where $|\mathcal{W}|$ is the size of the tokenizer's phone inventory. Standard smoothing techniques have been shown to help mitigate the n-gram estimation problem for language recognition tasks [41.25]. Other language modeling techniques, such as binary decision trees, have also been used successfully to model n-gram sequences with $n > 3$ [41.26].

Lattice-Based PRLM. As stated above, the standard 1-best PRLM model relies solely on the probability of the phone token sequence in its decision rule. This acts as an approximation of the acoustic hypothesis space generated by the underlying HMM phonetic models for a given input.

In [41.22], a better approximation using an extension of the 1-best PRLM model is proposed in which posterior probabilities of phone tokens are incorporated into the estimation of the language models and the observation likelihoods. In this model, estimates of expected counts derived from phone lattices are used in place of 1-best counts during LM training and scoring.

To derive this extension, the standard 1-best language modeling equation can be reformulated in log form as

$$\log P_L(W) = \sum_{\forall \hat{w}} \mathrm{C}(\hat{w}) \log P_L(w_i|w_{i-1}, \ldots, w_{i-n+1})\,, \tag{41.23}$$

where

$$\hat{w} = w_i w_{i-1} \ldots w_{i-n+1} \tag{41.24}$$

and $C(\hat{w})$ is the count of the n-gram $\hat{w}$ in the sequence W. In (41.23), the sum is performed over the *unique* n-grams $\hat{w}$ in W.

In the lattice formulation, n-gram counts from the 1-best hypothesis are replaced with expected counts from a phone lattice generated by the decoding of an input message:

$$E_{\mathcal{L}}[\log P_L(W)] = \sum_{\forall \hat{w}} E_{\mathcal{L}}\left[\mathrm{C}(\hat{w})\right] \log P_L(w_i|w_{i-1}\ldots, w_{i-n+1})\,.$$

Like the 1-best hypothesis, the phone lattice is only an approximation of the acoustic hypothesis space for a given speech utterance. There is evidence that both the quality of the phonetic models and the phone lattices used for language recognition are important [41.19,27]. Many methods of generating lattices have been proposed in the automatic speech recognition (ASR) literature [41.28–30]. The Viterbi n-best technique with a fixed number of trace-backs per state is often used [41.31].

Parallel PRLM (PPRLM)

An important aspect of the PRLM method, implied by (41.21), is that the operations of tokenization (by the phone recognizer) and language modeling are decoupled. As a consequence, it is not necessary for the phone

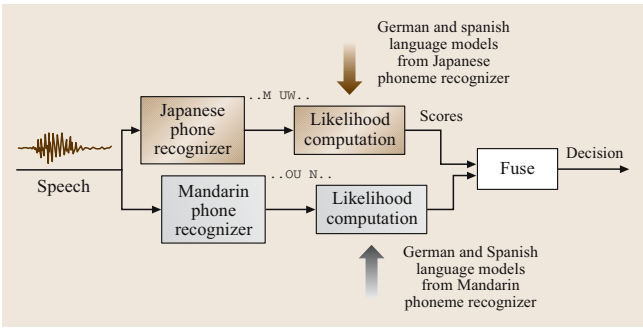


Fig. 41.8 PPRLM: *parallel* phone recognition followed by language modeling

recognizer to be trained in any of the target languages as long as the tokenization of the target languages is sufficiently distinct. In a natural extension of this observation, *Zissman* [41.15] proposed running phone recognizers for multiple languages in parallel in order to provide more-diversified coverage of the phones encountered in target languages. The resulting system configuration, commonly known as parallel PRLM (PPRLM), employs a bank of phone recognizers, each of which is followed by a set of language models trained from the corresponding token sequences that it generates. As with PRLM, the multiple phone recognizers are trained using phonetically or orthographically labeled speech, but need not be of any of the target languages. An example is shown in Fig. 41.8 where parallel Japanese and Mandarin phone recognizers are used to tokenize speech in a German versus Spanish language recognition task. The next section will describe methods by which the individual outputs are fused to produce a final score.

Language recognition systems based on PPRLM were found to outperform their competitors [41.5] and dominated the field of language recognition for a number of years. More recently, employing banks of lattice-based phone recognizers has led to further language recognition performance improvement [41.22]. Efforts have also been made to reduce the computation load of a PPRLM system by employing a single multilingual phone recognizer but to date these approaches have not proved to be more effective than PPRLM [41.32].

PPR-SVM-LATT

The PPR-SVM-LATT approach is an application of SVM classification for token streams [41.33]. For the PPR-SVM-LATT classifier, the phone token sequence for each utterance $\boldsymbol{X}$ in a language corpus is used to compute a vector of n-gram probabilities,

$$\boldsymbol{b} = \begin{pmatrix} c_1 p(\hat{w}_1|\boldsymbol{X}) \\ c_2 p(\hat{w}_2|\boldsymbol{X}) \\ \vdots \\ c_N p(\hat{w}_N|\boldsymbol{X}) \end{pmatrix} . \tag{41.25}$$

The probability $p(\hat{w}_i|\boldsymbol{X})$ represents the frequency of occurrence of n-gram $\hat{w}_i$ in the utterance $\boldsymbol{X}$, and $N = |\mathcal{W}|$ is the number of possible n-grams. The n-gram probabilities can be computed from counts from the 1-best phone sequence or from expected counts from phone lattices. Experiments have determined that expected counts from lattices produce better performance.

The values c_i in (41.25) are weights that normalize the components of the vector to a common scale. Typically, a weighting such as the term-frequency log likelihood ratio (TFLLR) [41.33] can be applied. For TFLLR,

$$c_i = \frac{1}{\sqrt{p(\hat{w}_i|\{\boldsymbol{X}_j\})}} , \tag{41.26}$$

the probability $p(\hat{w}_i|\{\boldsymbol{X}_j\})$ is calculated across all utterances of all languages in the training corpus.

After converting utterances to vectors, a simple linear kernel, $K(\boldsymbol{b}_1, \boldsymbol{b}_2) = \boldsymbol{b}_1^{\mathrm{T}} \boldsymbol{b}_2$, is used, where the $\boldsymbol{b}_i$ are the expansion of two utterances as in (41.25). Training and scoring are done in a manner analogous to the cepstral SVM.

41.4 System Fusion

The previous sections of this chapter described a variety of methods available for implementing a language recognition system. In practice, additional performance improvements can be obtained by running multiple systems in parallel and fusing the individual scores using a back-end. If the recognizers do not behave uniformly (i. e., their errors are not completely correlated) then combining the scores may make it possible to exploit complementary information that exists in the scores. Score fusion was first adopted as a means of combining the scores produced by a PPRLM language recognizer and has since been applied to fusing scores of differ-

ent types of recognizers, such as those described in Sects. 41.2 and 41.3.

41.4.1 Methods

For purpose of this discussion it is convenient to divide approaches to fusion into two types, rule based and classifier based. Rule-based fusion applies a fixed formula to the scores, such as a weighted-sum rule or product rule, where weights are either uniform or are chosen empirically using a set of development data. One example of a rule-based method for combining the scores in a PPRLM system will be discussed below. The second approach to fusion utilizes a development data set to train a classifier from a feature vector constructed from the individual system scores. In theory, the joint statistics among the scores (elements of the feature vector) can be exploited by the classifier to produce overall performance that is superior to that achieved with a rule-based back.end.

Product-Rule Fusion

A method for fusing the scores of a PPRLM language recognition system using the product rule was proposed by *Zissman* [41.5]. Assume a PPRLM language recognizer that uses K phone recognizers to detect L languages. For each test input, a set of KL (linear) scores is generated, with each phone recognizer producing L scores. For an input utterance $\boldsymbol{X}$, single per-language scores $y(\boldsymbol{X}|l)$ are computed by first normalizing the likelihoods $s_{k,l}$ produced by phone recognizer k and then multiplying the normalized per-language values across all phone recognizers:

$$y(\boldsymbol{X}|l) = \prod_{k}^{K} \frac{s_{k,l}}{\sum_{l}^{L} s_{k,l}} \,. \tag{41.27}$$

The normalization step puts scores across phone recognizers into a common range, and the multiplication step creates a final score. Probabilistically, the first step turns the raw phone recognizer scores into posteriors (with an assumption of equal priors) and the second step computes a joint probability of the observed data under the assumption that the information sources (individual PRLM systems) are independent. A potential drawback of product-rule fusion is that a single score close to zero may have an undesirably large impact on the final result. Despite the apparent inaccuracy of the independence assumption, the product-rule method proved to be an effective way of combining PPRLM scores [41.5].

Classifier Fusion

Classifier-based fusion for PPRLM language recognition systems was originally proposed by *Yan* and *Bernard* [41.34] using neural networks and by *Zissman* [41.35] using Gaussian classifiers, and these authors reported that performance was superior to that obtained with the product-rule approach. A block diagram of a Gaussian-based fusion approach for language recognition is shown in Fig. 41.9. The method consists of linear discriminant analysis (LDA) normalization followed by Gaussian classification. Although there can be no guarantee that this particular classifier-based fusion architecture produces optimum results in all language recognition applications, extensive development testing has shown that the technique produces reliably superior performance. Other investigators have reported results using neural network-based back-end fusion [41.22].

The generic block diagram in Fig. 41.9 shows a set of core language recognizers (e.g., GMM, SVM, PRLM, PPRLM), each of which produces a score or score vector for each language hypothesis given an input utterance. These scores are used to form the elements of a feature vector that is transformed via linear discriminant analysis, a method by which the feature vector is projected into a reduced dimensional space in such a way as to maximize interclass separability and minimize intraclass variability. In the figure, the transformed space is of dimension $L-1$, where L is the number of classes (languages) represented in the back-end training data. The transformed vector also has the desirable property that its features are decorrelated. It is important to recognize that the individual recognizers may produce different sized score vectors and that the scores may represent likelihoods of any languages, not just the target languages.

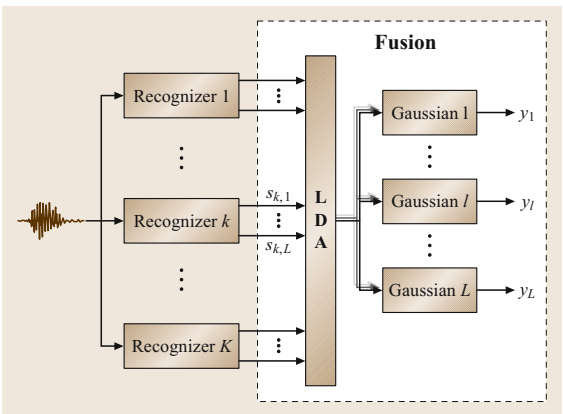


Fig. 41.9 Gaussian-based fusion for language recognition

The transformed feature vectors are then processed by a parallel bank of Gaussian classifiers. Each Gaussian has input dimension $L-1$, a diagonal grand covariance (i. e., pooled across all the back-end training data), and produces an output $y(\boldsymbol{X}|l)$.

41.4.2 Output Scores

The form of the final system scores depends on the nature of the language recognition application. For closed-set identification, the index of the maximum score, $\max_l y(\boldsymbol{X}|l)$, produced using either the rule- or classifier-based method is sufficient to identify the winning language hypothesis. For a detection task, where decisions are made using a fixed threshold, the output score must be formed in such a way as to be consistently meaningful across utterances. Typically, this is accomplished by interpreting the (linear) scores $y(\boldsymbol{X}|l)$ as likelihoods and forming an estimated likelihood ratio $r(\boldsymbol{X}|l)$ between the target language score and the sum (or average) of the others:

$$r(\boldsymbol{X}|l) = \frac{y(\boldsymbol{X}|l)}{\frac{1}{L-1}\sum_{j\neq l}^{L} y(\boldsymbol{X}|j)} \,. \tag{41.28}$$

The estimated likelihood ratio can then be used to set a threshold to achieve a specific application dependent operating point.

Language recognition applications often require that scores be reported as posterior probabilities rather than as likelihoods or likelihood ratios. This may be the case when the downstream consumer (human or machine) requires confidence scores or scores that have been mapped from unconstrained likelihoods or likelihood ratios to a normalized range (0–1 or 0–100). In principle, the scores $y(\boldsymbol{X}|l)$ produced by the back-end fusion for input utterance $\boldsymbol{X}$ can be used directly to generate posterior probabilities $P(l)$ for each target language l via Bayes' rule:

$$P(l|\boldsymbol{X}) = \frac{y(\boldsymbol{X}|l)P_l}{\sum_l^L y(\boldsymbol{X}|l)P_l} \,, \tag{41.29}$$

where P_l represents the prior probability of target language l. In practice, posteriors estimated this way may be unreliable due to the inaccurate assumptions made in constructing the overall system. Rather than treating the scores as true likelihoods, it is preferable to view them as raw scores and to estimate posteriors using another classifier.

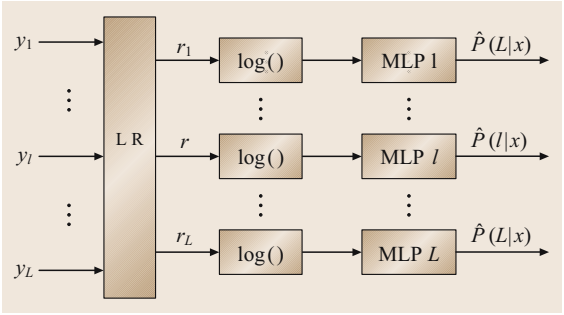


Fig. 41.10 Posterior probability estimation

A block diagram of a method that has been found useful for posterior probability estimation is shown in Fig. 41.10. The fusion back-end log-likelihood ratio estimates $\log r(\boldsymbol{X}|l)$ are treated as raw inputs to a bank of language-dependent single-input logistic regression classifiers (implemented using LNKnet [41.36] as multilayer perceptrons) that produce estimates $\hat{P}(l|\boldsymbol{X})$ of the posterior probabilities. These estimates can then be interpreted by humans as meaningful measures of confidence.

41.5 Performance Assessment

The language recognition approaches described in this chapter have been subject to extensive development, and this section describes their evaluation under conditions that follow protocols established by the US National Institute of Standards and Technology (NIST). Since 1996, NIST has coordinated several evaluations with the goal of assessing the existing state-of-the-art in language recognition technology. The most recent language recognition evaluation (LRE 2005) protocol will form the basis of the performance results presented in this section. A description of the 2005 NIST language recognition evaluation plan can be found in [41.37].

41.5.1 Task and Corpus

The task for LRE 2005 was the detection of the presence of one of seven target languages (English, Hindi, Japanese, Korean, Mandarin, Spanish, Tamil) in telephone speech utterances with nominal durations of 30, 10, and 3 seconds selected from unspecified lan-

guages and sources. The evaluation material contained 3,662 segments for each nominal duration. Although NIST evaluated and reported performance results under several conditions, its primary evaluation condition consisted of trials for which the segments had 30 second durations, were from one of the seven target languages, and were from material collected by the Oregon Health Sciences University (OHSU). Unless otherwise stated, the results presented here will be for the NIST primary evaluation condition. For a complete description of LRE 2005, see [41.38].

Training data for the systems described below was drawn from four telephone speech corpora: CallFriend, CallHome, Mixer, and Fisher, and cover 13 languages (for more information on these corpora, see [41.21]). CallFriend and CallHome contain conversations between acquaintances, while Mixer and Fisher contain conversations between strangers. For a detailed description of the use of this material for training the recognizers and the fusion back-end, see [41.23].

41.5.2 Systems

The fully fused system contained five core language recognizer components. For each recognizer, nonspeech frames were removed from the train and test utterances using a GMM-based speech activity detector.

GMM

The Gaussian mixture model (GMM)-based language recognizer used 2048 mixture components, shifted delta cepstral (SDC) coefficients, feature normalization across utterances to zero mean and unit variance on a per-feature basis, and feature mapping [41.3]. The SDC parameters N-d-P-k were set to 7-1-3-7 (Sect. 41.2.1), resulting in a 49-dimension feature vector computed over a 21-frame (210 ms) time span. In addition, gender-dependent language models were trained for each of the 13 languages in the training material. A background model was trained by pooling all the training material, and gender-dependent target language models were trained by adapting from the background model. This method of training permits fast scoring during recognition and was proposed by [41.39] for language recognition. The GMM system produced a total of 26 scores per test utterance.

SVM

The support vector machine (SVM)-based language recognizer used a generalized linear discriminant sequence (GLDS) kernel, expansion into feature space using monomials up to degree three, and SDC coefficients derived as for the GMM recognizer. The features were normalized across utterances to zero mean and unit variance on a per-feature basis. Thirteen language models were constructed from the training material and 13 scores were produced per test utterance.

PPRLM

The PPRLM-based language recognizer used six OGI-trained phone recognizers, a 39-dimensional MFCC feature vector containing cepstra, delta, and double delta coefficients, and trigram language modeling. Each phone recognizer's token sequences were used to train 13 language models, resulting in an output feature vector of dimension 78. The phone recognizers and language models were trained using Cambridge University engineering department's hidden Markov model toolkit (HTK) [41.40].

PPRLM-LATT

A PPRLM based language recognizer using lattices to generate probabilistic counts of phone frequencies for bigram and trigram language modeling. Phone recognizers were trained using material from CallHome (Arabic, Japanese, Mandarin) and Switchboard English (landline and cellular). A total of seven phone recognizers were trained, resulting in 91 (7×13) scores per input utterance. The tokenizers were selected based on results of development testing conducted in conjunction with LRE 2005 ([41.41]).

PPR-SVM-LATT

The PPR-SVM-LATT-based language recognizer used lattices to generate probabilistic counts of phone frequencies that are then classified with support vector machines. Probability vectors were constructed from unigrams and bigrams, and a linear kernel with TFLLR weighting was applied. Lattices of phones were produced using HMM models trained on six OGI languages, as in PPRLM. A total of 78 (6×13) scores were produced per input utterance.

The scores of the core language recognizers were stacked into a 286-dimension feature vector and fed to the fusion back-end for dimension reduction (linear discriminant analysis) and classification (seven Gaussians with pooled diagonal covariances). The outputs of the back-end were treated as estimated target language likelihoods and log likelihood ratios were formed by taking the log of the quantity in (41.28). These scores were treated as the final outputs of the system.

Amazon EX1010 Page 827

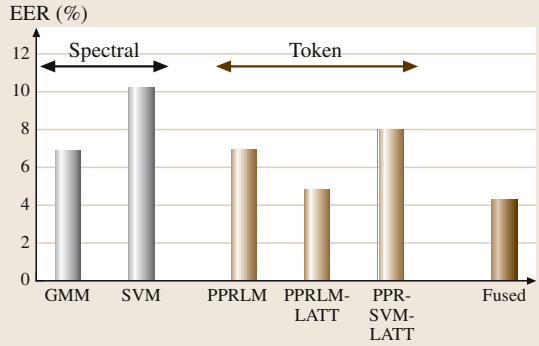


Fig. 41.11 Performance (percentage EER) of spectral, token, and fused language recognition systems on the 2005 NIST LRE primary condition test

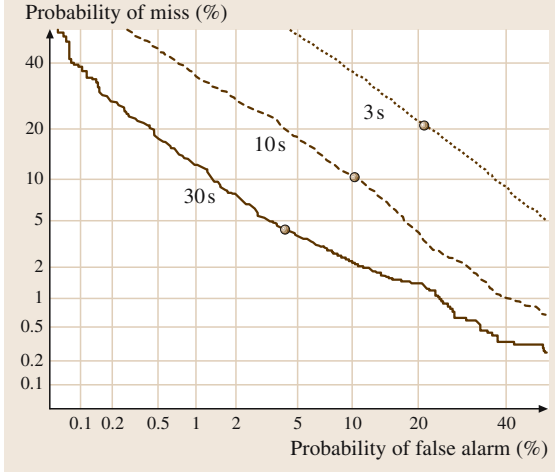


Fig. 41.12 Performance of five-system fusion on the LRE 2005 OHSU target language test segments for durations 30 s (*solid line*), 10 s (*dashed line*), and 3 s (*dotted line*). Filled circles indicate equal error points

41.5.3 Results

NIST language recognition evaluations require participants to provide two outputs for each test utterance, a hard decision and an arbitrarily scaled likelihood score. The hard decisions are used by NIST to evaluate a cost function while the scores are used to sweep out a detection error trade-off (DET) curve [41.42], a variant of the familiar receiver operating characteristic (ROC) curve. To compare detection performance across competing systems, it is convenient to use a summary statistic such as the equal error rate (EER), and this statistic has been adopted for comparison of the language recognition systems described in this section. Results for the NIST primary condition described above, given in percentage EER, are shown in Fig. 41.11. The recognizers have been categorized according to whether they are spectral (GMM or SVM) or token based (PPRLM, PPRLM-LATT, PPR-SVM-LATT). Individually, the best performing recognizer is PPRLM-LATT, while the SVM system produces the most errors. The remaining systems (GMM, PPRLM, and PPR-SVM-LATT) perform comparably. It is worth noting that fusing the scores of the spectral systems (SVM and GMM) results in a large performance gain [41.23]. Fusing the scores of all five recognizers produces the best overall result.

Figure 41.12 shows the DET plots for five-recognizer-system fusion for the LRE 2005 OHSU target language test segments with durations 30, 10, and 3 seconds. Table Table 41.1 gives the corresponding EERs for these DET curves. It is apparent that language recognition performance becomes increasingly challenging as the test segment duration decreases, although recent work has demonstrated that performance at shorter durations can be substantially improved [41.8].

41.5.4 Computational Considerations

Although the focus of the discussion of language recognition systems has been performance, any practical implementation of such a system needs to account for computational complexity. A comparison of process-

Table 41.1 Performance, measured as percentage equal error rate (EER) of full-fusion system for 30 s, 10 s, and 3 s test utterances

Duration	EER(%)
30	4.3
10	10.3
3	21.2

Table 41.2 Processing speed of selected language recognizers as measured with a 3 GHz Xeon processor with 2 GB RAM running Linux. The recognition task involved 12 target languages and five minute audio file. The PPRLM system contained six OGI-trained phone recognizers (no lattices). All factors refer to speeds faster than real time (RT)

Reognizer	Speed
GMM	17 RT
SVM	52 RT
PPRLM	2 RT

ing speed for three types of recognizers is shown in Table 41.2. Note that all factors indicate the degree to which the algorithms are *faster* than real time. All measurements were made using 12 language models and a single 5 minute audio file. The platform was a 3 GHz Xeon processor with 2 GB RAM and Linux OS. The GMM, SVM, and PPRLM systems were described in previous sections of this chapter. The PPRLM language recognizer used six OGI tokenizers and no lattices.

41.6 Summary

This chapter has presented a variety of methods for implementing automatic language recognition systems. Two fundamental approaches, based on either the spectral or phonotactic properties of speech signals, were successfully exploited for the language recognition task. For the spectral systems, these included GMM and SVM modeling using frame-based shifted delta cepstral coefficients. For phonotactic systems, variations of the classic parallel phone recognition followed by language modeling were described. Performance gains can be achieved by fusing the scores of multiple systems operating simultaneously, and a framework for implementing classifier based fusion was described. Finally, performance results were presented for the spectral, token, and fused systems using the standardized protocols developed by NIST. For the latest advances in the field of automatic language recognition, the reader is referred to the most recent proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), the Odyssey Speaker and Language Recognition Workshop, and the International Conference on Spoken Language Processing (Interspeech – ICSLP).

References

41.1 S. Davis, P. Mermelstein: Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences, IEEE Trans. Acoust. Speech Signal Process. (1980) pp. 357–366

41.2 H. Hermansky, N. Morgan, A. Bayya, P. Kohn: Compensation for the Effect of the Communication Channel in Auditory-Like Analysis of Speech, Proc. Eurospeech (1991) pp. 1367–1371

41.3 D.A. Reynolds: Channel robust speaker verification via feature mapping, Proceedings of the, International Conference on Acoustics Speech and Signal Processing (2003) pp. II–53, –56

41.4 P.A. Torres-Carrasquillo, E. Singer, M.A. Kohler, R.J. Greene, D.A. Reynolds, J.R. Deller Jr.: Approaches to language identification using Gaussian mixture models and shifted delta cepstral features, International Conference on Spoken Language Processing (2002) pp. 89–92

41.5 M. Zissman: Comparison of Four Approaches to Automatic Language Identification of Telephone Speech, IEEE Trans. Speech Audio Process. **4**(1), 31–44 (1996)

41.6 A. Dempster, N. Laird, D. Rubin: Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. **39**, 1–38 (1977)

41.7 D.A. Reynolds, T.F. Quatieri, R. Dunn: Speaker Verification Using Adapted Gaussian Mixture Models, Digital Signal Process. **10**(1-3), 19–41 (2000)

41.8 P. Matějka, L. Burget, P. Schwarz, J. Černocký: Brno University of Technology System for NIST 2005 Language Recognition Evaluation, Proc. IEEE 2006 Odyssey: The Speaker and Language Recognition Workshop (2006)

41.9 V.N. Vapnik: *Statistical Learning Theory* (Wiley, New York 1998)

41.10 R. Collobert, S. Bengio: SVMTorch: Support Vector Machines for Large-Scale Regression Problems, J. Mach. Learn. Res. **1**, 143–160 (2001)

41.11 W.M. Campbell: Generalized Linear Discriminant Sequence Kernels for Speaker Recognition, Proceedings of the International Conference on Acoustics Speech and Signal Processing (2002) pp. 161–164

41.12 J.C. Platt: Probabilities for SV Machines. In: *Advances in Large Margin Classifiers*, ed. by A.J. Smola, P.L. Bartlett, B. Schölkopf, D. Schuurmans (MIT, Cambridge 2000) pp. 61–74

41.13 W.M. Campbell, J.P. Campbell, D.A. Reynolds, E. Singer, P.A. Torres-Carasquillo: Support vector machines for speaker and language recognition, Comput. Speech. Lang. **20**(2-3), 210–229 (2006)

41.14 P.A. Torres-Carrasquillo, D.A. Reynolds, J.R. Deller Jr.: Language Identification Using Gaussian Mixture Model Tokenization, Proceedings of the International Conference on Acoustics Speech and Signal Processing (2002) pp. 757–760

41.15 M. Zissman, E. Singer: Automatic Language Identification of Telephone Speech Messages Using Phoneme Recognition and n -Gram Modeling, Proceedings of the International Conference on

Acoustics Speech and Signal Processing (1994) pp. 305–308
41.16 Y. Yan, E. Barnard: An Approach to Automatic Language Identification Based on Language-Dependent Phone Recognition, Proceedings of the International Conference on Acoustics Speech and Signal Processing (1995) pp. 3511–3514
41.17 A. House, E. Neuberg: Toward automatic identification of the language of an utterance. I, Preliminary methodological considerations, J. Acoust. Soc. Am. **62**, 708–713 (1977)
41.18 H. Hermansky: Perceptual linear predictive (PLP) analysis for speech, J. Acoust Soc. Am. **87**, 1738–1752 (1990)
41.19 D. Zhu, M. Adda-Decker, F. Antoine: Different Size Multilingual Phone Inventories and Context-Dependent Acoustic Models for Language Identification, Proc. Interspeech (2005) pp. 2833–2836
41.20 Y.K. Muthusamy, R.A. Cole, B.T. Oshika: The OGI Multi-language Telephone Speech Corpus, International Conference on Spoken Language Processing (1992) pp. 895–898
41.21 Linguistic Data Consortium, http://www.ldc.upenn.edu/ (2007)
41.22 J.L. Gauvain, A. Messaoudi, H. Schwenk: Language Recognition Using Phoneme Lattices, International Conference on Spoken Language Processing (2004) pp. 2833–2836
41.23 W. Campbell, T. Gleason, J. Navritil, D. Reynolds, W. Shen, E. Singer, P. Torres-Carrasquillo: Advanced Language Recognition using Cepstra and Phonotactics: MITLL System Performance on the NIST 2005 Language Recognition Evaluation, Proc. IEEE 2006 Odyssey: The Speaker and Language Recognition Workshop (2006)
41.24 Eastern European Speech Databases for Creation of Voice Driven Teleservices, http://www.fee.vutbr.cz/SPEECHDAT-E/ (2007)
41.25 E. Singer, P.A. Torres-Carrasquillo, T.P. Gleason, W.M. Campbell, D.A. Reynolds: Acoustic, Phonetic, and Discriminative Approaches to Automatic Language Identification, Proceedings of Eurospeech (2003) pp. 1345–1348
41.26 J. Navratil: Recent advances in phonotactic language recognition using binary-decision trees, International Conference on Spoken Language Processing, Vol. 2 (2006)
41.27 P. Matějka, P. Schwarz, J. Černocký, P. Chytil: Phonotactic Language Identification using High Quality Phoneme Recognition, Proceedings of Interspeech (2005) pp. 2833–2836
41.28 F. Weng, A. Stolcke, A. Sankar: New Developments in Lattice-based Search Strategies in SRI's H4 system, Proceedings of DARPA Speech Recognition Workshop (1998) p. 100
41.29 H. Ney, X. Aubert: A word graph algorithm for large vocabulary, continuous speech recognition, International Conference on Spoken Language Processing (1994) pp. 1355–1358
41.30 F. Weng, A. Stolcke, A. Sankar: Efficient Lattice Representation and Generation, International Conference on Spoken Language Processing (1998) pp. 100–100
41.31 R. Schwartz, S. Austin: A Comparison of several approximate algorithms for finding multiple (n-best) sentence hypotheses, Proceedings of the International Conference on Acoustics Speech and Signal Processing (1991) pp. 701–704
41.32 J. Navratil: Automatic Language Identification. In: *Multilingual Speech Processing*, ed. by T.S. Pittsburgh, K. Kirchhoff (Academic, New York 2006)
41.33 W.M. Campbell, J.P. Campbell, D.A. Reynolds, D.A. Jones, T.R. Leek: High-Level Speaker Verification with Support Vector Machines, Proceedings of the International Conference on Acoustics Speech and Signal Processing (2004) pp. I–73–I–76
41.34 Y. Yan, E. Barnard: Experiments for an Approach to Language Identification with Conversational Telephone Speech, Proceedings of the International Conference on Acoustics Speech and Signal Processing (1996) pp. 789–792
41.35 M.A. Zissman: Predicting, Diagnosing and Improving Automatic Language Identification Performance, Proceedings of Eurospeech (1997) pp. 51–54
41.36 R. Lippman, L. Kukolich: LNKnet Pattern Recognition Software Package, http://www.ll.mit.edu/IST/lnknet/ (2007)
41.37 A. Martin, G. Doddington: The 2005 NIST Language Recognition Evaluation Plan, http://www.nist.gov/speech/tests/lang/2005/LRE05EvalPlan-v5-2.pdf (2005)
41.38 A.F. Martin, A.N. Le: The Current State of Language Recognition: NIST 2005 Evaluation Results, Proc. IEEE 2006 Odyssey: The Speaker and Language Recognition Workshop (2006)
41.39 E. Wong, S. Sridharan: Methods to improve Gaussian mixture model based language identification system, International Conference on Spoken Language Processing (2002) pp. 93–96
41.40 Hidden Markov Model Toolkit, http://htk.eng.cam.ac.uk/ (2007)
41.41 W. Shen, W. Campbell, T. Gleason, D. Reynolds, E. Singer: Experiments with Lattice-based PPRLM Language Identification, Proc. IEEE 2006 Odyssey: The Speaker and Language Recognition Workshop (2006)
41.42 A. Martin, G. Doddington, T. Kamm, M. Ordowski, M. Przybocki: The DET Curve in Assessment of Detection Task Performance, Proceedings of Eurospeech (1997) pp. 1895–1898

Amazon EX1010 Page 830

42. Vector-Based Spoken Language Classification

H. Li, B. Ma, C.-H. Lee

This chapter presents a vector space characterization (VSC) approach to automatic spoken language classification. It is assumed that the space of all spoken utterances can be represented by a universal set of fundamental acoustic units common to all languages. We address research issues related to defining the set of fundamental acoustic units, modeling these units, transcribing speech utterances with these unit models and designing vector-based decision rules for spoken language classification. The proposed VSC approach is evaluated on the 1996 and 2003 National Institute of Standards and Technology (NIST) language recognition evaluation tasks. It is shown that the VSC framework is capable of incorporating any combination of existing vector-based feature representations and classifier designs. We will demonstrate that the VSC-based classification systems achieve competitively low error rates for both spoken language identification and verification.

The chapter is organized as follows. In Sect. 42.1, we introduce the concept of vector space characterization of spoken utterance and establish the notion of acoustic letter, acoustic word and spoken document. In Sect. 42.2 we discuss acoustic segment modeling in relation to augmented phoneme inventory. In Sect. 42.3, we discuss voice tokenization and spoken document vectorization. In Sect. 42.4, we discuss vector-based classifier design strategies. In Sect. 42.5, we report several experiments as the case study of classifier design, and the analytic study of front- and back-end. Finally in Sect. 42.6, we summarize the discussions.

42.1 **Vector Space Characterization** 826
42.2 **Unit Selection and Modeling** 827
42.2.1 Augmented Phoneme Inventory (API) 828
42.2.2 Acoustic Segment Model (ASM) 828
42.2.3 Comparison of Unit Selection 829
42.3 **Front-End: Voice Tokenization and Spoken Document Vectorization** 830
42.4 **Back-End: Vector-Based Classifier Design** 831
42.4.1 Ensemble Classifier Design 832
42.4.2 Ensemble Decision Strategy 833
42.4.3 Generalized VSC-Based Classification 833
42.5 **Language Classification Experiments and Discussion** 835
42.5.1 Experimental Setup 835
42.5.2 Language Identification 836
42.5.3 Language Verification 837
42.5.4 Overall Performance Comparison ... 838
42.6 **Summary** 838
References 839

Spoken language classification is the process of determining the language identity of a given spoken utterance. Like many frontiers in pattern recognition, spoken language classification has been formulated in the framework of statistical modeling, where a model is built for each spoken language by fitting the model parameters to maximize the observed data likelihood. A spoken language classification system typically consists of a front- and a back-end. The front-end extracts features in order to characterize spoken languages, while the back-end makes classification decision based on the feature pattern of a test utterance.

When human beings are constantly exposed to a language without being given any linguistic knowledge, they learn to determine the language identity by perceiving some of the language cues. This motivates us to explore useful language cues from phonotactic statistics of spoken languages. Suppose that the world's languages can be characterized by a universal set of acoustic units. Any speech segment can now be considered as a realization of a concatenation of such units analogous to representing a text passage as a sequence of basic symbols, such as English letters or Chinese characters. Using a collection

Amazon EX1010 Page 831

of models, one for each unit, a given speech utterance can be automatically transcribed into a sequence of sound alphabets or *acoustic letters* By grouping adjacent *acoustic letters* to form *acoustic words* we can convert any collection of speech segments into a *spoken document*. Similar to representing a text document as a term vector – as is done in information retrieval and text categorization – a spoken document can now be represented by a vector with key terms defined as the set of all acoustic words. This process is also referred to as vectorization, which leads us into several interesting topics, such as selection of acoustic letters and acoustic words, and dimensionality reduction of term vector.

With recent advances in machine learning techniques, vector space modeling has emerged as a promising alternative solution to multiclass speech classification problem. The vector space approach inherits several attractive properties that we discover in text-based information retrieval. For example, it handles the fusion of different types of language cues in a high-dimensional vector seamlessly. As opposed to similarity-based likelihood measurement, the vector space approach is motivated by discriminative training, which is aimed at minimizing misclassification error. In this chapter, we are interested in practical issues related to vectorization of spoken document and vector-based classifier design.

42.1 Vector Space Characterization

In a typical setting, spoken language classification is usually formulated as a probabilistic pattern recognition problem [42.1] in which the a posterior probability, $P(X|\lambda_l)$, of the l-th language to be considered with a model λ_l given an unknown utterance X, is computed. To make a decision, the language that gives the maximum $P(X|\lambda_l)$ is usually identified as the target language. Many algorithms developed in automatic speech and speaker recognition [42.2] have been adopted and extended to language recognition. Recent advances in acoustic and language modeling [42.3] have also contributed to the technological progress of pattern recognition approaches to spoken language classification.

On another front vector space characterization has become a prevailing paradigm in the information retrieval (IR) community since its introduction in the early 1970s [42.4]. Inspired by recent advances in machine learning, a wide variety of VSC approaches have been adopted successfully to text categorization (TC) [42.5], which essentially refers to the assigning of a category or a topic to a given text document based on the frequencies of co-occurrences of certain key terms [42.6] in the particular category or topic of interest. A similar VSC treatment of spoken documents has also been attempted [42.7]. Since the feature vectors to be considered here are often very high dimensional, sometimes in tens or even hundreds of thousands in some cases, a probabilistic characterization of these vectors is usually not an easy task due to several factors: the lack of training data, the difficulty to specify a good model, and the curse of dimensionality in the estimation of so many parameters with so little data. Nonetheless, vector-based classifier design has some of its own attractive properties that does deserve some attention even without using conventional statistical modeling techniques.

In this chapter, we explore VSC characterization of spoken utterances and the VSC-based spoken language classification system design. We consider that a particular spoken language will always contain a set of high-frequency function words, prefixes, and suffixes, which are realized as acoustic substrings in spoken utterances. Individually, those substrings may be shared across languages. Collectively, the pattern of their co-occurrences can facilitate the discrimination of one language from another.

Suppose that an utterance X, represented by a sequence of speech feature vectors $\boldsymbol{O}$, is decoded or tokenized, into a *spoken document*, $T(X)$, consisting of a series of I acoustic units, $T(X) = \{t_1 \ldots, t_i \ldots, t_I\}$, each unit is drawn from a universal inventory, $U = \{u_1, \ldots, u_j, \ldots u_J\}$, of J acoustic letters shared by all the spoken languages to be considered, such that $t_i \in U$. In the following, we will refer to the process of decoding speech into spoken documents as *tokenization*. We are then able to establish a collection of acoustic words by grouping units occurring in consecutive orders to obtain a vocabulary of M distinct words, $W = \{w_1, \ldots, w_m, \ldots w_M\}$, such that each w_m can be a single-letter word like (u_j), a double-letter word like $(u_j u_k)$, a triple-letter word like $(u_j u_k u_l)$, and so on. Usually the vocabulary size, M, is equal to the total number of n-gram patterns needed to form words, e.g., $M = J + J \times J + J \times J \times J$ if only up to three tokens are considered as a valid acoustic word. Next, we can use

some form of function $f(w_m)$, such as latent semantic indexing (LSI) [42.8], to evaluate the significance of having the word w_m in the document, $T(X)$. We are now ready to establish an M-dimension feature vector, $\boldsymbol{v} = [f(w_1), \ldots, f(w_m), \ldots f(w_M)]^{\mathrm{T}}$ with $\boldsymbol{x}^{\mathrm{T}}$ denoting the transpose of vector $\boldsymbol{x}$, for each spoken document.

It is clear that we need a not-too-small number of fundamental acoustic units to cover the acoustic variation in the sound space. However a large J will result in a feature vector with a very high dimension if we would like to cover as many unit combinations when forming acoustic words. For example, with a moderate value of $J = 256$, we have $M = 65\,792$ even when we only consider words less than or equal to two letters. This is already a very large dimensionality not commonly utilized in speech and language processing algorithms. Finally, a VSC-based classifier evaluates a goodness of fit, or score function $S_l(\boldsymbol{v}) = S(\boldsymbol{v}|\lambda_l)$, between a given vector, $\boldsymbol{v}$, and a model of the l-th spoken language, λ_l, to make a decision. Any vector-based classifier can be used to design spoken language identification and verification systems.

In language identification, we assume that each language to be recognized by the system has been registered and known to the system in advance. Therefore, it is a closed-set test. The objective is to identify the language l, among a pool of L candidates, that has the closest match or the highest score to a given test sample. System performance is often measured by the recognition error rate. On the other hand, in language verification, also known as language detection, the test language may or may not be known in advance to the system. This is therefore an open-set test. The objective of a verification test is to confirm whether a test sample belongs to the language that it is claimed to be. In this case, the test sample is compared against the model of the claimed language to produce a verification score. A decision is then made based upon whether the score is above or below a threshold. Two types of errors, *false alarms* (or false positives) and *misdetections* (or false negatives), are thus resulted. A wide range of error pairs can be obtained depending on the operating verification threshold of the system. To facilitate comparisons between different systems, an equal error rate (EER), indicating the error when the two rates are the same, is usually reported as the system performance. The NIST language recognition evaluation (LRE) is a series of open evaluations for benchmarking the progress of ongoing technology (http://www.nist.gov/speech/tests/index.htm).

Three sets of issues to be addressed in designing a VSC-based language classification system include: (i) fundamental unit selection and modeling, (ii) extraction of a spoken document vector, which can be considered as a front-end design, and (iii) vector-based classifier learning, which is labeled as a back-end design. These will be discussed in detail in the following three sections, respectively. Issues related to the accuracy of tokenization, the discrimination of feature vectors, and the performance of language classifiers will also be discussed in the remainder of the chapter. By having various combinations of the front- and back-ends, we have the flexibility to design a collection of systems that cover a wide spectrum of system performances and computation complexities.

42.2 Unit Selection and Modeling

The first issue to be considered is selection and modeling of the universal set of fundamental acoustic units. Spoken languages, despite sounding different from one another, do share some basic similarities in their acoustic characteristics. In the following, we list the 10 most common words in English and Chinese:

- *the* ... *of* ... *to* ... *a* ... *and* ... *in* ... *that* ... *for* ... *one* ... *is* ... (English)
- *de* (的) ... *yi* (一) ... *he* (和) ... *zai* (在) ... *shi* (是) ... *le* (了) ... *bu* (不) ... *you* (有) ... *zhe* (这) ... *ge* (个) ... (Mandarin)

By coincidence, the top-ranked English word, ("*the*"), and the most common Chinese Mandarin word, ("*de*"), share a similar pronunciation, resulting in a similar unit transcription if a common collection of sound unit models are used to decode both English and Mandarin speech segments containing this pair of words. Nonetheless, we can rely on the co-occurrence statistics of those words to discriminate one language from another. As discussed in Sect. 42.1, a decoder or tokenizer is needed to convert spoken utterances into sequences of fundamental acoustic units specified in an acoustic inventory. We believe that units that are not linked to a particular phonetic definition can be more universal, and are therefore conceptually easier to adopt. Such acoustic units are thus highly desirable for universal language characterization, especially for rarely observed languages or languages

without an orthographic system or a well-documented phonetic dictionary.

A number of variants have been developed along these lines, which were referred to as language-independent acoustic phone models. Hazen reported using 87 phones from the Oregon Graduate Institute multilanguage telephone speech corpus (OGI-TS) corpus [42.9]. *Berkling* [42.10] explored the possibility of finding and using only those phones that best discriminate between language pairs. *Berkling* [42.11] and *Corredor-Ardoy* [42.12] used phone clustering algorithms to find a common set of phones for languages. However, these systems are constrained to operate only when a phonetically transcribed database is available. On a separate front, a general effort to circumvent the need for phonetic transcription can be traced back to *Lee*'s work [42.13] in acoustic segment models (ASMs) in which a collection of ASMs were used to characterize the fundamental set of speech units, and constructed in an unsupervised manner without any linguistic definition. Acoustic words were thus formed by decoding word examples into sequences of such acoustic units, and then utilized for medium-vocabulary isolated-word recognition. A similar ASM approach has recently been adopted for language identification [42.14]. We first discuss the grouping of phone sets from multiple phone inventories of different languages to form a universal collection of units and corresponding phone models.

42.2.1 Augmented Phoneme Inventory (API)

Attempts have been made to derive a universal collection of phones to cover all sounds described in an international phonetic inventory, e.g., the international phonetic alphabet or Worldbet [42.15]. This is a challenging endeavor in practice. Note that these sounds overlap considerably across languages. One possible approximation is to form a superset of phonemes from several languages. We call this set the augmented phoneme inventory (API). This idea has been explored in one way or another in many studies [42.9–11]. A good inventory needs to cover phonetically as many targeted languages as possible. This method can be effective when phonemes from all the spoken languages form a closed set as studied by *Hazen* [42.9]. Results from human perceptual experiments have also shown a similar effect whereby listeners' language identification performance improved by increasing their exposure to multiple languages [42.16].

The API-based tokenization was recently studied [42.17] by using a set of all 124 phones from English, Korean, and Mandarin, plus four noise units, and extrapolating them to the other nine languages in the NIST LRE tasks. This set of 128 units is referred to as API I, which is a proprietary phone set defined for an internal database called the Institute for Infocomm Research language identification (IIR-LID) database. Many preliminary experiments were conducted using the IIR-LID database and the API I phone set. For example, we explored an API-based approach to universal language characterization [42.17], and a text categorization approach [42.7], which formed the basis for the vector-based feature extraction to be discussed in the next section. To expand the acoustic and phonetic coverage, we used another larger set of APIs with 258 phones, from the six languages defined by the OGI-TS database. These six languages all appear in the NIST LRE tasks. This set will be referred to as API II. A detailed breakdown of how the two phone sets were formed with phone set counts for each language is listed in Table 42.1.

Table 42.1 The languages and phone sets of API I and II

API I	Count	API II	Count
English	44	English	48
Mandarin	43	Mandarin	39
Korean	37	German	52
General	4	Hindi	51
		Japanese	32
		Spanish	36
Total	128	Total	258

Once the set of units are defined, models for each individual language can be obtained using conventional hidden Markov model (HMM) [42.18] tools trained on the speech examples specifically collected for the particular language. The collection of these phone models can be used to decode a given utterance by performing parallel phone recognition (PPR) [42.1]. The resulting multiple sequences of phone units can be used to form the spoken document vector corresponding to the given utterance. We will discuss both PPR and spoken document vectorization in later sections.

42.2.2 Acoustic Segment Model (ASM)

The conventional phone-based language characterization commonly used in automatic speech recognition and spoken language identification suffers from two major shortcomings. First, a combined phone set, such as API I mentioned above, from a limited set of multiple languages cannot be easily extended to cover new and rarely observed languages. Second, to train the acoustic mod-

els for each language, a collection of transcribed speech data is needed, but is often difficult to come by for all languages of interest. Furthermore, acoustic mismatches may exist among speech collected in different countries and under various acoustic conditions so that the collection of phone models may not work well for some intended languages under diverse test environments. To alleviate these difficulties, a data-driven method that does not rely on phonetically transcribed data is often preferred. This can be accomplished by constructing consistent acoustic segment models [42.13] intended to cover the entire sound space of all spoken languages in an unsupervised manner using the entire collection of speech training examples for all languages. The ASM framework takes advantage of the concept of language-independent acoustic phone models, and benefits from the unsupervised acoustic modeling technique.

Next, we present an ASM method to establish a universal representation of acoustic units for multiple languages [42.17]. As in any other hidden Markov modeling approaches, the initialization of ASMs is a critical factor to the success of ASM-based systems. Note that the unsupervised, data-driven procedure to obtain ASMs may result in many unnecessary small segments because of the lack of phonetic or prosodic constraints (e.g., the number of segments in a word and the duration of an ASM) during segmentation. This is especially severe in the case of segmenting a huge collection of speech utterances given by a large population of speakers from different language backgrounds. The API approach uses phonetically defined units in the sound inventory. It has the advantage of having phonetic constraints in the segmentation process. By using the API to bootstrap ASMs, we effectively incorporate some phonetic knowledge relating to a few languages in the initialization to guide the ASM training process, which is described as follows.

Step 1
Carefully select a few languages, typically with large amounts of labeled data, and train language-specific phone models. Choose a set of J models for bootstrapping.

Step 2
Use these J models to decode all training utterances in the training corpora. Assume the recognized sequences are *true* labels.

Step 3
Force-align and segment all utterances in the training corpora, using the available set of labels and HMMs.

Step 4
Group all segments corresponding to a specific label into a class. Use these segments to retrain an HMM.

Step 5
Repeat steps 2–4 several times until convergence.

In this procedure, we jointly optimize the J models as well as the segmentation of all utterances. This is equivalent to the commonly adopted segmental maximum likelihood (ML) and k-means HMM training algorithm [42.18] through iterative optimization of segmentation and maximization. We found that API-bootstrapped ASMs are more stable than the randomly initialized ASMs. There are also other ways of initializing ASMs, by imposing language-independent phonetic or prosodic constraints. The API-bootstrapped ASMs outperforms the API by a big margin in our 1996 NIST LRE task. Readers are referred to [42.17] for details of these ASM experiments. Using this single set of ASMs, we can now transcribe all spoken utterances from any language into sequences of units coming from a common alphabet. We call this process universal voice tokenization (UVT). Instead of using a single set of universal ASMs, we can also utilize multiple sets of fundamental units for parallel voice tokenization (PVT). For example, starting with the three-language API I units defined above, we can train three sets of language-dependent ASMs, bootstrapped from the phone models for each of the three languages. In this way, we can generate multiple sets of spoken documents, one from each language-specific tokenizer.

42.2.3 Comparison of Unit Selection

With an established acoustic inventory obtained from the API or the ASM method, we are now able to tokenize any given speech utterance into a token sequence $T(X)$, in a form similar to a text-like document. Note that ASMs are trained in a self-organized manner. We may not be able to establish a phonetic lexicon using ASMs and translate an ASM sequence into words. However, as far as language classification is concerned, we are more interested in a consistent tokenization than the underlying lexical characterization of a spoken utterance. The self-organizing ASM modeling approach offers a key advantage: we no longer require the training speech data to be phonetically transcribed.

In summary, the main difference between the API and ASM methods is the relaxation of phone transcription for segmentation. In the API approach, we train the phone models according to manually transcribed phone

Amazon EX1010 Page 835

labels while via ASMs, the segmentation is done in iterations using automatic recognition results. In this way, ASM gives us two advantages: (i) it allows us to adjust a set of API phones from a small number of selected languages towards a larger set of targeted languages; (ii) the ASM can be trained on the similar acoustic data as are used for the intended task, thus potentially minimizing mismatch between the test data and the API that is trained on a prior set of phonetically transcribed speech.

42.3 Front-End: Voice Tokenization and Spoken Document Vectorization

We are now ready to represent a spoken document or a spoken query by a vector whose dimensionality is equal to the size of the total number of useful features, including the statistics of the units and their co-occurrences. It is precisely with the usage of such high-dimensional vectors that we expect the discrimination capability of the feature vector to improve the performance even when only phonotactic features are used. The voice tokenization (VT) discussed here can typically be accomplished in a way similar to continuous phone recognition to decode a spoken utterance into a sequence of phone units. Readers are referred to [42.3] for an in-depth discussion of the modeling and decoding techniques.

Figure 42.1a,b illustrates a vectorization process that converts a spoken utterance into a document vector with the PVT and UVT front-ends. An unknown testing utterance is represented as a query vector by the front-end. For the PVT front-end with an F-language superset of APIs as initialization, the VSC vectorization (Fig. 42.1a) forms a large composite document vector, or supervector, $\boldsymbol{v} = [\boldsymbol{v}_1^{\mathrm{T}}, \ldots, \boldsymbol{v}_f^{\mathrm{T}}, \ldots, \boldsymbol{v}_F^{\mathrm{T}}]^{\mathrm{T}}$ by stacking F vectors, with each vector, $\boldsymbol{v}_f$, representing the document vector of the f-th language, obtained from the individual phone recognizers. On the other hand, for the UVT front-end, it constructs a single document vector $\boldsymbol{v}$ from the single phone recognizer. Figure 42.2a,b illustrate the language identification and verification processes using a four-language API superset example.

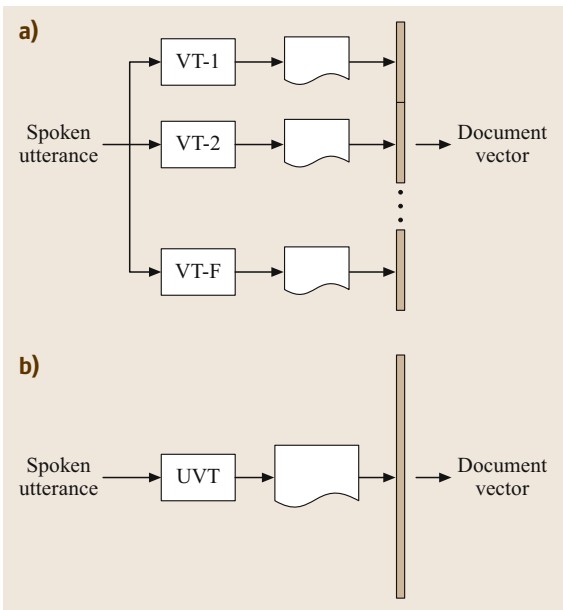


Fig. 42.1a,b Front-end: spoken utterance tokenization and vectorization **(a)** Spoken utterance tokenization and vectorization with PVT. **(b)** Spoken utterance tokenization and vectorization with UVT.

Since high-performance language-dependent acoustic and language models of phones are likely to be unnecessary for tokenization, we no longer need large amounts of training speech samples from each of the spoken languages. Instead, we only require a moderate-sized language-dependent training set of spoken documents in order to obtain a collection of spoken document vectors to be used to train a language-specific classifier for each language. When trigrams with 128 ASM units are incorporated, the vector dimensionality increases to over two million, which is well beyond the capability of current technology. It would be useful to find a balance between the *acoustic resolution*, i. e., the number of units J, needed to model the universal sound space for all languages, and the *language resolution*, i. e., the dimensionality of the spoken document vector M, needed to provide an adequate discriminative power for language identification. Readers are referred to [42.19] for a detailed study.

Many studies on vector-based document representation are available in information retrieval and text categorization literature [42.4, 5, 7, 20, 21]. In this chapter, we focus on language feature characterization that is used to discriminate between languages. Intuitively, sounds are heavily shared across different spoken languages because of the same human speech production mechanism that is used to produce them. The acoustic unit as proposed in Sect. 42.2 allows us to move away

Amazon EX1010 Page 836

from the conventional lexical descriptions of spoken languages. To account for the sequential, acoustic–phonotactic constraints, such as lexical constraints, we introduce the concept of an acoustic word (AW). An AW is typically smaller than a lexical word, and is composed of acoustic letters, such as the set of acoustic segment units, in an n-gram form. By exploiting the co-occurrence statistics of AWs, we can improve the discrimination power of the document vectors by incorporating acoustic words of different orders.

Suppose that we have a token sequence, $\{t_1, t_2, t_3, t_4\}$. We first derive the unigram statistics from the token sequence itself. We then compute the bigram statistics from $(\#t_1)$, (t_1t_2), (t_2t_3), (t_3t_4) and $(t_4\#)$ where the acoustic vocabulary is expanded to the token's right context. Similarly, we can extend this to the trigram statistics using $(\#t_1t_2)$, $(t_1t_2t_3)$, $(t_2t_3t_4)$ and $(t_3t_4\#)$ to account for both left and right contexts. The # sign is a place holder for free (or *do not care*) context. In the interest of manageability, we use only up to token trigrams. In this way, for an acoustic vocabulary of J tokens, we have potentially $J \times J$ bigram and $J \times J \times J$ trigram AWs to form a vocabulary of $M = J + J \times J + J \times J \times J$ AWs. Let us use the API I set defined in Sect. 42.2.1 to illustrate the concept of PVT-based vectorization. As shown in the left half of Table 42.1 that there are three subsets of units of sizes 44, 43, and 37 for English, Mandarin, and Korean, respectively. If we form AWs of up to two letters, this results in a lexicon of size $44^2 + 43^2 + 37^2 + 44 + 43 + 37 = 5278$, which is also the dimensionality of the document vectors.

This *bag-of-sounds* concept [42.22] is analogous to the *bag-of-words* paradigm originally formulated in the context of information retrieval and text categorization. In human languages, some words invariably occur more frequently than others. One of the most common ways to express this notion is known as the Zipf's law [42.23,24], which states that there is always a set of words that dominates most of the other words of a language in terms of their frequency of use. From some preliminary data, the theory can be extended to spoken words as well. This motivates the VSC-based approach.

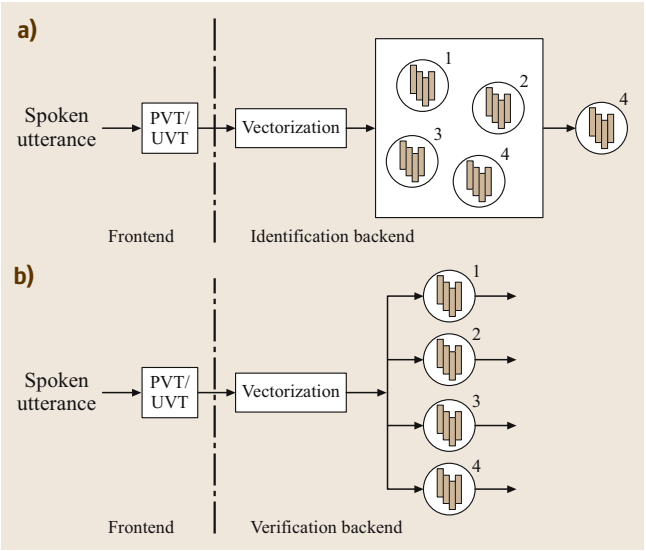


Fig. 42.2a,b Back-end: spoken language classification **(a)** Language identification using multiclass classification (four languages). **(b)** Language verification with independent Bayes decisions (four languages).

42.4 Back-End: Vector-Based Classifier Design

After a spoken utterance is tokenized and vectorized into a high-dimensional vector, language classification can be cast as a vector-based classification problem. There are many ways to construct the VSC back-end. The support vector machine (SVM) framework is a popular choice in many applications [42.5, 25]. Since the training of SVMs is usually optimized on a structural risk-minimization principle [42.26], SVMs have been shown to achieve good performances in several real-world tasks. In the rest of the chapter, we will use the SVM framework to illustrate the design of language classification systems. Other vector-based classifiers can also be utilized. Spoken language identification and verification system block diagrams for a four-language ($L = 4$) case are illustrated in Figs. 42.2a and b, respectively.

To train a two-class SVM, also known as a binary SVM, any input document vector $\boldsymbol{v}$ is labeled as Y_+ or Y_-, depending on whether the input belongs to the desired language category or not. Given a training database of D spoken document vectors, the binary SVM is a classifier of the form $g(\boldsymbol{v}) = \Omega^{\mathrm{T}}\psi(\boldsymbol{v}) + \omega_0$, characterized by a weight vector Ω, an offset ω_0, and a kernel function $\psi(\cdot)$. We can augment Ω with ω_0 and $\psi(\boldsymbol{v})$ with a unity, thus eliminating ω_0 in the rest of our discussion. SVM learning is usually posed as

an optimization problem with the goal of maximizing a margin, i. e., the distance between the separating hyperplane, $\Omega^{\mathrm{T}} \cdot \psi(\boldsymbol{v}) = 0$, and the nearest training vectors, such that if $g(\boldsymbol{v}) > 0$, then $\boldsymbol{v} \in Y_+$, and if $g(\boldsymbol{v}) \leq 0$, then $\boldsymbol{v} \in Y_-$. An extension of this formulation also allows for a wider margin at the cost of misclassifying some of the training examples. We used the SVM^{light} version 6.01 program to train all SVM models discussed in this chapter (http://svmlight.joachims.org/). This package allows us to explore both linear and nonlinear SVM kernels. A linear kernel SVM has $\psi(\boldsymbol{v}) = \boldsymbol{v}$. Other forms of kernels can also be used. Without loss of generality, we limit our discussion to linear-kernel SVM in the rest of this chapter.

42.4.1 Ensemble Classifier Design

When the number of classes L is greater than two, maximum margin classifier designs do not extend easily. Various approaches have been suggested. Some follow the concept of ensemble classifiers by having a collection of binary SVMs, such as *one-versus-rest*, *one-versus-one*, and their variants [42.27]; others build decision trees using top-down greedy algorithms or bottom-up clustering algorithms. We discuss these in detail in the present section.

Given L competing classes with all training vectors labeled as one of the L classes, the training set, T, is divided into a collection of L subsets, $T = \{T_1 \ldots, T_l \ldots, T_L\}$, with all samples that are labeled as class l belong to the set T_l.

One-Versus-Rest (OVR) SVM

This is conceptually the simplest multiclass ensemble classifier. We build L binary SVMs, each with the target l-th class as the positive category, with positive training set T_l, and the rest belonging to the negative category, with training set $\bar{T}_l = \{\boldsymbol{v}_i | i \neq l\}$ of negative examples. This approach is usually computationally expensive since we need to solve L quadratic program (QP) optimization problems, each involving D vectors with a large dimensionality, M. Given a test sample, the OVR decision function chooses the class that corresponds to the maximum value specified by the furthest hyperplane. As illustrated in Fig. 42.3a, the QP solutions result in the dotted line partitions while the decisions are made based on the solid line partitions. This gives errors as shown in the shaded area of Fig. 42.3a. Since there are usually more negative samples than positive ones, the OVR SVMs are usually hard to train because of the imbalance in the data distribution.

One-Versus-One (OVO) SVM

This method constructs binary SVMs for all $L \times (L-1)/2$ pairs of classes, one SVM for each pair of competing classes. When compared with OVR SVM, the OVO SVM method significantly reduces the computational needs in training because there are less training examples to train each pairwise SVM. In total, we need to solve $L \times (L-1)/2$ QP optimization problems. For each SVM, only data from the two competing classes are involved. Hence, we deal with a much smaller QP problem than that in OVR SVM. However, OVO SVM still leaves some undecided regions as shown in the shaded area of Fig. 42.3b.

Generalized SVM

This method is similar to OVR except that it modifies the support vector loss function to deal directly with multiple classes [42.28, 29]. Instead of solving a collection of independent SVMs, this method seeks to solve a single, monolithic QP problem of size $(L-1) \times D$, that

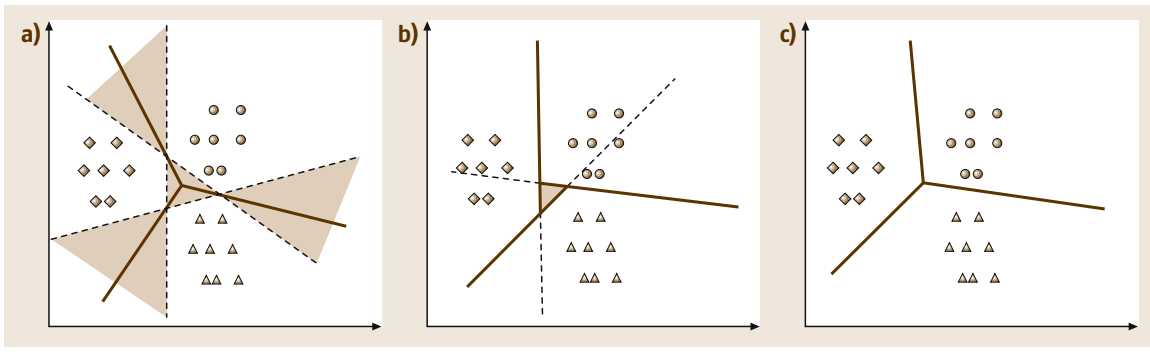


Fig. 42.3a–c SVM-based ensemble classifier design **(a)** One-versus-rest SVM. **(b)** One-versus-one SVM. **(c)** Generalized SVM.

Amazon EX1010 Page 838

maximizes the margins between all class pairs simultaneously. In general, it is computationally more expensive to solve a monolithic problem than multiple binary subproblems, such as is done by OVO SVMs, with the same amount of training data [42.25]. Generalized SVM can lead to L decision regions as illustrated in Fig. 42.3c. Although generalized SVMs are difficult to learn, other multiclass classifiers can serve this purpose. For example a maximum figure-of-merit (MFoM) learning algorithm has been shown to be effective in discriminative multiclass text categorization [42.6] and spoken language identification [42.7].

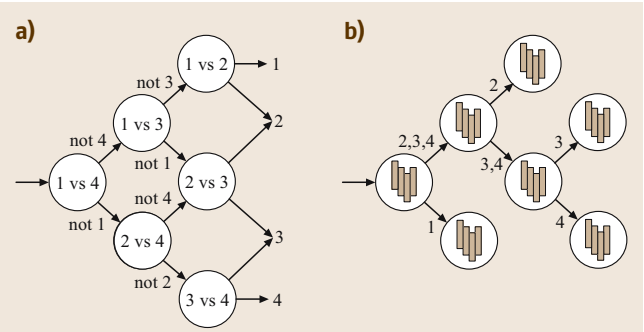


Fig. 42.4a,b Decision strategy **(a)** Direct acyclic graph. **(b)** Binary decision tree.

42.4.2 Ensemble Decision Strategy

The idea of an ensemble decision is to consult a large number of independently constructed classifiers to make a decision based on consensus. It provides a solution to the multiclass classification problem with independently trained binary classifiers. First, we construct many subordinate classifiers, each responsible for removing some uncertainty regarding the class assignment of the test sample; and secondly, we apply classification schemes to make collective decisions. To make a final decision for a multiclass classification problem with an ensemble of binary classifiers, two strategies have been well studied, the maximum wins (max. wins) (MW) and directed acyclic graph (DAG) strategies. The MW strategy makes a collective decision based on a number of individual decisions. Each decision is represented by an individual binary SVM. The collection of individual SVMs is arranged in a flat structure. In practice, we evaluate a test sample against all the $L\times(L-1)/2$ binary SVMs. The class that gains most of the winning votes is then chosen as the identified class.

A DAG is a graph whose edges have an orientation but with no cycles. The DAG strategy [42.30] uses a rooted binary directed acyclic graph which has $L\times(L-1)/2$ internal nodes with L leaves. Figure 42.4a gives an example of a four-class DAG classifier, with six internal nodes and four leaves. Each node represents one of the $L\times(L-1)/2$ binary SVM. To evaluate a test sample $\boldsymbol{v}$, starting at the root node, a binary decision is made at each node to eliminate one candidate class every time a node is visited. The node that survives all the eliminations is eventually chosen as the identified class. Readers can walk through the example in Fig. 42.4a. Thus, for a problem of L-class classification, there are $L-1$ steps of decisions before we arrive at an answer. Furthermore, a node or leaf in DAG is reachable by more than one possible path through the system. Therefore, the decision graph that the path traverses is a DAG, and not simply a tree. In this way, DAG allows for a more efficient representation of redundancies and repetitions that can occur in different branches of the tree, by allowing possible merging of different decision paths. The difference between a DAG and a decision tree can be observed by comparing the tree structures in Fig. 42.4b and the DAG in Fig. 42.4a.

Although both the MW and DAG strategies make decisions based on the same OVO SVMs, the DAG approach demonstrates two clear advantages: (i) with a decision tree architecture, DAG is a much more efficient strategy than the MW during testing. It only involves $(L-1)$ SVM operations, as opposed to $L\times(L-1)/2$ comparisons in MW; and (ii) DAG is amenable to a VC-style bound on the generalization error while bounds on the generalization errors have yet to be establish for the MW method [42.30].

While the MW and DAG strategies can produce accurate class predictions, they lack a structural representation for class similarities. With a large number of classes, we may want to have a meaningful organization of these classes. As shown in Fig. 42.4b, a decision tree presents the classes in a hierarchical structure of clusters and subclusters according to their proximity. It makes a final decision through steps of subdecisions for a test sample. At each node in the tree, a decision is made to reduce the uncertainty of the test samples. Starting from the root node through the successor nodes, we make the final decision when a leaf node is reached.

42.4.3 Generalized VSC-Based Classification

We have discussed designing ensemble classifiers and decision strategies based on high-dimension feature vectors and binary SVMs. In some cases, it is beneficial to

reduce feature dimensions in order to compensate for the lack of training vectors and potential measurement noise in feature extraction. Once we have a vector of lower dimension many well-known probabilistic modeling and machine learning techniques can be used to design classifiers and formulate decision rules.

We also consider an indirect vectorization mechanism based on the scores obtained from the ensemble classifiers discussed earlier. Since these scores characterize the distribution of the outputs of a variety of classifiers, they can be grouped together to form a score *supervector* describing an overall behavior of a score distribution over different classifiers on all competing classes. One example of such a supervector is to concatenate HMM state scores from all competing models to form an overall score vector. This approach has been shown to have good discrimination power in isolated-word recognition [42.31, 32]. Since these score supervectors are usually obtained from a finite set of ensemble classifiers, their dimensionality is usually much smaller than that of the spoken document vectors discussed in Sect. 42.3. These score supervectors are more amendable to using the conventional probabilistic modeling approaches. We call this combination framework generalized VSC-based classification; it uses high-dimensional feature vectors to generate scores from a collection of ensemble classifiers, and forms low-dimensional score supervectors to perform final classifier design and decisions. All the language identification and verification experiments to be discussed in Sect. 42.5 are based on this generalized VSC classification approach.

Feature Reduction

In many cases, it is desirable to reduce the dimensionality of the document vectors to a manageable size so that probabilistic models, such as a Gaussian mixture model (GMM), can be easily utilized. Many dimensionality reduction approaches, such as truncated singular value decomposition (SVD) [42.8] or principal component analysis (PCA) [42.33], have been studied. Suppose that we are given a database with D documents and M distinguished attributes, also known as terms. Let $\boldsymbol{A}$ denote the corresponding $M \times D$ document-term matrix with entries $a_{m,d}$ that represent the importance of the m-th term in the d-th document. SVD effectively reduces the dimensionality by finding the closest rank-R approximation to $\boldsymbol{A}$ in the Frobenius norm, while PCA finds the R-dimensional subspace that best represents the full data with respect to a minimum squared error. Although the SVD or the PCA method finds subspaces that are useful for representing the original high-dimensional vector space, there is no reason to assume that the resulting projections must be useful for discriminating between data in different classes [42.34].

Linear discriminant analysis finds a decision surface, also known as a hyperplane, that minimizes the *sample risk* or *misclassification error* for linearly separable classes. In the two-class case, the linear discriminant function [42.34] is expressed as $g(\boldsymbol{v}) = \Omega^{\mathrm{T}}\boldsymbol{v}$, with $g(\boldsymbol{v})$ representing the signed distance between $\boldsymbol{v}$ and the decision surface $\Omega^{\mathrm{T}}\boldsymbol{v} = 0$. In this way, from the perspective of dimensionality reduction, a multidimensional feature vector $\boldsymbol{v}$ is projected to a one-dimensional space by $g(\boldsymbol{v})$.

Vectorization with Ensemble Scores

If we employ a linear SVM for each of the subordinate classifiers, the outputs $g(\boldsymbol{v})$ from the SVMs then form a vector of signed distance. In this way, a high-dimensional document vector can be effectively reduced to a much lower dimensionality. From an OVR SVM ensemble classifier, we obtain a vector of $Q = L$ dimensionality; from an OVO SVM ensemble classifier, we arrive at a vector of $Q = L \times (L-1)/2$ dimensionality. Clearly, not only do we effectively reduce the dimensionality from a large M to a small Q, but we also represent the spoken document vector in a discriminative space of language pairs.

Studies show that an ensemble classifier performs well when the number of subordinate classifiers Q is set to around $10 \log_2 L$. In general, more subordinate classifiers improve performance, but at a higher computational cost [42.35]. Hence, from the perspective of expressiveness, OVO SVM ensemble classifier seems to be a better choice than OVR SVM when we are dealing with 15 languages as in *CallFriend*, where we have $105 = 15 \times (15-1)/2$ OVO SVM outputs versus 15 OVR SVM outputs (http://www.ldc.upenn.edu/Catalog/project_index.jsp).

Generalized Vector-Based Classifiers and Decision Rules

We mentioned above that generalized SVMs attempt to design ensemble classifiers by considering all possible partitions of the vector space as shown in Fig. 42.3c, which is usually a difficult problem. On the other hand, if we can define a class discriminant function, $S(\boldsymbol{v}|\lambda_l)$ for the l-th class with λ_l as the model for the target l-th class, and be able to consistently rank a goodness-of-fit score between the unknown vector $\boldsymbol{v}$ and the model λ_l, a decision rule for identification can easily be formulated as:

$$\hat{l} = \underset{l}{\operatorname{argmax}}\, S(\boldsymbol{v}|\lambda_l)\,. \tag{42.1}$$

Similarly, a verification decision can be made to accept the claimed class identity l if:

$$S(\boldsymbol{v}|\lambda_l) - S(\boldsymbol{v}|\bar{\lambda}_l) > \tau_l\,, \tag{42.2}$$

with $S(\boldsymbol{v}|\bar{\lambda}_l)$ denoting an antidiscriminant function score that is typically a score representing the collective contribution from all competing classes other than the l-th class, and τ_l denoting the verification threshold for the l-th class.

One good example is a linear discriminant function (LDF)-based score function $S(\boldsymbol{v}|\Omega_l) = \Omega_l{}^{\mathrm{T}}\boldsymbol{v}$ mentioned early in the SVM design, which is simply an inner product between the weight vector Ω_l of the l-th class and the unknown vector $\boldsymbol{v}$. Another commonly used example is to model each class by a GMM, so that we have $S(\boldsymbol{v}|\lambda_l)$, evaluating the likelihood of observing $\boldsymbol{v}$ under GMM λ_l. Artificial neural networks (ANNs) [42.36] are another popular approach that can be used to approximately map the input vectors to the output decisions. Results of using both LDF- and ANN-based generalized classifier design approaches have been recently reported in [42.37] for spoken language classification in the 2005 NIST LRE task. Classifier learning is based on the minimum classification error (MCE) and maximal figure-of-merit (MFoM) training principles which have been extensively studied in and successfully applied to speech recognition [42.38], and text categorization [42.6] tasks.

42.5 Language Classification Experiments and Discussion

We illustrate the VSC framework for the language identification results on the primary subset of 30 s test segments of the 1996 NIST LRE task. On the other hand, language verification benchmarks are established based on the 1996 and 2003 NIST LRE tasks.

42.5.1 Experimental Setup

The training sets for building models came from three sets of corpora, namely: (i) the three-language IIR-LID database [42.7] with English, Mandarin, and Korean; (ii) the six-language OGI-TS (multilanguage telephone speech, http://cslu.cse.ogi.edu/corpora/corpCurrent.html) database with English, German, Hindi, Japanese, Mandarin, and Spanish; and (iii) the 12-language Linguistic Data Consortium (LDC) *CallFriend* database. The overlap between the *CallFriend* database and the 1996 LRE data was removed from the training data as suggested http://www.nist.gov/speech/tests/index.htm for 2003 LRE. As English, Mandarin, and Spanish each have two accented versions in the *CallFriend* database, in all, we were given a 15-language set, which includes the three additional accents. The IIR-LID and OGI-TS databases were only used for bootstrapping the acoustic models as an initial set of phones. Both IIR-LID and OGI-TS databases are telephone speech with phonetic transcriptions. In addition, the *CallFriend* database was used for fully fledged ASM acoustic modeling, vectorization, and classifier design. It contains telephone conversations of the same 12 languages that are in the 1996 and 2003 NIST LRE tasks, but without phonetic transcriptions. The three databases were recorded independently of each other.

In the IIR-LID database, each language contains more than 150 hours of speech, while in the OGI-TS database, each language amounts to less than one hour of speech. Furthermore in the *CallFriend* database, each of the 15 language databases consists of 40 telephone conversations with each lasting approximately 30 min, giving a total of about 20 hours per language. For vectorization and classifier design, each conversation in the training set was segmented into overlapping sessions, resulting in about 12 000 sessions for each duration per language. For testing, there were three different duration settings: 3, 10, and 30 s. The 1996 NIST LRE evaluation data consist of 1503, 1501, and 1492 sessions of speech segments with 3, 10, and 30 s in length, respectively. The 2003 NIST LRE evaluation data consist of 1280 sessions per duration. The test data were labeled with the 12 main languages only, without considering their accent type.

We configured each ASM with a three-state left-to-right hidden Markov model. It was found that a 32-mixture Gaussian state density provides adequate results as compared with a 16- or 8-mixture Gaussian state [42.17]. It was also reported that the ASM set derived from API II (258-ASM) slightly outperforms that from API I (128-ASM) by having a broader acoustic coverage [42.19]. We will adopt the 128-ASM for language identification and the 258-ASM for language verification experiments.

Amazon EX1010 Page 841

Table 42.2 Error rates (ER%) and number of support vectorson 30 s NIST 1996 LRE

No. sessions per language	1000	2000	6000	12 000	Testing cost (no. SVM decisions)
DAG (ER%)	18.2	16.2	14.4	13.9	$L-1$
Max. wins (ER%)	19.0	16.7	15.1	14.5	$L\times(L-1)/2$
No. support vectors	1048	1457	1951	2142	

With 15 languages including accents in the *CallFriend* database, we trained $105 = 15\times(15-1)/2$ OVO SVMs to model the languages and accents. In language identification, we implemented both MW and DAG decision strategies based on OVO SVMs. The conventional language model (LM) scoring techniques used in the parallel phone recognition followed by language modeling (P-PRLM) approaches [42.1] can also be used as a back-end to contrast the VSC back-end discussed earlier. For language verification, to appreciate the contributions of the different front- and back-ends, we construct and test four combination systems using the PVT and UVT front-ends, and the VSC and LM back-ends. In the 1996 NIST LRE task, we built a system to identify 12 languages.

42.5.2 Language Identification

For language identification, the objective is to identify the most likely language given a test sample, as illustrated in Fig. 42.2a. This is usually considered as a closed-set problem assuming that all the languages to be decided are known to the system in advance. We follow the experiment setup in the NIST LRE tasks, and the evaluation is carried out on recorded telephony speech of 12 languages for the 1996 and 2003 NIST LRE: Arabic, English, Farsi, French, German, Hindi, Japanese, Korean, Mandarin, Spanish, Tamil, and Vietnamese.

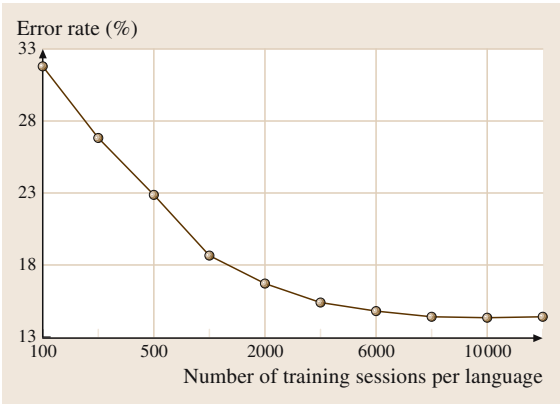


Fig. 42.5 Error rates (ER%) as a function of training set size evaluated on 30 s NIST 1996 LRE

It is well known that the size of the training set often affects the performance of a pattern classification system significantly. Let us study the effects of training set size on system performance using the DAG strategy. We proceed with the document vectors based on bigram AWs derived from the 128-ASM set. In Fig. 42.5, we reported the language identification error rate as a function of the number of training sessions (vectors). As the number of training sessions increase, the training cost of SVMs increase as well. The full corpus includes 12 000 spoken documents, or sessions, for each language/accent. The subset was randomly selected from the full corpus with an equal amount of data from each language. We observe that, when subset size grows beyond 8000 sessions per language, performance begins to saturate.

In Table 42.2, we compare the error rates using different training corpus sizes evaluated on the 30 s NIST 1996 LRE test set. We also report the number of resulting support vectors in the set of binary SVMs. When the number of sessions per language was 1000, there were 2000 training vectors for each language pair. The two-class SVMs gave about 1048 support vectors per binary SVM on average. Note that the number of support vectors increases as training corpus grows, but at a much slower rate than that of the training corpus size. When the training corpus size grows by 12 times, the number of support vectors only doubles. This explains the fact that, beyond 8000 sessions, increasing the training corpus size does not translate into substantial accuracy improvements.

As discussed in Sect. 42.4.1, both the MW and DAG SVM strategies make decisions based on the same OVO SVMs. We further compare the performance of the two strategies in Table 42.2. It is worth pointing out that it is much less expensive to train a set of OVO SVMs than a set of OVR SVMs [42.30]. From the perspective of computation cost in testing, the DAG strategy involves $L/2$ times less computation than MW does. The DAG approach also demonstrates superior performance to that obtained from the MW strategy in terms of accuracy consistency on all tests.

Amazon EX1010 Page 842

42.5.3 Language Verification

In the case of language verification, the goal is to verify whether a language identity claim is true or false. To this end, a typical VSC back-end consists of multiple independent Bayes decisions, each making a decision for a candidate language, as in Fig. 42.2b. Language verification is an open-set hypothesis testing problem. The language verification experiments discussed next follow the procedure in Sect. 42.4.3. We have described the two different front-ends, PVT and UVT, with both the VSC and LM back-ends. To gain further insight into the behavior of each of the front- and back-ends, it is desirable to investigate the performance of each of the four system combinations, namely PVT–LM, PVT–VSC, UVT–LM, and UVT–VSC, where the PVT/UVT front-ends are built on a set of universal ASMs.

Without loss of generality, we deployed the same 258-ASM in two different settings. First, the 258 ASMs were arranged in a six-language PVT front-end. They were redistributed according to their API II definitions into three languages. Second, they were lumped together in a single UVT front-end. The training of 258-ASM was discussed in Sect. 42.2.2.

The PVT/UVT front-end converts spoken documents into high-dimensional vectors as illustrated in Fig. 42.2a,b. The six-language PVT front-end generates vectors of 11 708 ($=48^2+39^2+52^2+51^2+32^2+36^2+48+39+52+51+32+36$) dimensions (PVT vectors), while the UVT front-end generates those of 66 822 $\left(=258^2+258\right)$ dimensions (UVT vectors).

We can use the vectors generated by the PVT and UVT front-ends directly. We can also use the scores generated by the binary classifiers. With the OVO SVMs, we further convert the document vectors into $105=15\times(15-1)/2$ attributes, with each attribute representing a signed distance between a document vector and the decision hyperplane of an OVO SVM. This score supervector represents 99.1% and 99.8% dimensionality reductions from the original PVT and UVT vector dimensions, respectively. We further train 512-mixture GMMs, λ^{x+} and λ^{x-}, for each of the languages, and report the equal error rates (EER in percentage) between misdetections and false-alarms.

The UVT–LM system follows the block diagram of the language-independent acoustic phone recognition approach [42.12]. The PVT–LM is implemented as in [42.1]. The LM back-end uses trigrams to derive phonotactic scores. The results on the 1996 and 2003 NIST LRE tasks are reported in Tables 42.3 and 42.4, respectively.

Table 42.3 Equal error rates (ERR%) comparison of four systems on NIST 1996 LRE

System	30 s	10 s	3 s
PVT–VSC	2.75	8.23	21.16
PVT–LM	2.92	8.39	18.61
UVT–VSC	4.87	11.18	22.38
UVT–LM	6.78	15.90	27.20

Table 42.4 Equal error rates (ERR%) comparison of four systems on NIST 2003 LRE

System	30 s	10 s	3 s
PVT–VSC	4.02	10.97	21.66
PVT–LM	4.62	11.30	21.18
UVT–VSC	6.81	13.75	24.44
UVT–LM	10.81	19.95	30.48

Before looking into the results, let us examine the combinative effect of the front- and back-ends. In these combinative systems, there are two unique front-end settings, PVT and UVT. The PVT converts an input spoken utterance into six spoken documents using the parallel front-end, while the UVT converts an input into a single document. The LM in the PVT–LM and that in the UVT–LM are different; the former has 6×12 n-gram language models while the latter has only 12 language models. That is to say, the former LM classifier is more complex, with a larger number of parameters. On the other hand, the VSC in the PVT–VSC and the VSC in the UVT–VSC are of different complexities as well. Although the dimensionality of the supervector from PVT is lower than that of UVT, the supervector is several times as dense as that of UVT because there are many low-occurrence AWs in the case of UVT, resulting in more-complex SVMs [42.26]. In other words, the VSC classifier in the PVT–VSC is more complex than that in UVT–VSC. In terms of the overall classifier back-end complexity, we rank the four systems from high to low as follows: PVT–VSC, PVT–LM or UVT–VSC, and UVT–LM.

We now summarize what we have discussed: (i) the VSC back-end demonstrates a clear advantage over the LM back-end for the 30 s trials, while LM works better for the 3 s trials in general. This can easily be explained by the fact that the VSC models are designed to capture higher-order phonotactics. As a result, VSC favors long utterances which provide a richer set of long span phonotactic information over short utterances; (ii) the system performance correlates highly with the complexity of system architectures. This can be found

consistently in Tables 42.3 and 42.4, where the PVT–VSC presents the best result with an EER of 2.75% and 4.02% for the 30 s 1996 and 2003 NIST LRE tasks, respectively. It is then followed by PVT–LM, UVT–VSC, and UVT–LM. Note that we can increase the system complexity by including more phone recognizers in PVT. We expect more phone recognizers to further improve the PVT–VSC system performance. As a general remark, the OVO SVM followed by the DAG decision strategy is shown to be computationally effective in both training and testing in language identification. The OVO SVM also delivers outstanding system performances in both identification and verification tasks.

42.5.4 Overall Performance Comparison

We summarize the overall performance of the SVM approach and compare its results with other state-of-the-art systems recently reported in the literature. Only the results of the 30 s segment testing, which represent the primary interest in the LRE tasks, are listed in Table 42.5. Here systems 1–3 are trained and tested on the same database configuration. Therefore, the results can be directly compared. We extract the corresponding results from Tables 42.3 and 42.4. Two sets of recently reported results are worth mentioning. They are listed under systems 4 and 5, and extracted from [42.39] and [42.40], respectively. It is clear from the results in Table 42.5 that the performance of the PVT–VSC system represents one of the best reported results on the 1996 and 2003 NIST LRE tasks.

Table 42.5 Equal error rates (ERR%) benchmark on 30 s NIST 1996/2003 LRE (in [42.40], Russian data are excluded)

	System	1996 LRE	2003 LRE
1	PVT–VSC	2.75	4.02
2	PVT–LM	2.92	4.62
3	UVT–VSC	4.87	6.81
4	Phone lattice [42.39]	3.20	4.00
5	Parallel PRLM [42.40]	5.60	6.60

The proposed VSC-based language classifier only compares phonotactic statistics from spoken documents. We have not explored the use of the acoustic scores resulting from the tokenization process. It was reported that there is a clear win in combining information about the acoustic scores along with the phonotactic statistics [42.12, 40, 41]. Furthermore, the fusion of phonotactic statistics at different levels of resolutions also improves overall performance [42.42]. We have good reasons to expect that fusion among our four combinative systems, or between our systems and other existing methods including the acoustic score classifier [42.41] and GMM tokenizer [42.43], will produce further improvements. This has been demonstrated by some recently reported results, such as the 2.70% EER on the 2003 NIST LRE in [42.39]. However, it is beyond the scope of this chapter to discuss classifier fusions.

42.6 Summary

We have presented a vector-based spoken language classification framework that addresses a range of questions from the theoretical issue of the nature of the mind to more-practical aspects such as design considerations. As a case study, this chapter covers three areas: (i) a vector space characterization strategy for representing spoken documents; (ii) a systematic discussion of language identification and verification formulation cast in the discriminative classifier design strategy; (iii) a number of practical issues for system implementation in applications such as the NIST LRE. Using a conventional SVM classifier design with the PVT front-end and VSC back-end combination, we achieved an EER of 2.75% and 4.02% in the 30 s 1996 and 2003 NIST LRE tasks, respectively. This represents one of the best reported results using a single classifier.

We have studied a number of practical issues in ensemble classifier design, such as the computational needs in the training and testing of different strategies. In language identification, we have successfully demonstrated through a case study that DAG is a better solution than MW in terms of computational efficiency and system accuracy. Multiclass SVM by itself is still an ongoing research issue. Moving forward, recent progress in discriminative classifier designs, such as MFoM [42.6] and decomposition method [42.25] for monolithic solutions to multiclass problems, will result in further improvements. In language verification, we introduced the concept of using SVM decision hyperplanes as the projection directions for dimensionality reduction. This allows us to carry out language verification under the classical GMM framework. The use of

OVO SVM outputs can also be seen as an implementation of the concept of error-correcting output coding (ECOC) [42.44, 45], which makes classification decision by a consensus among subordinate classifiers. One of the central issues in ECOC is choosing a good code. It will be interesting to study other ECOC codes beyond OVO SVMs.

One of the important properties of vectorization is that it handles the fusion of different types of language cues in a high dimensional vector seamlessly. We have successfully implemented the *bag-of-sounds* spoken document vectors using a mix of n-gram statistics of *acoustic letters*. We believe that this framework can be extended to accommodate heterogeneous attributes such as acoustic and prosodic features as well. Whereas fusion of classifier outputs has been the focus of recent research [42.40, 46], vector space modeling explores a new horizon where fusion can take place at the feature level. The same classifier design theory is applicable to other vector-based representation such as GMM *supervector* [42.47], maximum-likelihood linear regression (MLLR) vector [42.48], and generalized linear discriminant sequence (GLDS) vector representation [42.49].

In summary, vector-based spoken language classification benefits from progress in the areas of acoustic modeling, machine learning, and text-based information retrieval. We have successfully treated spoken language classification as a text categorization problem with the topic category being the language identity itself. The same theory also applies to other spoken document classification tasks, such as topic detection and tracking, and *query-by-example* spoken document retrieval. This is another highly anticipated research direction.

References

42.1 M.A. Zissman: Comparison of four approaches to automatic language identification of telephone speech, IEEE Trans. Speech Audio Process. **4**(1), 31–44 (1996), 1

42.2 C.-H. Lee, F.K. Soong, K.K. Paliwal (Eds.): *Automatic Speech and Speaker Recognition: Advanced Topics* (Kluwer Academic, Dordrecht 1996)

42.3 J.L. Gauvain, L. Lamel: Large-vocabulary continuous speech recognition: advances and applications, Proc. IEEE **88**(8), 1181–1200 (2000)

42.4 G. Salton: *The SMART Retrieval System* (Prentice-Hall, Englewood Cliffs 1971)

42.5 F. Sebastiani: Machine learning in automated text categorization, ACM Comput. Surv. **34**(1), 1–47 (2002)

42.6 S. Gao, W. Wu, C.-H. Lee, T.-S. Chua: A MFoM learning approach to robust multiclass multi-label text categorization, Proc. ICML (2004) pp. 42–49

42.7 S. Gao, B. Ma, H. Li, C.-H. Lee: A text-categorization approach to spoken language identification, Proc. Interspeech (2005) pp. 2837–2840

42.8 J.R. Bellegarda: Exploiting latent semantic information in statistical language modeling, Proc. IEEE **88**(8), 1279–1296 (2000)

42.9 T.J. Hazen: *Automatic Language Identification Using a Segment-based Approach* (MIT, Cambridge 1993), MS Thesis

42.10 K.M. Berkling, E. Barnard: Analysis of phoneme-based features for language identification, Proc. ICASSP (1994) pp. 289–292

42.11 K.M. Berkling, E. Barnard: Language identification of six languages based on a common set of broad phonemes, Proc. ICSLP (1994) pp. 1891–1894

42.12 C. Corredor-Ardoy, J.L. Gauvain, M. Adda-Decker, L. Lamel: Language identification with language-independent acoustic models, Proc. Eurospeech, Vol. 1 (1997) pp. 55–58

42.13 C.-H. Lee, F.K. Soong, B.-H. Juang: A segment model based approach to speech recognition, Proc. ICASSP (1988) pp. 501–504

42.14 A.K.V.S. Jayram, V. Ramasubramanian, T.V. Sreenivas: Language identification using parallel subword recognition, Proc. ICASSP (2003) pp. 32–35

42.15 J.L. Hieronymus: ASCII phonetic symbols for the world's languages: Worldbet, Technical Report AT&T Bell Labs (1994)

42.16 Y.K. Muthusamy, N. Jain, R.A. Cole: Perceptual benchmarks for automatic language identification, Proc. ICASSP (1994) pp. 333–336

42.17 B. Ma, H. Li, C.-H. Lee: An acoustic segment modeling approach to automatic language identification, Proc. Interspeech (2005) pp. 2829–2832

42.18 L.R. Rabiner: A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE **77**(2), 257–286 (1989)

42.19 H. Li, B. Ma, C.-H. Lee: A vector space modeling approach to spoken language identification, IEEE Trans. Audio Speech Language Process. **15**(1), 271–284 (2007)

42.20 H.K.J. Kuo, C.-H. Lee: Discriminative training of natural language call routers, IEEE Trans. Speech Audio Process. **11**(1), 24–35 (2003)

42.21 J. Chu-Carroll, B. Carpenter: Vector-based natural languagecall routing, Comput. Linguist. **25**(3), 361–388 (1999)

42.22 H. Li, B. Ma: A phonotactic language model for spoken language identification, Proc. ACL (2005) pp. 515–522

Amazon EX1010 Page 845

42.23 G.K. Zipf: *Human Behavior and the Principal of Least Effort, An Introduction to Human Ecology* (Addison-Wesley, Reading 1949)

42.24 K.S. Jones: A statistical interpretation of term specificity and its application in retrieval, J. Doc. **28**, 11–20 (1972)

42.25 C.-W. Hsu, C.-J. Lin: A comparison of methods for multiclass support vector machines, IEEE T. Neural Netw. **13**(2), 415–425 (2002)

42.26 V. Vapnik: *The Nature of Statistical Learning Theory* (Springer, Berlin, Heidelberg 1995)

42.27 A. Statnikov, C. Aliferis, I. Tsamardinos, D. Hardin, S. Levy: A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis, Bioinformatics **21**(5), 631–643 (2005)

42.28 J. Weston, C. Watkins: Multi-class support vector machines, Tech. Rep. CSD-TR-98-04 (University of London, London 1998)

42.29 Y. Lee, Y. Lin, G. Wahba: Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data, J. Am. Stat. Assoc. **99**(465), 67–81 (2004)

42.30 J.C. Platt, N. Cristianini, J. Shawe-Taylor: Large margin DAG's for multiclass classification, Advances in Neural Information Processing Systems, Vol. 12 (Cambridge, MIT Press 2000) pp. 547–553

42.31 S. Katagiri, C.-H. Lee: A New Hybrid Algorithm for Speech Recognition Based on HMM Segmentation and Discriminative Classification, IEEE Trans. Speech Audio Process. **1**(4), 421–430 (1993)

42.32 K.-Y. Su, C.-H. Lee: Speech Recognition using Weighted HMM and Subspace Projection Approaches, IEEE Trans. Speech Audio Process. **2**(1), 69–79 (1994)

42.33 M. Kobayashi, M. Aono: Vector space models for search and cluster mining. In: *Survey of Text Mining*, ed. by M.W. Berry (Springer, Berlin, Heidelberg 2003)

42.34 R.O. Duda, P.E. Hart, D.G. Stork: *Pattern Classification* (Wiley, New York 2001)

42.35 K. Crammer, Y. Singer: Improved Output Coding for Classification Using Continuous Relaxation, Proc. NIPS (2000) pp. 437–443

42.36 S. Haykin: *Neural Networks: A Comprehensive Foundation* (McMillan, London 1994)

42.37 J. Li, S. Yaman, C.-H. Lee, B. Ma, R. Tong, D. Zhu, H. Li: Language recognition based on score distribution feature vectors and discriminative classfier fusion, Proc IEEE Odyssey Speaker and Language Reognition Workshop (2006)

42.38 S. Katagiri, B.-H. Juang, C.-H. Lee: Pattern Recognition Using A Generalized Probabilistic Descent Method, Proc. IEEE **86**(11), 2345–2373 (1998)

42.39 J.L. Gauvain, A. Messaoudi, H. Schwenk: Language recognition using phone lattices, Proc. ICSLP (2004) pp. 1215–1218

42.40 E. Singer, P.A. Torres-Carrasquillo, T.P. Gleason, W.M. Campbell, D.A. Reynolds: Acoustic, phonetic and discriminative approaches to automatic language recognition, Proc. Eurospeech (2003) pp. 1345–1348

42.41 P.A. Torres-Carassquilo, E. Singer, M.A. Kohler, R.J. Greene, D.A. Reynolds, J.R. Deller Jr.: Approaches to language identification using Gaussian mixture models and shifted delta cepstral features, Proc. ICSLP (2002) pp. 89–92

42.42 B.P. Lim, H. Li, B. Ma: Using local and global phonotactic features in Chinese dialect identification, Proc. ICASSP (2005) pp. 577–580

42.43 P.A. Torres-Carrasquillo, D.A. Reynolds, R.J. Deller Jr.: Language identification using Gaussian mixture model tokenization, Proc. ICASSP (2002) pp. 757–760

42.44 T.G. Dietterich, G. Bakiri: Solving multiclass learning problems via error-correcting output codes, J Artif. Intell. Res. **2**, 263–286 (1995)

42.45 H. Li, B. Ma, R. Tong: Vector-Based Spoken Language Recognition using Output Coding, Proc. Interspeech (2006)

42.46 R. Tong, B. Ma, D. Zhu, H. Li, E.S. Chng: Integrating acoustic, prosodic and phonotactic features for spoken language identification, Proc. ICASSP **1**, 205–208 (2006)

42.47 W.M. Campbell, D.E. Sturim, D.A. Reynolds: Support vector machines using GMM Supervectors for speaker recognition, IEEE Signal Process. Lett. **13**(5), 308–311 (2006)

42.48 A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, A. Venkataraman: MLLR transforms as features in speaker recognition, Proc. Interspeech (2005) pp. 2425–2428

42.49 W.M. Campbell, J.P. Campbell, D.A. Reynolds, E. Singer, P.A. Torres-Carrasquillo: Support vector machines for speaker and language recognition, Comput. Speech. Lang. **20**(2-3), 210–229 (2005)

Amazon EX1010 Page 846

Part H Speech Enhancement

Part H Speech Enhancement

Ed. by J. Chen, S. Gannot, J. Benesty

43 Fundamentals of Noise Reduction
J. Chen, Murray Hill, USA
J. Benesty, Montreal, Canada
Y. Huang, Murray Hill, USA
E. J. Diethorn, Basking Ridge, USA

44 Spectral Enhancement Methods
I. Cohen, Haifa, Israel
S. Gannot, Ramat-Gan, Israel

45 Adaptive Echo Cancelation for Voice Signals
M. M. Sondhi, Basking Ridge, USA

46 Dereverberation
Y. Huang, Murray Hill, USA
J. Benesty, Montreal, Canada
J. Chen, Murray Hill, USA

47 Adaptive Beamforming and Postfiltering
S. Gannot, Ramat-Gan, Israel
I. Cohen, Haifa, Israel

48 Feedback Control in Hearing Aids
A. Spriet, Leuven, Belgium
S. Doclo, Leuven, Belgium
M. Moonen, Leuven, Belgium
J. Wouters, Leuven, Belgium

49 Active Noise Control
S. M. Kuo, DeKalb, USA
D. R. Morgan, Murray Hill, USA

Amazon EX1010 Page 847

43. Fundamentals of Noise Reduction

J. Chen, J. Benesty, Y. Huang, E. J. Diethorn

The existence of noise is inevitable. In all applications that are related to voice and speech, from sound recording, telecommunications, and telecollaborations, to human–machine interfaces, the signal of interest that is picked up by a microphone is generally contaminated by noise. As a result, the microphone signal has to be *cleaned up* with digital signal-processing tools before it is stored, analyzed, transmitted, or played out. The cleaning process, which is often referred to as either noise reduction or speech enhancement, has attracted a considerable amount of research and engineering attention for several decades. Remarkable advances have already been made, and this area is continuing to progress, with the aim of creating processors that can extract the desired speech signal as if there is no noise. This chapter presents a methodical overview of the state of the art of noise-reduction algorithms. Based on their theoretical origin, the algorithms are categorized into three fundamental classes: filtering techniques, spectral restoration, and model-based methods. We outline the basic ideas underlying these approaches, discuss their characteristics, explain their intrinsic relationships, and review their advantages and disadvantages.

43.1 **Noise** 843
43.2 **Signal Model and Problem Formulation** .. 845
43.3 **Evaluation of Noise Reduction** 846
43.3.1 Signal-to-Noise Ratio 846
43.3.2 Noise-Reduction Factor and Gain Function 846
43.3.3 Speech-Distortion Index and Attenuation Frequency Distortion .. 847
43.4 **Noise Reduction via Filtering Techniques** 847
43.4.1 Time-Domain Wiener Filter 847
43.4.2 A Suboptimal Filter 851
43.4.3 Subspace Method 852
43.4.4 Frequency-Domain Wiener Filter ... 855
43.4.5 Short-Time Parametric Wiener Filter 857
43.5 **Noise Reduction via Spectral Restoration** 857
43.5.1 MMSE Spectral Estimator 857
43.5.2 MMSE Spectral Amplitude and Phase Estimators 859
43.5.3 Maximum A Posteriori (MAP) Spectral Estimator 861
43.5.4 Maximum-Likelihood Spectral Amplitude Estimator 861
43.5.5 Maximum-Likelihood Spectral Power Estimator 862
43.5.6 MAP Spectral Amplitude Estimator . 863
43.6 **Speech-Model-Based Noise Reduction** 863
43.6.1 Harmonic-Model-Based Noise Reduction 864
43.6.2 Linear-Prediction-Based Noise Reduction 864
43.6.3 Hidden-Markov-Model-Based Noise Reduction 866
43.7 **Summary** 868
References 869

43.1 Noise

Since we live in a natural environment where noise is inevitable and ubiquitous, speech signals are generally immersed in noise and can seldom be acquired and processed in pure form. Noise can profoundly affect human-to-human and human-to-machine communications, including changing a talker's speaking pattern, modifying the characteristics of the speech signal, degrading speech quality and intelligibility, and affecting the listener's perception and machine's processing of the recorded speech. In order to make voice communication feasible, natural, and comfortable in the presence of noise regardless of the noise level, it is desirable to develop digital signal processing techniques to *clean* the microphone signal before it is stored,

transmitted, or played out. This problem has been a major challenge for many researchers and engineers for decades [43.1].

Generally speaking, noise is a term used to signify any unwanted signal that interferes with measurement, processing, and communication of the desired information-bearing speech signal. This broad-sense definition, however, is too encompassing as it masks many important technical aspects of the real problem. To enable better modeling and removal of the effects of noise, it is advantageous to break the general definition of noise into the following four subcategories: *additive noise* originating from various ambient sound sources, *interfering signals* from concurrent competing speakers, *reverberation* caused by multipath propagation introduced by an enclosure, and *echo* resulting from coupling between loudspeakers and microphones. Combating these four problems has led to the developments of diverse acoustic signal processing techniques. They include noise reduction (or speech enhancement), source separation, speech dereverberation, and echo cancelation and suppression, each of which is a rich subject of research. This chapter will focus on techniques to eliminate or mitigate the effects of additive noise, while the approaches to the remaining three problems will be addressed in other chapters in Parts H and I. So, from now on, we shall narrow the definition of noise to additive noise.

Under this narrowed definition, the observed microphone signal can be modeled as a superposition of the clean speech and noise. The objective of noise reduction, then, becomes to restore the original clean speech from the mixed signal. This can be described as a parameter estimation problem and the optimal estimate of the clean speech can be achieved by optimizing some criterion, such as mean-square error (MSE) between the clean speech and its estimate, signal-to-noise ratio (SNR), the a posteriori probability of the clean speech given its noisy observations, etc. Unfortunately, an optimal estimate formed from a signal processing perspective does not necessarily correspond to the best quality when perceived by the human ear. This inconsistency between objective measures and subjective quality judgement has forced researchers to rethink performance criteria for noise reduction. The objective of the problem has subsequently been broadened, which can be summarized as to achieve one or more of the following three primary goals:

1. to improve objective performance criteria such as intelligibility, SNR, etc.;
2. to improve the perceptual quality of degraded speech; and
3. as a preprocessor, to increase robustness of other speech processing applications (such as speech coding, echo cancelation, automatic speech recognition, etc.) to noise.

The choice of a different goal will lead to a distinct speech estimate. It is very hard to satisfy all three goals at the same time.

With a specified goal (performance criterion), the difficulty and complexity of the noise-reduction problem can vary tremendously, depending on the number of microphone channels. In general, the larger the number of microphones, the easier the noise-reduction problem. For example, when an array of microphones can be used, a beam can be formed and steered to a desired direction. As a result, signals propagating from the desired direction will be passed through without degradation, while signals originating from all other directions will either suffer a certain amount of attenuation or be completely rejected [43.2–4]. In the two-microphone case, with one microphone picking up the noisy signal and the other measuring the noise field, we can use the second microphone signal as a noise reference and eliminate the noise in the first microphone by means of adaptive cancelation [43.5–9]. However, most of today's communication terminals are equipped with only one microphone. In this case, noise reduction becomes a very difficult problem since no reference of the noise is accessible and the clean speech cannot be preprocessed prior to being corrupted by noise. Due to its wide range of potential applications, however, this single-channel noise-reduction problem has attracted a significant amount of research attention, and is also the focus of this chapter.

Pioneering research on the single-channel noise reduction was started more than 40 years ago with two patents by *Schroeder* [43.10, 11]. He proposed an analog implementation of spectral magnitude subtraction. This work, however, has not received much public attention, probably because it was never published in journals or conferences. About 15 years later, *Boll*, in his informative paper [43.12], reinvented the spectral subtraction method but in the digital domain. Almost at the same time, *Lim* and *Oppenheim*, in their landmark work [43.13], systematically formulated the noise-reduction problem and studied and compared the different algorithms known at that time. Their work demonstrated that noise reduction is not only useful in improving the quality of noise-corrupted speech,

but also useful in increasing both the quality and intelligibility of linear prediction coding (LPC)-based parametric speech coding systems. It is this work that has sparkled a huge amount of research attention on the problem.

By and large, the developed techniques can be broadly classified into three categories, i. e., filtering technique, spectral restoration, and model-based methods. The basic principle underlying the filtering technique is to pass the noisy speech through a linear filter/transformation. Since speech and noise normally have very different characteristics, the filter/transformation can be designed to significantly attenuate the noise level while leaving the clean speech relatively unchanged. The representative algorithms in this category include Wiener filters [43.12–18], subspace methods [43.19–26], etc. Comparatively, the spectral restoration technique treats noise reduction as a robust spectral estimation problem, i. e., estimating the spectrum of the clean speech from that of the noise-corrupted speech. Many algorithms have been developed for this purpose, such as the minimum-mean-square-error (MMSE) estimator [43.27,28], the maximum-likelihood (ML) estimator [43.29], and the maximum a posteriori (MAP) estimator [43.30], to name a few. Similar to the spectral restoration technique, these model-based approaches also formulate noise reduction as a parameter estimation problem. The difference between the two is that, in the model-based methods, a mathematical model is used to represent human speech production and parameter estimation is carried out in the model space, which normally has a much lower dimensionality than the original signal space. Typical algorithms in this group include harmonic-model-based methods [43.13, 14], linear-prediction-model-based Kalman filtering approaches [43.31–33], and hidden-Markov-model-based statistical methods [43.34–36]. This chapter attempts to briefly summarize all these techniques and review recent advances that have significantly improved the performance of noise reduction. We will outline the basic ideas underlying these approaches, analyze their performance, discuss their intrinsic relationships, and review their advantages and disadvantages.

43.2 Signal Model and Problem Formulation

The noise-reduction problem considered in this chapter is to recover a speech signal of interest $x(n)$ from the noisy observation

$$y(n) = x(n) + v(n)\,, \tag{43.1}$$

where $v(n)$ is the unwanted additive noise, which is assumed to be a zero-mean random process (white or colored) and uncorrelated with $x(n)$.

In vector/matrix form, the signal model (43.1) is written as

$$\boldsymbol{y}(n) = \boldsymbol{x}(n) + \boldsymbol{v}(n)\,, \tag{43.2}$$

where

$$\boldsymbol{y}(n) = [\; y(n) \;\; y(n-1) \;\; \cdots \;\; y(n-L+1) \;]^{\mathrm{T}}\,,$$

is a vector consisting of the L most-recent samples of the noisy speech signal, superscript ^{T} denotes transpose of a vector or a matrix, and $\boldsymbol{x}(n)$ and $\boldsymbol{v}(n)$ are defined in a similar way to $\boldsymbol{y}(n)$. In this case, the noise-reduction problem is formulated as one of estimating $\boldsymbol{x}(n)$ from the observation $\boldsymbol{y}(n)$.

Applying an L-point discrete Fourier transform (DFT) to both sides of (43.2), we have the following relationship in the frequency domain:

$$Y(n, \mathrm{i}\omega_k) = X(n, \mathrm{i}\omega_k) + V(n, \mathrm{i}\omega_k)\,, \tag{43.3}$$

where

$$Y(n, \mathrm{i}\omega_k) = \sum_{l=0}^{L-1} w(l) y(n-L+l+1)\, \mathrm{e}^{-\mathrm{i}\omega_k l}\,,$$

is the short-time DFT of the noisy speech at time instant n, $\omega_k = 2\pi k/L$, $k = 0, 1, \cdots, L-1$, $w(l)$ is a window function (e.g., Hamming window, Hann window), and $X(n, \mathrm{i}\omega_k)$ and $V(n, \mathrm{i}\omega_k)$ are the short-time DFTs of the clean speech and noise signals, defined in a similar way to $Y(n, \mathrm{i}\omega_k)$. Based on this relationship, the noise-reduction problem can be expressed in the frequency domain as one of estimating $X(n, \mathrm{i}\omega_k)$ from $Y(n, \mathrm{i}\omega_k)$.

43.3 Evaluation of Noise Reduction

The goal of noise reduction is to attenuate the background noise while keeping the desired speech signal unchanged. In order to verify whether this goal has been fulfilled after application of an algorithm, we need some performance criterion. Generally speaking, there are two categories of performance measures, i. e., subjective and objective ones. Subjective measures rely on human listeners' judgements and are typically used for evaluating speech quality. Commonly used subjective tests include a so-called naturalness test [43.37], mean-opinion-score (MOS) tests [43.37, 38], categorical estimation (CE) [43.39], and magnitude estimation [43.39]. In contrast, objective measures, often derived either from pure signal processing considerations or from combinations of signal processing and human speech perception, are independent of listeners' preferences. As we mentioned earlier, one major goal of noise reduction is to improve the quality of degraded speech. As far as speech quality is concerned, the subjective method should be the most appropriate performance criterion because it is the listener's judgment that ultimately counts. However, subjective evaluation is often labor intensive and time consuming and the results are expensive to obtain. In addition, it may not be able to provide timely guidance on algorithm optimization. Comparatively, objective measures are often economic to use. Moreover, some objective criteria can also be integrated into optimization cost functions to guide algorithm design. For these reasons, great efforts have been devoted to the development of objective measures [43.40]. Some widely accepted and used objective measures include signal-to-noise ratio (SNR), log-spectral distance [43.41, 42], Itakura distance, and Itakura–Saito distance [43.40, 43, 44]. Before discussing noise-reduction algorithms, we list some objective measures that are helpful in understanding the performance behavior of noise-reduction algorithms.

43.3.1 Signal-to-Noise Ratio

Signal-to-noise ratio (SNR) is one of the most often used terms in the field of noise reduction; it quantifies how noisy a signal is. This ratio is defined as the signal intensity relative to the intensity of additive background noise and is usually measured in decibels (dB). With the signal model shown in (43.1), the SNR of the microphone signal $y(n)$ is given by

$$\mathrm{SNR} \triangleq \frac{\sigma_x^2}{\sigma_v^2} = \frac{E[x^2(n)]}{E[v^2(n)]}\,, \tag{43.4}$$

where $E[\cdot]$ denotes statistical expectation. A frequency-domain counterpart of the above definition can also be formulated according to the well-known Parseval's relation

$$\mathrm{SNR} = \frac{\int_{-\pi}^{\pi} P_x(\omega)\,\mathrm{d}\omega}{\int_{-\pi}^{\pi} P_v(\omega)\,\mathrm{d}\omega}\,, \tag{43.5}$$

where $P_x(\omega)$ and $P_v(\omega)$ are, respectively, the power spectral densities of the signal $x(n)$ and noise $v(k)$, and ω is the angular frequency. Apparently, the higher the SNR value, the cleaner the signal.

In the context of noise reduction, both the terms a priori SNR and a posteriori SNR are often used. The former usually refers to the SNR of the observed noisy signal, and the latter regards the SNR of the noise-reduced signal. The difference between the two is called the SNR improvement. The higher the SNR improvement, the more effective the noise-reduction algorithm.

43.3.2 Noise-Reduction Factor and Gain Function

The primary issue that we must determine with noise reduction is how much noise is actually attenuated. The noise-reduction factor is a measure of this, and is defined as the ratio between the original noise intensity and the intensity of the residual noise remaining in the noise-reduced speech. Consider the observed signal given in (43.1). If, after noise reduction, the residual noise is denoted by $v_\mathrm{r}(n)$, the noise-reduction factor is mathematically written as

$$\xi_\mathrm{nr} \triangleq \frac{E[v^2(n)]}{E\left[v_\mathrm{r}^2(n)\right]}\,. \tag{43.6}$$

This factor is greater than one when noise is indeed reduced. The larger the value of ξ_nr, the greater the noise reduction.

It is known that a speech waveform consists of a sequence of different events. Its characteristics, therefore, fluctuate over time and frequency. Often, the characteristics of ambient noise also vary across frequency bands. This indicates that we cannot achieve uniform noise-reduction performance over the frequency range of interest; rather, it changes from one frequency band to another. Although the noise-reduction factor gives us an idea of the overall noise attenuation, it does not tell us the noise-reduction behavior from frequency to frequency. A more-useful and insightful frequency-dependent measure of noise reduction can be defined as

the ratio between the spectral densities of the original and residual noise, i. e.,

$$\Psi_{\mathrm{nr}}(\omega) \triangleq \frac{E[|V(\mathrm{i}\omega)|^2]}{E\left[|V_{\mathrm{r}}(\mathrm{i}\omega)|^2\right]} = \frac{P_v(\omega)}{P_{v_{\mathrm{r}}}(\omega)} , \tag{43.7}$$

where $V(\mathrm{i}\omega)$, $V_{\mathrm{r}}(\mathrm{i}\omega)$, $P_v(\omega)$, and $P_{v_{\mathrm{r}}}(\omega)$ are the Fourier spectra and power spectral densities of $v(k)$ and $v_{\mathrm{r}}(k)$, respectively. This measure reflects the noise-reduction gain as a function of frequency. It is therefore logical to call it the noise-reduction gain function.

43.3.3 Speech-Distortion Index and Attenuation Frequency Distortion

Noise reduction is often achieved at a price of speech distortion. The degree of speech distortion is a very complicated issue that depends on many factors, such as the amount of a priori knowledge about the speech and noise that can be accessed. The speech-distortion index is a measure that was introduced to quantify the speech distortion due to a noise-reduction algorithm. With the signal model given in (43.1), if we denote the speech component in the speech estimate as $\hat{x}_{\mathrm{nr}}(n)$, the speech-distortion index is defined as [43.18]

$$\varphi_{\mathrm{sd}} \triangleq \frac{E\left\{\left[x(n)-\hat{x}_{\mathrm{nr}}(n)\right]^2\right\}}{E[x^2(n)]} . \tag{43.8}$$

This parameter is lower bounded by 0 and expected to be upper bounded by 1. A larger value of φ_{sd} means more speech distortion. This index, therefore, should be kept as small as possible.

To measure speech distortion over frequency, we can borrow the concept of attenuation frequency distortion from telecommunication theory. The attenuation frequency distortion is a measure that was developed to assess how a telephone channel preserves the fidelity of a speech signal. It is defined as the change in amplitude of the transmitted signal over a voice band. Here we adjust this concept and define the attenuation frequency distortion for noise reduction as

$$\begin{aligned}\Phi_{\mathrm{sd}}(\omega) &\triangleq \frac{E\left[|X(\mathrm{i}\omega)|^2-|\hat{X}_{\mathrm{nr}}(\mathrm{i}\omega)|^2\right]}{E[|X(\mathrm{i}\omega)|^2]} \\ &= \frac{P_x(\omega)-P_{\hat{x}_{\mathrm{nr}}}(\omega)}{P_x(\omega)} ,\end{aligned} \tag{43.9}$$

where $X(\mathrm{i}\omega)$ and $P_x(\omega)$ are the Fourier spectrum and power spectral density of the clean speech $x(n)$, and $\hat{X}_{\mathrm{nr}}(\mathrm{i}\omega)$ and $P_{\hat{x}_{\mathrm{nr}}}(\omega)$ are, respectively, the Fourier spectrum and power spectral density of the speech component in the noise-reduced signal. Note that whereas (43.8) is a *coherent* measure of distortion, (43.9) is an *incoherent* measure. Therefore, Φ_{sd} is not simply a frequency-domain version of φ_{sd}; they are fundamentally different.

43.4 Noise Reduction via Filtering Techniques

We are now ready to discuss noise-reduction algorithms. Let us first consider filtering techniques. The basic principle behind this class of techniques is to design a linear filter/transformation so that, when the noisy speech is passed through such a filter/transformation, the noise component will be attenuated. The representative algorithms include the time- and frequency-domain Wiener filters, the subspace method, and the parametric Wiener filter.

43.4.1 Time-Domain Wiener Filter

The Wiener filter is one of the most fundamental approaches for noise reduction and can be formulated in either the time or frequency domains. The time-domain Wiener filter is obtained by minimizing the mean-square error (MSE) between the signal of interest and its estimate.

With the signal model given in (43.1), an estimate of the clean speech can be obtained by passing the noisy signal $y(n)$ through a temporal filer, i. e.,

$$\hat{x}(n) = \boldsymbol{h}^{\mathrm{T}}\boldsymbol{y}(n) , \tag{43.10}$$

where

$$\boldsymbol{h} = [\, h_0 \;\; h_1 \;\; \cdots \;\; h_{L-1} \,]^{\mathrm{T}} \tag{43.11}$$

is a finite impulse response (FIR) filter of length L. The error signal between the clean speech sample and its estimate at time n is defined as

$$e_x(n) \triangleq x(n)-\hat{x}(n) = x(n)-\boldsymbol{h}^{\mathrm{T}}\boldsymbol{y}(n) . \tag{43.12}$$

The MSE criterion is then written as

$$J_x(\boldsymbol{h}) \triangleq E\left\{e_x^2(n)\right\} . \tag{43.13}$$

Consider the particular filter,

$$\boldsymbol{h}_1 = [\, 1 \;\; 0 \;\; \cdots \;\; 0 \,]^{\mathrm{T}} . \tag{43.14}$$

This means that the observed signal $y(n)$ will pass this filter unaltered (no noise reduction). Thus the corresponding MSE is

$$J_x(\boldsymbol{h}_1) = E\{v^2(n)\} = \sigma_v^2 \,. \tag{43.15}$$

The optimal estimate $\hat{x}_o(n)$ of the clean speech sample $x(n)$ tends to contain less noise than the observation sample $y(n)$, and the optimal filter that forms $\hat{x}_o(n)$ is the Wiener filter, which is obtained as

$$\boldsymbol{h}_o = \arg\min_{\boldsymbol{h}} J_x(\boldsymbol{h}) \,. \tag{43.16}$$

In principle, for the optimal filter $\boldsymbol{h}_o$, we should have

$$J_x(\boldsymbol{h}_o) < J_x(\boldsymbol{h}_1) = \sigma_v^2 \,. \tag{43.17}$$

In other words, the Wiener filter should be able to reduce the level of noise in the noisy speech signal $y(n)$.

From (43.16), we easily find the Wiener–Hopf equations

$$\boldsymbol{R}_y \boldsymbol{h}_o = \boldsymbol{r}_{yx} \,, \tag{43.18}$$

where

$$\boldsymbol{R}_y = E\{\boldsymbol{y}(n)\boldsymbol{y}^T(n)\} \tag{43.19}$$

is the correlation matrix of the observed signal $y(n)$ and

$$\boldsymbol{r}_{yx} = E\{\boldsymbol{y}(n)x(n)\} \tag{43.20}$$

is the cross-correlation vector between the noisy and clean speech signals.

It is seen from (43.18) that both $\boldsymbol{R}_y$ and $\boldsymbol{r}_{yx}$ need to be known in order to compute the Wiener filter $\boldsymbol{h}_o$. The correlation matrix $\boldsymbol{R}_y$ can be directly estimated from the observation signal $y(n)$. However, $x(n)$ is unobservable; as a result, an estimate of $\boldsymbol{r}_{yx}$ may seem difficult to obtain. But since speech and noise are uncorrelated, we can readily derive

$$\boldsymbol{r}_{yx} = E\{\boldsymbol{y}(n)y(n)\} - E\{\boldsymbol{v}(n)v(n)\} = \boldsymbol{r}_{yy} - \boldsymbol{r}_{vv} \,. \tag{43.21}$$

Now $\boldsymbol{r}_{yx}$ depends on two correlation vectors: $\boldsymbol{r}_{yy}$ and $\boldsymbol{r}_{vv}$. The vector $\boldsymbol{r}_{yy}$, which is the first column of $\boldsymbol{R}_y$, can also be directly estimated from $y(n)$. The vector $\boldsymbol{r}_{vv}$ can be measured during intervals where the speech signal is absent. As a result, the Wiener–Hopf equations given in (43.18) can be rewritten as

$$\boldsymbol{R}_y \boldsymbol{h}_o = \boldsymbol{r}_{yy} - \boldsymbol{r}_{vv} \,. \tag{43.22}$$

If we assume that the matrix $\boldsymbol{R}_y$ is full rank, which is the case in most practical situations, the Wiener filter can be obtained by solving either (43.18) or (43.22), i. e.,

$$\boldsymbol{h}_o = \boldsymbol{R}_y^{-1}\boldsymbol{r}_{yx} = \boldsymbol{R}_y^{-1}\boldsymbol{r}_{yy} - \boldsymbol{R}_y^{-1}\boldsymbol{r}_{vv} = \boldsymbol{h}_1 - \boldsymbol{R}_y^{-1}\boldsymbol{r}_{vv} \,. \tag{43.23}$$

If we define two normalized correlation matrices

$$\tilde{\boldsymbol{R}}_x \triangleq \frac{\boldsymbol{R}_x}{\sigma_x^2} \,, \quad \tilde{\boldsymbol{R}}_v \triangleq \frac{\boldsymbol{R}_v}{\sigma_v^2} \,, \tag{43.24}$$

where $\boldsymbol{R}_x$ and $\boldsymbol{R}_v$ are, respectively, the correlation matrices of the clean speech and noise, which are defined similarly to $\boldsymbol{R}_y$ in (43.19), the Wiener filter in (43.23) can be further expressed as

$$\begin{aligned}\boldsymbol{h}_o &= \left[\boldsymbol{I} - \boldsymbol{R}_y^{-1}\boldsymbol{R}_v\right]\boldsymbol{h}_1 \\ &= \left[\boldsymbol{I} - (\mathrm{SNR}\cdot\tilde{\boldsymbol{R}}_x + \tilde{\boldsymbol{R}}_v)^{-1}\tilde{\boldsymbol{R}}_v\right]\boldsymbol{h}_1 \,,\end{aligned}$$

where $\boldsymbol{I}$ is the identity matrix and SNR is the signal-to-noise ratio of the microphone signal as defined in (43.4).

Now, as SNR approaches infinity, we have

$$\lim_{\mathrm{SNR}\to\infty} \boldsymbol{h}_o = \boldsymbol{h}_1 \,. \tag{43.25}$$

This is expected, since noise reduction is not needed in this case. At the other extreme, where SNR approaches 0, we have

$$\lim_{\mathrm{SNR}\to 0} \boldsymbol{h}_o = \boldsymbol{0} \,, \tag{43.26}$$

where the vector $\boldsymbol{0}$ has the same size as $\boldsymbol{h}_o$ and consists of all zeros. So, in the absence of any speech signal, the Wiener filter passes nothing.

Now we are ready to check whether the Wiener filter can actually reduce the level of noise as expected. To do this, let us examine the noise-reduction factor. Substituting $\boldsymbol{h}_o$ into (43.10), we obtain the optimal speech estimate

$$\hat{x}_o(n) = \boldsymbol{h}_o^T\boldsymbol{y}(n) = \boldsymbol{h}_o^T\boldsymbol{x}(n) + \boldsymbol{h}_o^T\boldsymbol{v}(n) \,. \tag{43.27}$$

It is seen that there are two terms in the right-hand side of (43.27), where the first, $\boldsymbol{h}_o^T\boldsymbol{x}(n)$, is the clean speech filtered by the Wiener filter, and the second, $\boldsymbol{h}_o^T\boldsymbol{v}(n)$, represents the residual noise. So, the noise-reduction factor, according to (43.6), can be written as

$$\xi_{\mathrm{nr}}(\boldsymbol{h}_o) = \frac{E\left\{\left[\boldsymbol{h}_1^T\boldsymbol{v}(n)\right]^2\right\}}{E\left\{\left[\boldsymbol{h}_o^T\boldsymbol{v}(n)\right]^2\right\}} = \frac{\boldsymbol{h}_1^T\boldsymbol{R}_v\boldsymbol{h}_1}{\boldsymbol{h}_o^T\boldsymbol{R}_v\boldsymbol{h}_o} \,. \tag{43.28}$$

Substituting $\boldsymbol{h}_o = \boldsymbol{R}_y^{-1}\boldsymbol{r}_{yx} = \boldsymbol{R}_y^{-1}\boldsymbol{r}_{xx} = \boldsymbol{R}_y^{-1}\boldsymbol{R}_x\boldsymbol{h}_1$ into (43.28), we get

$$\xi_{\mathrm{nr}}(\boldsymbol{h}_o) = \frac{\boldsymbol{h}_1^T\boldsymbol{R}_v\boldsymbol{h}_1}{\left(\boldsymbol{h}_1^T\boldsymbol{R}_x\boldsymbol{R}_y^{-1}\right)\boldsymbol{R}_v\left(\boldsymbol{R}_y^{-1}\boldsymbol{R}_x\boldsymbol{h}_1\right)} \,, \tag{43.29}$$

which is a function of all the three correlation matrices $\boldsymbol{R}_x$, $\boldsymbol{R}_v$, and $\boldsymbol{R}_y$. Using the generalized eigenvalue

decomposition [43.45], we can decompose the three correlation matrices into the following form:

$$\boldsymbol{R}_x = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{B}\,, \tag{43.30a}$$

$$\boldsymbol{R}_v = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}\,, \tag{43.30b}$$

$$\boldsymbol{R}_y = \boldsymbol{B}^{\mathrm{T}}(\boldsymbol{I}+\boldsymbol{\Lambda})\boldsymbol{B}\,, \tag{43.30c}$$

where $\boldsymbol{B}$ is an invertible square matrix, and

$$\boldsymbol{\Lambda} = \mathrm{diag}\left(\lambda_1 \;\; \lambda_2 \;\; \cdots \;\; \lambda_L\right) \tag{43.31}$$

is a diagonal matrix with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L \geq 0$. Substituting (43.30c) into (43.29), we can deduce that

$$\xi_{\mathrm{nr}}(\boldsymbol{h}_{\mathrm{o}}) = \frac{\sum\limits_{i=1}^{L} b_{i1}^2}{\sum\limits_{i=1}^{L} \frac{\lambda_i^2}{(1+\lambda_i)^2} b_{i1}^2}\,, \tag{43.32}$$

where b_{i1}, $i = 1, 2, \cdots, L$, forms the first column of $\boldsymbol{B}$ and satisfies $\sum_{i=1}^{L} b_{i1}^2 = \sigma_v^2$.

Also, with the matrix decomposition in (43.30c), the SNR of the observation signal can be expressed as

$$\mathrm{SNR} = \frac{\boldsymbol{h}_1^{\mathrm{T}}\boldsymbol{R}_x\boldsymbol{h}_1}{\boldsymbol{h}_1^{\mathrm{T}}\boldsymbol{R}_v\boldsymbol{h}_1} = \frac{\sum\limits_{i=1}^{L} \lambda_i b_{i1}^2}{\sum\limits_{i=1}^{L} b_{i1}^2}\,. \tag{43.33}$$

Using (43.33), we can rewrite (43.32) as

$$\begin{aligned}
\xi_{\mathrm{nr}}(\boldsymbol{h}_{\mathrm{o}}) &= \frac{1}{\mathrm{SNR}} \frac{\sum\limits_{i=1}^{L} \lambda_i b_{i1}^2}{\sum\limits_{i=1}^{L} \frac{\lambda_i^2}{(1+\lambda_i)^2} b_{i1}^2} \\
&= \frac{1}{\mathrm{SNR}} \frac{\sum\limits_{i=1}^{L} \frac{(1+\lambda_i)^2}{(1+\lambda_i)^2} \lambda_i b_{i1}^2}{\sum\limits_{i=1}^{L} \frac{\lambda_i^2}{(1+\lambda_i)^2} b_{i1}^2} \\
&= \frac{1}{\mathrm{SNR}} \left(\frac{\sum\limits_{i=1}^{L} \frac{\lambda_i+\lambda_i^3}{(1+\lambda_i)^2} b_{i1}^2}{\sum\limits_{i=1}^{L} \frac{\lambda_i^2}{(1+\lambda_i)^2} b_{i1}^2} + 2 \right)\,.
\end{aligned} \tag{43.34}$$

Using the fact that $\lambda_i + \lambda_i^3 \geq \lambda_i^3$, we easily deduce from (43.34) that

$$\xi_{\mathrm{nr}}(\boldsymbol{h}_{\mathrm{o}}) \geq \frac{1}{\mathrm{SNR}} \left(\frac{\sum\limits_{i=1}^{L} \frac{\lambda_i^3}{(1+\lambda_i)^2} b_{i1}^2}{\sum\limits_{i=1}^{L} \frac{\lambda_i^2}{(1+\lambda_i)^2} b_{i1}^2} + 2 \right)\,. \tag{43.35}$$

Now, before we continue with our discussion on the noise-reduction factor, we first give a lemma.

Lemma 1. With λ_i ($i = 1, 2, \cdots, L$ and $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L \geq 0$) being defined in (43.30c) and $\mu > 0$, we have

$$\begin{aligned}
&\left[\sum_{i=1}^{L} \frac{\lambda_i^3}{(\lambda_i+\mu)^2} q_i^2\right] \sum_{i=1}^{L} q_i^2 \\
&\geq \left[\sum_{i=1}^{L} \frac{\lambda_i^2}{(\lambda_i+\mu)^2} q_i^2\right] \sum_{i=1}^{L} \lambda_i q_i^2\,,
\end{aligned} \tag{43.36}$$

where q_i can be any real numbers.

Proof. This inequality can be proved by way of induction.

- Basic Step: If $L = 2$,

$$\begin{aligned}
&\left(\sum_{i=1}^{2} \frac{\lambda_i^3}{(\lambda_i+\mu)^2} q_i^2\right) \sum_{i=1}^{2} q_i^2 \\
&= \frac{\lambda_1^3}{(\lambda_1+\mu)^2} q_1^4 + \frac{\lambda_2^3}{(\lambda_2+\mu)^2} q_2^4 \\
&\quad + \left(\frac{\lambda_1^3}{(\lambda_1+\mu)^2} + \frac{\lambda_2^3}{(\lambda_2+\mu)^2}\right) q_1^2 q_2^2\,.
\end{aligned}$$

Since $\lambda_1 \geq \lambda_2 \geq 0$, it is trivial to show that

$$\begin{aligned}
&\frac{\lambda_1^3}{(\lambda_1+\mu)^2} + \frac{\lambda_2^3}{(\lambda_2+\mu)^2} \\
&\geq \frac{\lambda_1^2\lambda_2}{(\lambda_1+\mu)^2} + \frac{\lambda_1\lambda_2^2}{(\lambda_2+\mu)^2}\,,
\end{aligned}$$

where $=$ holds when $\lambda_1 = \lambda_2$. Therefore

$$\begin{aligned}
&\left(\sum_{i=1}^{2} \frac{\lambda_i^3}{(\lambda_i+\mu)^2} q_i^2\right) \sum_{i=1}^{2} q_i^2 \\
&\geq \frac{\lambda_1^3}{(\lambda_1+\mu)^2} q_1^4 + \frac{\lambda_2^3}{(\lambda_2+\mu)^2} q_2^4 \\
&\quad + \left(\frac{\lambda_1^2\lambda_2}{(\lambda_1+\mu)^2} + \frac{\lambda_1\lambda_2^2}{(\lambda_2+\mu)^2}\right) q_1^2 q_2^2 \\
&= \left(\sum_{i=1}^{2} \frac{\lambda_i^2}{(\lambda_i+\mu)^2} q_i^2\right) \sum_{i=1}^{2} \lambda_i q_i^2\,,
\end{aligned}$$

so the property is true for $L = 2$, where $=$ holds when $\lambda_1 = \lambda_2$ or at least one of q_1 and q_2 is equal to 0.

- Inductive Step: Assume that the property is true for $L = m$, i. e.,

$$\left(\sum_{i=1}^{m}\frac{\lambda_i^3}{(\lambda_i+\mu)^2}q_i^2\right)\sum_{i=1}^{m}q_i^2 \geq \left(\sum_{i=1}^{m}\frac{\lambda_i^2}{(\lambda_i+\mu)^2}q_i^2\right)\sum_{i=1}^{m}\lambda_i q_i^2 \,. \tag{43.37}$$

We must prove that it is also true for $L = m+1$. As a matter of fact,

$$\begin{aligned}
&\left(\sum_{i=1}^{m+1}\frac{\lambda_i^3}{(\lambda_i+\mu)^2}q_i^2\right)\sum_{i=1}^{m+1}q_i^2 \\
&= \left(\sum_{i=1}^{m}\frac{\lambda_i^3}{(\lambda_i+\mu)^2}q_i^2 + \frac{\lambda_{m+1}^3}{(\lambda_{m+1}+\mu)^2}q_{m+1}^2\right) \\
&\quad\times\left(\sum_{i=1}^{m}q_i^2 + q_{m+1}^2\right) \\
&= \left(\sum_{i=1}^{m}\frac{\lambda_i^3}{(\lambda_i+\mu)^2}q_i^2\right)\sum_{i=1}^{m}q_i^2 \\
&\quad+\frac{\lambda_{m+1}^3}{(\lambda_{m+1}+\mu)^2}q_{m+1}^4 \\
&\quad+\sum_{i=1}^{m}\left(\frac{\lambda_i^3}{(\lambda_i+\mu)^2}+\frac{\lambda_{m+1}^3}{(\lambda_{m+1}+\mu)^2}\right)q_i^2 q_{m+1}^2 \,.
\end{aligned} \tag{43.38}$$

Using the induction hypothesis, and also the fact that

$$\begin{aligned}
&\frac{\lambda_i^3}{(\lambda_i+\mu)^2}+\frac{\lambda_{m+1}^3}{(\lambda_{m+1}+\mu)^2} \\
&\geq \frac{\lambda_i^2\lambda_{m+1}}{(\lambda_i+\mu)^2}+\frac{\lambda_i\lambda_{m+1}^2}{(\lambda_{m+1}+\mu)^2} \,,
\end{aligned} \tag{43.39}$$

we obtain

$$\begin{aligned}
&\left(\sum_{i=1}^{m+1}\frac{\lambda_i^3}{(\lambda_i+\mu)^2}q_i^2\right)\sum_{i=1}^{m+1}q_i^2 \\
&\geq \left(\sum_{i=1}^{m}\frac{\lambda_i^2}{(\lambda_i+\mu)^2}q_i^2\right)\sum_{i=1}^{m}\lambda_i q_i^2 \\
&\quad+\frac{\lambda_{m+1}^3}{(\lambda_{m+1}+\mu)^2}q_{m+1}^4 \\
&\quad+\sum_{i=1}^{m}\left(\frac{\lambda_i^2\lambda_{m+1}}{(\lambda_i+\mu)^2}+\frac{\lambda_i\lambda_{m+1}^2}{(\lambda_{m+1}+\mu)^2}\right)q_i^2 q_{m+1}^2 \\
&= \left(\sum_{i=1}^{m+1}\frac{\lambda_i^2}{(\lambda_i+\mu)^2}q_i^2\right)\sum_{i=1}^{m+1}\lambda_i q_i^2 \,,
\end{aligned} \tag{43.40}$$

where $=$ holds when all the λ_is corresponding to nonzero q_i are equal, where $i = 1, 2, \ldots, n+1$. That completes the proof.

From *Lemma 1*, if we set $\mu = 1$ and $q_i = b_{i1}$, we obtain the following inequality (see also the Appendix in [43.18]):

$$\frac{\sum_{i=1}^{L}\frac{\lambda_i^3}{(1+\lambda_i)^2}b_{i1}^2}{\sum_{i=1}^{L}\frac{\lambda_i^2}{(1+\lambda_i)^2}b_{i1}^2} \geq \frac{\sum_{i=1}^{L}\lambda_i b_{i1}^2}{\sum_{i=1}^{L}b_{i1}^2} = \mathrm{SNR} \,, \tag{43.41}$$

with equality if and only if all the λ_is corresponding to nonzero b_{i1} are equal, where $i = 1, 2, \cdots, L$. It follows immediately that

$$\xi_{\mathrm{nr}}(\boldsymbol{h}_{\mathrm{o}}) \geq \frac{\mathrm{SNR}+2}{\mathrm{SNR}} \,. \tag{43.42}$$

The right-hand side of (43.42) is always greater than 1 since SNR is nonnegative. This shows that noise reduction is always feasible with the Wiener filter. It can be checked from (43.42) that the lower bound of the noise-reduction factor is a monotonically decreasing function on SNR. It approaches infinity when the SNR comes close to 0 and approaches 1 as the SNR approaches infinity. This indicates that more noise reduction can be achieved with the Wiener filter as the SNR decreases, which is of course desirable since, as the SNR drops, there will be more noise to eliminate.

The speech-distortion index due to the Wiener filter, according to (43.8), can be written as

$$\varphi_{\mathrm{sd}}(\boldsymbol{h}_{\mathrm{o}}) = \frac{(\boldsymbol{h}_1-\boldsymbol{h}_{\mathrm{o}})^{\mathrm{T}}\boldsymbol{R}_x(\boldsymbol{h}_1-\boldsymbol{h}_{\mathrm{o}})}{\boldsymbol{h}_1^{\mathrm{T}}\boldsymbol{R}_x\boldsymbol{h}_1} \,. \tag{43.43}$$

Obviously, we have

$$\varphi_{\mathrm{sd}}(\boldsymbol{h}_{\mathrm{o}}) \geq 0 \,. \tag{43.44}$$

Substituting (43.30c) into (43.43), we derive

$$\begin{aligned}
\varphi_{\mathrm{sd}}(\boldsymbol{h}_{\mathrm{o}}) &= \frac{\sum_{i=1}^{L}\frac{\lambda_i}{(1+\lambda_i)^2}b_{i1}^2}{\sum_{i=1}^{L}\lambda_i b_{i1}^2} \\
&\leq \frac{\sum_{i=1}^{L}\frac{\lambda_i}{1+2\lambda_i}b_{i1}^2}{\sum_{i=1}^{L}\frac{\lambda_i+2\lambda_i^2}{1+2\lambda_i}b_{i1}^2} \leq \frac{1}{2\mathrm{SNR}+1} \,,
\end{aligned} \tag{43.45}$$

where we have used the following inequality:

$$\frac{\sum_{i=1}^{L}\frac{\lambda_i^2}{1+2\lambda_i}b_{i1}^2}{\sum_{i=1}^{L}\frac{\lambda_i}{1+2\lambda_i}b_{i1}^2} \geq \frac{\sum_{i=1}^{L}\lambda_i b_{i1}^2}{\sum_{i=1}^{L}b_{i1}^2} = \text{SNR}\,. \tag{43.46}$$

This inequality can be proved via induction (following the same analysis steps in the proof of *Lemma 1*), which is left to the reader's investigation. Figure 43.1 illustrates expressions (43.42) and (43.45).

From the previous analysis we see that, while noise reduction is feasible with the Wiener filter, speech attenuation is also unavoidable. In general, the more the noise is reduced, the more the speech is attenuated. A key question is whether the Wiener filter can improve SNR. To answer this question, we give the following proposition.

Proposition 1. With the Wiener filter given in (43.18) and (43.23), the a posteriori SNR (defined after the Wiener filter) is always greater than or at least equal to the a priori SNR.

Proof. If the noise $v(n)$ is zero, we already see that the Wiener filter has no effect on the speech signal. Now we consider the case where the noise is not zero. From (43.33), we know that the a priori SNR is

$$\text{SNR} = \frac{\sum_{i=1}^{L}\lambda_i b_{i1}^2}{\sum_{i=1}^{L}b_{i1}^2}\,. \tag{43.47}$$

After the Wiener filter, the a posteriori SNR can be written as

$$\text{SNR}_\text{o} = \frac{\boldsymbol{h}_\text{o}^\text{T}\boldsymbol{R}_x\boldsymbol{h}_\text{o}}{\boldsymbol{h}_\text{o}^\text{T}\boldsymbol{R}_v\boldsymbol{h}_\text{o}}\,. \tag{43.48}$$

Substituting $\boldsymbol{h}_\text{o} = \boldsymbol{R}_y^{-1}\boldsymbol{R}_x\boldsymbol{h}_1$ into (43.48), we obtain

$$\text{SNR}_\text{o} = \frac{\boldsymbol{h}_1^\text{T}\boldsymbol{R}_x\boldsymbol{R}_y^{-1}\boldsymbol{R}_x\boldsymbol{R}_y^{-1}\boldsymbol{R}_x\boldsymbol{h}_1}{\boldsymbol{h}_1^\text{T}\boldsymbol{R}_x\boldsymbol{R}_y^{-1}\boldsymbol{R}_v\boldsymbol{R}_y^{-1}\boldsymbol{R}_x\boldsymbol{h}_1}\,. \tag{43.49}$$

Using the matrix decomposition given in (43.30c), we deduce that

$$\text{SNR}_\text{o} = \frac{\sum_{i=1}^{L}\frac{\lambda_i^3}{(\lambda_i+1)^2}b_{i1}^2}{\sum_{i=1}^{L}\frac{\lambda_i^2}{(\lambda_i+1)^2}b_{i1}^2}\,. \tag{43.50}$$

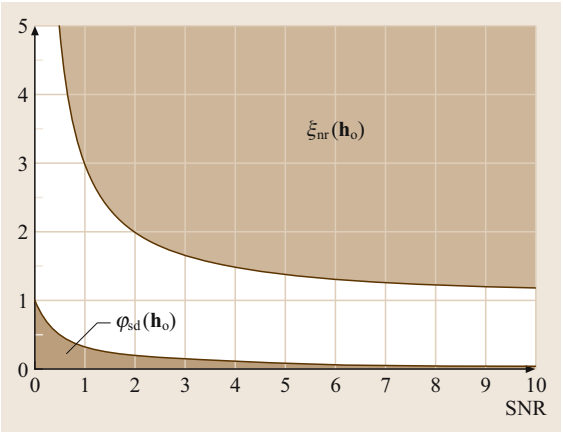


Fig. 43.1 Illustration of the areas where $\xi_{\text{nr}}(\boldsymbol{h}_\text{o})$ and $\varphi_{\text{sd}}(\boldsymbol{h}_\text{o})$ take their values as a function of the SNR

Following the inequality given in (43.41), we immediately have

$$\text{SNR}_\text{o} \geq \text{SNR}\,. \tag{43.51}$$

This completes the proof of the proposition. Therefore, we see that the time-domain Wiener filter increases the SNR of the observed signal. In other words, the Wiener filter always reduces noise.

43.4.2 A Suboptimal Filter

From the previous analysis, we see that the time-domain Wiener filter achieves noise reduction by paying a price of speech distortion. A key question immediately arises: if a given application requires high-quality speech, is there any way that we can control the compromise between noise reduction and speech distortion? To answer this question, we describe a suboptimal filter.

The Wiener filter given in (43.23) has a nice physical interpretation: it is the sum of two filters, i. e., $\boldsymbol{h}_1$ and $-\boldsymbol{R}_y^{-1}\boldsymbol{r}_{vv}$, each of which serves a different purpose. The objective of the first is to create a replica of the noisy signal, while the aim of the second is to generate a noise estimate. So the Wiener filter actually achieves noise reduction in two steps: it first creates an optimal noise estimate, and then subtracts that estimate from the noisy observations. Obviously, if we introduce a parameter to control the amount of noise to be subtracted, we can manage the tradeoff between noise reduction and speech distortion. Therefore, we construct the following

filter:

$$\boldsymbol{h}_{\text{sub}} = \boldsymbol{h}_1 - \alpha \boldsymbol{R}_y^{-1} \boldsymbol{r}_{vv} \,, \tag{43.52}$$

where $\alpha \geq 0$ is a real number. Note that $\boldsymbol{h}_{\text{sub}}$ is no longer the solution of the MMSE criterion, so we shall call it a suboptimal filter.

Substituting $\boldsymbol{h}_{\text{sub}}$ into (43.13), we can derive the corresponding MSE due to the suboptimal filter, i. e.,

$$\begin{aligned} J_x(\boldsymbol{h}_{\text{sub}}) &= E\left\{\left[x(n) - \boldsymbol{h}_{\text{sub}}^{\text{T}} \boldsymbol{y}(n)\right]^2\right\} \\ &= \sigma_v^2 - \alpha(2-\alpha)\boldsymbol{r}_{vv}^{\text{T}} \boldsymbol{R}_y^{-1} \boldsymbol{r}_{vv} \,. \end{aligned} \tag{43.53}$$

In order to have noise reduction, α must be chosen in such a way that $J_x(\boldsymbol{h}_{\text{sub}}) < J_x(\boldsymbol{h}_1)$, therefore,

$$0 < \alpha < 2 \,. \tag{43.54}$$

The noise-reduction factor of the suboptimal filter can be written as

$$\xi_{\text{nr}}(\boldsymbol{h}_{\text{sub}}) = \frac{E\left\{\left[\boldsymbol{h}_1^{\text{T}} \boldsymbol{v}(n)\right]^2\right\}}{E\left\{\left[\boldsymbol{h}_{\text{sub}}^{\text{T}} \boldsymbol{v}(n)\right]^2\right\}} \,. \tag{43.55}$$

With the matrix decomposition given in (43.30c), we can further rewrite this factor in the form

$$\xi_{\text{nr}}(\boldsymbol{h}_{\text{sub}}) = \frac{\sum\limits_{i=1}^{L} b_{i1}^2}{\sum\limits_{i=1}^{L} \frac{(\lambda_i+1-\alpha)^2}{(1+\lambda_i)^2} b_{i1}^2} \,. \tag{43.56}$$

Similarly, we can express the speech-distortion index as

$$\begin{aligned} \varphi_{\text{sd}}(\boldsymbol{h}_{\text{sub}}) &= \frac{E\left\{\left[x(n) - \boldsymbol{h}_{\text{sub}}^{\text{T}} \boldsymbol{x}(n)\right]^2\right\}}{\sigma_x^2} \\ &= \alpha^2 \frac{\sum\limits_{i=1}^{L} \frac{\lambda_i}{(1+\lambda_i)^2} b_{i1}^2}{\sum\limits_{i=1}^{L} \lambda_i b_{i1}^2} = \alpha^2 \varphi_{\text{sd}}(\boldsymbol{h}_{\text{o}}) \,. \end{aligned} \tag{43.57}$$

So the ratio between the speech-distortion indices corresponding to the two filters $\boldsymbol{h}_{\text{sub}}$ and $\boldsymbol{h}_{\text{o}}$ depends on α only.

In order to have less distortion with the suboptimal filter $\boldsymbol{h}_{\text{sub}}$ than with the Wiener filter $\boldsymbol{h}_{\text{o}}$, we must find α in such a way that,

$$\varphi_{\text{sd}}(\boldsymbol{h}_{\text{sub}}) < \varphi_{\text{sd}}(\boldsymbol{h}_{\text{o}}) \,. \tag{43.58}$$

The condition on α should be $-1 < \alpha < 1$. Finally, if α is taken as $0 < \alpha < 1$, the suboptimal filter $\boldsymbol{h}_{\text{sub}}$ reduces the level of noise in the observed signal $y(n)$ while causing less distortion than the Wiener filter $\boldsymbol{h}_{\text{o}}$. For the extreme case $\alpha = 0$, we obtain $\boldsymbol{h}_{\text{sub}} = \boldsymbol{h}_1$, so there is no noise reduction but no speech distortion as well. At the other extreme $\alpha = 1$, we have $\boldsymbol{h}_{\text{sub}} = \boldsymbol{h}_{\text{o}}$, hence noise reduction is maximized and so is speech distortion.

The a posteriori SNR, after the suboptimal filter, can be written

$$\text{SNR}_{\text{sub}} = \frac{\boldsymbol{h}_{\text{sub}}^{\text{T}} \boldsymbol{R}_x \boldsymbol{h}_{\text{sub}}}{\boldsymbol{h}_{\text{sub}}^{\text{T}} \boldsymbol{R}_v \boldsymbol{h}_{\text{sub}}} = \frac{\sum\limits_{i=1}^{L} \frac{\lambda_i(\lambda_i+1-\alpha)^2}{(\lambda_i+1)^2} b_{i1}^2}{\sum\limits_{i=1}^{L} \frac{(\lambda_i+1-\alpha)^2}{(\lambda_i+1)^2} b_{i1}^2} \,. \tag{43.59}$$

As long as $\alpha \in (0, 1)$, it can be shown that

$$\text{SNR}_{\text{sub}} \geq \text{SNR} \,. \tag{43.60}$$

The proof of this can easily be obtained by following the proof of *Proposition 1*. Therefore, the suboptimal filter can also improve SNR, although less effectively than the Wiener filter.

43.4.3 Subspace Method

We have seen that the Wiener filter, derived from the MMSE criterion, can achieve the maximum amount of noise reduction (as measured by the definition of the noise-reduction factor), but it causes significant speech distortion at the same time. In the previous section, we discussed a suboptimal filter, which was constructed empirically from the Wiener filter. While it makes good sense for managing the compromise between noise reduction and speech distortion, the suboptimal filter has no optimality properties associated with it. We now investigate a class of techniques called subspace methods, which, like the Wiener filter, acquire a clean speech estimate by applying a linear transformation to a vector of noisy speech observations. Similar to the Wiener filter, the subspace approach is also formulated from the MMSE criterion with mathematical rigor; hence it is also an optimal estimator. However, it differs from the Wiener filter in that its linear transformation is derived from a constrained optimization problem while the Wiener filter is deduced from an unconstrained one.

Consider the signal model given in (43.2). An estimate of the clean speech can be obtained by applying a linear transformation to the noisy speech vector, i. e.,

$$\hat{\boldsymbol{x}}(n) = \boldsymbol{H}\boldsymbol{y}(n) \,, \tag{43.61}$$

where $\boldsymbol{H}$ is a matrix of size $L \times L$. The error signal obtained by this estimation is written as

$$\begin{aligned}\boldsymbol{e}(n) &\triangleq \hat{\boldsymbol{x}}(n) - \boldsymbol{x}(n) = \boldsymbol{H}\boldsymbol{y}(n) - \boldsymbol{x}(n) \\ &= (\boldsymbol{H} - \boldsymbol{I})\boldsymbol{x}(n) + \boldsymbol{H}\boldsymbol{v}(n) = \boldsymbol{e}_x(n) + \boldsymbol{e}_v(n) \,,\end{aligned} \tag{43.62}$$

where

$$\boldsymbol{e}_x(n) \triangleq (\boldsymbol{H} - \boldsymbol{I})\boldsymbol{x}(n) \tag{43.63}$$

and

$$\boldsymbol{e}_v(n) \triangleq \boldsymbol{H}\boldsymbol{v}(n) \tag{43.64}$$

represent, respectively, the speech distortion due to the linear transformation and the residual noise. It is immediately seen that there are three criteria to estimate $\boldsymbol{H}$:

1. minimizing the energy of $\boldsymbol{e}(n)$;
2. minimizing the energy of $\boldsymbol{e}_v(n)$ while limiting the level of speech distortion;
3. minimizing the energy of $\boldsymbol{e}_x(n)$ while limiting the level of residual noise.

The first case will lead to the Wiener solution. The only difference between the Wiener filter discussed in Sect. 43.4.1 and this estimator is that the former obtains the current speech estimate using only past and current noisy speech samples, while the latter uses not only past and present observation samples, but also future values. The second criterion is rarely used in practice because the resulting estimator produces nonstationary residual noise (due to the nonstationarity of speech signals), which is usually intolerable to the human perception system. The third case will lead to the so-called subspace method, which, like the Wiener filter, has been widely investigated.

Mathematically, the optimal linear transformation in the subspace technique can be described as

$$\begin{aligned}&\boldsymbol{H}_{\mathrm{o}} = \arg\min_{\boldsymbol{H}} \mathrm{tr}\left\{E\left[\boldsymbol{e}_x(n)\boldsymbol{e}_x^{\mathrm{T}}(n)\right]\right\} \\ &\text{subject to } \mathrm{tr}\left\{E\left[\boldsymbol{e}_v(n)\boldsymbol{e}_v^{\mathrm{T}}(n)\right]\right\} \le L\sigma^2 \,.\end{aligned} \tag{43.65}$$

If we use a Lagrange multiplier to adjoin the constraint to the cost function, (43.65) can be rewritten as

$$\boldsymbol{H}_{\mathrm{o}} = \arg\min_{\boldsymbol{H}} \mathcal{L}(\boldsymbol{H}, \mu) \,, \tag{43.66}$$

where

$$\begin{aligned}&\mathcal{L}(\boldsymbol{H}, \mu) \\ &= \mathrm{tr}\left\{E\left[\boldsymbol{e}_x(n)\boldsymbol{e}_x^{\mathrm{T}}(n)\right]\right\} \\ &\quad + \mu\left(\mathrm{tr}\left\{E\left[\boldsymbol{e}_v(n)\boldsymbol{e}_v^{\mathrm{T}}(n)\right]\right\} - L\sigma^2\right) \\ &= \mathrm{tr}\left[(\boldsymbol{H} - \boldsymbol{I})\boldsymbol{R}_x(\boldsymbol{H} - \boldsymbol{I})^{\mathrm{T}}\right] \\ &\quad + \mu\left[\mathrm{tr}\left(\boldsymbol{H}\boldsymbol{R}_v\boldsymbol{H}^{\mathrm{T}}\right) - L\sigma^2\right] \,,\end{aligned} \tag{43.67}$$

$\boldsymbol{R}_x$ and $\boldsymbol{R}_v$ are, respectively, the covariance matrices of the clean speech and noise, and $\mu > 0$ is the Lagrange multiplier. Using the fact that

$$\frac{\partial}{\partial \boldsymbol{H}} \mathrm{tr}(\boldsymbol{R}_x\boldsymbol{H}) = \frac{\partial}{\partial \boldsymbol{H}} \mathrm{tr}(\boldsymbol{H}\boldsymbol{R}_x) = \boldsymbol{R}_x^{\mathrm{T}} = \boldsymbol{R}_x \,, \tag{43.68a}$$

$$\frac{\partial}{\partial \boldsymbol{H}} \mathrm{tr}\left(\boldsymbol{H}\boldsymbol{R}_x\boldsymbol{H}^{\mathrm{T}}\right) = 2\boldsymbol{H}\boldsymbol{R}_x \,, \tag{43.68b}$$

$$\frac{\partial}{\partial \boldsymbol{H}} \mathrm{tr}\left(\boldsymbol{H}\boldsymbol{R}_v\boldsymbol{H}^{\mathrm{T}}\right) = 2\boldsymbol{H}\boldsymbol{R}_v \,, \tag{43.68c}$$

we can readily deduce that

$$\frac{\partial}{\partial \boldsymbol{H}} \mathcal{L}(\boldsymbol{H}, \mu) = 2\boldsymbol{H}\boldsymbol{R}_x - 2\boldsymbol{R}_x + 2\mu\boldsymbol{H}\boldsymbol{R}_v \,. \tag{43.69}$$

Equating the right-hand side of (43.69) to zero, we obtain the solution to (43.66):

$$\boldsymbol{H}_{\mathrm{o}} = \boldsymbol{R}_x(\boldsymbol{R}_x + \mu\boldsymbol{R}_v)^{-1} \,, \tag{43.70}$$

where μ satisfies

$$\mathrm{tr}\left[\boldsymbol{R}_x(\boldsymbol{R}_x + \mu\boldsymbol{R}_v)^{-1}\boldsymbol{R}_v(\boldsymbol{R}_x + \mu\boldsymbol{R}_v)^{-1}\boldsymbol{R}_x\right] = L\sigma^2 \,. \tag{43.71}$$

To implement the optimal transformation given in (43.70), we need to compute the inverse of the matrix sum $\boldsymbol{R}_x + \mu\boldsymbol{R}_v$. The most straightforward way to do this is the direct method, which computes the inverse of the matrix explicitly, but this approach is not very robust particularly when the matrix is not well conditioned. Another popular method is based on either eigenvalue decomposition [43.19–25] or generalized eigenvalue decomposition [43.26]. For example, with the generalized eigenvalue decomposition as defined in (43.30c), we can rewrite the optimal transformation as

$$\boldsymbol{H}_{\mathrm{o}} = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{\Lambda}(\boldsymbol{\Lambda} + \mu\boldsymbol{I})^{-1}\boldsymbol{B}^{-\mathrm{T}} \,. \tag{43.72}$$

Therefore, the estimation of the clean speech can be decoupled into three steps: applying the transformation $\boldsymbol{B}^{-\mathrm{T}}$ to the noisy signal, modifying the noisy signal in the transformed domain by a gain function $\boldsymbol{\Lambda}(\boldsymbol{\Lambda} + \mu\boldsymbol{I})^{-1}$, and applying $\boldsymbol{B}^{\mathrm{T}}$ to transform the modified components back to the original domain.

Another interesting interpretation of (43.72) is to divide the vector space into two subspaces: the speech subspace corresponding to all the λ_i for $\lambda_i > 0$ and the noise subspace associated with all λ_i for $\lambda_i = 0$. Suppose that the dimension of the speech subspace is M. We then can rewrite (43.72) as

$$\boldsymbol{H}_{\mathrm{o}} = \boldsymbol{B}^{\mathrm{T}} \begin{pmatrix} \boldsymbol{\Sigma} & \boldsymbol{0}_{M\times K} \\ \boldsymbol{0}_{K\times M} & \boldsymbol{0}_{K\times K} \end{pmatrix} \boldsymbol{B}^{-\mathrm{T}} \,, \tag{43.73}$$

where $K = L - M$ is the order of the noise subspace and

$$\boldsymbol{\Sigma} = \mathrm{diag}\left(\frac{\lambda_1}{\lambda_1+\mu}, \frac{\lambda_2}{\lambda_2+\mu}, \cdots, \frac{\lambda_M}{\lambda_M+\mu}\right) \tag{43.74}$$

is an $M \times M$ diagonal matrix. We now clearly see that noise reduction with the subspace method is achieved by nulling the noise subspace and cleaning the speech-plus-noise subspace via a reweighted reconstruction.

Now we are ready to check if the linear transformation given in (43.70) can really reduce the level of noise. Let us examine the noise-reduction factor, which can be written, according to (43.6), as

$$\xi_{\mathrm{nr}}(\boldsymbol{H}_{\mathrm{o}}) = \frac{\mathrm{tr}\{\boldsymbol{R}_v\}}{\mathrm{tr}\left\{E\left[\boldsymbol{e}_v(n)\boldsymbol{e}_v^{\mathrm{T}}(n)\right]\right\}} . \tag{43.75}$$

Substituting (43.70) into (43.64), we can derive

$$\begin{aligned}&\mathrm{tr}\left\{E\left[\boldsymbol{e}_v(n)\boldsymbol{e}_v^{\mathrm{T}}(n)\right]\right\}\\ &= \mathrm{tr}\left\{\boldsymbol{R}_x(\boldsymbol{R}_x+\mu\boldsymbol{R}_v)^{-1}\boldsymbol{R}_v(\boldsymbol{R}_x+\mu\boldsymbol{R}_v)^{-1}\boldsymbol{R}_x\right\} .\end{aligned} \tag{43.76}$$

Following the matrix-decomposition procedure given in (43.30c), we easily deduce that

$$\mathrm{tr}\{\boldsymbol{R}_v\} = \mathrm{tr}\{\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}\} = \mathrm{tr}\{\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}\} = \sum_{i=1}^{L}\sum_{j=1}^{L} b_{ij}^2 , \tag{43.77}$$

and

$$\begin{aligned}\mathrm{tr}\left\{E\left[\boldsymbol{e}_v(n)\boldsymbol{e}_v^{\mathrm{T}}(n)\right]\right\} &= \mathrm{tr}\{\boldsymbol{B}^{\mathrm{T}}\boldsymbol{\Lambda}(\boldsymbol{\Lambda}+\mu\boldsymbol{I})^{-2}\boldsymbol{\Lambda}\boldsymbol{B}\}\\ &= \mathrm{tr}\{\boldsymbol{\Lambda}(\boldsymbol{\Lambda}+\mu\boldsymbol{I})^{-2}\boldsymbol{\Lambda}\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}\}\\ &= \sum_{i=1}^{L}\sum_{j=1}^{L}\frac{\lambda_i^2}{(\lambda_i+\mu)^2}b_{ij}^2 .\end{aligned} \tag{43.78}$$

Therefore, we have

$$\xi_{\mathrm{nr}}(\boldsymbol{H}_{\mathrm{o}}) = \frac{\sum_{i=1}^{L}\sum_{j=1}^{L} b_{ij}^2}{\sum_{i=1}^{L}\sum_{j=1}^{L}\frac{\lambda_i^2}{(\lambda_i+\mu)^2}b_{ij}^2} . \tag{43.79}$$

Since $\mu > 0$ and $\lambda_i \geq 0$ for $i = 1, 2, \cdots, L$, it is obvious that

$$\xi_{\mathrm{nr}}(\boldsymbol{H}_{\mathrm{o}}) > 1 . \tag{43.80}$$

This shows that noise reduction is always feasible with the subspace method.

Since the energy of the speech distortion $\boldsymbol{e}_x(n)$ is always greater than zero, the power of the clean speech is also attenuated when we reduce the noise with the subspace method. We then ask the same question about SNR as we did for the Wiener filter: can the subspace method improve SNR? To answer this question, we give the following proposition.

Proposition 2. For the linear transformation given in (43.70), if $\mu > 0$, the a posteriori SNR with the subspace method is always greater than or equal to the a priori SNR.

Proof. If the noise is zero, we can easily check that the optimal transformation matrix $\boldsymbol{H}_{\mathrm{o}}$ will be the identity matrix, and hence will not change the input speech. If the noise is not zero, the a priori SNR can be written as

$$\mathrm{SNR} = \frac{\mathrm{tr}\{\boldsymbol{R}_x\}}{\mathrm{tr}\{\boldsymbol{R}_v\}} . \tag{43.81}$$

After applying the linear transformation $\boldsymbol{H}_{\mathrm{o}}$, the a posteriori SNR can be expressed as

$$\begin{aligned}\mathrm{SNR}_{\mathrm{o}} &= \frac{\mathrm{tr}\left\{E\left[\boldsymbol{H}_{\mathrm{o}}\boldsymbol{x}(n)\boldsymbol{x}^{\mathrm{T}}(n)\boldsymbol{H}_{\mathrm{o}}^{\mathrm{T}}\right]\right\}}{\mathrm{tr}\left\{E\left[\boldsymbol{H}_{\mathrm{o}}\boldsymbol{v}(n)\boldsymbol{v}^{\mathrm{T}}(n)\boldsymbol{H}_{\mathrm{o}}^{\mathrm{T}}\right]\right\}}\\ &= \frac{\mathrm{tr}\left\{\boldsymbol{H}_{\mathrm{o}}\boldsymbol{R}_x\boldsymbol{H}_{\mathrm{o}}^{\mathrm{T}}\right\}}{\mathrm{tr}\left\{\boldsymbol{H}_{\mathrm{o}}\boldsymbol{R}_v\boldsymbol{H}_{\mathrm{o}}^{\mathrm{T}}\right\}} .\end{aligned} \tag{43.82}$$

It follows then that

$$\begin{aligned}\frac{\mathrm{SNR}_{\mathrm{o}}}{\mathrm{SNR}} &= \frac{\mathrm{tr}\{\boldsymbol{R}_v\}\mathrm{tr}\left\{\boldsymbol{H}_{\mathrm{o}}\boldsymbol{R}_x\boldsymbol{H}_{\mathrm{o}}^{\mathrm{T}}\right\}}{\mathrm{tr}\{\boldsymbol{R}_x\}\mathrm{tr}\left\{\boldsymbol{H}_{\mathrm{o}}\boldsymbol{R}_v\boldsymbol{H}_{\mathrm{o}}^{\mathrm{T}}\right\}}\\ &= \frac{\mathrm{tr}\{\boldsymbol{R}_v\}\mathrm{tr}\left\{\boldsymbol{R}_x[\boldsymbol{R}_x+\mu\boldsymbol{R}_v]^{-1}\boldsymbol{R}_x[\boldsymbol{R}_x+\mu\boldsymbol{R}_v]^{-1}\boldsymbol{R}_x\right\}}{\mathrm{tr}\{\boldsymbol{R}_x\}\mathrm{tr}\left\{\boldsymbol{R}_x[\boldsymbol{R}_x+\mu\boldsymbol{R}_v]^{-1}\boldsymbol{R}_v[\boldsymbol{R}_x+\mu\boldsymbol{R}_v]^{-1}\boldsymbol{R}_x\right\}} .\end{aligned} \tag{43.83}$$

Following the matrix-decomposition procedure given in (43.30c) and using some simple algebra, we find that

$$\begin{aligned}\frac{\mathrm{SNR}_{\mathrm{o}}}{\mathrm{SNR}} &= \frac{\mathrm{tr}\{\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}\}\mathrm{tr}\{\boldsymbol{B}^{\mathrm{T}}\boldsymbol{\Lambda}(\boldsymbol{\Lambda}+\mu\boldsymbol{I})^{-1}\boldsymbol{\Lambda}(\boldsymbol{\Lambda}+\mu\boldsymbol{I})^{-1}\boldsymbol{\Lambda}\boldsymbol{B}\}}{\mathrm{tr}\{\boldsymbol{B}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{B}\}\mathrm{tr}\{\boldsymbol{B}^{\mathrm{T}}\boldsymbol{\Lambda}(\boldsymbol{\Lambda}+\mu\boldsymbol{I})^{-2}\boldsymbol{\Lambda}\boldsymbol{B}\}}\\ &= \frac{\mathrm{tr}\{\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}\}\mathrm{tr}\{\boldsymbol{\Lambda}(\boldsymbol{\Lambda}+\mu\boldsymbol{I})^{-1}\boldsymbol{\Lambda}(\boldsymbol{\Lambda}+\mu\boldsymbol{I})^{-1}\boldsymbol{\Lambda}\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}\}}{\mathrm{tr}\{\boldsymbol{\Lambda}\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}\}\mathrm{tr}\{\boldsymbol{\Lambda}(\boldsymbol{\Lambda}+\mu\boldsymbol{I})^{-2}\boldsymbol{\Lambda}\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}\}}\\ &= \frac{\sum_{i=1}^{L}\sum_{j=1}^{L} b_{ij}^2 \cdot \sum_{i=1}^{L}\sum_{j=1}^{L}\frac{\lambda_i^3}{(\lambda_i+\mu)^2}b_{ij}^2}{\sum_{i=1}^{L}\sum_{j=1}^{L}\lambda_i b_{ij}^2 \cdot \sum_{i=1}^{L}\sum_{j=1}^{L}\frac{\lambda_i^2}{(\lambda_i+\mu)^2}b_{ij}^2} .\end{aligned} \tag{43.84}$$

Now using the *Lemma 1* and setting $q_i^2 = \sum_{j=1}^{L} b_{ij}^2$, we obtain that

$$\sum_{i=1}^{L}\sum_{j=1}^{L} b_{ij}^2 \cdot \sum_{i=1}^{L}\sum_{j=1}^{L} \frac{\lambda_i^3}{(\lambda_i+\mu)^2} b_{ij}^2 \geq \sum_{i=1}^{L}\sum_{j=1}^{L} \lambda_i b_{ij}^2 \cdot \sum_{i=1}^{L}\sum_{j=1}^{L} \frac{\lambda_i^2}{(\lambda_i+\mu)^2} b_{ij}^2 \,. \tag{43.85}$$

It follows immediately that

$$\mathrm{SNR_o} \geq \mathrm{SNR} \,, \tag{43.86}$$

with equality if and only if $\lambda_1 = \lambda_2 = \cdots = \lambda_L$. Note that this condition is the same as that obtained previously for the Wiener filter, which should not come as a surprise since both techniques are formulated in a similar way.

From the previous analysis, we see that $\mathrm{SNR_o}$ depends not only on the speech and noise characteristics, but also on the value of μ. With some algebra, it can be shown that

$$\lim_{\mu\to 0} \mathrm{SNR_o} = \lim_{\mu\to 0} \frac{\sum_{i=1}^{L}\sum_{j=1}^{L} \frac{\lambda_i^3}{(\lambda_i+\mu)^2} b_{ij}^2}{\sum_{i=1}^{L}\sum_{j=1}^{L} \frac{\lambda_i^2}{(\lambda_i+\mu)^2} b_{ij}^2} = \mathrm{SNR} \,, \tag{43.87}$$

which is the lower bound of the a posteriori SNR. When $\mu \to +\infty$, we see that $\lambda_i + \mu \to \mu$. Therefore,

$$\lim_{\mu\to +\infty} \mathrm{SNR_o} = \frac{\sum_{i=1}^{L}\sum_{j=1}^{L} \lambda_i^3 b_{ij}^2}{\sum_{i=1}^{L}\sum_{j=1}^{L} \lambda_i^2 b_{ij}^2} \leq \sum_{i=1}^{L} \lambda_i \,, \tag{43.88}$$

which is the upper bound of the a posteriori SNR.

43.4.4 Frequency-Domain Wiener Filter

The Wiener filter can also be formulated in the frequency domain. One way of deriving such a filter is to transform the time-domain Wiener filter into the frequency domain using the so-called overlap-add technique. In this case, the time-domain Wiener filter and its frequency-domain counterpart have exactly the same performance. More often, however, the frequency-domain Wiener filter is formulated by directly estimating the clean speech spectrum from the noisy speech spectrum. The resulting filter differs in two aspects from the time-domain Wiener filter: first, the former is a causal filter, while the latter can be noncausal; second, the former is a full-band technique, while the latter is a subband technique, where each subband filter is independent of the filters corresponding to other frequency bands.

Let us consider the signal model in (43.3). The frequency-domain subband Wiener filter is derived by the criterion [43.46]

$$H_\mathrm{o}(\mathrm{i}\omega_k) = \arg \min_{H(\mathrm{i}\omega_k)} J_X[H(\mathrm{i}\omega_k)] \,, \tag{43.89}$$

where

$$J_X[H(\mathrm{i}\omega_k)] = E\left[|X(n,\mathrm{i}\omega_k) - H(\mathrm{i}\omega_k)Y(n,\mathrm{i}\omega_k)|^2\right]$$

is the MSE between the speech spectrum and its estimate at frequency ω_k. Differentiating $J_X[H(\mathrm{i}\omega_k)]$ with respect to $H(\mathrm{i}\omega_k)$ and equating the result to zero, we easily deduce the Wiener filter,

$$H_\mathrm{o}(\mathrm{i}\omega_k) = \frac{E\left[|X(n,\mathrm{i}\omega_k)|^2\right]}{E\left[|Y(n,\mathrm{i}\omega_k)|^2\right]} = \frac{P_x(\omega_k)}{P_y(\omega_k)} \,, \tag{43.90}$$

where

$$P_x(\omega_k) = \frac{1}{L} E\left[|X(n,\mathrm{i}\omega_k)|^2\right]$$

and

$$P_y(\omega_k) = \frac{1}{L} E\left[|Y(n,\mathrm{i}\omega_k)|^2\right]$$

are the power spectral densities (PSDs) of $x(n)$ and $y(n)$, respectively. It can be seen from this expression that the frequency-domain Wiener filter $H_\mathrm{o}(\mathrm{i}\omega_k)$ is nonnegative and real valued. Therefore, it only modifies the amplitude of the noisy speech spectra, while leaving the phase components unchanged. Since $H_\mathrm{o}(\mathrm{i}\omega_k)$ is real valued, we shall, from now on, drop the symbol i from its expression, which should not cause any confusion.

We see from (43.90) that, in order to obtain the Wiener filter, we need to know the PSDs of both the noisy and clean speech signals. The former can be directly estimated from the noisy observation signal $y(n)$. But $x(n)$ is not accessible; hence it may seem difficult to estimate the PSD of the clean speech. However, since speech and noise are assumed to be uncorrelated, we have

$$P_y(\omega_k) = P_x(\omega_k) + P_v(\omega_k) \,, \tag{43.91}$$

where $P_v(\omega_k)$ is the PSD of $v(n)$. Therefore, the Wiener filter can be written as

$$H_\mathrm{o}(\omega_k) = \frac{P_y(\omega_k) - P_v(\omega_k)}{P_y(\omega_k)} \,. \tag{43.92}$$

Now we see the Wiener filter depends on the PSDs of both the noisy observation and the noise signal, where the noise PSD can be estimated during the absence of speech.

The optimal estimate of the clean speech spectrum, using $H_o(\omega_k)$, is

$$\begin{aligned}\hat{X}_o(n, i\omega_k) &= H_o(\omega_k)Y(n, i\omega_k) \\ &= H_o(\omega_k)X(n, i\omega_k) + H_o(\omega_k)V(n, i\omega_k) .\end{aligned} \tag{43.93}$$

Applying the inverse DFT (IDFT) to (43.93), we can obtain the optimal estimate of the speech samples $\hat{x}_o(n)$.

The power of the estimated clean speech can be evaluated according to Parseval's relation, i. e.,

$$\begin{aligned}&E\left[\hat{x}_o^2(n)\right] \\ &= \sum_{k=0}^{L-1} \frac{1}{L} E\left[|\hat{X}_o(n, i\omega_k)|^2\right] \\ &= \sum_{k=0}^{L-1} H_o^2(\omega_k) P_y(\omega_k) \\ &= \sum_{k=0}^{L-1} \frac{P_x^2(\omega_k)}{P_y^2(\omega_k)} P_x(\omega_k) + \sum_{k=0}^{L-1} \frac{P_x^2(\omega_k)}{P_y^2(\omega_k)} P_v(\omega_k) ,\end{aligned} \tag{43.94}$$

which is the sum of two terms. The first is the power of the filtered clean speech and the second is the power of the residual noise.

If the noise is not zero, we can write the noise-reduction factor of the frequency-domain Wiener filter, according to (43.6) and (43.94), as

$$\xi_{nr}[H(\omega_k)] = \frac{\sum_{k=0}^{L-1} P_v(\omega_k)}{\sum_{k=0}^{L-1} \frac{P_x^2(\omega_k)}{P_y^2(\omega_k)} P_v(\omega_k)} . \tag{43.95}$$

Since $P_x^2(\omega_k) \le P_y^2(\omega_k)$, we can easily verify that

$$\xi_{nr}[H(\omega_k)] \ge 1 . \tag{43.96}$$

This indicates that the Wiener filter can reduce the noise level (unless there is no noise). Similarly, we can check that the power of the filtered clean speech is always less than the power of the original clean speech. So, the frequency-domain Wiener filter, just like its time-domain counterpart, also reduces noise at the price of attenuating the clean speech. It is key to know, then, whether this Wiener filter can improve SNR. We have the following proposition:

Proposition 3. With the Wiener filter given in (43.90), the a posteriori SNR (defined after the Wiener filter) is always greater than or equal to the a priori SNR.

Proof. If there is no noise, we see that the Wiener filter has no effect on SNR. Now we consider the generic case where noise is not zero. In the frequency domain, the a priori SNR can be expressed as

$$\text{SNR} = \frac{\sum_{k=0}^{L-1} P_x(\omega_k)}{\sum_{k=0}^{L-1} P_v(\omega_k)} . \tag{43.97}$$

Similarly, we can write, after Wiener filtering, the a posteriori SNR according to (43.94) as

$$\text{SNR}_o = \frac{\sum_{k=0}^{L-1} \frac{P_x^2(\omega_k)}{P_y^2(\omega_k)} P_x(\omega_k)}{\sum_{k=0}^{L-1} \frac{P_x^2(\omega_k)}{P_y^2(\omega_k)} P_v(\omega_k)} . \tag{43.98}$$

Now let us denote

$$\begin{aligned}\phi(\omega_k) = &\sum_{k=0}^{L-1} \frac{P_x^2(\omega_k)}{P_y^2(\omega_k)} P_x(\omega_k) \sum_{k=0}^{L-1} P_v(\omega_k) \\ &- \sum_{k=0}^{L-1} \frac{P_x^2(\omega_k)}{P_y^2(\omega_k)} P_v(\omega_k) \sum_{k=0}^{L-1} P_x(\omega_k) .\end{aligned}$$

It follows immediately that

$$\begin{aligned}&\phi(\omega_k) \\ &= \sum_{k=0}^{L-1} \sum_{j=0}^{L-1} \frac{P_x^3(\omega_k)}{P_y^2(\omega_k)} P_v(\omega_j) \\ &\quad - \sum_{k=0}^{L-1} \sum_{j=0}^{L-1} \frac{P_x^2(\omega_k)}{P_y^2(\omega_k)} P_v(\omega_k) P_x(\omega_j) \\ &= \sum_{k=0}^{L-1} \sum_{j=0}^{L-1} \frac{P_x^2(\omega_k)}{P_y^2(\omega_k)} [P_x(\omega_k)P_v(\omega_j) - P_v(\omega_k)P_x(\omega_j)] \\ &= \sum_{k=0}^{L-1} \sum_{j>k}^{L-1} \left[\frac{P_x^2(\omega_k)}{P_y^2(\omega_k)} - \frac{P_x^2(\omega_j)}{P_y^2(\omega_j)}\right] \\ &\quad \times [P_x(\omega_k)P_v(\omega_j) - P_v(\omega_k)P_x(\omega_j)] .\end{aligned} \tag{43.99}$$

Using (43.91), we can rewrite $\phi(\omega_k)$ as

$$\begin{aligned}&\phi(\omega_k)\\&=\sum_{k=0}^{L-1}\sum_{j>k}^{L-1}\frac{1}{P_y(\omega_k)P_y(\omega_j)}\left[\frac{P_x(\omega_k)}{P_y(\omega_k)}+\frac{P_x(\omega_j)}{P_y(\omega_j)}\right]\\&\quad\times[P_x(\omega_k)P_v(\omega_j)-P_v(\omega_k)P_x(\omega_j)]^2\,.\end{aligned}\tag{43.100}$$

Since $P_x(\omega_k)\geq 0$, $P_v(\omega_k)\geq 0$, and $P_y(\omega_k)\geq 0$, it is easy to see that $\phi(\omega_k)$ is greater than, or at least equal to 0. Therefore, we have

$$\begin{aligned}&\sum_{k=0}^{L-1}\frac{P_x^2(\omega_k)}{P_y^2(\omega_k)}P_x(\omega_k)\sum_{k=0}^{L-1}P_v(\omega_k)\\&\geq\sum_{k=0}^{L-1}\frac{P_x^2(\omega_k)}{P_y^2(\omega_k)}P_v(\omega_k)\sum_{k=0}^{L-1}P_x(\omega_k)\,,\end{aligned}$$

which means that

$$\mathrm{SNR_o}\geq\mathrm{SNR}\,,$$

where we see from (43.100) that equality is attained if and only if $P_x(\omega_0)/P_v(\omega_0)=P_x(\omega_1)/P_v(\omega_1)=\cdots=P_x(\omega_{L-1})/P_v(\omega_{L-1})$. In other words, the frequency-domain Wiener filter will increase SNR unless all subband SNRs are equal, which is understandable. When all subband SNRs are equal, both speech and noise have the same PSD under the given resolution condition. In this case, the Wiener filter is not able to distinguish noise from speech, and hence is not able to increase SNR. In all other cases, the subband Wiener filter will improve SNR.

43.4.5 Short-Time Parametric Wiener Filter

In implementations of the Wiener filter [(43.90) or (43.92)], some approximations are required since the PSDs of the noisy speech and noise signals are not known and must be estimated. One way is to replace the PSD with the instantaneous magnitude-squared spectral amplitude, yielding

$$H_{\mathrm{MS}}(\omega_k)=\frac{|Y(n,\mathrm{i}\omega_k)|^2-|V(n,\mathrm{i}\omega_k)|^2}{|Y(n,\mathrm{i}\omega_k)|^2}\,.\tag{43.101}$$

With such a filter, the clean speech estimate can be written as

$$\hat{X}(n,\mathrm{i}\omega_k)=H_{\mathrm{MS}}(\omega_k)Y(\mathrm{i}\omega_k)\,.\tag{43.102}$$

To enable more flexibility in manipulating the trade-off between noise suppression and speech distortion, (43.101) has been extended to a more-generic form referred to as the parametric Wiener filtering technique [43.13–15], namely

$$H_{\mathrm{PW}}(\omega_k)=\left[\frac{|Y(n,\mathrm{i}\omega_k)|^p-\eta|V(n,\mathrm{i}\omega_k)|^p}{|Y(n,\mathrm{i}\omega_k)|^p}\right]^{1/q}\,,\tag{43.103}$$

where p and q are some positive numbers, and η is a parameter introduced to control the amount of noise to be reduced. More-aggressive noise reduction can be achieved with $\eta>1$; this is, of course, at the expense of distortion artifacts. If we are more concerned with speech distortion, we can choose $\eta<1$. Commonly used configurations of (p,q,η) include $(1,1,1)$, $(2,1,1)$, and $(2,1/2,1)$.

With the short-time parametric Wiener filter, the estimate of the clean speech spectrum is written as

$$\hat{X}(n,\mathrm{i}\omega_k)=H_{\mathrm{PW}}(\omega_k)Y(n,\mathrm{i}\omega_k)\,.\tag{43.104}$$

Note that for the general configuration of (p,q,η), the estimator in (43.104) has no optimality properties associated with it. However, this does not diminish its usefulness. As a matter of fact, (43.104) has been widely used as a benchmark to which other noise reduction techniques are compared [43.13, 14, 27, 29].

The analysis of noise reduction, speech distortion, and SNR of the parametric Wiener filter can be done by following the analysis for the frequency-domain Wiener filter. This will be left to the reader's investigation.

43.5 Noise Reduction via Spectral Restoration

In the frequency domain, noise reduction can be formulated as a robust spectral estimation problem, i. e., estimation of the clean speech spectrum from the noisy speech spectrum. We now develop good estimators of clean speech within the framework of estimation theory.

43.5.1 MMSE Spectral Estimator

The minimum-mean-square-error (MMSE) estimator has its roots in Bayesian estimation theory. Let us consider the signal model given in (43.3). Since both clean speech and noise are assumed to be zero-mean random

processes, their short-time Fourier transform (STFT) coefficients, i. e., $X(n, \mathrm{i}\omega_k)$ and $V(n, \mathrm{i}\omega_k)$, are also zero-mean random variables. If we decompose the clean speech spectrum into real and imaginary parts as

$$X(n, \mathrm{i}\omega_k) = X_\mathrm{R} + \mathrm{i}X_\mathrm{I} ,$$

and assume that the two random variables X_R and X_I are independent, then an MMSE (conditional) estimator of $X(n, \mathrm{i}\omega_k)$ is written as

$$\begin{aligned}
&\widehat{X}_{\mathrm{MMSE}}(n, \mathrm{i}\omega_k) \\
&= E\left[X(n, \mathrm{i}\omega_k)|Y(n, \mathrm{i}\omega_k)\right] \\
&= E\left[X_\mathrm{R}|Y(n, \mathrm{i}\omega_k)\right] + \mathrm{i}E\left[X_\mathrm{I}|Y(n, \mathrm{i}\omega_k)\right] .
\end{aligned} \tag{43.105}$$

Let us define

$$\begin{aligned}
\widehat{X}_{\mathrm{R,MMSE}} &\triangleq E\left[X_\mathrm{R}|Y(n, \mathrm{i}\omega_k)\right] , \\
\widehat{X}_{\mathrm{I,MMSE}} &\triangleq E\left[X_\mathrm{I}|Y(n, \mathrm{i}\omega_k)\right] .
\end{aligned} \tag{43.106}$$

Using the conditional and joint probability density functions (pdfs), we can write the two estimators in (43.106) as

$$\begin{aligned}
&\widehat{X}_{\mathrm{R,MMSE}} \\
&= \int X_\mathrm{R} p[X_\mathrm{R}|Y(n, \mathrm{i}\omega_k)]\,\mathrm{d}X_\mathrm{R} \\
&= \frac{\iint X_\mathrm{R} p[Y(n, \mathrm{i}\omega_k)|X_\mathrm{R}, X_\mathrm{I}]p(X_\mathrm{R}, X_\mathrm{I})\,\mathrm{d}X_\mathrm{R}\,\mathrm{d}X_\mathrm{I}}{\iint p[Y(n, \mathrm{i}\omega_k)|X_\mathrm{R}, X_\mathrm{I}]p(X_\mathrm{R}, X_\mathrm{I})\,\mathrm{d}X_\mathrm{R}\,\mathrm{d}X_\mathrm{I}} ,
\end{aligned} \tag{43.107}$$

and

$$\begin{aligned}
&\widehat{X}_{\mathrm{I,MMSE}} \\
&= \int X_\mathrm{I} p[X_\mathrm{I}|Y(n, \mathrm{i}\omega_k)]\,\mathrm{d}X_\mathrm{I} \\
&= \frac{\iint X_\mathrm{I} p[Y(n, \mathrm{i}\omega_k)|X_\mathrm{R}, X_\mathrm{I}]p(X_\mathrm{R}, X_\mathrm{I})\,\mathrm{d}X_\mathrm{R}\,\mathrm{d}X_\mathrm{I}}{\iint p[Y(n, \mathrm{i}\omega_k)|X_\mathrm{R}, X_\mathrm{I}]p(X_\mathrm{R}, X_\mathrm{I})\,\mathrm{d}X_\mathrm{R}\,\mathrm{d}X_\mathrm{I}} .
\end{aligned} \tag{43.108}$$

The noise spectrum $V(n, \mathrm{i}\omega_k)$ is often assumed to be a complex Gaussian random variable. Now suppose that the real and imaginary parts of the speech spectrum can also be modeled as a Gaussian distribution. Let

$$\sigma_x^2 \triangleq E\left[|X(n, \mathrm{i}\omega_k)|^2\right]$$

and

$$\sigma_v^2 \triangleq E\left[|V(n, \mathrm{i}\omega_k)|^2\right]$$

denote, respectively, the variances of the clean speech and noise spectra. The pdf of the speech spectrum is given by

$$p(X_\mathcal{C}) = \frac{1}{\sqrt{\pi\sigma_x^2}} \exp\left(-\frac{X_\mathcal{C}^2}{\sigma_x^2}\right) , \quad \mathcal{C} \in \{\mathrm{R}, \mathrm{I}\} . \tag{43.109}$$

In such a condition, the conditional pdf $p[Y(n, \mathrm{i}\omega_k)|X_\mathrm{R}, X_\mathrm{I}]$ and the joint pdf $p(X_\mathrm{R}, X_\mathrm{I})$ can be expressed as

$$\begin{aligned}
p[Y(n, \mathrm{i}\omega_k)|X_\mathrm{R}, X_\mathrm{I}] &= p[V(n, \mathrm{i}\omega_k)] \\
&= \frac{1}{\pi\sigma_v^2} \exp\left[-\frac{|V(n, \mathrm{i}\omega_k)|^2}{\sigma_v^2}\right]
\end{aligned} \tag{43.110}$$

and

$$\begin{aligned}
p(X_\mathrm{R}, X_\mathrm{I}) &= p(X_\mathrm{R})p(X_\mathrm{I}) \\
&= \frac{1}{\pi\sigma_x^2} \exp\left[-\frac{|X(n, \mathrm{i}\omega_k)|^2}{\sigma_x^2}\right] .
\end{aligned} \tag{43.111}$$

Substituting (43.110) and (43.111) into (43.107) and (43.108), with some simple mathematical manipulation, we can deduce that

$$\widehat{X}_{\mathrm{MMSE}}(n, \mathrm{i}\omega_k) = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_v^2} Y(n, \mathrm{i}\omega_k) . \tag{43.112}$$

If we introduce the concept of narrow-band a priori SNR, defined as

$$\zeta_k \triangleq \frac{\sigma_x^2}{\sigma_v^2} , \tag{43.113}$$

we can write (43.112) in the following form:

$$\begin{aligned}
\widehat{X}_{\mathrm{MMSE}}(n, \mathrm{i}\omega_k) &= H_{\mathrm{MMSE}}(\omega_k)Y(n, \mathrm{i}\omega_k) \\
&= \frac{\zeta_k}{1 + \zeta_k} Y(n, \mathrm{i}\omega_k) .
\end{aligned} \tag{43.114}$$

The form of (43.114) is identical to that of the estimate [given in (43.93)] obtained with the Wiener filter. This is expected since for Gaussian variables, the MMSE waveform estimator is always a linear function of the noisy observation [43.47], and so the two optimal MMSE solutions coincide.

In the above derivation, we assumed that the real and imaginary parts of the STFT coefficients of both the clean speech and noise signals can be modeled by a Gaussian distribution. This assumption generally holds for noise and approximately holds for speech when the analysis frame size is large and the span of the correlation of the speech signal is much shorter than the frame size. However, in the noise-reduction application, the frame size is typically selected between 10 and 40 ms.

Many studies have investigated the use of non-Gaussian distributions and found that the real and imaginary parts of the STFT coefficients of speech are better modeled with either the Gamma or Laplacian probability distributions when the frame size is short. In the Gamma case, the pdf of $X_{\mathcal{C}}$ [$\mathcal{C} \in \{\mathrm{R}, \mathrm{I}\}$] is given by

$$p(X_{\mathcal{C}}) = \frac{1}{2\sqrt{\pi}} \left(\frac{3}{2\sigma_x^2}\right)^{1/4} \frac{1}{\sqrt{|X_{\mathcal{C}}|}} \exp\left(-\sqrt{\frac{3}{2\sigma_x^2}}|X_{\mathcal{C}}|\right) . \tag{43.115}$$

In the Laplacian case, we have

$$p(X_{\mathcal{C}}) = \frac{1}{\sqrt{\sigma_x^2}} \exp\left(-\frac{2|X_{\mathcal{C}}|}{\sigma_x^2}\right) . \tag{43.116}$$

If we define the instantaneous narrow-band a posteriori SNR as

$$\gamma_{\mathcal{C},k} \triangleq \frac{Y_{\mathcal{C}}^2}{\sigma_v^2} , \tag{43.117}$$

where $Y_{\mathcal{C}}$ is defined in a similar way to $X_{\mathcal{C}}$, the MMSE estimator for $X_{\mathcal{C}}$ in both Gamma- and Laplacian-distribution conditions can be written in the form

$$\widehat{X}_{\mathcal{C},\mathrm{MMSE}} = H(\zeta_k, \gamma_{\mathcal{C},k}) Y_{\mathcal{C}} , \tag{43.118}$$

where $H(\zeta_k, \gamma_{\mathcal{C},k})$ is a gain function. For the Gamma distribution, the gain function is [43.48]

$$\begin{aligned} H(\zeta_k, \gamma_{\mathcal{C},k}) &= \left[\exp\left(\frac{\epsilon_{\mathcal{C}-}^2}{4}\right) D_{-1.5}(\epsilon_{\mathcal{C}-}) - \exp\left(\frac{\epsilon_{\mathcal{C}+}^2}{4}\right) D_{-1.5}(\epsilon_{\mathcal{C}+})\right] \\ &\Big/ \left\{(\epsilon_{\mathcal{C}+} - \epsilon_{\mathcal{C}-})\left[\exp\left(\frac{\epsilon_{\mathcal{C}-}^2}{4}\right) D_{-1.5}(\epsilon_{\mathcal{C}-}) + \exp\left(\frac{\epsilon_{\mathcal{C}+}^2}{4}\right) D_{-1.5}(\epsilon_{\mathcal{C}+})\right]\right\} , \end{aligned} \tag{43.119}$$

where

$$\epsilon_{\mathcal{C}\pm} \triangleq \frac{\sqrt{3}}{2\sqrt{\zeta_k}} \pm \sqrt{2\gamma_{\mathcal{C},k}} , \tag{43.120}$$

and $D_r(z)$ is the parabolic cylinder function [43.49]. For the Laplacian speech model, the gain function is written as [43.50, 51]

$$H(\zeta_k, \gamma_{\mathcal{C},k}) = \frac{2\left[\kappa_{\mathcal{C}+}\mathrm{erfcx}(\kappa_{\mathcal{C}+}) - \kappa_{\mathcal{C}-}\mathrm{erfcx}(\kappa_{\mathcal{C}-})\right]}{(\kappa_{\mathcal{C}+} - \kappa_{\mathcal{C}-})\left[\mathrm{erfcx}(\kappa_{\mathcal{C}+}) + \mathrm{erfcx}(\kappa_{\mathcal{C}-})\right]} , \tag{43.121}$$

where

$$\kappa_{\mathcal{C}\pm} \triangleq \frac{1}{\sqrt{\zeta_k}} \pm \sqrt{\gamma_{\mathcal{C},k}} , \tag{43.122}$$

and erfcx(z) is the scaled complementary error function, defined as

$$\mathrm{erfcx}(z) \triangleq \frac{2}{\sqrt{\pi}} \exp(z^2) \int_z^\infty \exp(-t^2)\,\mathrm{d}t . \tag{43.123}$$

Combining the real and imaginary spectral estimators, we can easily construct the MMSE estimator of the clean speech for both Gamma and Laplacian distributions.

43.5.2 MMSE Spectral Amplitude and Phase Estimators

The previous section discussed an MMSE estimator that estimates the real and imaginary spectral components and then combines the results together to form an estimate of the clean speech spectrum. Now, in this section, we take a slightly different perspective and address MMSE estimators of amplitude and phase spectra.

Consider the decomposition of the complex spectrum into amplitude and phase components, i. e.,

$$Y(n, \mathrm{i}\omega_k) = Y_k \exp\left(\mathrm{i}\theta_{Y_k}\right) , \tag{43.124a}$$

$$X(n, \mathrm{i}\omega_k) = X_k \exp\left(\mathrm{i}\theta_{X_k}\right) , \tag{43.124b}$$

where Y_k and X_k denote, respectively, the amplitudes of the noisy and clean speech spectra, and θ_{Y_k} and θ_{X_k} are their corresponding phase components. The problem of noise reduction then becomes one of designing two signal estimators that make decisions separately on the magnitude and phase spectra from the observed signal.

The MMSE estimator of the spectral amplitude X_k is given by the conditional mean

$$\begin{aligned} \widehat{X}_{k,\mathrm{MMSE}} &= \int_0^\infty X_k\, p[X_k | Y(n, \mathrm{i}\omega_k)]\,\mathrm{d}X_k \\ &= \frac{\int_0^\infty \int_0^\pi X_k\, p\left[Y(n,\mathrm{i}\omega_k)|X_k,\theta_{X_k}\right] p\left(X_k,\theta_{X_k}\right)\mathrm{d}X_k\,\mathrm{d}\theta_{X_k}}{\int_0^\infty \int_0^\pi p\left[Y(n,\mathrm{i}\omega_k)|X_k,\theta_{X_k}\right] p\left(X_k,\theta_{X_k}\right)\mathrm{d}X_k\,\mathrm{d}\theta_{X_k}} . \end{aligned} \tag{43.125}$$

If both noise and speech spectra are modeled as a Gaussian distribution, the conditional pdf $p[Y(n, \mathrm{i}\omega_k)|X_k, \theta_{X_k}]$ can easily be deduced as

$$p\left[Y(n, \mathrm{i}\omega_k)|X_k, \theta_{X_k}\right] = \frac{1}{\pi\sigma_v^2} \exp\left[-\frac{|V(n, \mathrm{i}\omega_k)|^2}{\sigma_v^2}\right] . \tag{43.126}$$

Amazon EX1010 Page 864

It is known that, for complex Gaussian random variables with zero mean, their amplitude and phase components are statistically independent [43.47]. Therefore, we have

$$p(X_k, \theta_{X_k}) = p(X_k)p(\theta_{X_k}) , \tag{43.127}$$

where $p(X_k)$ is a Rayleigh density,

$$p(X_k) = \frac{2X_k}{\sigma_x^2} \exp\left(-\frac{X_k^2}{\sigma_x^2}\right) , \tag{43.128}$$

and $p(\theta_{X_k})$ is a uniform density with

$$p(\theta_{X_k}) = \frac{1}{2\pi} . \tag{43.129}$$

Substituting (43.126) and (43.127) into (43.125) gives

$$\widehat{X}_{k,\mathrm{MMSE}} = \frac{\int_0^\infty X_k^2 \exp\left[-\frac{1}{\sigma_v^2}\left(Y_k^2 + \frac{\sigma_x^2+\sigma_v^2}{\sigma_v^2}\right)\right] I_0\left(\frac{2X_kY_k}{\sigma_v^2}\right) \mathrm{d}X_k}{\int_0^\infty X_k \exp\left[-\frac{1}{\sigma_v^2}\left(Y_k^2 + \frac{\sigma_x^2+\sigma_v^2}{\sigma_v^2}\right)\right] I_0\left(\frac{2X_kY_k}{\sigma_v^2}\right) \mathrm{d}X_k} . \tag{43.130}$$

One can check that the numerator and denominator of (43.130) are in the form of the second- and first-order moments, respectively, of the Rician density function [43.52, 53].

With some mathematical manipulation, the estimator (43.130) can be expressed in the following form [43.27],

$$\widehat{X}_{k,\mathrm{MMSE}} = \Gamma\left(\frac{3}{2}\right)\frac{\sqrt{\vartheta_k}}{\gamma_k}\exp\left(-\frac{\vartheta_k}{2}\right) \times\left[(1+\vartheta_k)I_0\left(\frac{\vartheta_k}{2}\right)+\vartheta_k I_1\left(\frac{\vartheta_k}{2}\right)\right]Y_k , \tag{43.131}$$

where $\Gamma(\cdot)$ denotes the gamma function, with $\Gamma(3/2) = \sqrt{\pi}/2$, $I_0(\cdot)$ and $I_1(\cdot)$ are, respectively, the modified Bessel functions of zero and first order, and

$$\vartheta_k \triangleq \frac{\zeta_k}{1+\zeta_k}\gamma_k \tag{43.132}$$

with

$$\zeta_k \triangleq \frac{\sigma_x^2}{\sigma_v^2} = \frac{P_x(\omega_k)}{P_v(\omega_k)} \tag{43.133}$$

and

$$\gamma_k \triangleq \frac{Y_k^2}{\sigma_v^2} , \tag{43.134}$$

denoting, respectively, the subband a priori and the instantaneous subband a posteriori SNR [43.27].

Now let us estimate the phase component of the clean speech spectrum from the noisy speech spectrum. Ephraim and Malah formulated an MMSE complex exponential estimator [43.27], i. e.,

$$\left[\exp\left(\mathrm{i}\hat{\theta}_{X_k}\right)\right]_{\mathrm{MMSE}} = \arg\min_{\exp\left(\mathrm{i}\hat{\theta}_{X_k}\right)} E\left[\left|\exp\left(\mathrm{i}\theta_{X_k}\right)-\exp\left(\mathrm{i}\hat{\theta}_{X_k}\right)\right|^2\right] . \tag{43.135}$$

It turns out that the solution of this MMSE estimator is

$$\left[\exp\left(\mathrm{i}\hat{\theta}_{X_k}\right)\right]_{\mathrm{MMSE}} = \exp\left(\mathrm{i}\theta_{Y_k}\right) . \tag{43.136}$$

That is, the exponential of the noisy phase θ_{Y_k} is the MMSE exponential estimator of the signal's phase.

Also derived in [43.27] was an optimal estimator of the principle value of the signal's phase, i. e.,

$$\hat{\theta}_{X_k,\mathrm{o}} = \arg\min_{\hat{\theta}_{X_k}} E\left[1-\cos\left(\theta_{X_k}-\hat{\theta}_{X_k}\right)\right] . \tag{43.137}$$

The solution of this estimator is

$$\hat{\theta}_{X_k,\mathrm{o}} = \theta_{Y_k} , \tag{43.138}$$

which indicates that the noisy phase is an optimal estimate of the principle value of the signal's phase.

Combining the amplitude and phase estimators, we can construct an estimate of the clean speech spectrum as

$$\widehat{X}_{\mathrm{MMSE}}(n, \mathrm{i}\omega_k) = \widehat{X}_{k,\mathrm{MMSE}}\exp\left(\mathrm{i}\theta_{Y_k}\right) = H_{\mathrm{MMSE}}(\zeta_k, \gamma_k)Y(n, \mathrm{i}\omega_k) , \tag{43.139}$$

where the gain function $H_{\mathrm{MMSE}}(\zeta_k, \gamma_k)$ is

$$H_{\mathrm{MMSE}}(\zeta_k, \gamma_k) = \Gamma\left(\frac{3}{2}\right)\frac{\sqrt{\vartheta_k}}{\gamma_k}\exp\left(-\frac{\vartheta_k}{2}\right) \times\left[(1+\vartheta_k)I_0\left(\frac{\vartheta_k}{2}\right)+\vartheta_k I_1\left(\frac{\vartheta_k}{2}\right)\right] . \tag{43.140}$$

In high-SNR conditions (i. e., $\zeta_k \gg 1$), (43.140) can be simplified as

$$H_{\mathrm{MMSE}}(\zeta_k, \gamma_k) \approx \frac{\zeta_k}{1+\zeta_k} . \tag{43.141}$$

In such a situation, the MMSE estimate of the clean speech spectrum can be written as

$$\hat{X}_{\mathrm{MMSE}}(n, \mathrm{i}\omega_k) = \frac{\zeta_k}{1+\zeta_k}Y(n, \mathrm{i}\omega_k) = \frac{P_x(\omega_k)}{P_y(\omega_k)}Y(n, \mathrm{i}\omega_k) . \tag{43.142}$$

This result is the same as the frequency-domain Wiener filter. Therefore, the MMSE amplitude estimator converges to the Wiener filter in high-SNR conditions.

43.5.3 Maximum A Posteriori (MAP) Spectral Estimator

We have discussed two MMSE spectral estimators: one optimizes the real and imaginary parts of the clean speech spectrum separately, and the other deals with spectral amplitude and phase separately. We now consider the joint estimation of the spectral amplitude and phase using the MAP criterion. The MAP estimator can be written as

$$\widehat{X}_{k,\mathrm{MAP}} = \arg\max_{X_k} J_{\mathrm{MAP}}\left(X_k, \theta_{X_k}\right) , \tag{43.143}$$

and

$$\hat{\theta}_{X_k,\mathrm{MAP}} = \arg\max_{\theta_{X_k}} J_{\mathrm{MAP}}\left(X_k, \theta_{X_k}\right) , \tag{43.144}$$

where the MAP cost function is expressed as

$$J_{\mathrm{MAP}}(X_k, \theta_{X_k}) = \ln\left\{p\left[X_k, \theta_{X_k}\middle|Y(n, \mathrm{i}\omega)\right]\right\} . \tag{43.145}$$

For the optimization problems given in (43.143) and (43.144), the MAP cost function above can be equivalently written as

$$\begin{aligned} &J_{\mathrm{MAP}}\left(X_k, \theta_{X_k}\right) \\ &= \ln\left\{p\left[Y(n, \mathrm{i}\omega)\middle|X_k, \theta_{X_k}\right]\right] p\left(X_k, \theta_{X_k}\right)\right\} . \end{aligned} \tag{43.146}$$

If we assume that both speech and noise spectra can be modeled by Gaussian distribution, according to (43.126) and (43.127), we can readily deduce that

$$\begin{aligned} &p\left[Y(n, \mathrm{i}\omega)\middle|X_k, \theta_{X_k}\right]\right] p(X_k, \theta_{X_k}) \\ &= \frac{X_k}{\pi^2\sigma_x^2\sigma_v^2} \\ &\quad \times \exp\left[-\frac{X_k^2}{\sigma_x^2} - \frac{\left|Y(n, \mathrm{i}\omega_k) - X_k \exp\left(\mathrm{i}\theta_{X_k}\right)\right|^2}{\sigma_v^2}\right] . \end{aligned} \tag{43.147}$$

Substituting (43.147) into (43.146), differentiating the MAP cost function with respect to X_k, and equating the result to zero, we obtain [43.30]

$$\widehat{X}_{k,\mathrm{MAP}} = \frac{\zeta_k + \sqrt{\zeta_k^2 + 2(1+\zeta_k)\zeta_k/\gamma_k}}{2(1+\zeta_k)} Y_k , \tag{43.148}$$

where ζ_k and γ_k denote, respectively, the subband a priori and instantaneous subband a posteriori SNR as defined in Sect. 43.5.1.

Similarly, if we substitute (43.147) into (43.146), differentiate the MAP cost function with respect to θ_X, and equate the result to zero, we obtain

$$\hat{\theta}_{X_k,\mathrm{MAP}} = \theta_{Y_k} . \tag{43.149}$$

So, this MAP estimate of the speech phase, which is identical to the MMSE phase estimator, is the phase of the noisy speech.

Combining (43.148) and (43.149), we achieve the MAP estimate of the clean speech spectrum

$$\widehat{X}_{\mathrm{MAP}}(n, \mathrm{i}\omega_k) = H_{\mathrm{MAP}}(\omega_k) Y(n, \mathrm{i}\omega_k) , \tag{43.150}$$

where

$$H_{\mathrm{MAP}}(\omega_k) = \frac{\zeta_k + \sqrt{\zeta_k^2 + 2(1+\zeta_k)\zeta_k/\gamma_k}}{2(1+\zeta_k)} . \tag{43.151}$$

When $\zeta_k \gg 1$, we can approximately have

$$H_{\mathrm{MAP}}(\omega_k) \approx \frac{\zeta_k}{1+\zeta_k} . \tag{43.152}$$

Therefore, in high-SNR conditions, this MAP estimator will be the same as the MMSE estimator and the Wiener filter.

43.5.4 Maximum-Likelihood Spectral Amplitude Estimator

We have seen from previous analysis that the noise-reduction problem is essentially a matter of recovering the spectral amplitude since the optimal estimate of the signal's phase turned out to be the noisy phase. One may still want to ask: can noise reduction performance be augmented by recovering the signal's phase? To answer this question, many perception experiments have been conducted and the results have shown that the human perception system is relatively insensitive to phase corruption. At least, within the context of noise reduction, it has been shown that speech distortion resulting from phase corruption is generally imperceptible when subband SNR at any ω_k is greater than about 6 dB [43.15]. In addition, *Wang* and *Lim* [43.54] compared the degree of distortion to reconstructed speech using noisy estimates of spectral magnitude and phase. Using listening tests, their results showed that the subjective quality of speech increases dramatically with an increase in the accuracy of the magnitude estimate, whereas speech quality increases only marginally with increases in the accuracy of the phase estimate.

From both the signal-processing and perception points of view, we see that the use of noisy phase as

the signal's phase is good enough for speech enhancement in most applications. As a result, the single-channel noise-reduction problem is generally formulated to acquire an estimate of the amplitude of the clean speech spectrum. Now let us consider how to estimate the amplitude of the clean speech spectrum. One straightforward way of doing this is through spectral substraction [43.12] or parametric spectral subtraction [43.17]. These techniques will be extensively discussed in the following chapter. Here we consider the maximum likelihood (ML) method, which is one of the most popularly used statistical estimators due to its asymptotic optimal property, i. e., the estimation variance can achieve the Cramèr–Rao lower bound (CRLB) when the number of observation samples approaches infinity.

Let us rewrite the signal model given in (43.3) in the form

$$\begin{aligned} Y(n,\mathrm{i}\omega_k) &= X(n,\mathrm{i}\omega_k)+V(n,\mathrm{i}\omega_k) \\ &= X_k\exp\left(\mathrm{i}\theta_{X_k}\right)+V(n,\mathrm{i}\omega_k)\,. \end{aligned} \tag{43.153}$$

Then the ML estimator for X_k can be written as

$$\widehat{X}_{k,\mathrm{ML}} = \arg\max_{X_k} \mathcal{L}\,, \tag{43.154}$$

where

$$\begin{aligned} \mathcal{L} &= \ln\left\{p\left[Y(n,\mathrm{i}\omega_k)|X_k\right]\right\} \\ &= \ln\left\{\int_{-\pi}^{\pi} p\left[Y(n,\mathrm{i}\omega_k)|X_k,\theta_{X_k}\right]p\left(\theta_{X_k}\right)\,\mathrm{d}\theta_{X_k}\right\} \end{aligned} \tag{43.155}$$

is the log-likelihood cost function. If the noise spectrum $V(n,\mathrm{i}\omega_k)$ is assumed to be a Gaussian process, the conditional pdf $p[Y(n,\mathrm{i}\omega_k)|X_k,\theta_{X_k}]$ is given by (43.126). Substituting (43.126) into (43.155) and assuming that θ_{X_k} is uniformly distributed between $-\pi$ and π, we can deduce that

$$\begin{aligned} &p\left[Y(n,\mathrm{i}\omega_k)|X_k\right] \\ &= \frac{1}{\pi\sigma_v^2}\exp\left[-\frac{Y_k^2+X_k^2}{\sigma_v^2}\right]I_0\left[\frac{2X_kY_k}{\sigma_v^2}\right]\,. \end{aligned} \tag{43.156}$$

Substituting (43.156) into (43.155), differentiating the log-likelihood cost function with respect to X_k, and equating the result to zero, we can obtain the ML estimate of the spectral amplitude. Unfortunately, the above ML estimator is difficult to implement since the Bessel function is not reducible. However, the modified Bessel function of order zero, when its argument is large, can be approximated as

$$I_0(|\alpha|) = \frac{1}{\sqrt{2\pi|\alpha|}}\exp(|\alpha|)\,,\quad \text{for } |\alpha|\gg 1\,. \tag{43.157}$$

So, when $2X_kY_k/\sigma_v^2 \gg 1$ (this is equivalent to saying that the a priori SNR is high), the conditional pdf $p\left[Y(n,\mathrm{i}\omega_k)|X_k,\theta_X\right]$ can be approximated as

$$\begin{aligned} &p[Y(n,\mathrm{i}\omega_k)|X_k] \\ &\approx \frac{1}{2\pi\sigma_v\sqrt{\pi X_kY_k}}\exp\left(-\frac{Y_k^2-2X_kY_k+X_k^2}{\sigma_v^2}\right)\,. \end{aligned} \tag{43.158}$$

Substituting this approximation into (43.155), we can readily derive an approximate ML estimator for the spectral amplitude, i. e.,

$$\widehat{X}_{k,\mathrm{MLA}} = \frac{Y_k+\sqrt{Y_k^2-\sigma_v^2}}{2}\,. \tag{43.159}$$

In this case, the estimate of the clean speech spectrum can be expressed as

$$\begin{aligned} \hat{X}_{\mathrm{MLA}}(n,\mathrm{i}\omega_k) &= \widehat{X}_{k,\mathrm{MLA}}\exp\left(\theta_{Y_k}\right) \\ &= H_{\mathrm{MLA}}(\omega_k)Y(n,\mathrm{i}\omega_k)\,, \end{aligned} \tag{43.160}$$

where

$$H_{\mathrm{MLA}}(\omega_k) = \frac{1+\sqrt{(Y_k^2-\sigma_v^2)/Y_k^2}}{2}$$

is a gain filter.

43.5.5 Maximum-Likelihood Spectral Power Estimator

The ML method can also be applied to estimate the spectral power. For the signal model given in (43.3), if we assume both the clean and noise signals are Gaussian processes, then $Y(n,\mathrm{i}\omega_k)$ is also a Gaussian random variable whose pdf is given by

$$p[Y(n,\mathrm{i}\omega_k)] = \frac{1}{\pi\left(\sigma_x^2+\sigma_v^2\right)}\exp\left(-\frac{Y_k^2}{\sigma_x^2+\sigma_v^2}\right)\,. \tag{43.161}$$

The log-likelihood function of $Y(n,\mathrm{i}\omega_k)$, given the speech power σ_x^2 and noise power σ_v^2, can be written as

$$\begin{aligned} \mathcal{L} &= \ln\left\{p\left[Y(n,\mathrm{i}\omega_k)|\sigma_x^2,\sigma_v^2\right]\right\} \\ &= -\ln(\pi)-\ln\left(\sigma_x^2+\sigma_v^2\right)-\frac{Y_k^2}{\sigma_x^2+\sigma_v^2}\,. \end{aligned} \tag{43.162}$$

Differentiating the log-likelihood function $\mathcal{L}$ with respect to σ_x^2 and equating the result to zero, we can readily derive the ML spectral power estimator,

$$\hat{\sigma}^2_{X,\mathrm{ML}} = Y_k^2 - \sigma_v^2 \,. \tag{43.163}$$

An estimate of the clean speech spectrum can then be constructed as

$$\begin{aligned}\hat{X}_{\mathrm{MLP}}(n, \mathrm{i}\omega_k) &= \sqrt{\hat{\sigma}^2_{X,\mathrm{ML}}}\exp\left(\mathrm{i}\theta_{Y_k}\right) \\ &= H_{\mathrm{MLP}}(\omega_k)Y(n, \mathrm{i}\omega_k)\,,\end{aligned} \tag{43.164}$$

where

$$H_{\mathrm{MLP}}(\omega_k) = \sqrt{\frac{Y_k^2 - \sigma_v^2}{Y_k^2}} \tag{43.165}$$

is a gain filter.

43.5.6 MAP Spectral Amplitude Estimator

In the previous subsections, we developed two ML estimators that assume the clean speech spectrum is fixed but unknown. We now depart from this philosophy to investigate a particular maximum a posteriori (MAP) estimator, namely,

$$\widehat{X}_{k,\,\mathrm{MAP}} = \arg\max_{X_k} J_{\mathrm{MAP}}(X_k)\,, \tag{43.166}$$

where the MAP cost function is written as

$$\begin{aligned}&J_{\mathrm{MAP}}(X_k) \\ &= \ln\{p[X_k|Y(n, \mathrm{i}\omega)]\} \\ &= \ln\left\{\int_{-\pi}^{\pi} p\left[Y(n, \mathrm{i}\omega)|X_k, \theta_{X_k}\right]p\left(X_k, \theta_{X_k}\right)p\left(\theta_{X_k}\right)\mathrm{d}\theta_{X_k}\right\}.\end{aligned} \tag{43.167}$$

Following (43.126) and (43.127), if we define

$$\rho_k = \frac{\sigma_x^2\sigma_v^2}{2\left(\sigma_x^2 + \sigma_v^2\right)} \tag{43.168}$$

and

$$\varrho_k = \sqrt{\rho_k \vartheta_k}\,, \tag{43.169}$$

where ϑ_k is defined in (43.132), we find that

$$\begin{aligned}&p[X_k|Y(n, \mathrm{i}\omega)] \\ &= \frac{X_k}{\rho_k}\exp\left(-\frac{X_k^2 + \varrho_k^2}{2\rho_k}\right)I_0\left(\frac{X_k\varrho_k}{\rho_k}\right)\,.\end{aligned} \tag{43.170}$$

Using the approximation given in (43.157), we can further write (43.170) in the form

$$p[X_k|Y(n, \mathrm{i}\omega)] \approx \frac{\sqrt{X_k}}{\sqrt{2\pi\rho_k\varrho_k}}\exp\left[-\frac{(X_k - \varrho_k)^2}{2\rho_k}\right]\,. \tag{43.171}$$

Substituting (43.171) into (43.167), differentiating the cost function with respect to X_k, and equating the result to zero, we find

$$\widehat{X}_{k,\mathrm{MAP}} = \frac{\zeta_k + \sqrt{\zeta_k^2 + (1+\zeta_k)\zeta_k/\gamma_k}}{2(1+\zeta_k)}Y_k\,, \tag{43.172}$$

where ζ_k is the frequency-domain subband a priori SNR (43.133) and γ_k is the instantaneous subband a posteriori SNR (43.133). Together with the noisy phase, we can construct an estimate of the clean speech spectrum via the approximate MAP method as

$$\widehat{X}_{\mathrm{MAPA}}(n, \mathrm{i}\omega_k) = H_{\mathrm{MAPA}}(\omega_k)Y(n, \mathrm{i}\omega_k)\,, \tag{43.173}$$

where

$$H_{\mathrm{MAPA}}(\omega_k) = \frac{\zeta_k + \sqrt{\zeta_k^2 + (1+\zeta_k)\zeta_k/\gamma_k}}{2(1+\zeta_k)}\,. \tag{43.174}$$

43.6 Speech-Model-Based Noise Reduction

From Von Kempelen's and Wheatstone's speaking machines, to *Stewart*'s vocoder, to *Dunn*'s electrical vocal tract, and then to *Atal* and *Itakura*'s linear prediction (LP) technique [43.55–60], it took scientists nearly two centuries to develop mathematical models that can represent human speech production and characterize the changing characteristics of speech. The discovery of speech models, particularly discrete ones such as the harmonic model, LP model, and hidden Markov model (HMM), has opened a new era for speech processing. For example, with the use of discrete speech models, noise reduction can be converted into a parameter-estimation problem in a much lower-dimensional space. In this way, we can fully take advantage of our knowledge of speech production to enhance a speech signal corrupted by additive noise.

43.6.1 Harmonic-Model-Based Noise Reduction

The harmonic (or sinusoidal) model has been applied with certain success to many speech processing problems [43.14, 61–66]. In this model, the speech signal is represented as a sum of harmonically related sine waves with frequencies given by pitch harmonics. Specifically, at time n, the speech signal $x(n)$ is represented as

$$x(n) = \sum_{i=1}^{Q} A_i(n) \cos(2\pi f_i n + \phi_i) \,, \tag{43.175}$$

where $A_i(n)$, f_i, and ϕ_i are, respectively, the time-varying amplitude, frequency, and initial phase of the i-th tone, and Q denotes the total number of harmonics. For voiced speech, the f_i are multiples of the fundamental frequency f_0, i. e., $f_i = i f_0$. Hence the model (43.175) reduces to

$$x(n) = \sum_{i=1}^{Q} A_i(n) \cos(2\pi i f_0 n + \phi_i) \,. \tag{43.176}$$

With the above speech source model, the microphone observation signal $y(n)$ can be decomposed into two parts: harmonics of speech, and noise. Since speech has energy only at harmonics, while noise, in general, has energy in the frequency regions between harmonics, comb filtering can be applied to reduce noise while preserving the speech periodicity, provided that the information of the fundamental frequency is known. In other words, an estimate of the clean speech can be obtained as [43.67]

$$\hat{x}(n) = y(n) * h_{\mathrm{COMB}}(n) \,, \tag{43.177}$$

where

$$h_{\mathrm{COMB}}(n) = \sum_{i=-K}^{K} h_i \delta(n - N_i) \tag{43.178}$$

is the impulse response of the comb filter, $\delta(n)$ denotes the unit impulse function, h_i are the filter coefficients, which satisfy $\sum_{i=-K}^{K} h_i = 1$,

$$N_i = \begin{cases} -\sum_{j=0}^{i-1} T_j \,, & i > 0 \\ 0 \,, & i = 0 \\ -\sum_{j=i}^{-1} T_j \,, & i < 0 \end{cases} \tag{43.179}$$

and T_j is the instantaneous pitch period, which contains the point of the speech waveform that is multiplied by the filter coefficient h_i. It is seen that the length of the comb filter is $2K+1$ pitch periods.

With accurate and reliable algorithms to update both the filter coefficients h_i and the time-varying pitch T_j, the comb filtering operation given in (43.177) can significantly attenuate the background noise presented in the noisy speech signal. However, one major drawback with this approach is that comb filtering should be restricted only to vowels or vowel-like sounds. If applied to the unvoiced speech, it usually introduces audible speech distortion, which is undesirable. Many efforts have been devoted to improving this method, such as dealing with voiced and unvoiced speech separately, or using a more accurate harmonic-plus-noise model (HNM). But, in general, this method suffers higher speech distortion than filtering, spectral restoration, and other model-based noise-reduction techniques.

43.6.2 Linear-Prediction-Based Noise Reduction

The basic idea underlying the linear prediction (LP) model [also called the autoregressive (AR) model] is that, during a short-time stationary frame of speech, the production system can be described by an all-pole system driven by an excitation sequence [43.41, 59, 60, 68]. The excitation can be either a pulse train (for voiced speech) or random noise (for unvoiced sound). For ease of analysis, let us first assume that the excitation is random noise. In this situation, the speech signal $x(n)$, at time n can be written as

$$\begin{aligned} x(n) &= \sum_{p=1}^{P} a_{x,p} x(n-p) + g_x u(n) \\ &= \boldsymbol{a}_x^{\mathrm{T}} \boldsymbol{x}(n-1) + g_x u(n) \,, \end{aligned} \tag{43.180}$$

where $a_{x,p}$ $(p = 1, 2, \cdots, P)$ are the LP coefficients, which are assumed constant over the analysis speech frame, P is the order of the linear prediction, g_x represents the gain of the excitation, $u(n)$ is a Gaussian random process with zero mean and unit variance, and

$$\begin{aligned} \boldsymbol{a}_x &= [\, a_{x,1} \;\; a_{x,2} \;\; \cdots \;\; a_{x,P} \,]^{\mathrm{T}} \,, \\ \boldsymbol{x}(n-1) &= [\, x(n-1) \;\; x(n-2) \;\; \cdots \;\; x(n-P) \,]^{\mathrm{T}} \,. \end{aligned} \tag{43.181}$$

It is seen from (43.180) that the clean speech $x(n)$ depends on a total of $2P+1$ parameters, namely the P coefficients in the prediction vector $\boldsymbol{a}_x$, the single gain factor g_x, and the P speech samples in the initial $\boldsymbol{x}$ vector; we denote this initial vector by $\boldsymbol{x}(0)$. With this LP

speech model, the microphone signal observed in the presence of noise can be written as

$$y(n) = x(n) + v(n) = \boldsymbol{a}_x^{\mathrm{T}} \boldsymbol{x}(n-1) + g_x u(n) + v(n) \,. \tag{43.182}$$

It becomes immediately clear that the noise-reduction problem can be formulated as one of estimating the parameters $\boldsymbol{a}_x$, g_x, and $\boldsymbol{x}(0)$ from the noisy observation $y(n)$. Relatively, the initial vector plays a less important role than the LP coefficients and gain factor, so we can simply assume that this vector consists of all zeros or the first P samples of the noisy speech signal. Therefore, the noise-reduction problem is indeed a matter of estimating the LP coefficients and gain factor, which together are called the LP model parameters.

Following the same line of ideas discussed in Sect. 43.5, we can formulate different estimators such as MMSE, ML, MAP, etc., to estimate the LP model parameters of the clean speech from the noisy observations. One widely investigated method is the sequential MAP estimator originally developed by *Lim* and *Oppenheim* [43.13, 69]. This method optimizes the LP model parameters and the clean speech estimate in an iterative way until some criterion is satisfied. Suppose that at time n we have a frame of observation signal $\boldsymbol{y}(n)$ and denote $\hat{\boldsymbol{x}}_i(n)$, $\hat{\boldsymbol{a}}_{x,i}$, and $\hat{g}_{x,i}$, respectively, as the estimate of the clean speech vector $\boldsymbol{x}(n)$, the LP vector $\boldsymbol{a}_x$, and the gain factor g_x from the ith iteration. The sequential MAP technique estimates the LP model parameters and the clean speech vector based on the following steps:

1. Obtain a MAP estimate of the LP model parameters based on the previous estimate of the model parameters and clean speech, i. e.,

$$\left(\hat{\boldsymbol{a}}_{x,i}, \hat{g}_{x,i}\right) = \arg\max_{\boldsymbol{a}_x, g_x} p\left[\boldsymbol{a}_x, g_x | \boldsymbol{y}(n), \hat{\boldsymbol{x}}_{i-1}, \boldsymbol{x}(0)\right] \,. \tag{43.183}$$

2. Obtain a MAP estimate of the clean speech with the newly estimated model parameters, i. e.,

$$\hat{\boldsymbol{x}}_i(n) = \arg\max_{\boldsymbol{x}(n)} p\left[\boldsymbol{x}(n) | \boldsymbol{y}(n), \hat{\boldsymbol{a}}_{x,i}, \hat{g}_{x,i}\right] \,. \tag{43.184}$$

With the assumption that all the unknown parameters are random with a Gaussian pdf, it can be shown that the first step is equivalent to estimating the LP model parameters from $\hat{\boldsymbol{x}}_{i-1}$ using the correlation method of linear prediction analysis, and the second step achieves the clean speech estimate through the optimal Wiener filter

$$H_{\mathrm{o},i}(\omega) = \frac{\hat{P}_x(\omega, i)}{\hat{P}_x(\omega, i) + \hat{P}_v(\omega, i)} \,, \tag{43.185}$$

where

$$\hat{P}_x(\omega, i) = \frac{g_x^2}{\left|1 - \hat{\boldsymbol{a}}_{x,i}^{\mathrm{T}} \boldsymbol{f}\right|^2} \,,$$
$$\boldsymbol{f} = [\exp(-\mathrm{i}\omega) \;\; \exp(-\mathrm{i}2\omega) \;\; \cdots \;\; \exp(-\mathrm{i}P\omega)]^{\mathrm{T}} \,, \tag{43.186}$$

and $\hat{P}_v(\omega, i) = \sigma_v^2$ if $v(n)$ is white Gaussian noise. We see that the second step is the same as the frequency-domain Wiener filter as discussed in Sect. 43.4.4. The only difference is that in Sect. 43.4.4, the power spectrum is estimated via periodogram, while here it is computed from the LP model. It has been shown that, if the above two steps are repeated until some specified error criterion is satisfied, this iterative technique can increase the joint likelihood of $\hat{\boldsymbol{a}}_{x,i}$ and $\hat{\boldsymbol{x}}_i(n)$ with each iteration, thereby provide an optimal estimator in the MMSE sense. The above method can also be easily extended to the colored-noise case [43.69].

Another intensively investigated estimation method is based on the Kalman filtering technique. Putting (43.180) into matrix/vector form, we get

$$\begin{aligned} \boldsymbol{x}(n) &= \boldsymbol{F}\boldsymbol{x}(n-1) + \boldsymbol{g}u(n) \,, \\ y(n) &= \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}(n) + v(n) \,, \end{aligned} \tag{43.187}$$

where

$$\boldsymbol{F} = \begin{pmatrix} a_{x,1} & a_{x,2} & \cdots & a_{x,P-1} & a_{x,P} \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \cdots & \cdots & \vdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \tag{43.188}$$

and

$$\boldsymbol{g} = \boldsymbol{c} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \end{bmatrix}^{\mathrm{T}} \,. \tag{43.189}$$

If we assume that all the parameters $a_{x,p}$ ($p = 1, 2, \cdots, P$), g_x, and σ_v^2 are known, an optimal estimate of $\boldsymbol{x}(n)$ can be obtained using the Kalman filter [43.31]. Briefly, the solution of using Kalman filter can be sum-

marized as

$$\hat{\boldsymbol{x}}(n)=\boldsymbol{F}\hat{\boldsymbol{x}}(n-1)+\boldsymbol{k}(n)[y(n)-\boldsymbol{c}^{\mathrm{T}}\hat{\boldsymbol{x}}(n-1)]\,, \tag{43.190a}$$

$$\boldsymbol{k}(n)=\boldsymbol{M}(n|n-1)\boldsymbol{c}\left[\boldsymbol{c}^{\mathrm{T}}\boldsymbol{M}(n|n-1)\boldsymbol{c}+\sigma_v^2\right]^{-1}, \tag{43.190b}$$

$$\boldsymbol{M}(n|n-1)=\boldsymbol{F}\boldsymbol{M}(n-1)\boldsymbol{F}^{\mathrm{T}}+\sigma_u^2\boldsymbol{g}\boldsymbol{g}^{\mathrm{T}}\,, \tag{43.190c}$$

$$\boldsymbol{M}(n)=[\boldsymbol{I}-\boldsymbol{k}(n)\boldsymbol{c}^{\mathrm{T}}]\boldsymbol{M}(n|n-1)\,. \tag{43.190d}$$

where $\hat{\boldsymbol{x}}(n)$ is the estimate of $\boldsymbol{x}(n)$, $\boldsymbol{k}(n)$ is the Kalman gain vector, $\boldsymbol{M}(n|n-1)=E\{[\boldsymbol{x}(n)-\boldsymbol{F}\hat{\boldsymbol{x}}(n-1)][\boldsymbol{x}(n)-\boldsymbol{F}\hat{\boldsymbol{x}}(n-1)]^{\mathrm{T}}\}$ is the predicted state-error covariance matrix, and $\boldsymbol{M}(n|n)=E\{[\boldsymbol{x}(n)-\hat{\boldsymbol{x}}(n)][\boldsymbol{x}(n)-\hat{\boldsymbol{x}}(n)]^{\mathrm{T}}\}$ is the filtered state-error covariance matrix. After Kalman filtering, the estimate of the clean speech at time n is

$$\hat{x}(n)=\boldsymbol{c}^{\mathrm{T}}\hat{\boldsymbol{x}}(n)\,. \tag{43.191}$$

In this Kalman filtering technique, we make the following assumptions:

1. Within the analysis speech frame, both the LP coefficients and the excitation are constant.
2. The LP model parameters of the clean speech are known.
3. The excitation can be modeled as a Gaussian random process.
4. The observation noise $v(n)$ is a white Gaussian process.

This technique was first proposed by *Paliwal* and *Basu* in [43.31]. It can be shown that, when the aforementioned four assumptions hold, the method can significantly reduce the observation noise. However, the four assumptions can hardly be satisfied in practical environments, making the technique difficult to implement. Many efforts have been spent to relax some of these assumptions, thereby making the technique more practical. Broadly speaking, these efforts can be classified into four categories:

1. Assuming LP model parameters are time varying but can be modeled as stationary processes [43.70, 71]. The resulting methods are often called time-varying AR-based techniques, which are meant to remedy the first assumption.
2. The LP model parameters of the clean speech are normally not known in real applications. Several methods have been developed to estimate these parameters from the noisy speech. The most representative of these is the iterative or estimate-maximize algorithm [43.33, 72, 73], where the LP model and noise parameters are optimized iteratively, and the enhanced speech is obtained as a byproduct of the parameter estimation. Apparently, this group of efforts are to remedy the second assumption.
3. As we know, the excitation for unvoiced sound can be modeled as random noise, but it should be a pulse train for voiced speech. To model the excitation better, methods such as multiple variances [43.73], algebraic code excitation [43.74], and mixture Gaussian noise [43.73,75] have been developed. These are meant to remedy the third assumption.
4. In the case that $v(n)$ is not white but colored, we can model it with an LP model with a lower order than that of the speech, i. e.,

$$v(n)=\sum_{q=1}^{Q}a_{v,q}v(n-q)+g_v\eta(n)\,. \tag{43.192}$$

In matrix/vector form, (43.192) can be written as

$$\begin{aligned}\boldsymbol{v}(n)&=\boldsymbol{F}_v\boldsymbol{v}(n-1)+\boldsymbol{g}_v\eta(n)\,,\\ v(n)&=\boldsymbol{c}_v^{\mathrm{T}}\boldsymbol{v}(n)\,,\end{aligned} \tag{43.193}$$

where $\boldsymbol{F}_v$, $\boldsymbol{g}_v$, and $\boldsymbol{c}_v$ are defined similarly to their counterparts in (43.187). Now if (43.187) and (43.193) are combined, we get the augmented state equations:

$$\begin{aligned}\tilde{\boldsymbol{x}}(n)&=\tilde{\boldsymbol{F}}\tilde{\boldsymbol{x}}(n-1)+\tilde{\boldsymbol{G}}\tilde{\boldsymbol{u}}(n)\,,\\ y(n)&=\tilde{\boldsymbol{c}}^{\mathrm{T}}\tilde{\boldsymbol{x}}(n)\,,\end{aligned} \tag{43.194}$$

where

$$\tilde{\boldsymbol{x}}(n)=[\,\boldsymbol{x}^{\mathrm{T}}(n)\;\;\boldsymbol{v}^{\mathrm{T}}(n)\,]^{\mathrm{T}}\,.$$

$$\tilde{\boldsymbol{F}}=\begin{pmatrix}\boldsymbol{F}&\boldsymbol{0}\\ \boldsymbol{0}&\boldsymbol{F}_v\end{pmatrix}\,,$$

$$\tilde{\boldsymbol{G}}=\begin{pmatrix}\boldsymbol{g}&\boldsymbol{0}\\ \boldsymbol{0}&\boldsymbol{g}_v\end{pmatrix}\,,$$

$$\tilde{\boldsymbol{u}}(n)=[\,u(n)\;\;\eta(n)\,]^{\mathrm{T}}\,,$$

$$\tilde{\boldsymbol{c}}=\left[\boldsymbol{c}^{\mathrm{T}}\;\;\boldsymbol{c}_v^{\mathrm{T}}\right]^{\mathrm{T}}\,.$$

Then, the Kalman filter given in (43.190a) can easily be generalized to estimate both speech and noise.

43.6.3 Hidden-Markov-Model-Based Noise Reduction

The hidden Markov model (HMM) is a statistical model that uses a finite number of states and the associated

state transitions to jointly model the temporal and spectral variation of signals. It has long been used for speech modeling with applications to speech recognition [43.76] and noise reduction [43.36]. For noise reduction, the HMM-based method is basically similar to the statistical estimators described in Sect. 43.5. However, in Sect. 43.5, the estimators assume explicit knowledge of the joint probability distribution of the clean speech and noise signals, so that the conditional expected value of the clean speech (or its sample spectrum), given the noisy speech, can be evaluated, while in the HMM-based approach such statistical knowledge is not required to be known. Instead, the HMM method conquers the problem in two steps. In the first step, which is often called the training process, the probability distributions of the clean speech and the noise process are estimated from given training sequences. The estimated distributions are then applied in the second step to construct the desired speech estimators.

Typically, for a noise-reduction problem, we only require two ergodic HMMs: one for modeling the clean speech signal and the other for modeling the noise process. Consider HMMs with N states and M mixtures. Let $\boldsymbol{O} \triangleq \{\boldsymbol{o}_0, \boldsymbol{o}_1, \cdots, \boldsymbol{o}_t, \cdots, \boldsymbol{o}_T\}$ be the observed feature sequence, where $\boldsymbol{o}_t$ is a K-dimensional feature vector (e.g., consisting of K consecutive samples of the speech signal or the speech spectrum) and T is the duration of the sequence. Let $\boldsymbol{S} \triangleq \{s_0, s_1, \cdots, s_t, \cdots, s_T\}$, $s_t \in \{1, 2, \cdots, N\}$, be a sequence of states corresponding to $\boldsymbol{O}$. The pdf for $\boldsymbol{O}$ is written

$$p(\boldsymbol{O}) = \sum_{\boldsymbol{S}} p(\boldsymbol{O}, \boldsymbol{S}|\lambda) = \sum_{\boldsymbol{S}} \prod_{t=0}^{T} \alpha_{s_{t-1}s_t} b_{s_t}(\boldsymbol{o}_t) \,, \tag{43.195}$$

where $\alpha_{s_{t-1}s_t}$ is the transition probability from state s_{t-1} at time $t-1$ to state s_t at time t, $\alpha_{s_{-1}s_0} \triangleq \pi_{s_0}$ denotes the initial state probability, $b_{s_t}(\boldsymbol{o}_t)$ is the pdf of the observation vector, and λ denotes the HMM parameter set, i. e.,

$$\lambda \triangleq (\boldsymbol{\Pi}, \boldsymbol{A}, \boldsymbol{b}) \tag{43.196}$$

with

$$\boldsymbol{\Pi} = \{\pi_1, \pi_2, \cdots, \pi_N\} \tag{43.197}$$

being the set of initial state probabilities,

$$\boldsymbol{A} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1N} \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2N} \\ \vdots & \vdots & \cdots & \vdots \\ \alpha_{N1} & \alpha_{N2} & \cdots & \alpha_{\mathrm{NN}} \end{pmatrix} \tag{43.198}$$

being the set of state transition probabilities, and

$$\boldsymbol{b} = \{\, b_1(\boldsymbol{o}_t) \;\; b_2(\boldsymbol{o}_t) \;\; \cdots \;\; b_N(\boldsymbol{o}_t) \,\} \tag{43.199}$$

defining the pdf of the observed signal vector $\boldsymbol{O}$. For Gaussian-mixture autoregressive (AR) HMMs, the pdf $b_{s_t}(\boldsymbol{o}_t)$ is given by

$$b_{s_t}(\boldsymbol{o}_t) = \sum_{\kappa_t} c_{\kappa_t|s_t} b(\boldsymbol{o}_t|s_t, \kappa_t) \,, \tag{43.200}$$

where $\kappa_t \in [1, 2, \cdots, M]$ denotes the mixture component chosen at time t, $c_{\kappa_t|s_t}$ is the probability of choosing the mixture component κ_t given that the process is in state s_t, and $b(\boldsymbol{o}_t|s_t, \kappa_t)$ is the pdf of the observation vector $\boldsymbol{o}_t$ given κ_t and s_t. For a zero-mean Gaussian AR process with order P, we have

$$b(\boldsymbol{o}_t|s_t = n, \kappa_t = m) = \frac{\exp\left(-\frac{1}{2}\boldsymbol{o}_t^{\mathrm{T}} \boldsymbol{\Sigma}_{n,m}^{-1} \boldsymbol{o}_t\right)}{\sqrt{(2\pi)^K \det(\boldsymbol{\Sigma}_{n,m})}} \,, \tag{43.201}$$

where $\boldsymbol{\Sigma}_{n,m}$ is the covariance matrix, which is given by

$$\boldsymbol{\Sigma}_{n,m} = \sigma_{n,m}^2 \left(\boldsymbol{A}_{n,m}^H \boldsymbol{A}_{n,m}\right)^{-1} \,, \tag{43.202}$$

$\sigma_{n,m}^2$ is the variance of the AR process, and $\boldsymbol{A}_{n,m}$ is a $K \times K$ lower-triangular Toeplitz matrix with its first $P+1$ entries in the first column being the AR coefficients, i. e., $\boldsymbol{a}_{n,m} = [a_{n,m}(0), a_{n,m}(1), \cdots, a_{n,m}(P)]^{\mathrm{T}}$, and $a_{n,m}(0) = 1$ [43.34, 77]. In this case, the HMM parameter set λ consists of the following parameters: the initial state probabilities $\boldsymbol{\Pi}$, the state transition probabilities $\boldsymbol{A}$, and the AR parameters $\boldsymbol{a}_{n,m}$ and $\sigma_{n,m}^2$. Given a training data sequence, an ML estimate of λ (for both clean speech and noise) can be obtained using the Baum re-estimation algorithm [43.77]. Alternatively, the segmental k-means algorithm can be used to optimize the parameter set along the dominant state and mixture sequence [43.41].

Once the model parameters for both the clean speech and the noise signals are estimated, statistical estimators of clean speech can be constructed based on different criteria. For example, the MAP estimator can be formulated as

$$\hat{\boldsymbol{x}}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{x}} p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{s}_y, \lambda_x, \lambda_b) \,, \tag{43.203}$$

where $\boldsymbol{x} \triangleq \{\boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_t\}$ denotes a sequence of the clean speech signal, $\boldsymbol{y} \triangleq \{\boldsymbol{y}_0, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_\tau\}$ (with $\tau \geq t$) is the observation signal sequence, $\boldsymbol{s}_y$ denotes the composite state sequence of the noisy signal, and λ_x and λ_b are, respectively, the HMM model parameters for the clean speech and noise signals.

Amazon EX1010 Page 872

Since the composite state sequence $\boldsymbol{s}_y$ is not known, direct maximization of $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{s}_y, \lambda_x, \lambda_b)$ is not trivial. An alternative method is to achieve the maximization iteratively using the estimation-maximization (EM) algorithm [43.78]. Let i denote the iterative index (initially set to zero). The EM algorithm obtains a new speech estimate by maximizing the cost function

$$J[\boldsymbol{x}(i+1)] \triangleq \sum_{\boldsymbol{s}_y} \sum_{\boldsymbol{\kappa}} p\left[\boldsymbol{s}_y, \boldsymbol{\kappa}|\hat{\boldsymbol{x}}(i)\right] \ln p[\boldsymbol{s}_y, \boldsymbol{\kappa}, \boldsymbol{x}(i+1)|\boldsymbol{y}] \,, \tag{43.204}$$

where $\boldsymbol{\kappa} \triangleq \{\boldsymbol{\kappa}_0, \boldsymbol{\kappa}_1, \cdots, \boldsymbol{\kappa}_\tau\}$ is a sequence of mixture components corresponding to $\boldsymbol{y}$ and $\boldsymbol{s}_y$. The solution to the above optimization problem is

$$\hat{\boldsymbol{x}}_t(i+1) = \left[\sum_{n=1}^{N} \sum_{m=1}^{M} q_t[n, m|\hat{\boldsymbol{x}}(i)] H_{n,m}^{-1}\right]^{-1} \boldsymbol{y}_t \,, \tag{43.205}$$

where $q_t[n, m|\hat{\boldsymbol{x}}(i)] = P[q_t = n, \kappa_t = m|\hat{\boldsymbol{x}}(i)]$ is the conditional probability of being in state n and choosing mixture m at time frame t given an estimate of the clean speech, $\hat{\boldsymbol{x}}(i)$, and $H_{n,m}$ is a Wiener filter for the output Gaussian process from state n and mixture m, which can be constructed from the speech and noise covariance matrices given in (43.202) [43.34]. The iterative MAP estimator given in (43.205) can also be implemented in the frequency domain so as to exploit the efficiency of the FFT. In the frequency domain, the estimator can be written as

$$\hat{\boldsymbol{x}}_{t,\omega}(i+1) = \left[\sum_{n=1}^{N} \sum_{m=1}^{M} q_t[n, m|\hat{\boldsymbol{x}}(i)] H_{n,m}^{-1}(\omega)\right]^{-1} \boldsymbol{y}_{t,\omega} \,, \tag{43.206}$$

where $\hat{\boldsymbol{x}}_{t,\omega}(i+1)$ and $\boldsymbol{y}_{t,\omega}$ are the Fourier transforms of $\hat{\boldsymbol{x}}_t(i+1)$ and $\boldsymbol{y}_t$, respectively, and $H_{n,m}(\omega)$ is the frequency-domain counterpart of $H_{n,m}$, which can be constructed from the AR (LPC) coefficients of the speech and noise signals as in (43.185). Briefly, the iterative MAP estimator given in (43.205) and (43.206) can be summarized as follows.

1. The conditional probability $q_t[n, m|\hat{\boldsymbol{x}}(i)]$ [being in state n and mixture m at time t given an estimate of the clean speech, i. e., $\hat{\boldsymbol{x}}(i)$] is evaluated for all possible states n, mixture m, and time frames t (using the forward–backward algorithm [43.76]).
2. For each state and mixture pair (n, m), there is a set of AR (LPC) coefficients. This set of coefficients is used to form a Wiener filter $H_{n,m}^{-1}(\omega)$.
3. A new estimate of the clean speech is obtained by filtering the noisy speech through a weighted sum of Wiener filters (the weights are $q_t[n, m|\hat{\boldsymbol{x}}(i)]$).
4. The new speech estimate is used to re-estimate the conditional probability $q_t[n, m|\hat{\boldsymbol{x}}(i+1)]$.

These four steps are repeated until some preset convergence criterion is satisfied.

Similarly, we can formulate the MMSE estimator presented in [43.79]. Note that the HMM-based estimators rely solely on the statistics of the clean speech inferred from the training data. However, some speech properties, such as the time-varying gain (energy contour), cannot be reliably predicted in the training sequences and these properties have to be directly estimated from the noisy speech [43.36].

The advantage of the HMM-based estimators is that they do not need explicit knowledge of the speech and noise distributions. In addition, they can tolerate some nonstationarity of the noise signal, depending on the number of states and mixtures used in the noise HMM. However, distortion will arise when the characteristics of the noise are not represented in the training set.

43.7 Summary

Noise reduction has long been a major research focus of speech processing for communications. The fundamental objective of noise reduction is to mitigate or eliminate the effects of noise on the desired speech signal. The fulfillment of this objective, however, is not trivial. In general, the formulation of noise reduction varies from one application to another, and the difficulty and complexity of the problem depend on many factors such as the nature of the noise signal, the number of microphones available for noise reduction, and the nature of the speech communication system. This chapter was devoted to a particular yet ubiquitous problem, i. e., single-channel noise reduction. We formulated this particular problem as a signal/parameter estimation problem. Through theoretical analysis, we demonstrated, in terms of

noise attenuation and signal-to-noise ratio (SNR) improvement, that noise reduction is feasible in the single-microphone scenario. However, the price for this is speech distortion. In general, the more the noise reduction, the more the speech distortion. We reviewed the state of the art of noise-reduction techniques. The algorithms were classified into three broad categories, i. e., filtering techniques, spectral restoration, and speech-model-based methods. The basic principle and idea underlying each group of algorithms were outlined and their strengths and shortcomings were also discussed.

References

43.1 J. Benesty, S. Makino, J. Chen (Eds.): *Speech Enhancement* (Springer, Berlin, Heidelberg 2005)

43.2 D.H. Johnson, D.E. Dudgeon: *Array Signal Processing: Concepts and Techniques* (Prentice Hall, Upper Saddle River 1993)

43.3 M. Brandstein, D. Ward (Eds.): *Microphone Arrays: Signal Processing Techniques and Applications* (Springer, Berlin, Heidelberg 2001)

43.4 Y. Huang, J. Benesty (Eds.): *Audio Signal Processing for Next-Generation Multimedia Communication Systems* (Kluwer Academic, Boston 2004)

43.5 B. Widrow, J.R. Glover, J.M. McCool, J. Kaunitz, C.S. Williams, R.H. Hearn, J.R. Zeidler, E. Dong, R.C. Goodwin: Adaptive noise canceling: principles and applications, Proc. IEEE **63**, 1692–1716 (1975)

43.6 B. Widrow, S.D. Stearns: *Adaptive Signal Processing* (Prentice Hall, Englewood Cliffs 1985)

43.7 M.M. Goulding, J.S. Bird: Speech enhancement for mobile telephony, IEEE Trans. Veh. Technol. **39**, 316–326 (1990)

43.8 H.J. Kushner: On closed-loop adaptive noise cancellation, IEEE Trans. Automat. Contr. **43**, 1103–1107 (1998)

43.9 A.S. Abutaled: An adaptive filter for noise canceling, IEEE Trans. Circuits Syst. **35**, 1201–1209 (1998)

43.10 M. R. Schroeder: U.S. Patent No. 3180936, filed Dec. 1, 1960, issued Apr. 27, 1965

43.11 M. R. Schroeder: U.S. Patent No. 3403224, filed May 28, 1965, issued Sept. 24, 1968

43.12 S.F. Boll: Suppression of acoustic noise in speech using spectral subtraction, IEEE Trans. Acoust. Speech Signal Process. **ASSP-27**, 113–120 (1979)

43.13 J.S. Lim, A.V. Oppenheim: Enhancement and bandwidth compression of noisy speech, Proc. IEEE **67**, 1586–1604 (1979)

43.14 J.S. Lim (Ed.): *Speech Enhancement* (Prentice Hall, Englewood Cliffs 1983)

43.15 P. Vary: Noise suppression by spectral magnitude estimation–mechanism and theoretical limits, Signal Process. **8**, 387–400 (1985)

43.16 R. Martin: Noise power spectral density estimation based on optimal smoothing and minimum statistics, IEEE Trans. Speech Audio Process. **9**, 504–512 (2001)

43.17 W. Etter, G.S. Moschytz: Noise reduction by noise-adaptive spectral magnitude expansion, J. Audio Eng. Soc. **42**, 341–349 (1994)

43.18 J. Chen, J. Benesty, Y. Huang, S. Doclo: New insights into the noise reduction Wiener filter, IEEE Trans. Speech Audio Process. **14**, 1218–1234 (2006)

43.19 Y. Ephraim, H.L. Van Trees: A signal subspace approach for speech enhancement, IEEE Trans. Speech Audio Process. **3**, 251–266 (1995)

43.20 M. Dendrinos, S. Bakamidis, G. Garayannis: Speech enhancement from noise: A regenerative approach, Speech Commun. **10**, 45–57 (1991)

43.21 P.S.K. Hansen: *Signal Subspace Methods for Speech Enhancement*, Ph.D. Dissertation (Tech. Univ. Denmark, Lyngby 1997)

43.22 S.H. Jensen, P.C. Hansen, S.D. Hansen, J.A. Sørensen: Reduction of broad-band noise in speech by truncated QSVD, IEEE Trans. Speech Audio Process. **3**, 439–448 (1995)

43.23 H. Lev-Ari, Y. Ephraim: Extension of the signal subspace speech enhancement approach to colored noise, IEEE Trans. Speech Audio Process. **10**, 104–106 (2003)

43.24 A. Rezayee, S. Gazor: An adaptive KLT approach for speech enhancement, IEEE Trans. Speech Audio Process. **9**, 87–95 (2001)

43.25 U. Mittal, N. Phamdo: Signal/noise KLT based approach for enhancing speech degraded by colored noise, IEEE Trans. Speech Audio Process. **8**, 159–167 (2000)

43.26 Y. Hu, P.C. Loizou: A generalized subspace approach for enhancing spech corrupted by colored noise, IEEE Trans. Speech Audio Process. **11**, 334–341 (2003)

43.27 Y. Ephraim, D. Malah: Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator, IEEE Trans. Acoust. Speech Signal Process. **32**, 1109–1121 (1984)

43.28 Y. Ephraim, D. Malah: Speech enhancement using a minimum mean-square error log-spectral amplitude estimator, IEEE Trans. Acoust. Speech Signal Process. **ASSP-33**, 443–445 (1985)

43.29 R.J. McAulay, M.L. Malpass: Speech enhancement using a soft-decision noise suppression filter, IEEE Trans. Acoust. Speech Signal Process. **28**, 137–145 (1980)

43.30 P.J. Wolfe, S.J. Godsill: Simple alternatives to the Ephraim and Malah suppression rule for speech ehancemnet, Proc. IEEE ICASSP **2001**, 496–499 (2001)

43.31 K.K. Paliwal, A. Basu: A speech enhancement method based on Kalman filtering, Proc. IEEE ICASSP **1987**, 177–180 (1987)

43.32 J.D. Gibson, B. Koo, S.D. Gray: Filtering of colored noise for speech enhancement and coding, IEEE Trans. Signal Process. **39**, 1732–1742 (1991)

43.33 S. Gannot, D. Burshtein, E. Weinstein: Iterative and sequential Kalman filter-based speech enhancement algorithms, IEEE Trans. Speech Audio Process. **6**, 373–385 (1998)

43.34 Y. Ephraim, D. Malah, B.-H. Juang: On the application of hidden Markov models for enhancing noisy speech, IEEE Trans. Acoust. Speech Signal Process. **37**, 1846–1856 (1989)

43.35 Y. Ephraim: A Bayesian estimation approach for speech enhancement using hidden Markov models, IEEE Trans. Signal Process. **40**, 725–735 (1992)

43.36 Y. Ephraim: Statstical-model-based speech enhancement systems, Proc. IEEE **80**, 1526–1555 (1992)

43.37 D. Klatt: Review of test-to-speech conversion for English, J. Acoust. Soc. Am. **82**, 737–793 (1987)

43.38 U. Jekosch: Speech quality assessment and evaluation, Proc. Eurospeech **1993**, 1387–1394 (1993)

43.39 C. Delogu, P. Paoloni, P. Pocci, C. Sementina: Quality evaluation of text-to-speech synthesizers using magnitude estimation, categorical estimation, pair comparison and reaction time methods, Proc. Eurospeech **1991**, 353–356 (1991)

43.40 S.R. Quackenbush, T.P. Barnwell, M.A. Clements: *Objective Measures of Speech Quality* (Prentice Hall, Englewood Cliffs 1988)

43.41 L.R. Rabiner, B.H. Juang: *Fundamentals of Speech Recognition* (Prentice Hall, Englewood Cliffs 1993)

43.42 D. Mansour, B.H. Juang: A family of distortion meansures based upon projection operation for robust speech recognition, IEEE Trans. Acoust. Speech Signal Process. **37**, 1659–1671 (1989)

43.43 F. Itakura, S. Saito: A statistical method for estimation of speech spectral density and formant frequencies, Electron. Commun. Jpn. **53A**, 36–43 (1970)

43.44 G. Chen, S.N. Koh, I.Y. Soon: Enhanced Itakura measure incorporating masking properties of human auditory system, Signal Process. **83**, 1445–1456 (2003)

43.45 K. Fukunaga: *Introduction to Statistial Pattern Recognition* (Academic, San Diego 1990)

43.46 N. Wiener: *Extrapolation, Interpolation, and Smoothing of Stationary Time Series* (Wiley, New York 1949)

43.47 H.L. Van Trees: *Dection, Estimation, and Modulation Theory*, Part I (Wiley, New York 1968)

43.48 R. Martin: Speech enhancement using MMSE short time spectral estimation with Gamma distributed speech priors, Proc. IEEE ICASSP **2002**, I253–I256 (2002)

43.49 I.S. Gradshteyn, I.M. Ryzhik, A. Jeffery, D. Zwillinger (Eds.): *Table of Integrals, Series, and Products* (Academic, San Diego 2000)

43.50 C. Breithaupt, R. Martin: MMSE estimation fo magnitude-square DFT coefficients with supergaussian priors, Proc. IEEE ICASSP **2003**, I848–I851 (2003)

43.51 I. Cohen: Speech enhancement using supergaussian speech models and noncausal a priori SNR estimation, Speech Commun. **47**, 336–350 (2005)

43.52 S.O. Rice: Stasitical properties of a sinewave plus random noise, Bell System Tech. J. **0**, 109–157 (1948)

43.53 D. Middleton, R. Esposito: Simultaneous optimum detection and estimation of signals in noise, IEEE Trans. Inform. Theory **IT-14**, 434–444 (1968)

43.54 D.L. Wang, J.S. Lim: The unimportance of phase in speech enhancement, IEEE Trans. Acoust. Speech Signal Process. **ASSP-30**, 679–681 (1982)

43.55 H. Dudley, T.H. Tarnoczy: The speaking machine of Wolfgang von Kempelen, J. Acoust. Soc. Am. **22**, 151–166 (1950)

43.56 Sir R. Paget: *Human Speech* (Harcourt, London, New York 1930)

43.57 J.Q. Stewart: An electrical analogue of the vocal cords, Nature **110**, 311–312 (1922)

43.58 H.K. Dunn: The calculation of vowel resonances, and an electrical vocal tract, J. Acoust. Soc. Am. **22**, 740–753 (1950)

43.59 B.S. Atal, L.S. Hanauer: Speech analysis and synthesis by linear prediction of the speech wave, J. Acoust. Soc. Am. **50**, 637–655 (1971)

43.60 F. Itakura: Minimum prediction residual principle applied to speech recognition, IEEE Trans. Acoust. Speech Signal Process. **ASSP-23**, 67–72 (1975)

43.61 T.W. Parsons: Separation of speech from interfering speech by means of harmonic selection, J. Acoust. Soc. Am. **60**, 911–918 (1976)

43.62 R.H. Frazier, S. Samsam, L.D. Braida, A.V. Oppenheim: Enhancement of speech by adaptive filtering, Proc. IEEE ICASSP **1976**, 251–253 (1976)

43.63 R.J. McAulay, T.F. Quatieri: Mid-rate coding based on sinusoidal representation of speech, Proc IEEE ICASSP **1985**, 945–948 (1985)

43.64 D.P. Morgan, E.B. George, L.T. Lee, S.M. Kay: Cochannel speaker separation by harmonic enhancement and suprresion, IEEE Trans. Speech Audio Process. **5**, 405–424 (1997)

43.65 E.B. George, M.J.T. Smith: Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model, IEEE Trans. Speech Audio Process. **5**, 389–406 (1997)

43.66 D. O'Brien, A.I.C. Monaghan: Concatenative synthesis based on a harmonic model, IEEE Trans. Speech Audio Process. **9**, 11–20 (2001)

43.67 J.S. Lim, A.V. Oppenheim, L.D. Braida: Evaluation of an adpative comb filtering method for enhancing speech degraded by white noise addition,

Amazon EX1010 Page 875

IEEE Trans. Acoust. Speech Signal Process. **ASSP-26**, 354–358 (1978)
43.68 J. Makhoul: Linear prediction: A tutorial review, Proc. IEEE **63**, 561–580 (1975)
43.69 J.R. Deller, J.G. Proakis, J.H.L. Hansen: *Discrete-Time Processing of Speech Signals* (Macmillan, New York 1993)
43.70 K.M. Malladi, R.V. Rajakumar: Estimation of time-varying AR models of speech through Gauss-Markov modeling, Proc. IEEE ICASSP **6**, 305–308 (2003)
43.71 M. Niedźwiecki, K. Cisowski: Adaptive scheme for elimination fo broadband noise and impulsive disturbance from AR and ARMA signals, IEEE Trans. Signal Process. **44**, 528–537 (1996)
43.72 B. Koo, J.D. Gibson: Filtering of colored noise for speech enhancement and coding, Proc. IEEE ICASSP **1989**, 345–352 (1989)
43.73 B. Lee, K.Y. Lee, S. Ann: An EM-based approach for parameter enhancement with an application to speech signals, Signal Process. **46**, 1–14 (1995)
43.74 Z. Goh, K.C. Tan, B.T.G. Tan: Kalman-filtering speech enhancement method based on a voiced-unvoiced speech model, IEEE Trans. Speech Audio Process. **7**, 510–524 (1999)
43.75 C. Li, S.V. Andersen: Intergrating Kalman filtering and multi-pulse coding for speech enhancement with a non-stationary model of the speech signal, Proc. IEEE ICASSP **2004**, 2300–2304 (2004)
43.76 L.R. Rabiner: A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE **77**, 257–286 (1989)
43.77 B.H. Juang, L.R. Rabiner: Mixture autoregressive hidden Markov models for speech signals, IEEE Trans. Acoust. Speech Signal Process. **ASSP-33**, 1404–1413 (1985)
43.78 A.P. Dempster, N.M. Laird, D.B. Rubin: Maximum likelihood from incomplete data via the EM algorithm, J. Roy. Stat. Soc. B **39**, 1–38 (1977)
43.79 H. Sameti, H. Sheikhzadeh, L. Deng, R.L. Brennan: HMM-based strategies for enhancement of speech signals embedded in nonstationary noise, IEEE Trans. Speech Audio Process. **6**, 445–455 (1998)

Amazon EX1010 Page 876

44. Spectral Enhancement Methods

I. Cohen, S. Gannot

In this chapter, we focus on the statistical methods that constitute a speech spectral enhancement system and describe some of their fundamental components. We begin in Sect. 44.2 by formulating the problem of spectral enhancement. In Sect. 44.3, we address the time–frequency correlation of spectral coefficients for speech and noise signals, and present statistical models that conform with these characteristics. In Sect. 44.4, we present estimators for speech spectral coefficients under speech presence uncertainty based on various fidelity criteria. In Sect. 44.5, we address the problem of speech presence probability estimation. In Sect. 44.6, we present useful estimators for the a priori signal-to-noise ratio (SNR) under speech presence uncertainty. We present the decision-directed approach, which is heuristically motivated, and the recursive estimation approach, which is based on statistical models and follows the rationale of Kalman filtering. In Sect. 44.7, we describe the improved minima-controlled recursive averaging (IMCRA) approach for noise power spectrum estimation. In Sect. 44.8, we provide a detailed example of a speech enhancement algorithm, and demonstrate its performance in environments with various noise types. In Sect. 44.9, we survey the main types of spectral enhancement components, and discuss the significance of the choice of statistical model, fidelity criterion, a priori SNR estimator, and noise spectrum estimator. Some concluding comments are made in Sect. 44.10.

44.1 **Spectral Enhancement** 874

44.2 **Problem Formulation** 875

44.3 **Statistical Models** 876

44.4 **Signal Estimation** 879
- 44.4.1 MMSE Spectral Estimation 879
- 44.4.2 MMSE Log-Spectral Amplitude Estimation 881

44.5 **Signal Presence Probability Estimation** ... 881

44.6 **A Priori SNR Estimation** 882
- 44.6.1 Decision-Directed Estimation 882
- 44.6.2 Causal Recursive Estimation 883
- 44.6.3 Relation Between Causal Recursive Estimation and Decision-Directed Estimation . 886
- 44.6.4 Noncausal Recursive Estimation 887

44.7 **Noise Spectrum Estimation** 888
- 44.7.1 Time-Varying Recursive Averaging . 888
- 44.7.2 Minima-Controlled Estimation 889

44.8 **Summary of a Spectral Enhancement Algorithm** 891

44.9 **Selection of Spectral Enhancement Algorithms** 896
- 44.9.1 Choice of a Statistical Model and Fidelity Criterion 896
- 44.9.2 Choice of an A Priori SNR Estimator 897
- 44.9.3 Choice of a Noise Estimator 898

44.10 **Conclusions** 898

References 899

The problem of spectral enhancement of noisy speech signals from a single microphone has attracted considerable research effort for over 30 years. It is a problem with numerous applications ranging from speech recognition, to hearing aids and hands-free mobile communication. In this chapter, we present the fundamental components that constitute a speech spectral enhancement system. We describe statistical models that take into consideration the time correlation between successive spectral components of the speech signal, and present estimators for the speech spectral coefficients based on various fidelity criteria. We address the problem of a priori SNR estimation under speech presence uncertainty, and noise power spectrum estimation. We also provide a detailed design example of a speech enhancement algorithm.

Amazon EX1010 Page 877

44.1 Spectral Enhancement

Spectral enhancement of noisy speech has been a challenging problem for many researchers for over 30 years, and is still an active research area (see e.g., [44.1–3] and references therein). This problem is often formulated as the estimation of speech spectral components from a speech signal degraded by statistically independent additive noise. In this chapter we consider spectral enhancement methods for single-channel set-ups, assuming that only one-microphone noisy output is available for the estimation. The situation of one-microphone setups is particularly difficult under nonstationary noise and a low signal-to-noise ratio (SNR), since no reference signal is available for the estimation of the background noise.

A variety of different approaches for spectral enhancement of noisy speech signals have been introduced over the years. One of the earlier methods, and perhaps the most well-known approach, is spectral subtraction [44.4, 5], in which an estimate of the short-term power spectral density of the clean signal is obtained by subtracting an estimate of the power spectral density of the background noise from the short-term power spectral density of the degraded signal. The square root of the resulting estimate is considered an estimate of the spectral magnitude of the speech signal. Subsequently, an estimate of the signal is obtained by combining the spectral magnitude estimate with the complex exponential of the phase of the noisy signal. This method generally results in random narrowband fluctuations in the residual noise, also known as musical tones, which is annoying and disturbing to the perception of the enhanced signal. Many variations have been developed to cope with the musical residual noise phenomena [44.4, 6–9], including spectral subtraction techniques based on masking properties of the human auditory system [44.10, 11].

The spectral subtraction method makes minimal assumptions about the signal and noise, and when carefully implemented, produces enhanced signals that may be acceptable for certain applications. Statistical methods [44.12–16] are designed to minimize the expected value of some distortion measure between the clean and estimated signals. This approach requires the presumption of reliable statistical models for the speech and noise signals, the specification of a perceptually meaningful distortion measure, and a mathematically tractable derivation of an efficient signal estimator. A statistical speech model and perceptually meaningful distortion measure, which are the most appropriate for spectral enhancement, have not yet been determined. Hence, statistical methods for spectral enhancement mainly differ in their statistical model [44.12, 14, 15], distortion measure [44.17–19], and the particular implementation of the spectral enhancement algorithm [44.2].

Spectral enhancement based on hidden Markov processes (HMPs) try to circumvent the assumption of specific distributions for the speech and noise processes [44.20–23]. The probability distributions of the two processes are first estimated from long training sequences of clean speech and noise samples, and then used jointly with a given distortion measure to derive an estimator for the speech signal. Normally, vectors generated from a given sequence of states are assumed to be statistically independent. However, the HMP can be extended to take into account the time–frequency correlation of speech signals by using nondiagonal covariance matrices for each subsource, and by assuming that a sequence of vectors generated from a given sequence of states is a nonzero-order autoregressive process [44.21, 24]. HMP-based speech enhancement relies on the type of training data [44.25]. It works best with the type of noise used during training, and often worse with other types of noise. Furthermore, improved performance generally entails more-complex models and greater computational requirements. While hidden Markov models have been successfully applied to automatic recognition of clean speech signals [44.26, 27], they were not found to be sufficiently refined models for speech enhancement applications [44.3].

Subspace methods [44.28–31] attempt to decompose the vector space of the noisy signal into a signal-plus-noise subspace and a noise subspace. Spectral enhancement is performed by removing the noise subspace and estimating the speech signal from the remaining subspace. The signal subspace decomposition can be achieved by either using the Karhunen–Loève transform (KLT) via eigenvalue decomposition of a Toeplitz covariance estimate of the noisy vector [44.28, 30], or by using the singular value decomposition of a data matrix [44.32, 33]. Linear estimation in the signal-plus-noise subspace is performed with the goal of minimizing signal distortion while masking the residual noise by the signal. A perceptually motivated signal subspace approach takes into account the masking properties of the human auditory system and reduces the perceptual effect of the residual noise [44.34, 35].

Amazon EX1010 Page 878

44.2 Problem Formulation

Let $x(n)$ and $d(n)$ denote speech and uncorrelated additive noise signals, respectively, where n is a discrete-time index. The observed signal $y(n)$, given by $y(n) = x(n) + d(n)$, is transformed into the time-frequency domain by applying the short-time Fourier transform (STFT). Specifically,

$$Y_{tk} = \sum_{n=0}^{N-1} y(n+tM)h(n)\mathrm{e}^{-\mathrm{i}\frac{2\pi}{N}nk}\,, \tag{44.1}$$

where t is the time frame index ($t = 0, 1, \ldots, k$) is the frequency bin index ($k = 0, 1, \ldots, N-1$), $h(n)$ is an analysis window of size N (e.g., Hamming window), and M is the framing step (number of samples separating two successive frames). Given an estimate $\hat{X}_{tk}$ for the STFT of the clean speech (Fig. 44.1), an estimate for the clean speech signal is obtained by applying the inverse STFT,

$$\hat{x}(n) = \sum_{t}\sum_{k=0}^{N-1} \hat{X}_{tk}\tilde{h}(n-tM)\mathrm{e}^{\mathrm{i}\frac{2\pi}{N}k(n-tM)}\,, \tag{44.2}$$

where $\tilde{h}(n)$ is a synthesis window that is biorthogonal to the analysis window $h(n)$ [44.36], and the inverse STFT is efficiently implemented by using the weighted overlap-add method [44.37] (see also Sect. 44.8).

The spectral enhancement problem is generally formulated as deriving an estimator $\hat{X}_{tk}$ for the speech spectral coefficients, such that the expected value of a certain distortion measure is minimized. Let $d(X_{tk}, \hat{X}_{tk})$ denote a distortion measure between X_{tk} and its estimate $\hat{X}_{tk}$, and let ψ_t represent the information set that can be employed for the estimation at frame t (e.g., the noisy data observed through time t). Let H_1^{tk} and H_0^{tk} denote, respectively, hypotheses of signal presence and absence in the noisy spectral coefficient Y_{tk}:

$$\begin{aligned} H_1^{tk}: &\quad Y_{tk} = X_{tk} + D_{tk} \\ H_0^{tk}: &\quad Y_{tk} = D_{tk}\,. \end{aligned}$$

Let $\hat{p}_{tk} = P(H_1^{tk}|\psi_t)$ denote an estimate for the signal presence probability, $\hat{\lambda}_{tk} = E\{|X_{tk}|^2|H_1^{tk}, \psi_t\}$ denote an estimate for the variance of a speech spectral coefficient X_{tk} under H_1^{tk}, and $\widehat{\sigma_{tk}^2} = E\{|Y_{tk}|^2|H_0^{tk}, \psi_t\}$ denote an estimate for the variance of a noise spectral coefficient D_{tk}. Then, we consider an estimator for X_{tk} which minimizes the expected distortion given $\hat{p}_{tk}$, $\hat{\lambda}_{tk}$, $\widehat{\sigma_{tk}^2}$ and the noisy spectral coefficient Y_{tk}:

$$\min_{\hat{X}_{tk}} E\left\{d\left(X_{tk}, \hat{X}_{tk}\right)\middle|\hat{p}_{tk}, \hat{\lambda}_{tk}, \widehat{\sigma_{tk}^2}, Y_{tk}\right\}\,. \tag{44.3}$$

In particular, restricting ourselves to a squared error distortion measure of the form

$$d\left(X_{tk}, \hat{X}_{tk}\right) = \left|g(\hat{X}_{tk}) - \tilde{g}(X_{tk})\right|^2\,, \tag{44.4}$$

where $g(X)$ and $\tilde{g}(X)$ are specific functions of X (e.g., X, $|X|$, $\log|X|$, $\mathrm{e}^{\mathrm{i}\angle X}$), the estimator $\hat{X}_{tk}$ is calculated from

$$\begin{aligned} g(\hat{X}_{tk}) &= E\left\{\tilde{g}(X_{tk})\middle|\hat{p}_{tk}, \hat{\lambda}_{tk}, \widehat{\sigma_{tk}^2}, Y_{tk}\right\} \\ &= \hat{p}_{tk}E\left\{\tilde{g}(X_{tk})\middle|H_1^{tk}, \hat{\lambda}_{tk}, \widehat{\sigma_{tk}^2}, Y_{tk}\right\} \\ &\quad + (1-\hat{p}_{tk})E\left\{\tilde{g}(X_{tk})\middle|H_0^{tk}, Y_{tk}\right\}\,. \end{aligned} \tag{44.5}$$

Hence, the design of a particular estimator for X_{tk} requires the following specifications:

- Functions $g(X)$ and $\tilde{g}(X)$, which determine the fidelity criterion of the estimator
- A conditional probability density function (pdf) $p(X_{tk}|\lambda_{tk}, H_1^{tk})$ for X_{tk} under H_1^{tk} given its variance λ_{tk}, which determines the statistical model
- An estimator $\hat{\lambda}_{tk}$ for the speech spectral variance
- An estimator σ_{tk}^2 for the noise spectral variance
- An estimator $\hat{p}_{tk|t-1} = P(H_1^{tk}|\psi_{t-1})$ for the a priori signal presence probability, where ψ_{t-1} represents the information set known prior to having the measurement Y_{tk}

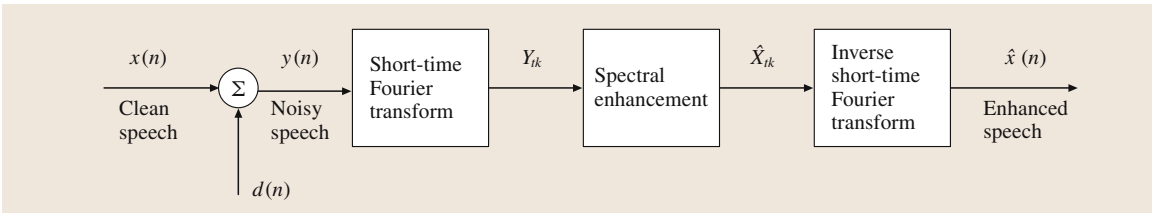


Fig. 44.1 Spectral enhancement approach

Given the a priori signal presence probability $\hat{p}_{tk|t-1}$, the (a posteriori) signal presence probability can be obtained from Bayes' rule:

$$\hat{p}_{tk} = P\left(H_1^{tk}\middle|Y_{tk}, \psi_{t-1}\right) = \left[1 + \frac{\left(1-\hat{p}_{tk|t-1}\right)p\left(Y_{tk}|H_0^{tk}, \psi_{t-1}\right)}{\hat{p}_{tk|t-1}\,p\left(Y_{tk}|H_1^{tk}, \psi_{t-1}\right)}\right]^{-1} . \quad (44.6)$$

In the following sections we present statistical models for speech signals in the STFT domain, and address the estimation problem of the speech spectral coefficient X_{tk} given $\hat{\lambda}_{tk}$, $\widehat{\sigma_{tk}^2}$, and $\hat{p}_{tk}$. Then we consider the estimation of the speech spectral variance λ_{tk}, the noise spectral variance σ_{tk}^2, and the speech presence probability $P(H_1^{tk})$, and describe an example of a speech enhancement algorithm.

44.3 Statistical Models

In this section, we present statistical models that take into account the time correlation between successive spectral components of the speech signal. To see graphically the relation between successive spectral components of a speech signal, in comparison with a noise signal, we present scatter plots for successive spectral magnitudes, and investigate the sample autocorrelation coefficient sequences (ACSs) of the STFT coefficients along time trajectories (the frequency bin index k is held fixed). We consider a speech signal that is constructed from six different utterances, without intervening pauses. The utterances, half from male speakers and half from female speakers, are taken from the TIMIT database [44.38]. (A corpus of phonemically and lexically transcribed speech of American English speakers of different sexes and dialects. The speech was recorded at Texas Instruments (TI) and transcribed at Massachusetts Institute of Technology (MIT), hence the corpus' name.) The speech signal is sampled at 16 kHz, and transformed into the STFT domain using Hamming analysis windows of 512 samples (32 ms) length, and 256 samples framing step (50% overlap between successive frames).

Figure 44.2 shows an example of scatter plots for successive spectral magnitudes of white Gaussian noise (WGN) and speech signals. It implies that 50% overlap between successive frames does not yield a significant correlation between the spectral magnitudes of the WGN signal. However, successive spectral magnitudes of the speech signal are highly correlated. Figure 44.3 shows the ACSs of the speech spectral components along time trajectories, for various frequency bins and framing steps. The 95% confidence limits [44.39] are depicted as horizontal dotted lines. In order to prevent an upward bias of the autocovariance estimates due to irrelevant (nonspeech) spectral components, the ACSs are computed from spectral components whose magnitudes are within 30 dB of the maximal magnitude. Specifically, the sample autocorrelation coefficients of the spectral magnitudes are calculated by

$$\rho_m = \frac{\sum_{t\in\mathcal{T}}\left(A_{tk}-\overline{A_k}\right)\left(A_{t+m,k}-\overline{A_k}\right)}{\sum_{t\in\mathcal{T}}\left(A_{tk}-\overline{A_k}\right)^2} , \quad (44.7)$$

where $A_{tk} \triangleq |X_{tk}|$ denotes the magnitude of X_{tk},

$$\overline{A_k} = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} A_{tk}$$

denotes the sample mean, m is the lag in frames, and $\mathcal{T}$ represents the set of relevant spectral components

$$\mathcal{T} = \left\{t \,\middle|\, A_{tk} \geq 10^{-30/20}\max_t\{A_{tk}\}\right\} .$$

The corresponding sample autocorrelation coefficients of the spectral phases are obtained by

$$\varrho_m = \frac{\sum_{t\in\mathcal{T}}\varphi_{tk}\varphi_{t+m,k}}{\sum_{t\in\mathcal{T}}\varphi_{tk}^2} , \quad (44.8)$$

where φ_{tk} denotes the phase of X_{tk}. Figure 44.4 shows the variation of the correlation between successive spectral magnitudes on frequency and on overlap between successive frames. Figures 44.3 and 44.4 demonstrate that, for speech signals, successive spectral magnitudes are highly correlated, while the correlation is generally larger at lower frequencies, and increases as the overlap between successive frames increases. As a comparison, the variation of ρ_1 on the overlap between frames is also shown for a realization of WGN (Fig. 44.4b, dotted line). It implies that, for a sufficiently large framing step ($M \geq N/2$, i. e., overlap

between frames $\leq 50\%$), successive spectral components of the *noise* signal, but clearly not of the *speech* signal, can be assumed uncorrelated. For smaller framing steps, the correlation between successive spectral noise components also has to be taken into consideration. Furthermore, since the length of the analysis window cannot be too large (its typical length is 20–40 ms [44.12]), adjacent Fourier expansion coefficients of the noise signal, D_{tk} and $D_{t,k+1}$, as well as adjacent coefficients of the speech signal, X_{tk} and $X_{t,k+1}$, are also correlated to a certain degree. Nevertheless, it is commonly assumed that expansion coefficients in different frequency bins are statistically independent [44.12, 15, 16, 40]. This allows one to formulate independent estimation problems for each frequency bin k, which greatly simplifies the resulting algorithms. In view of this discussion, we employ statistical models in the STFT domain that rely on the following set of assumptions [44.41].

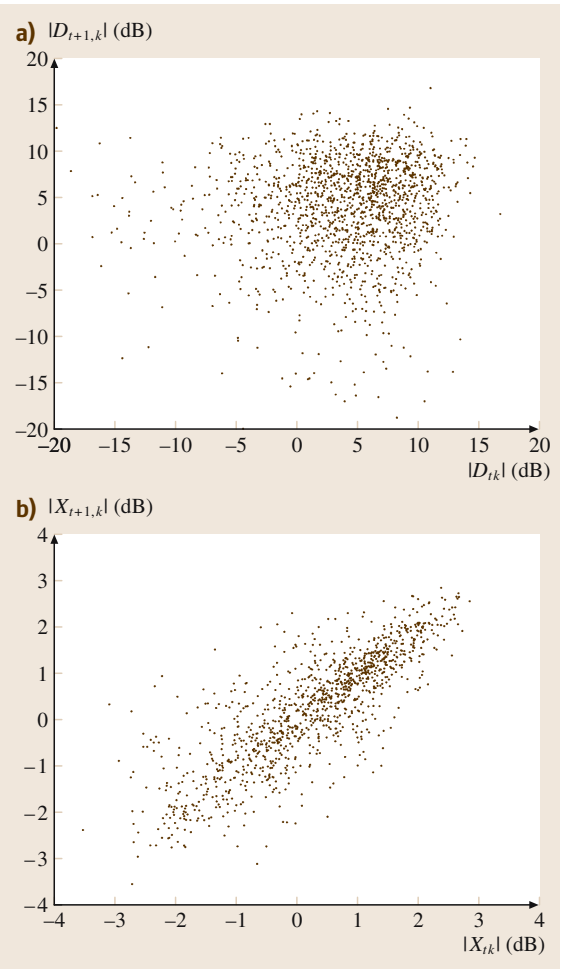


Fig. 44.2a,b Scatter plots for successive spectral magnitudes of **(a)** a white Gaussian noise signal, and **(b)** a speech signal at a center frequency of 500 Hz ($k = 17$). The overlap between successive frames is 50% (after [44.15])

1. The noise spectral coefficients $\{D_{tk}\}$ are zero-mean statistically independent Gaussian random variables. The real and imaginary parts of D_{tk} are independent and identically distributed (iid) random variables $\mathcal{N}(0, \sigma_{tk}^2/2)$.
2. Given $\{\lambda_{tk}\}$ and the state of speech presence in each time–frequency bin (H_1^{tk} or H_0^{tk}), the speech spectral coefficients $\{X_{tk}\}$ are generated by

$$X_{tk} = \sqrt{\lambda_{tk}}\, V_{tk} \,, \tag{44.9}$$

where $\{V_{tk}|H_0^{tk}\}$ are identically zero, and $\{V_{tk}|H_1^{tk}\}$ are statistically independent complex random variables with zero mean, unit variance, and iid real and imaginary parts:

$$\begin{aligned} H_1^{tk}: &\quad E\{V_{tk}\} = 0,\, E\left\{|V_{tk}|^2\right\} = 1 \,, \\ H_0^{tk}: &\quad V_{tk} = 0 \,. \end{aligned} \tag{44.10}$$

3. The pdf of V_{tk} under H_1^{tk} is determined by the specific statistical model. Let $V_{\mathrm{R}tk} = \mathrm{Re}\{V_{tk}\}$ and $V_{\mathrm{I}tk} = \mathrm{Im}\{V_{tk}\}$ denote, respectively, the real and imaginary parts of V_{tk}. Let $p(V_{\rho tk}|H_1^{tk})$ denote the pdf of $V_{\rho tk}$ ($\rho \in \{\mathrm{R}, \mathrm{I}\}$) under H_1^{tk}. Then, for a Gaussian model

$$p\left(V_{\rho tk}|H_1^{tk}\right) = \frac{1}{\sqrt{\pi}} \exp\left(-V_{\rho tk}^2\right) , \tag{44.11}$$

for a gamma model

$$\begin{aligned} &p\left(V_{\rho tk}|H_1^{tk}\right) \\ &= \frac{\sqrt[4]{3}}{2\sqrt{\pi}\sqrt[4]{2}} |V_{\rho tk}|^{-1/2} \exp\left(-\frac{\sqrt{3}|V_{\rho tk}|}{\sqrt{2}}\right) , \end{aligned} \tag{44.12}$$

and for a Laplacian model

$$p\left(V_{\rho tk}|H_1^{tk}\right) = \exp(-2|V_{\rho tk}|) \,. \tag{44.13}$$

4. The sequence of speech spectral variances $\{\lambda_{tk}|t = 0, 1, \ldots\}$ is a random process, which is generally correlated with the sequence of speech spectral magnitudes $\{A_{tk}|t = 0, 1, \ldots\}$. However, given λ_{tk}, A_{tk} is statistically independent of $A_{t'k'}$ for all $t \neq t'$ and $k \neq k'$.

Clearly, the first assumption does not hold when the overlap between successive frames is too large (Fig. 44.4b,

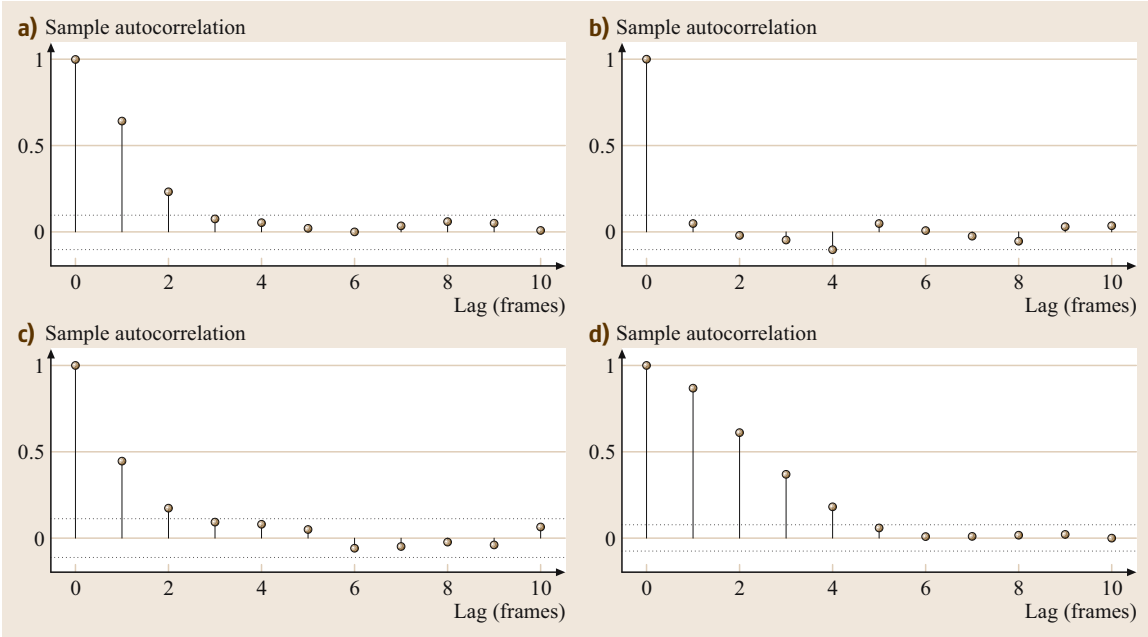
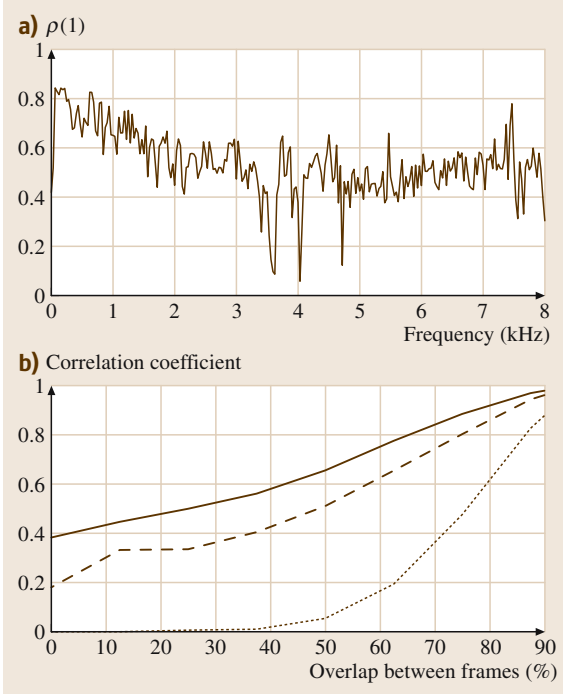


Fig. 44.3a–d Sample autocorrelation coefficient sequences (ACSs) of clean-speech STFT coefficients along time trajectories, for various frequency bins and framing steps. The *dotted lines* represent 95% confidence limits. **(a)** ACS of the spectral magnitude at frequency bin $k = 17$ (center frequency 500 Hz), framing step $M = N/2$ (50% overlap between frames). **(b)** ACS of the spectral phase, $k = 17$, $M = N/2$. **(c)** ACS of the spectral magnitude, $k = 65$ (center frequency 2 kHz), $M = N/2$. **(d)** ACS of the spectral magnitude, $k = 17$, $M = N/4$ (75% overlap between frames) (after [44.15])



implemented in accordance with this assumption (e.g., the overlap is not greater than 50%). The second assumption implies that the speech spectral coefficients $\{X_{tk}|H_1^{tk}\}$ are conditionally zero-mean statistically independent random variables given their variances $\{\lambda_{tk}\}$. The real and imaginary parts of X_t under H_1^t are conditionally iid random variables given λ_{tk}, satisfying

$$p\left(X_{\rho tk}|\lambda_{tk}, H_1^{tk}\right) = \frac{1}{\sqrt{\lambda_{tk}}} p\left(V_{\rho tk} = \frac{X_{\rho tk}}{\sqrt{\lambda_{tk}}} \middle| H_1^{tk}\right), \quad (44.14)$$

where $\rho \in \{\mathrm{R}, \mathrm{I}\}$. The last assumption allows one to take into account the time correlation between

Fig. 44.4a,b Variation of the correlation coefficient between successive spectral magnitudes. **(a)** Typical variation of ρ_1 with frequency for a speech signal and 50% overlap between frames. **(b)** Typical variation of ρ_1 with overlap between frames for a speech signal at center frequencies of 1 kHz (*solid line*) and 2 kHz (*dashed line*), and for a realization of white Gaussian noise (*dotted line*) (after [44.15])

successive spectral coefficients of the speech signal, while still considering the scalar estimation problem formulated in (44.3). Note that successive spectral coefficients are correlated, since the random processes $\{X_{tk}|t=0,1,\dots\}$ and $\{\lambda_{tk}|t=0,1,\dots\}$ are not independent.

44.4 Signal Estimation

In this section, we derive estimators for X_{tk} using various fidelity criteria, assuming that $\hat{p}_{tk}$, $\hat{\lambda}_{tk}$, and $\widehat{\sigma_{tk}^2}$ are given. Fidelity criteria that are of particular interest for speech enhancement applications are minimum mean-squared error (MMSE) [44.5], MMSE of the spectral amplitude (MMSE-SA) [44.12], and MMSE of the log-spectral amplitude (MMSE-LSA) [44.17,42]. The MMSE estimator is derived by substituting into (44.5) the functions

$$g(\hat{X}_{tk}) = \hat{X}_{tk}$$
$$\tilde{g}(X_{tk}) = \begin{cases} X_{tk}\,, & \text{under } H_1^{tk} \\ G_{\min} Y_{tk}\,, & \text{under } H_0^{tk}\,, \end{cases} \tag{44.15}$$

where $G_{\min} \ll 1$ represents a constant attenuation factor, which retains the noise naturalness during speech absence [44.2,42].

The MMSE-SA estimator is obtained by using the functions

$$g(\hat{X}_{tk}) = |\hat{X}_{tk}|\,,$$
$$\tilde{g}(X_{tk}) = \begin{cases} |X_{tk}|\,, & \text{under } H_1^{tk} \\ G_{\min}|Y_{tk}|\,, & \text{under } H_0^{tk}\,. \end{cases} \tag{44.16}$$

The MMSE-LSA estimator is obtained by using the functions

$$g(\hat{X}_{tk}) = \log|\hat{X}_{tk}|\,,$$
$$\tilde{g}(X_{tk}) = \begin{cases} \log|X_{tk}|\,, & \text{under } H_1^{tk} \\ \log(G_{\min}|Y_{tk}|)\,, & \text{under } H_0^{tk}\,. \end{cases} \tag{44.17}$$

The last two estimators are insensitive to the estimation error of φ_{tk}, the phase of X_{tk}. Therefore, they are combined with the following constrained optimization problem [44.12]:

$$\min_{\hat{\varphi}_{tk}} E\{|\mathrm{e}^{\mathrm{i}\varphi_{tk}} - \mathrm{e}^{\mathrm{i}\hat{\varphi}_{tk}}|^2\} \quad \text{subject to } |\mathrm{e}^{\mathrm{i}\hat{\varphi}_{tk}}| = 1\,. \tag{44.18}$$

This yields an estimator for the complex exponential of the phase, constrained not to affect the spectral magnitude estimate. Alternatively, an estimate for the spectral phase $\hat{\varphi}_{tk}$ can be obtained by minimizing the expected value of the following distortion measure [44.12]

$$d_{\varphi}(\varphi_{tk}, \hat{\varphi}_{tk}) \triangleq 1 - \cos(\varphi_{tk} - \hat{\varphi}_{tk})\,. \tag{44.19}$$

This measure is invariant under modulo 2π transformation of the estimation error $\varphi_{tk} - \hat{\varphi}_{tk}$, and for small estimation errors it closely resembles the squared-error distortion measure, since $1 - \cos\beta \approx \beta^2/2$ for $\beta \ll 1$. The constrained optimization problem (44.18) and the distortion measure (44.19) both yield an estimator $\mathrm{e}^{\mathrm{i}\hat{\varphi}_{tk}} = Y_{tk}/|Y_{tk}|$, which is simply the complex exponential of the noisy signal [44.12].

44.4.1 MMSE Spectral Estimation

Let

$$\xi_{tk} \triangleq \frac{\lambda_{tk}}{\sigma_{tk}^2}\,, \quad \gamma_{\rho tk} \triangleq \frac{Y_{\rho tk}^2}{\sigma_{tk}^2}\,, \tag{44.20}$$

represent the a priori and a posteriori SNRs, respectively ($\rho \in \{\mathrm{R}, \mathrm{I}\}$), and let $G_{\mathrm{MSE}}(\xi, \gamma\rho)$ denote a gain function that satisfies

$$E\left\{X_{\rho tk}\middle|H_1^{tk}, \lambda_{tk}, \sigma_{tk}^2, Y_{\rho tk}\right\} = G_{\mathrm{MSE}}(\xi_{tk}, \gamma_{\rho tk}) Y_{\rho tk}\,. \tag{44.21}$$

Then, substituting (44.15) and (44.21) into (44.5), we have

$$\hat{X}_{tk} = \hat{p}_{tk}\big[G_{\mathrm{MSE}}(\hat{\xi}_{tk}, \hat{\gamma}_{\mathrm{R}tk}) Y_{\mathrm{R}tk} + \mathrm{i} G_{\mathrm{MSE}}(\hat{\xi}_{tk}, \hat{\gamma}_{\mathrm{I}tk}) Y_{\mathrm{I}tk}\big] + (1 - \hat{p}_{tk}) G_{\min} Y_{tk}\,. \tag{44.22}$$

The specific expression for $G_{\mathrm{MSE}}(\xi, \gamma_\rho)$ depends on the particular statistical model:

$$G_{\mathrm{MSE}}(\xi, \gamma_\rho) = \frac{1}{Y_\rho} \int X_\rho\, p(X_\rho|H_1, \lambda, \sigma^2, Y_\rho)\, \mathrm{d}X_\rho = \frac{1}{Y_\rho} \int X_\rho \frac{p(Y_\rho|X_\rho, \sigma^2)\, p(X_\rho|\lambda, H_1)}{p(Y_\rho|\lambda, \sigma^2)}\, \mathrm{d}X_\rho\,.$$

For a Gaussian model, the gain function is independent of the a posteriori SNR. It is often referred to as Wiener

filter, given by [44.5]

$$G_{\mathrm{MSE}}(\xi) = \frac{\xi}{1+\xi} \,. \tag{44.23}$$

For a gamma model, the gain function is given by [44.40]

$$\begin{aligned} G_{\mathrm{MSE}}(\xi, \gamma_\rho) = \frac{1}{\sqrt{8\gamma_\rho}} & \left[\exp\left(\frac{C_{\rho-}^2}{4}\right) D_{-1.5}(C_{\rho-}) \right. \\ & \left. - \exp\left(\frac{C_{\rho+}^2}{4}\right) D_{-1.5}(C_{\rho+}) \right] \\ & \times \left[\exp\left(\frac{C_{\rho-}^2}{4}\right) D_{-0.5}(C_{\rho-}) \right. \\ & \left. + \exp\left(\frac{C_{\rho+}^2}{4}\right) D_{-0.5}(C_{\rho+}) \right]^{-1} , \end{aligned} \tag{44.24}$$

where $C_{\rho+}$ and $C_{\rho-}$ are defined by

$$C_{\rho\pm} \triangleq \frac{\sqrt{3}}{2\sqrt{\xi}} \pm \sqrt{2\gamma_\rho} \,, \tag{44.25}$$

and $D_p(z)$ denotes the parabolic cylinder function [44.44, (9.240)]. For a Laplacian speech model, the gain function is given by [44.45]

$$\begin{aligned} & G_{\mathrm{MSE}}(\xi, \gamma_\rho) \\ & = \frac{1}{\sqrt{\gamma_\rho}} [L_{\rho+}\mathrm{erfcx}(L_{\rho+}) - L_{\rho-}\mathrm{erfcx}(L_{\rho-})] \\ & \quad \times [\mathrm{erfcx}(L_{\rho+}) + \mathrm{erfcx}(L_{\rho-})]^{-1} \,, \end{aligned} \tag{44.26}$$

where $L_{\rho+}$ and $L_{\rho-}$ are defined by

$$L_{\rho\pm} \triangleq \frac{1}{\sqrt{\xi}} \pm \sqrt{\gamma_\rho} \,, \tag{44.27}$$

and $\mathrm{erfcx}(x)$ is the scaled complementary error function, defined by

$$\mathrm{erfcx}(x) \triangleq \mathrm{e}^{x^2} \frac{2}{\sqrt{\pi}} \int_x^\infty \mathrm{e}^{-t^2} \, \mathrm{d}t \,. \tag{44.28}$$

Note that when the signal is surely absent (i. e., when $\hat{p}_{tk} = 0$), the resulting estimator $\hat{X}_{tk}$ reduces to a constant attenuation of Y_{tk} (i. e., $\hat{X}_{tk} = G_{\min} Y_{tk}$). This retains the noise naturalness, and is closely related to the *spectral floor* proposed by *Berouti* et al. [44.6].

Figure 44.5 displays parametric gain curves describing $G_{\mathrm{MSE}}(\xi, \gamma_\rho)$ for several values of γ_ρ, which result from (44.23), (44.24), and (44.26). It shows that the spectral gains are monotonically increasing functions of the a priori SNR when the a posteriori SNR is kept constant. For gamma and Laplacian models, the spectral gains are also monotonically increasing functions of the a posteriori SNR, when the a priori SNR is kept constant.

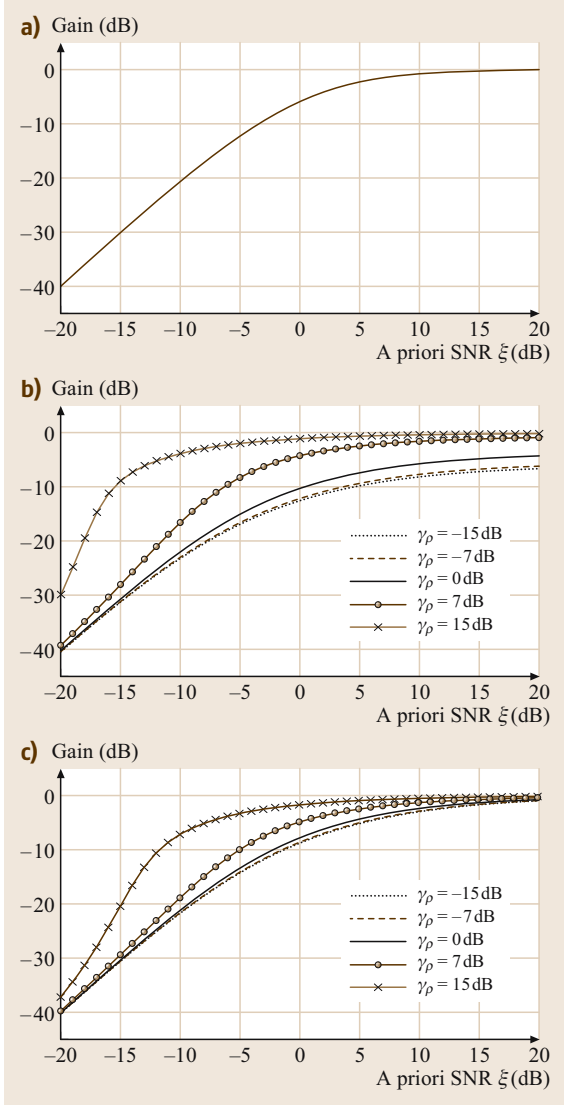


Fig. 44.5a–c Parametric gain curves describing the MMSE gain function $G_{\mathrm{MSE}}(\xi, \gamma_\rho)$ for different speech models. **(a)** Gain curve for a Gaussian model, obtained by (44.23). **(b)** Gain curves for a gamma model, obtained by (44.24). **(c)** Gain curves for a Laplacian model, obtained by (44.26) (after [44.43])

44.4.2 MMSE Log-Spectral Amplitude Estimation

In speech enhancement applications, estimators that minimize the mean-squared error of the log-spectral amplitude have been found advantageous to MMSE spectral estimators [44.12, 17, 46]. An MMSE-LSA estimator is obtained by substituting (44.17) into (44.5). It is difficult, or even impossible, to find analytical expressions for an MMSE-LSA estimator under a gamma or Laplacian model. However, assuming a Gaussian model and combing the resulting amplitude estimate with the phase of the noisy spectral coefficient Y_{tk} yields [44.42]

$$\hat{X}_{tk} = \left[G_{\mathrm{LSA}}(\hat{\xi}_{tk}, \hat{\gamma}_{tk})\right]^{\hat{p}_{tk}} G_{\min}^{1-\hat{p}_{tk}} Y_{tk} \,, \tag{44.29}$$

where $\hat{\gamma}_{tk}$ denotes an estimate for the a posteriori SNR

$$\hat{\gamma}_{tk} = \hat{\gamma}_{\mathrm{R}tk} + \hat{\gamma}_{\mathrm{I}tk} \,, \tag{44.30}$$

and $G_{\mathrm{LSA}}(\xi, \gamma)$ represents the LSA gain function under H_1^{tk} which was derived by *Ephraim* and *Malah* [44.17]

$$G_{\mathrm{LSA}}(\xi, \gamma) \triangleq \frac{\xi}{1+\xi} \exp\left(\frac{1}{2}\int_{\vartheta}^{\infty} \frac{\mathrm{e}^{-x}}{x}\,\mathrm{d}x\right) \,, \tag{44.31}$$

where ϑ is defined by $\vartheta \triangleq \xi\gamma/(1+\xi)$. Similar to the MMSE spectral estimator, the MMSE-LSA estimator reduces to a constant attenuation of Y_{tk} when the signal is surely absent (i. e., $\hat{p}_{tk} = 0$ implies $\hat{X}_{tk} = G_{\min} Y_{tk}$). However, the characteristics of these estimators when the signal is present are readily distinctive. Figure 44.6 displays parametric gain curves describing $G_{\mathrm{LSA}}(\xi, \gamma)$ for several values of γ. For a fixed value of the a posteriori SNR, the LSA gain is a monotonically increasing function of ξ. However, for a fixed value of ξ, the LSA gain is a monotonically *decreasing* function of γ. Note that the gain function $G_{\mathrm{MSE}}(\xi, \gamma_\rho)$ for a Gaussian model is independent of the a posteriori SNR, while for gamma and Laplacian speech models $G_{\mathrm{MSE}}(\xi, \gamma_\rho)$ is an *increasing* function of the a posteriori SNR (Fig. 44.5). The behavior of $G_{\mathrm{LSA}}(\xi, \gamma)$ is related to the useful mechanism that counters the musical noise phenomenon [44.47]. Local bursts of the a posteriori SNR, during noise-only frames, are *pulled down* to the average noise level, thus avoiding local buildup of noise whenever it exceeds its average characteristics. As a result, the MMSE-LSA estimator generally produces lower levels of residual musical noise, when compared with the MMSE spectral estimators.

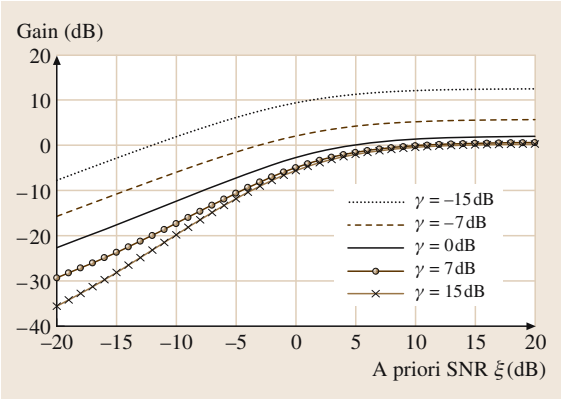


Fig. 44.6 Parametric gain curves describing the MMSE log-spectral amplitude gain function $G_{\mathrm{LSA}}(\xi, \gamma)$ for a Gaussian model, obtained by (44.31)

44.5 Signal Presence Probability Estimation

In this section, we derive an efficient estimator $\hat{p}_{tk|t-1}$ for the a priori speech presence probability. This estimator employs a soft-decision approach to compute three parameters based on the time–frequency distribution of the estimated a priori SNR $\hat{\xi}_{tk}$. The parameters exploit the strong correlation of speech presence in neighboring frequency bins of consecutive frames.

Let ζ_{tk} denote a recursive average of the a priori SNR with a time constant α_ζ,

$$\zeta_{tk} = \alpha_\zeta \zeta_{t-1,k} + (1-\alpha_\zeta)\hat{\xi}_{t-1,k} \,. \tag{44.32}$$

By applying *local* and *global* averaging windows in the frequency domain, we obtain respectively local and global averages of the a priori SNR

$$\zeta_{tk}^{\chi} = \sum_{i=-w_\chi}^{w_\chi} h_\chi(i)\zeta_{t,k-i} \,, \tag{44.33}$$

where the superscript χ designates either *local* or *global*, and h_χ is a normalized window of size $2w_\chi + 1$. We define two parameters, P_{tk}^{local} and P_{tk}^{global}, which represent the relation between the above averages and the like-

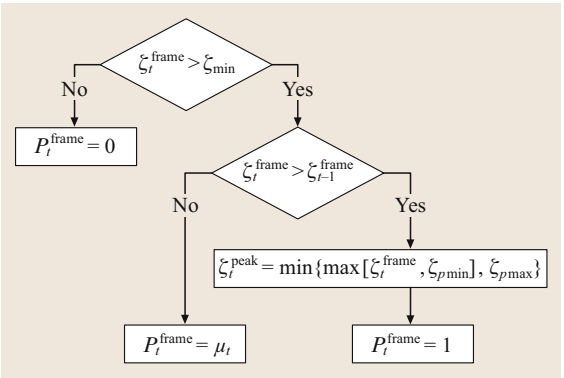


Fig. 44.7 Block diagram for computing P^{frame} (a parameter representing the likelihood of speech in a given frame) (after [44.42])

lihood of speech in the k-th frequency bin of the t-th frame. These parameters are given by

$$P_{tk}^{\chi} = \begin{cases} 0\,, & \text{if } \zeta_{tk}^{\chi} \le \zeta_{\min} \\ 1\,, & \text{if } \zeta_{tk}^{\chi} \ge \zeta_{\max} \\ \frac{\log(\zeta_{tk}^{\chi}/\zeta_{\min})}{\log(\zeta_{\max}/\zeta_{\min})}\,, & \text{otherwise}\,, \end{cases} \tag{44.34}$$

where $\zeta_{\min}$ and $\zeta_{\max}$ are empirical constants, maximized to attenuate noise while maintaining weak speech components.

In order to attenuate noise further in noise-only frames, we define a third parameter, P_t^{frame}, which is based on the speech energy in neighboring frames. An averaging of ζ_{tk} in the frequency domain (possibly over a certain frequency band) yields

$$\zeta_t^{\text{frame}} = \underset{1\le k\le N/2}{\text{mean}}\{\zeta_{tk}\}\,. \tag{44.35}$$

To prevent clipping of speech onsets or weak components, speech is assumed whenever ζ_t^{frame} increases over time. Moreover, the transition from H_1 to H_0 is delayed,

Table 44.1 Values of the parameters used in the implementation of the speech presence probability estimator, for a sampling rate of 16 kHz

$\alpha_\zeta = 0.7$	$\zeta_{\min} = -10\,\text{dB}$	$\zeta_{p\,\min} = 0\,\text{dB}$
$w_{\text{local}} = 1$	$\zeta_{\max} = -5\,\text{dB}$	$\zeta_{p\,\max} = 10\,\text{dB}$
$w_{\text{global}} = 15$	$p_{\min} = 0.005$	
h_{local}, h_{global}: Hann windows		

which reduces the misdetection of weak speech tails, by allowing for a certain decrease in the value of ζ_t^{frame}. Figure 44.7 describes a block diagram for computing P_t^{frame}, where

$$\mu_t \triangleq \begin{cases} 0\,, & \text{if } \zeta_t^{\text{frame}} \le \zeta_t^{\text{peak}}\zeta_{\min} \\ 1\,, & \text{if } \zeta_t^{\text{frame}} \ge \zeta_t^{\text{peak}}\zeta_{\max} \\ \frac{\log(\zeta_t^{\text{frame}}/\zeta_t^{\text{peak}}/\zeta_{\min})}{\log(\zeta_{\max}/\zeta_{\min})}\,, & \text{otherwise}\,, \end{cases} \tag{44.36}$$

represents a soft transition from *speech* to *noise*, ζ_t^{peak} is a confined peak value of ζ_t^{frame}, and $\zeta_{p\,\min}$ and $\zeta_{p\,\max}$ are empirical constants that determine the delay of the transition. Typical values of parameters used for a sampling rate of 16 kHz are summarized in Table 44.1.

The proposed estimate for the a priori speech presence probability is obtained by

$$\hat{p}_{tk|t-1} = P_{tk}^{\text{local}}\, P_{tk}^{\text{global}}\, P_t^{\text{frame}}\,. \tag{44.37}$$

Accordingly, $\hat{p}_{tk|t-1}$ is smaller if either previous frames, or recent neighboring frequency bins, do not contain speech. When $\hat{p}_{tk|t-1} \to 0$, the conditional speech presence probability $\hat{p}_{tk} \to 0$ by (44.6), and consequently the signal estimator $\hat{X}_{tk}$ reduces to $\hat{X}_{tk} = G_{\min} Y_{tk}$. Therefore, to reduce the possibility of speech distortion we generally restrict $\hat{p}_{tk|t-1}$ to be larger than a threshold $p_{\min}$ ($p_{\min} > 0$).

44.6 A Priori SNR Estimation

In this section, we address the problem of estimating the speech spectral variance λ_{tk} assuming that $\hat{p}_{tk|t-1}$ and $\widehat{\sigma_{tk}^2}$ are given. We present the decision-directed, causal and noncausal estimators for the a priori SNR $\xi_{tk} = \lambda_{tk}/\sigma_{tk}^2$ under speech presence uncertainty. The a priori SNR ξ_{tk} is estimated for each spectral component and each analysis frame due to the nonstationarity of the speech signal.

44.6.1 Decision-Directed Estimation

Ephraim and *Malah* [44.12] proposed a decision-directed approach, which provides a very useful

estimation method for the a priori SNR [44.47,48]. Accordingly, if speech presence is assumed ($p_{tk} \equiv 1$), then the expression

$$\Xi_{tk} = \alpha \frac{|\hat{X}_{t-1,k}|^2}{\hat{\sigma}^2_{t-1,k}} + (1-\alpha) \max\left\{\hat{\gamma}_{tk} - 1, 0\right\} \tag{44.38}$$

can be substituted for the a priori SNR. The first term, $|\hat{X}_{t-1,k}|^2/\hat{\sigma}^2_{t-1,k}$, represents the a priori SNR resulting from the processing of the previous frame. The second term, $\max\{\hat{\gamma}_{tk} - 1, 0\}$, is a maximum-likelihood estimate for the a priori SNR, based entirely on the current frame. The parameter α ($0 < \alpha < 1$) is a weighting factor that controls the trade-off between noise reduction and transient distortion brought into the signal [44.12, 47]. A larger value of α results in a greater reduction of the musical noise phenomena, but at the expense of attenuated speech onsets and audible modifications of transient components. As a compromise, a value 0.98 of α was determined by simulations and informal listening tests [44.12].

Under speech presence uncertainty, according to [44.12, 49], the expression in (44.38) estimates a *nonconditional a priori* SNR $\eta_{tk} \triangleq E\{|X_{tk}|^2\}/\sigma^2_{tk}$. The a priori SNR $\xi_{tk} = E\{|X_{tk}|^2|H_1^{tk}\}/\sigma^2_{tk}$ is related to η_{tk} by

$$\eta_{tk} = \frac{E\{|X_{tk}|^2|H_1^{tk}\}P(H_1^{tk})}{\sigma^2_{tk}} = \xi_{tk}\, p_{tk|t-1}\,. \tag{44.39}$$

Therefore the estimate for ξ_{tk} should supposedly be given by

$$\hat{\xi}_{tk} = \frac{\Xi_{tk}}{\hat{p}_{tk|t-1}}\,. \tag{44.40}$$

However, the division by $\hat{p}_{tk|t-1}$ may deteriorate the performance of the speech enhancement system [44.50,51]. In some cases, it introduces interaction between the estimated $p_{tk|t-1}$ and the a priori SNR, that adversely affects the total gain for noise-only bins, resulting in an unnaturally structured residual noise [44.52]. To some extent, the noise structuring can be eliminated by utilizing a voice activity detector (VAD) and applying a uniform attenuation factor to frames that do not contain speech [44.49]. Yet, VADs are difficult to tune and their reliability is often insufficient for weak speech components and low input SNR.

Let $\hat{X}_{tk|H_1} = \hat{X}_{tk}|_{\hat{p}_{tk}=1}$ denote an estimate for $\hat{X}_{tk}$ under the hypothesis of speech presence. Then an alternative a priori SNR estimator under speech presence uncertainty is given by [44.42]

$$\hat{\xi}_{tk} = \alpha \frac{|\hat{X}_{t-1,k|H_1}|^2}{\hat{\sigma}^2_{t-1,k}} + (1-\alpha) \max\left\{\hat{\gamma}_{tk} - 1, 0\right\}. \tag{44.41}$$

Notice that for $\hat{p}_{t-1,k|t-2} \neq 1$, this yields a different estimate than either Ξ_{tk} or $\Xi_{tk}/\hat{p}_{tk|t-1}$. In [44.50, 51], it was suggested to simply estimate the a priori SNR by Ξ_{tk}, rather than $\Xi_{tk}/\hat{p}_{tk|t-1}$. However, the use of $\hat{X}_{t-1,k|H_1}$ in (44.41) boosts the gain up when speech is present, which provides a compensation for not dividing by $\hat{p}_{tk|t-1}$.

To show that under speech presence uncertainty it is advantageous to estimate the a priori SNR by the expression in (44.41) rather than by $\Xi_{tk}/\hat{p}_{tk|t-1}$, we assume that an estimate $\hat{p}_{tk|t-1}$ for the a priori speech presence probability is given, and that Ξ_{tk} and $\hat{\xi}_{tk}$ have been calculated by (44.38) and (44.41), respectively. By definition, if H_1^{tk} is true, then the spectral estimate $\hat{X}_{tk}$ should degenerate to $\hat{X}_{tk|H_1}$, and the a priori SNR estimate should coincide with Ξ_{tk}. On the contrary, if H_0^{tk} is true, then $\hat{X}_{tk}$ should reduce to $G_{\min} Y_{tk}$, or equivalently the a priori SNR estimate should be as small as possible. Indeed, if H_1^{tk} is true then

$$\hat{\xi}_{tk}|_{H_1} \approx \Xi_{tk}|_{H_1} \leq \left.\frac{\Xi_{tk}}{\hat{p}_{tk|t-1}}\right|_{H_1}, \tag{44.42}$$

where we have used that under H_1^{tk} the spectral estimate $\hat{X}_{t-1,k}$ is approximately the same as $\hat{X}_{t-1,k|H_1}$ (if H_1^{tk} is true then $H_1^{t-1,k}$ is likely to be true as well, due to the strong correlation of speech presence in successive frames). On the other hand, if H_0^{tk} is true, then $\hat{p}_{tk|t-1}$ is expected to approach zero, and $\hat{\xi}_{tk}$ is likely to be much smaller than $\Xi_{tk}/\hat{p}_{tk|t-1}$:

$$\hat{\xi}_{tk}|_{H_0} \approx \alpha G^2_{\min} \ll \left.\frac{\Xi_{tk}}{\hat{p}_{tk|t-1}}\right|_{H_0} \approx \frac{\alpha G^2_{\min}}{\hat{p}_{tk|t-1}}\,. \tag{44.43}$$

Therefore, under speech presence uncertainty the decision-directed a priori SNR estimator is more favorably modified as in (44.41), rather than dividing Ξ_{tk} by the a priori speech presence probability $\hat{p}_{tk|t-1}$.

44.6.2 Causal Recursive Estimation

In this section, we present a causal conditional estimator $\hat{\xi}_{tk|t} = \hat{\lambda}_{tk|t}/\widehat{\sigma^2_{tk}}$ for the a priori SNR given the noisy measurements up to frame t. The estimator combines two steps, a *propagation* step and an *update* step, following the rational of Kalman filtering, to predict and update the estimate for λ_{tk} recursively as new data arrive.

Let $\mathcal{X}_0^\tau = \{X_{tk} | t = 0, \ldots, \tau, k = 0, \ldots, N-1\}$ represent the set of clean-speech spectral coefficients up to frame τ, and let $\lambda_{tk|\tau} \triangleq E\{|X_{tk}|^2 | H_1^{tk}, \mathcal{X}_0^\tau\}$ denote the *conditional* variance of X_{tk} under H_1^{tk} given the clean spectral coefficients up to frame τ. Assuming that an estimate $\hat{\lambda}_{tk|t-1}$ for the one-frame-ahead conditional variance of X_{tk} is available, an estimate for $\lambda_{tk|t}$ can be obtained by calculating its conditional mean under H_1^{tk} given Y_{tk} and $\hat{\lambda}_{tk|t-1}$. By definition, $\lambda_{tk|t} = |X_{tk}|^2$. Hence,

$$\begin{aligned}\hat{\lambda}_{tk|t} &= E\left\{|X_{tk}|^2 \middle| H_1^{tk}, \hat{\lambda}_{tk|t-1}, Y_{tk}\right\} \\ &= E\left\{X_{\mathrm{R}tk}^2 \middle| H_1^{tk}, \hat{\lambda}_{tk|t-1}, Y_{\mathrm{R}tk}\right\} \\ &\quad + E\left\{X_{\mathrm{I}tk}^2 \middle| H_1^{tk}, \hat{\lambda}_{tk|t-1}, Y_{\mathrm{I}tk}\right\}, \end{aligned} \tag{44.44}$$

where we have used that $X_{\mathrm{R}tk}$ is independent of $Y_{\mathrm{I}tk}$, and $X_{\mathrm{I}tk}$ is independent of $Y_{\mathrm{R}tk}$. Let $G_{\mathrm{SP}}(\xi, \gamma_\rho)$ represent the MMSE gain function in the spectral power domain for $Y_{\rho tk} \neq 0$ [44.43]:

$$E\left\{X_{\rho tk}^2 \middle| H_1^{tk}, \hat{\lambda}_{tk|t-1}, Y_{\rho tk}\right\} = G_{\mathrm{SP}}\left(\hat{\xi}_{tk|t-1}, \gamma_{\rho tk}\right) Y_{\rho tk}^2 , \tag{44.45}$$

where $\rho \in \{\mathrm{R}, \mathrm{I}\}$. Then the specific expression for $G_{\mathrm{SP}}(\xi, \gamma_\rho)$ depends on the particular statistical model. For a Gaussian model, the spectral power gain function is given by

$$G_{\mathrm{SP}}(\xi, \gamma_\rho) = \frac{\xi}{1+\xi}\left(\frac{1}{2\gamma_\rho} + \frac{\xi}{1+\xi}\right) . \tag{44.46}$$

For a gamma model [44.43],

$$\begin{aligned} G_{\mathrm{SP}}(\xi, \gamma_\rho) = \frac{3}{8\gamma_\rho} &\left[\exp\left(\frac{C_{\rho-}^2}{4}\right) D_{-2.5}(C_{\rho-}) \right. \\ &\left. + \exp\left(\frac{C_{\rho+}^2}{4}\right) D_{-2.5}(C_{\rho+})\right] \\ &\times \left[\exp\left(\frac{C_{\rho-}^2}{4}\right) D_{-0.5}(C_{\rho-}) \right. \\ &\left. + \exp\left(\frac{C_{\rho+}^2}{4}\right) D_{-0.5}(C_{\rho+})\right]^{-1} , \end{aligned} \tag{44.47}$$

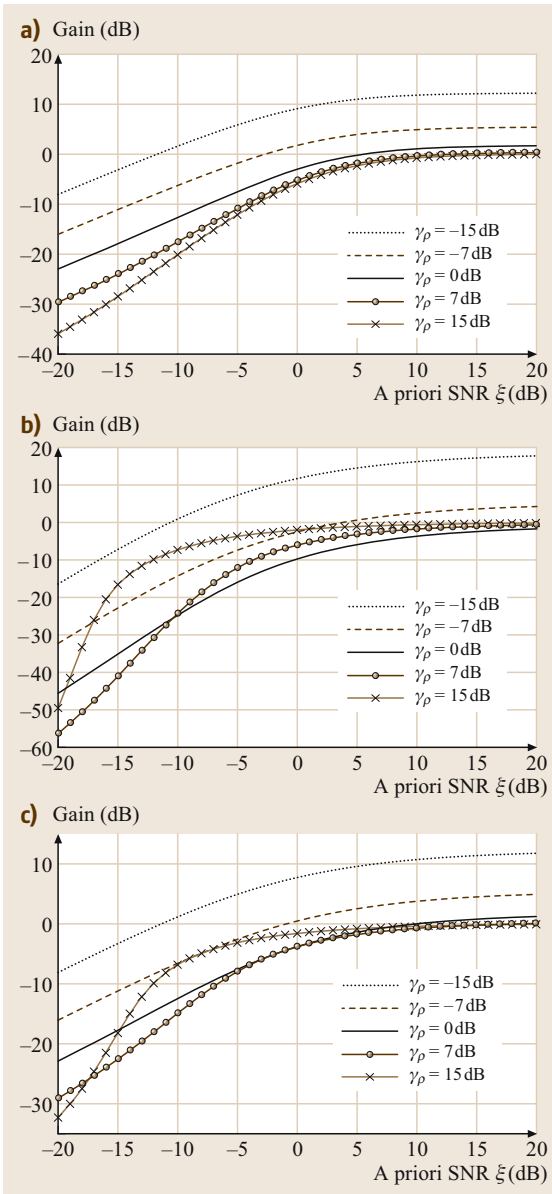


Fig. 44.8a–c Parametric gain curves describing the MMSE spectral power gain function $G_{\mathrm{SP}}(\xi, \gamma_\rho)$ for different speech models. **(a)** Gain curves for a Gaussian model, obtained by (44.46). **(b)** Gain curves for a gamma model, obtained by (44.47). **(c)** Gain curves for a Laplacian model, obtained by (44.48) (after [44.43])

where $C_{\rho\pm}$ are defined by (44.25). For a Laplacian model [44.43],

$$G_{\mathrm{SP}}(\xi,\gamma_\rho) = \frac{1}{\gamma_\rho}[\mathrm{erfcx}(L_{\rho+})+\mathrm{erfcx}(L_{\rho-})]^{-1} \times\left[\left(L_{\rho+}^2+0.5\right)\mathrm{erfcx}(L_{\rho+}) + \left(L_{\rho-}^2+0.5\right)\mathrm{erfcx}(L_{\rho-}) - \frac{(L_{\rho+}+L_{\rho-})}{\sqrt{\pi}}\right], \quad (44.48)$$

where $L_{\rho\pm}$ are defined by (44.27). Figure 44.8 shows parametric gain curves, which describe the functions $G_{\mathrm{SP}}(\xi,\gamma_\rho)$ for several values of γ_ρ, resulting from (44.46), (44.47), and (44.48). When γ_ρ is kept constant, $G_{\mathrm{SP}}(\xi,\gamma_\rho)$ is a monotonically increasing function of ξ. When ξ is kept constant, $G_{\mathrm{SP}}(\xi,\gamma_\rho)$ is a monotonically decreasing function of γ_ρ for a Gaussian model, but is not a monotonic function of γ_ρ for gamma or Laplacian models.

Equation (44.45) does not hold in the case $Y_{\rho tk}\to 0$, since $G_{\mathrm{SP}}(\xi,\gamma_\rho)\to\infty$ as $\gamma_\rho\to 0$, and the conditional variance of $X_{\rho tk}$ is generally not zero. For $Y_{\rho tk}=0$ (or practically for $Y_{\rho tk}$ smaller in magnitude than a predetermined threshold) we use the following expressions [44.43]: for a Gaussian model

$$E\left\{X_{\rho tk}^2\middle|H_1^{tk},\hat{\lambda}_{tk|t-1},Y_{\rho tk}=0\right\}=\frac{\hat{\xi}_{tk|t-1}}{1+\hat{\xi}_{tk|t-1}}\sigma_{tk}^2, \quad (44.49)$$

for a gamma model

$$E\left\{X_{\rho tk}^2\middle|H_1^{tk},\hat{\lambda}_{tk|t-1},Y_{\rho tk}=0\right\} = \frac{3D_{-2.5}\left(\frac{\sqrt{3}}{2\sqrt{\hat{\xi}_{tk|t-1}}}\right)}{8D_{-0.5}\left(\frac{\sqrt{3}}{2\sqrt{\hat{\xi}_{tk|t-1}}}\right)}\sigma_{tk}^2 \quad (44.50)$$

and for a Laplacian model

$$E\left\{X_{\rho tk}^2\middle|H_1^{tk},\hat{\lambda}_{tk|t-1},Y_{\rho tk}=0\right\} = \sqrt{\frac{2}{\pi}}\frac{\exp\left(\frac{1}{2\hat{\xi}_{tk|t-1}}\right)D_{-3}\left(\sqrt{\frac{2}{\hat{\xi}_{tk|t-1}}}\right)}{\mathrm{erfcx}\left(\frac{1}{\sqrt{\hat{\xi}_{tk|t-1}}}\right)}\sigma_{tk}^2. \quad (44.51)$$

From (44.45–44.51), we can define a function $f(\xi,\sigma^2,Y_\rho^2)$ such that

$$\frac{1}{\sigma_{tk}^2}E\left\{X_{\rho tk}^2\middle|H_1^{tk},\hat{\lambda}_{tk|t-1},Y_{\rho tk}\right\} = f\left(\hat{\xi}_{tk|t-1},\sigma_{tk}^2,Y_{\rho tk}^2\right) \quad (44.52)$$

for all $Y_{\rho tk}$. Substituting (44.52) into (44.44), we obtain an estimate for $\xi_{tk|t}$ given by

$$\hat{\xi}_{tk|t}=f\left(\hat{\xi}_{tk|t-1},\widehat{\sigma_{tk}^2},Y_{\mathrm{R}tk}^2\right)+f\left(\hat{\xi}_{tk|t-1},\widehat{\sigma_{tk}^2},Y_{\mathrm{I}tk}^2\right). \quad (44.53)$$

Equation (44.53) is the update step of the recursive estimation, since we start with an estimate $\hat{\xi}_{tk|t-1}$ that relies on the noisy observations up to frame $t-1$, and then update the estimate by using the additional information Y_{tk}.

To formulate the propagation step, we assume that we are given at frame $t-1$ estimates for the speech spectral coefficient $X_{t-1,k}$ and its spectral variance $\lambda_{t-1,k}$, conditioned on $\mathcal{Y}_0^{t-1}=\{Y_{\tau k}|\tau=0,\ldots,t-1,k=0,\ldots,N-1\}$. Then, these estimates can be *propagated* in time to obtain an estimate for λ_{tk}. Since λ_{tk} is correlated with both $\lambda_{t-1,k}$ and $|X_{t-1,k}|$, it was proposed in [44.15] to use an estimate of the form

$$\hat{\lambda}_{tk|t-1} = \max\left\{(1-\mu)\hat{\lambda}_{t-1,k|t-1}+\mu\left|\hat{X}_{t-1,k|H_1}\right|^2,\lambda_{\min}\right\}, \quad (44.54)$$

where μ ($0\le\mu\le 1$) is related to the degree of nonstationarity of the random process $\{\lambda_{tk}|t=0,1,\ldots\}$, and $\lambda_{\min}$ is a lower bound on the variance of X_{tk}. In the case of a pseudostationary process, μ is set to a small value, since $\hat{\lambda}_{tk|t-1}\approx\hat{\lambda}_{t-1,k|t-1}$. In the case of a nonstationary process, μ is set to a larger value, since the variances at successive frames are less correlated, and the relative importance of $\hat{\lambda}_{t-1,k|t-1}$ to predict $\hat{\lambda}_{tk|t-1}$ decreases. Dividing both sides of (44.54) by $\hat{\sigma}_{t-1,k}^2$, we obtain the *propagation* step

$$\hat{\xi}_{tk|t-1} = \max\left\{(1-\mu)\hat{\xi}_{t-1,k|t-1}+\mu\frac{|\hat{X}_{t-1,k|H_1}|^2}{\hat{\sigma}_{t-1,k}^2},\xi_{\min}\right\}, \quad (44.55)$$

where $\xi_{\min}$ is a lower bound on the a priori SNR.

The steps of the causal recursive a priori SNR estimation are summarized in Table 44.2. The algorithm is initialized at frame $t=-1$ with $\hat{X}_{-1,k|H_1}=0$ and

Table 44.2 Summary of the causal recursive a priori SNR estimation

Initialization:
$\hat{X}_{-1,k\|H_1} = 0,\ \hat{\xi}_{-1,k\|-1} = \xi_{\min}$, for all k.
For all short-time frames $t = 0, 1, \ldots$
For all frequency bins $k = 0, \ldots, N-1$
Propagation step:
Compute $\hat{\xi}_{tk\|t-1}$ using (44.55)
Update step:
Compute $\hat{\xi}_{tk\|t}$ using (44.53)
Spectral estimation:
Compute $\hat{X}_{tk\|H_1} = \hat{X}_{tk}\big\|_{\hat{p}_{tk}=1}$ using, e.g., (44.22) or (44.29)

$\hat{\xi}_{-1,k|-1} = \xi_{\min}$ for all k. Then, for $t = 0, 1, \ldots$, the propagation and update steps are iterated to obtain estimates for the nonstationary a priori SNR. The spectral gain function employed for the computation of $\hat{X}_{tk|H_1}$ is determined by the particular choice of the distortion measure.

44.6.3 Relation Between Causal Recursive Estimation and Decision-Directed Estimation

The causal conditional estimator $\hat{\xi}_{tk|t}$ for the a priori SNR is closely related to the decision-directed estimator of *Ephraim* and *Malah* [44.12]. The decision-directed estimator under speech presence uncertainty is given by (44.41) where α $(0 \le \alpha \le 1)$ is a weighting factor that controls the trade-off between the noise reduction and the transient distortion introduced into the signal.

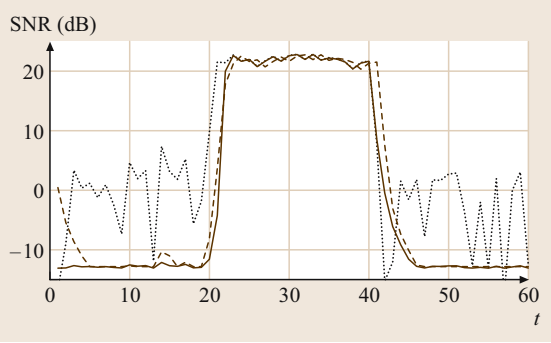


Fig. 44.9 SNRs in successive short-time frames: a posteriori SNR γ_{tk} (*dotted line*), decision-directed a priori SNR $\hat{\xi}^{\mathrm{DD}}_{tk|t}$ (*dashed line*), and causal recursive a priori SNR estimate $\hat{\xi}^{\mathrm{RE}}_{tk|t}$ (*solid line*) (after [44.15])

The update step (44.53) of the causal recursive estimator under a Gaussian model can be written as [44.15]

$$\hat{\xi}_{tk|t} = \alpha_{tk}\hat{\xi}_{tk|t-1} + (1-\alpha_{tk})(\hat{\gamma}_{tk} - 1)\,, \tag{44.56}$$

where α_{tk} is given by

$$\alpha_{tk} \triangleq 1 - \frac{\hat{\xi}^2_{tk|t-1}}{\left(1+\hat{\xi}_{tk|t-1}\right)^2}\,. \tag{44.57}$$

Substituting (44.55) into (44.56) and (44.57) with $\mu \equiv 1$, and applying the lower-bound constraint to $\hat{\xi}_{tk|t}$ rather than $\hat{\xi}_{tk|t-1}$, we have

$$\hat{\xi}_{tk|t} = \max\left\{\alpha_{tk}\frac{\left|\hat{X}_{t-1,k|H_1}\right|^2}{\hat{\sigma}^2_{t-1,k}} + (1-\alpha_{tk})\left(\hat{\gamma}_{tk}-1\right), \xi_{\min}\right\}\,, \tag{44.58}$$

$$\alpha_{tk} = 1 - \frac{\left|\hat{X}_{t-1,k|H_1}\right|^4}{\left(\hat{\sigma}^2_{t-1,k} + \left|\hat{X}_{t-1,k|H_1}\right|^2\right)^2}\,. \tag{44.59}$$

The expression (44.58) with $\alpha_{tk} \equiv \alpha$ is actually a practical form of the decision-directed estimator,

$$\hat{\xi}^{\mathrm{DD}}_{tk|t} = \max\left\{\alpha\frac{\left|\hat{X}_{t-1,k|H_1}\right|^2}{\hat{\sigma}^2_{t-1,k}} + (1-\alpha)\left(\hat{\gamma}_{tk}-1\right), \xi_{\min}\right\}\,, \tag{44.60}$$

that includes a lower-bound constraint to further reduce the level of residual musical noise [44.47]. Accordingly, a special case of the causal recursive estimator with $\mu \equiv 1$ degenerates to a *decision-directed* estimator with a *time-varying frequency-dependent* weighting factor α_{tk}.

It is interesting to note that the weighting factor α_{tk}, given by (44.59), is monotonically decreasing as a function of the instantaneous SNR, $|\hat{X}_{t-1,k|H_1}|^2/\hat{\sigma}^2_{t-1,k}$. A decision-directed estimator with a larger weighting factor is indeed preferable during speech absence (to reduce musical noise phenomena), while a smaller weighting factor is more advantageous during speech presence (to reduce signal distortion) [44.47]. The above special case of the causal recursive estimator conforms to such a desirable behavior. Moreover, the general form of the causal recursive estimator provides an additional degree of freedom for adjusting the value of μ in (44.55) to the degree of spectral nonstationarity. This may produce even further improvement in the performance.

The different behaviors of the causal recursive estimator $\hat{\xi}^{\mathrm{RE}}_{tk|t}$ (Table 44.2) and the decision-directed estimator $\hat{\xi}^{\mathrm{DD}}_{tk|t}$ (44.60) are illustrated in the example

Amazon EX1010 Page 890

of Fig. 44.9. The analyzed signal contains only white Gaussian noise during the first and last 20 frames, and in between it contains an additional sinusoidal component at the displayed frequency with 0 dB SNR. (Note that the SNR is computed in the time domain, whereas the a priori and a posteriori SNRs are computed in the time–frequency domain. Therefore, the latter SNRs may increase at the displayed frequency well above the average SNR.) The signal is transformed into the STFT domain using half-overlapping Hamming windows. The a priori SNR estimates, $\hat{\xi}^{\mathrm{RE}}_{tk|t}$ and $\hat{\xi}^{\mathrm{DD}}_{tk|t}$, are obtained by using the parameters $\xi_{\min} = -25\,\mathrm{dB}$, $\alpha = 0.98$, $\mu = 0.9$. The spectral estimate $\hat{X}_{tk|H_1}$ is recursively obtained by applying $G_{\mathrm{LSA}}(\hat{\xi}_{tk|t}, \hat{\gamma}_{tk})$ to the noisy spectral measurements (44.29, 31).

Figure 44.9 shows that, when the a posteriori SNR γ_{tk} is sufficiently low, the causal recursive a priori SNR estimate is smoother than the decision-directed estimate, which helps reducing the level of musical noise. When γ_{tk} increases, the response of the a priori SNR $\hat{\xi}^{\mathrm{RE}}_{tk|t}$ is initially slower than $\hat{\xi}^{\mathrm{DD}}_{tk|t}$, but then builds up faster to the a posteriori SNR. When γ_{tk} is sufficiently high, $\hat{\xi}^{\mathrm{DD}}_{tk|t}$ follows the a posteriori SNR with a delay of one frame, whereas $\hat{\xi}^{\mathrm{RE}}_{tk|t}$ follows the a posteriori SNR instantaneously. When γ_{tk} decreases, the response of $\hat{\xi}^{\mathrm{RE}}_{tk|t}$ is immediate, while that of $\hat{\xi}^{\mathrm{DD}}_{tk|t}$ is delayed by one frame. As a consequence, when compared with the decision-directed estimator, the causal recursive estimator produces a lower level of musical noise while not increasing the audible distortion in the enhanced signal [44.15].

44.6.4 Noncausal Recursive Estimation

In some important applications, e.g., digital voice recording, surveillance, speech recognition and speaker identification, a delay of a few short-term frames between the enhanced speech and the noisy observation is tolerable. In such cases, a noncausal estimation approach may produce less signal distortion and less musical residual noise than a causal estimation approach. In this section, we present a noncausal conditional estimator $\hat{\xi}_{tk|t+L}$ for the a priori SNR, given the noisy measurements up to frame $t+L$, where $L > 0$ denotes the admissible time delay in frames. Similar to the causal estimator, the noncausal estimator combines update and propagation steps to recursively estimate λ_{tk} as new data arrive. However, future spectral measurements are also employed in the process to better predict the spectral variances of the clean speech.

Let $\lambda'_{tk|t+L} \triangleq E\{|X_{tk}|^2 | \mathcal{Y}_0^{t-1}, \mathcal{Y}_{t+1}^{t+L}\}$ denote the conditional spectral variance of X_{tk} given $\mathcal{Y}_0^{t+L}$ excluding the noisy measurement at frame t. Let $\lambda_{tk|[t+1,t+L]} \triangleq E\{|X_{tk}|^2 | \mathcal{Y}_{t+1}^{t+L}\}$ denote the conditional spectral variance of X_{tk} given the subsequent noisy measurements $\mathcal{Y}_{t+1}^{t+L}$. Then, similar to (44.53), the estimate for ξ_{tk} given $\hat{\xi}'_{tk|t+L} \triangleq \hat{\lambda}'_{tk|t+L}/\widehat{\sigma_{tk}^2}$ and Y_t can be updated by

$$\hat{\xi}_{tk|t+L} = f\left(\hat{\xi}'_{tk|t+L}, \widehat{\sigma_{tk}^2}, Y^2_{\mathrm{R}tk}\right) + f\left(\hat{\xi}'_{tk|t+L}, \widehat{\sigma_{tk}^2}, Y^2_{\mathrm{I}tk}\right) . \tag{44.61}$$

To obtain an estimate for $\lambda'_{tk|t+L}$, we employ the estimates $\hat{X}_{t-1,k|H_1}$ and $\hat{\lambda}_{t-1,k|t+L-1}$ from the previous frame, and derive an estimate for λ_{tk} from the measurements $\mathcal{Y}_{t+1}^{t+L}$. Suppose an estimate $\hat{\lambda}_{tk|[t+1,t+L]}$ is given, we propagate the estimates from frame $t-1$ to frame t by [44.15, 53]

$$\hat{\lambda}'_{tk|t+L} = \max\left\{\mu\left|\hat{X}_{t-1,k|H_1}\right|^2 + (1-\mu)\left[\mu'\hat{\lambda}_{t-1,k|t+L-1} + (1-\mu')\hat{\lambda}_{tk|[t+1,t+L]}\right], \lambda_{\min}\right\} , \tag{44.62}$$

where μ ($0 \le \mu \le 1$) is related to the stationarity of the random process $\{\lambda_{tk} | t = 0, 1, \dots\}$, and μ' ($0 \le \mu' \le 1$) is associated with the reliability of the estimate $\hat{\lambda}_{tk|[t+1,t+L]}$ in comparison with that of $\hat{\lambda}_{t-1,k|t+L-1}$. Dividing both sides of (44.62) by $\hat{\sigma}^2_{t-1,k}$, we have the following *backward–forward propagation* step:

$$\hat{\xi}'_{tk|t+L} = \max\left\{\mu\frac{\left|\hat{X}_{t-1,k|H_1}\right|^2}{\hat{\sigma}^2_{t-1,k}} + (1-\mu)\left[\mu'\hat{\xi}_{t-1,k|t+L-1} + (1-\mu')\hat{\xi}_{tk|[t+1,t+L]}\right], \xi_{\min}\right\} . \tag{44.63}$$

An estimate for the a priori SNR ξ_{tk} given the measurements $\mathcal{Y}_{t+1}^{t+L}$ is obtained by

$$\hat{\xi}_{tk|[t+1,t+L]} = \begin{cases} \frac{1}{L}\sum_{n=1}^{L}\hat{\gamma}_{t+n,k} - \beta_f , & \text{if positive} , \\ 0 , & \text{otherwise} , \end{cases} \tag{44.64}$$

where β_f ($\beta_f \ge 1$) is an oversubtraction factor to compensate for a sudden increase in the noise level. This estimator is an anticausal version of the maximum-likelihood a priori SNR estimator suggested in [44.12].

Table 44.3 Summary of the noncausal recursive a priori SNR estimation

Initialization:
$\hat{X}_{-1,k\|H_1} = 0,\ \hat{\xi}_{-1\|L-1} = \xi_{\min}$.
For all short-time frames $t = 0, 1, \ldots$
For all frequency bins $k = 0, \ldots, N-1$
Backward estimation:
Compute $\hat{\xi}_{tk\|[t+1,t+L]}$ using (44.64)
Backward–forward propagation:
Compute $\hat{\xi}'_{tk\|t+L}$ using (44.63)
Update step:
Compute $\hat{\xi}_{tk\|t+L}$ using (44.61)
Spectral estimation:
Compute $\hat{X}_{tk\|H_1} = \hat{X}_{tk}\big\|_{\hat{p}_{tk}=1}$ using, e.g., (44.22) or (44.29)

The steps of the noncausal recursive a priori SNR estimation are summarized in Table 44.3. The algorithm is initialized at frame $t = -1$ with $\hat{X}_{-1,k|H_1} = 0$ and $\hat{\xi}_{-1|L-1} = \xi_{\min}$. Then, for $t = 0, 1, \ldots$, the propagation and update steps are iterated to obtain estimates for the a priori SNR and the speech spectral components.

Figure 44.10 demonstrates the behavior of the noncausal recursive estimator in the same example of Fig. 44.9. The noncausal a priori SNR estimate $\hat{\xi}^{\mathrm{RE}}_{tk|t+3}$ is obtained with the parameters $\xi_{\min} = -25\,\mathrm{dB}$, $\mu = \mu' = 0.9$, $\beta_f = 2$, and $L = 3$ frames delay. A comparison of Figs. 44.9 and 44.10 indicates that the differences between the causal and noncausal recursive estimators are primarily noticeable during onsets

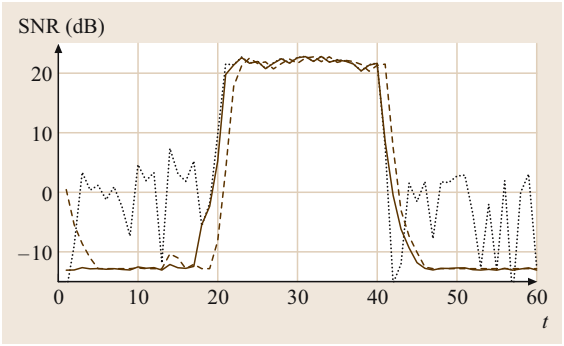


Fig. 44.10 SNRs in successive short-time frames: a posteriori SNR γ_{tk} (*dotted line*), decision-directed a priori SNR $\hat{\xi}^{\mathrm{DD}}_{tk|t}$ (*dashed line*), and noncausal recursive a priori SNR estimate $\hat{\xi}^{\mathrm{RE}}_{tk|t+3}$ with three-frame delay (*solid line*) (after [44.15])

of signal components. Clearly, the *causal* a priori SNR estimator, as well as the decision-directed estimator, cannot respond too fast to an abrupt increase in γ_{tk}, since it necessarily implies an increase in the level of musical residual noise. By contrast, the *noncausal* estimator, having a few subsequent spectral measurements at hand, is capable of discriminating between speech onsets and irregularities in γ_{tk} corresponding to noise only. Therefore, in comparison with the decision-directed estimator, the noncausal a priori SNR estimator produces even lower levels of musical noise and signal distortion [44.15, 53].

44.7 Noise Spectrum Estimation

In this section, we derive an estimator for the noise power spectrum under speech presence uncertainty. The noise estimate is obtained by averaging past spectral power values of the noisy measurement, and multiplying the result by a constant factor that compensates the bias. The recursive averaging is carried out using a time-varying frequency-dependent smoothing parameter that is adjusted by the speech presence probability.

44.7.1 Time-Varying Recursive Averaging

A common noise estimation technique is to average past spectral power values of the noisy measurement recursively during periods of speech absence, and hold the estimate during speech presence. Specifically,

$$\begin{aligned} H_0^{tk} &: \overline{\sigma}^2_{t+1,k} = \alpha_{\mathrm{d}}\overline{\sigma}^2_{tk} + (1-\alpha_{\mathrm{d}})|Y_{tk}|^2 \\ H_1^{tk} &: \overline{\sigma}^2_{t+1,k} = \overline{\sigma}^2_{tk}\,, \end{aligned} \tag{44.65}$$

where α_{d} $(0 < \alpha_{\mathrm{d}} < 1)$ denotes a smoothing parameter. Under speech presence uncertainty, we can employ the conditional speech presence probability, and carry out the recursive averaging by

$$\begin{aligned} \overline{\sigma}^2_{t+1,k} = {} & \tilde{p}_{tk}\overline{\sigma}^2_{tk} \\ & + (1-\tilde{p}_{tk})\left[\alpha_{\mathrm{d}}\overline{\sigma}^2_{tk} + (1-\alpha_{\mathrm{d}})|Y_{tk}|^2\right], \end{aligned} \tag{44.66}$$

where $\tilde{p}_{tk}$ is an estimator for the conditional speech presence probability $p_{tk} = P(H_1^{tk}|Y_{tk})$. Equivalently, the

recursive averaging can be obtained by

$$\overline{\sigma}^2_{t+1,k} = \tilde{\alpha}_{tk}\overline{\sigma}^2_{tk} + \big(1-\tilde{\alpha}_{tk}\big)|Y_{tk}|^2 \,, \tag{44.67}$$

where

$$\tilde{\alpha}_{tk} \triangleq \alpha_{\mathrm{d}} + (1-\alpha_{\mathrm{d}})\tilde{p}_{tk} \tag{44.68}$$

is a time-varying frequency-dependent smoothing parameter. The smoothing parameter $\tilde{\alpha}_{tk}$ is adjusted by the speech presence probability, which is estimated based on the noisy measurement.

Here we make a distinction between the estimator $\hat{p}_{tk}$ in (44.3), used for estimating the clean speech, and the estimator $\tilde{p}_{tk}$, which controls the adaptation of the noise spectrum. Clearly, deciding speech is absent (H_0) when speech is present (H_1) is more destructive when estimating the speech than when estimating the noise. Hence, different decision rules are employed [44.42], and generally we tend to employ estimators that satisfy $\hat{p}_{tk} \geq \tilde{p}_{tk}$. Given an estimator $\tilde{p}_{tk|t-1}$ for the a priori speech presence probability, the conditional speech presence probability estimate can be obtained from Bayes' rule, which under a Gaussian model reduces to [44.12]

$$\tilde{p}_{tk} = \left[1 + \frac{\big(1-\tilde{p}_{tk|t-1}\big)\big(1+\hat{\xi}_{tk}\big)}{\tilde{p}_{tk|t-1}\exp\left(\frac{\hat{\xi}_{tk}\hat{\gamma}_{tk}}{1+\hat{\xi}_{tk}}\right)}\right]^{-1} . \tag{44.69}$$

In the next subsection we present an estimator $\tilde{p}_{tk|t-1}$ that enables noise spectrum estimation during speech activity. Both $\tilde{p}_{tk|t-1}$ and $\hat{p}_{tk|t-1}$ are biased toward higher values, since deciding that speech is absent when speech is present results ultimately in the attenuation of speech components. Whereas, the alternative false decision, up to a certain extent, merely introduces some level of residual noise. Accordingly, we include a bias compensation factor in the noise estimator

$$\hat{\sigma}^2_{t+1,k} = \beta\overline{\sigma}^2_{t+1,k} \tag{44.70}$$

such that the factor β ($\beta \geq 1$) compensates the bias when speech is absent

$$\beta \triangleq \left.\frac{\sigma^2_{tk}}{E\big\{\overline{\sigma}^2_{tk}\big\}}\right|_{\xi_{tk}=0} . \tag{44.71}$$

The value of β is completely determined by the particular estimator for the a priori speech absence probability [44.54]. We note that the noise estimate is based on a variable time segment in each subband, which takes into account the probability of speech presence. The time segment is longer in subbands that contain frequent *speech* portions, and shorter in subbands that contain frequent *silence* portions. This feature has been considered [44.55] a desirable characteristic of the noise estimator, which improves its robustness and tracking capability.

44.7.2 Minima-Controlled Estimation

In this section, we present an estimator $\tilde{p}_{tk|t-1}$ that is controlled by the minima values of a smoothed power spectrum of the noisy signal. In contrast to the minimum statistics (MS) and related methods [44.56, 57], the smoothing of the noisy power spectrum is carried out in both time and frequency. This takes into account the strong correlation of speech presence in neighboring frequency bins of consecutive frames [44.42]. Furthermore, the procedure comprises two iterations of smoothing and minimum tracking. The first iteration provides a rough voice activity detection in each frequency band. Then, the smoothing in the second iteration excludes relatively strong speech components, which makes the minimum tracking during speech activity robust, even when using a relatively large smoothing window. A larger smoothing window decreases the variance of the minima values, but also widens the peaks of the speech activity power. An alternative solution is to modify the smoothing in time and frequency based on a smoothed a posteriori SNR [44.56].

Let α_{s} ($0 < \alpha_{\mathrm{s}} < 1$) be a smoothing parameter, and let b denote a normalized window function of length $2w+1$, i. e., $\sum_{i=-w}^{w} b_i = 1$. The frequency smoothing of the noisy power spectrum in each frame is defined by

$$S^{\mathrm{f}}_{tk} = \sum_{i=-w}^{w} b_i |Y_{t,k-i}|^2 . \tag{44.72}$$

Subsequently, smoothing in time is performed by a first-order recursive averaging:

$$S_{tk} = \alpha_{\mathrm{s}} S_{t-1,k} + (1-\alpha_{\mathrm{s}}) S^{\mathrm{f}}_{tk} . \tag{44.73}$$

In accordance with the MS method, the minima values of S_{tk} are picked within a finite window of length D, for each frequency bin:

$$S^{\min}_{tk} \triangleq \min\{S_{\tau k} \mid t-D+1 \leq \tau \leq t\} . \tag{44.74}$$

It follows [44.56] that there exists a constant factor $B_{\min}$, independent of the noise power spectrum, such that

$$E\big\{S^{\min}_{tk} \big| \xi_{tk} = 0\big\} = B^{-1}_{\min}\sigma^2_{tk} . \tag{44.75}$$

The factor $B_{\min}$ represents the bias of a minimum noise estimate, and generally depends on the values of D, α_{s},

w and the spectral analysis parameters (type, length and overlap of the analysis windows). The value of $B_{\min}$ can be estimated by generating a white Gaussian noise, and computing the inverse of the mean of $S_{tk}^{\min}$. This also takes into account the time–frequency correlation of the noisy periodogram $|Y_{tk}|^2$. Notice that the value of $B_{\min}$ is fixed, whereas in [44.56] it is estimated for each frequency band and each frame.

Let $\gamma_{tk}^{\min}$ and ζ_{tk} be defined by

$$\gamma_{tk}^{\min} \triangleq \frac{|Y_{tk}|^2}{B_{\min} S_{tk}^{\min}} ,$$
$$\zeta_{tk} \triangleq \frac{S_{tk}}{B_{\min} S_{tk}^{\min}} . \tag{44.76}$$

Under a Gaussian model, the probability density functions of $\gamma_{tk}^{\min}$ and ζ_{tk}, in the absence of speech, can be approximated by exponential and chi-square densities, respectively [44.54]:

$$p\left(\gamma_{tk}^{\min} \mid H_0^{tk}\right) \approx \mathrm{e}^{-\gamma_{tk}^{\min}} u\left(\gamma_{tk}^{\min}\right) , \tag{44.77}$$

$$p\left(\zeta_{tk} \mid H_0^{tk}\right) \approx \frac{1}{\left(\frac{2}{m}\right)^{m/2} \Gamma\left(\frac{m}{2}\right)} \zeta_{tk}^{m/2-1} \times \exp\left(-\frac{m\zeta_{tk}}{2}\right) u(\zeta_{tk}) , \tag{44.78}$$

where $\Gamma(\cdot)$ is the gamma function, and m is the equivalent degrees of freedom. Based on the first iteration smoothing and minimum tracking, a rough decision about speech presence is given by

$$I_{tk} = \begin{cases} 1 , & \text{if } \gamma_{tk}^{\min} < \gamma_0 \text{ and } \zeta_{tk} < \zeta_0 \\ & \text{(speech is absent)} \\ 0 , & \text{otherwise} \\ & \text{(speech is present)} . \end{cases} \tag{44.79}$$

The thresholds γ_0 and ζ_0 are set to satisfy a certain significance level ϵ:

$$\mathcal{P}\left(\gamma_{tk}^{\min} \geq \gamma_0 \mid H_0^{tk}\right) < \epsilon , \tag{44.80}$$

$$\mathcal{P}\left(\zeta_{tk} \geq \zeta_0 \mid H_0^{tk}\right) < \epsilon . \tag{44.81}$$

From (44.77) and (44.78) we have

$$\gamma_0 = -\log(\epsilon) , \tag{44.82}$$

$$\zeta_0 = \frac{1}{m} F_{\chi^2;m}^{-1}(1-\epsilon) , \tag{44.83}$$

where $F_{\chi^2;m}(x)$ denotes the standard chi-square cumulative distribution function, with m degrees of freedom. Typically, we use $\epsilon = 0.01$ and $m = 32$, so $\gamma_0 = 4.6$ and $\zeta_0 = 1.67$.

The second iteration of smoothing is conditional on the rough speech activity detection of the first iteration. It includes only the power spectral components, which have been identified as containing primarily noise. We set the initial condition for the first frame by $\tilde{S}_{0,k} = S_{0,k}^{\mathrm{f}}$. Then, for $t > 0$ the smoothing in frequency, employing the above voice activity detector, is obtained by

$$\tilde{S}_{tk}^{\mathrm{f}} = \begin{cases} \dfrac{\sum_{i=-w}^{w} b_i I_{t,k-i} |Y_{t,k-i}|^2}{\sum_{i=-w}^{w} b_i I_{t,k-i}} , & \text{if } \sum_{i=-w}^{w} I_{t,k-i} \neq 0 \\ \tilde{S}_{t-1,k} , & \text{otherwise} . \end{cases} \tag{44.84}$$

Smoothing in time is given, as before, by a first-order recursive averaging

$$\tilde{S}_{tk} = \alpha_{\mathrm{s}} \tilde{S}_{t-1,k} + (1-\alpha_{\mathrm{s}}) \tilde{S}_{tk}^{\mathrm{f}} . \tag{44.85}$$

The minima values of $\tilde{S}_{tk}$ are picked within a finite window of length D, for each frequency bin

$$\tilde{S}_{tk}^{\min} \triangleq \min\left\{\tilde{S}_{\tau k} \mid t - D + 1 \leq \tau \leq t\right\} .$$

Accordingly, $\tilde{S}_{tk}^{\min}$ represents minima tracking that is conditional on the rough speech activity detection of the first iteration. We note that keeping the strong speech components out of the smoothing process enables improved minimum tracking. In particular, a larger smoothing parameter (α_{s}) and smaller minima search window (D) can be used. This reduces the variance of the minima values [44.56], and shortens the delay when responding to a rising noise power, which eventually improves the tracking capability of the noise estimator.

Let $\tilde{\gamma}_{tk}^{\min}$ and $\tilde{\zeta}_{tk}$ be defined by

$$\tilde{\gamma}_{tk}^{\min} \triangleq \frac{|Y_{tk}|^2}{B_{\min} \tilde{S}_{tk}^{\min}} ,$$
$$\tilde{\zeta}_{tk} \triangleq \frac{S_{tk}}{B_{\min} \tilde{S}_{tk}^{\min}} . \tag{44.86}$$

Since we use a relatively small significance level in the first iteration ($\epsilon = 0.01$), the influence of the voice activity detector in noise-only periods can be neglected. That is, the effect of excluding strong *noise* components from the smoothing process is negligible. Accordingly, the conditional distributions of $\tilde{\gamma}_{tk}^{\min}$ and $\tilde{\zeta}_{tk}$, in the absence of speech, are approximately the same as those of $\gamma_{tk}^{\min}$ and ζ_{tk} (44.77, 78).

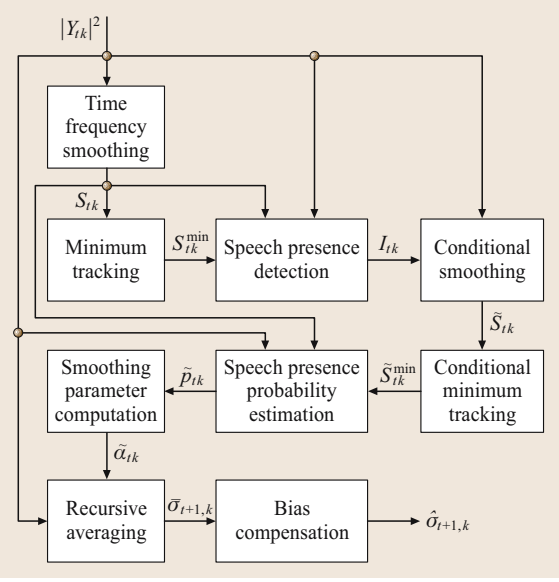


Fig. 44.11 Block diagram of the IMCRA noise spectrum estimation

An estimator for the a priori speech presence probability is given by

$$\tilde{p}_{tk|t-1} = \begin{cases} 0\,, & \text{if } \tilde{\gamma}_{tk}^{\min} \le 1 \\ & \text{and } \tilde{\zeta}_{tk} < \zeta_0 \\ \left(\tilde{\gamma}_{tk}^{\min} - 1\right)/(\gamma_1 - 1)\,, & \text{if } 1 < \tilde{\gamma}_{tk}^{\min} < \gamma_1 \\ & \text{and } \tilde{\zeta}_{tk} < \zeta_0 \\ 1\,, & \text{otherwise}\,. \end{cases} \tag{44.87}$$

The threshold γ_1 is set to satisfy a certain significance level ϵ_1 ($\epsilon_1 > \epsilon$):

$$P\left(\tilde{\gamma}_{tk}^{\min} > \gamma_1 \middle| H_0^{tk}\right) < \epsilon_1 \quad \Rightarrow \quad \gamma_1 \approx -\log(\epsilon_1)\,. \tag{44.88}$$

Typically $\epsilon_1 = 0.05$, and $\gamma_1 = 3$.

The a priori speech presence probability estimator assumes that speech is present ($\tilde{p}_{tk|t-1} = 1$) whenever $\tilde{\zeta}_{tk} \ge \zeta_0$ or $\tilde{\gamma}_{tk}^{\min} \ge \gamma_1$. That is, whenever the local measured power, S_{tk}, or the instantaneous measured power, $|Y_{tk}|^2$, are relatively high compared to the noise power $B_{\min}\tilde{S}_{tk}^{\min} \approx \sigma_{tk}^2$. The estimator assumes that speech is absent ($\tilde{p}_{tk|t-1} = 0$) whenever both the local and instantaneous measured powers are relatively low compared to the noise power ($\tilde{\gamma}_{tk}^{\min} \le 1$ and $\tilde{\zeta}_{tk} < \zeta_0$). In between, the estimator provides a soft transition between speech absence and speech presence, based on the value of $\tilde{\gamma}_{tk}^{\min}$.

The main objective of combining conditions on both $\tilde{\gamma}_{tk}^{\min}$ and $\tilde{\zeta}_{tk}$ is to prevent an increase in the estimated noise during weak speech activity, especially when the input SNR is low. Weak speech components can often be extracted using the condition on $\tilde{\zeta}_{tk}$. Sometimes, speech components are so weak that $\tilde{\zeta}_{tk}$ is smaller than ζ_0. In that case, most of the speech power is still excluded from the averaging process using the condition on $\tilde{\gamma}_{tk}^{\min}$. The remaining speech components can hardly affect the noise estimator, since their power is relatively low compared to that of the noise.

A block diagram of the improved minima-controlled recursive averaging (IMCRA) [44.54] noise spectrum estimation is described in Fig. 44.11. Typical values of parameters used in the implementation of the IMCRA noise estimator for a sampling rate of 16 kHz are summarized in Table 44.4. The noise spectrum estimate, $\hat{\sigma}_{tk}^2$, is initialized at the first frame by $\hat{\sigma}_{0,k}^2 = |Y_{0,k}|^2$. Then, at each frame t ($t \ge 0$), it is used, together with the current observation Y_{tk}, for estimating the noise power spectrum at the next frame, $t+1$. The bias compensation factor β is given by [44.54]

$$\beta = \frac{\gamma_1 - 1 - \mathrm{e}^{-1} + \mathrm{e}^{-\gamma_1}}{\gamma_1 - 1 - 3\,\mathrm{e}^{-1} + (\gamma_1 + 2)\,\mathrm{e}^{-\gamma_1}}\,. \tag{44.89}$$

In particular, for $\gamma_1 = 3$, we have $\beta = 1.47$. The value of $\tilde{\alpha}_{tk}$ is updated for each frequency bin and time frame, using the speech presence probability $\tilde{p}_{tk}$, and (44.68).

Table 44.4 Values of the parameters used in the implementation of the IMCRA noise estimator, for a sampling rate of 16 kHz

$w = 1$	$\alpha_s = 0.9$	$\alpha_d = 0.85$	$\beta = 1.47$
$B_{\min} = 1.66$	$\zeta_0 = 1.67$	$\gamma_0 = 4.6$	$\gamma_1 = 3$
$D = 120$	b: Hann window		

44.8 Summary of a Spectral Enhancement Algorithm

In this section, we present an example of a speech enhancement algorithm, which is based on an MMSE log-spectral amplitude estimation under a Gaussian model, IMCRA noise estimation, and decision-directed

Table 44.5 Summary of a speech enhancement algorithm

Initialization at the first frame for all frequency-bins $k = 1, \ldots, N/2$:
$\hat{\sigma}_{0k}^2 = \lvert Y_{0k}\rvert^2;\ \overline{\sigma}_{0k}^2 = \lvert Y_{0k}\rvert^2;\ S_{0k} = S_{0k}^{\mathrm{f}};\ S_{0k}^{\min} = S_{0k}^{\mathrm{f}};\ S_k^{\min\_\mathrm{sw}} = S_{0k}^{\mathrm{f}};\ \tilde{S}_{0k} = S_{0k}^{\mathrm{f}};\ \tilde{S}_{0k}^{\min} = S_{0k}^{\mathrm{f}};\ \tilde{S}_k^{\min\_\mathrm{sw}} = S_{0k}^{\mathrm{f}};\ \zeta_{0k} = 0.$
Let $\ell = 0$. % ℓ is a counter for frames within a subwindow ($0 \le \ell \le V$).
For all short-time frames $t = 0, 1, \ldots$
For all frequency bins $k = 1, \ldots, N/2$
Compute the a posteriori SNR $\hat{\gamma}_{tk}$ using (44.20) and (44.30), and the a priori SNR $\hat{\xi}_{tk}$ using (44.41), with the initial condition $\hat{\xi}_{0k} = \alpha + (1-\alpha)\max\{\hat{\gamma}_{0k} - 1, 0\}$.
Compute the conditional spectral estimate under the hypothesis of speech presence $\hat{X}_{tk\lvert H_1} = G_{\mathrm{LSA}}(\hat{\xi}_{tk}, \hat{\gamma}_{tk}) Y_{tk}$ using (44.29) and (44.31).
Compute the smoothed power spectrum S_{tk} using (44.72) and (44.73), and update its running minimum: $S_{tk}^{\min} = \min\{S_{t-1,k}^{\min}, S_{tk}\};\ S_k^{\min\_\mathrm{sw}} = \min\{S_k^{\min\_\mathrm{sw}}, S_{tk}\}$.
Compute the indicator function I_{tk} for the voice activity detection using (44.76) and (44.79).
Compute the conditional smoothed power spectrum $\tilde{S}_{tk}$ using (44.84) and (44.85), and update its running minimum: $\tilde{S}_{tk}^{\min} = \min\{\tilde{S}_{t-1,k}^{\min}, \tilde{S}_{tk}\};\ \tilde{S}_k^{\min\_\mathrm{sw}} = \min\{\tilde{S}_k^{\min\_\mathrm{sw}}, \tilde{S}_{tk}\}$.
Compute the a priori speech presence probability $\tilde{p}_{tk\lvert t-1}$ using (44.86) and (44.87), the speech presence probability $\tilde{p}_{tk}$ using (44.69), and the smoothing parameter $\tilde{\alpha}_{tk}$ using (44.68).
Update the noise spectrum estimate $\hat{\sigma}_{t+1,k}^2$ using (44.67) and (44.70).
Compute P_{tk}^{local} and P_{tk}^{global} using (44.32–44.34), and P_t^{frame} using the block diagram in Fig. 44.7.
Compute the a priori speech presence probability $\hat{p}_{tk\lvert t-1}$ using (44.37), and the speech presence probability $\hat{p}_{tk}$ using (44.69) by substituting $\tilde{p}_{tk\lvert t-1}$ with $\hat{p}_{tk\lvert t-1}$.
Compute the speech spectral estimate $\hat{X}_{tk}$ using (44.29).
Let $\ell = \ell + 1$.
If $\ell = V$
For all frequency bins k
Store $S_k^{\min\_\mathrm{sw}}$, set $S_{tk}^{\min}$ to the minimum of the last U stored values of $S_k^{\min\_\mathrm{sw}}$, and let $S_k^{\min\_\mathrm{sw}} = S_{tk}$.
Store $\tilde{S}_k^{\min\_\mathrm{sw}}$, set $\tilde{S}_{tk}^{\min}$ to the minimum of the last U stored values of $\tilde{S}_k^{\min\_\mathrm{sw}}$, and let $\tilde{S}_k^{\min\_\mathrm{sw}} = \tilde{S}_{tk}$.
Let $\ell = 0$.

a priori SNR estimation. The performance of the algorithm is demonstrated on speech signals degraded by various additive noise types.

The implementation of the speech enhancement algorithm is summarized in Table 44.5. For each time frame t we recursively estimate the STFT coefficients of the clean speech $\{X_{tk} | k = 1, \ldots, N/2\}$ from the noisy STFT coefficients $\{Y_{tk} | k = 1, \ldots, N/2\}$, where N is the length of the analysis window. We typically use a Hamming window of 32 ms length and a framing step of 8 ms (i. e., $N = 512$ and $M = 128$ for a sampling rate of 16 kHz). In the first frame ($t = 0$) we compute $\{Y_{0k} | k = 0, \ldots, N-1\}$ by applying the discrete Fourier transform to a short-time section of the noisy data

$$\boldsymbol{y}_0 = \left[y(0)h(0)\ \ y(1)h(1)\ \ \ldots\ \ y(N-1)h(N-1) \right]^{\mathrm{T}} ,$$

where $h(n)$ is the analysis window. In the following frames ($t > 0$), the section of noisy data is updated with M additional samples

$$\boldsymbol{y}_t = \left[y(tM)h(0)\ \ \ y(1+tM)h(1) \right.$$
$$\left. \ldots\ \ \ y(N-1+tM)h(N-1) \right]^{\mathrm{T}} ,$$

and subsequently $\{Y_{tk} | k = 0, \ldots, N-1\}$ is computed by applying the discrete Fourier transform to $\boldsymbol{y}_t$. Since the speech signal $x(n)$ is assumed to be real, once we estimate $\{X_{tk} | k = 1, \ldots, N/2\}$, the spectral coefficients for $N/2 < k \le N-1$ are obtained by $\hat{X}_{tk} = \hat{X}_{t,N-k}^*$, where * denotes complex conjugation. The DC component $\hat{X}_{to}$ is set to zero, and a sequence $\{\hat{x}_t(n) | n = 0, \ldots, N-1\}$ is obtained by applying the inverse discrete Fourier transform to $\{\hat{X}_{tk} | k = 0, \ldots, N-1\}$:

$$\hat{x}_t(n) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{X}_{tk}\, \mathrm{e}^{\mathrm{i}\frac{2\pi}{N}nk} . \qquad (44.90)$$

Employing the weighted overlap-add method [44.37], we compute the following sequence

$$o_t(n) = \begin{cases} o_{t-1}(n+M) + N\tilde{h}(n)\hat{x}_t(n)\,, \\ \qquad\qquad \text{for } 0 \le n \le N-M-1 \\ N\tilde{h}(n)\hat{x}_t(n)\,, \quad \text{for } N-M \le n \le N-1\,, \end{cases} \tag{44.91}$$

where $\tilde{h}(n)$ is the synthesis window. Then, according to (44.2), for each frame t, we obtain M additional samples of the enhanced speech signal:

$$\hat{x}(n+tM) = o_t(n)\,, \quad n = 0, \ldots, M-1\,. \tag{44.92}$$

The synthesis window $\tilde{h}(n)$ should satisfy the completeness condition [44.36]

$$\sum_t \tilde{h}(n-tM)h(n-tM) = \frac{1}{N} \quad \text{for all } n\,. \tag{44.93}$$

Given analysis and synthesis windows that satisfy (44.93), any signal $x(n) \in \ell_2(\mathbb{Z})$ can be perfectly reconstructed from its STFT coefficients X_{tk}. However, for $M < N$ (over-redundant STFT representation) and for a given analysis window $h(n)$, there might be an infinite number of solutions to (44.93). A reasonable choice of a synthesis window is the one with minimum energy [44.36, 58], given by

$$\tilde{h}(n) = \frac{h(n)}{N\sum_\ell h^2(n-\ell M)}\,. \tag{44.94}$$

The estimator for the a priori speech presence probability, $\tilde{p}_{tk|t-1}$ in (44.87), requires two iterations of time–frequency smoothing (S_{tk}, $\tilde{S}_{tk}$) and minimum tracking ($S_{tk}^{\min}$, $\tilde{S}_{tk}^{\min}$). The minimum tracking is implemented by the method proposed in [44.56, 59], which provides a flexible balance between the computational complexity and the update rate of the minima values. Accordingly, we divide the window of D samples into U subwindows of V samples ($UV = D$). Whenever V samples are read, the minimum of the current subwindow is determined and stored for later use. The overall minimum is obtained as the minimum of past samples within the current subwindow and the U previous subwindow minima. Typical values of D and V correspond to 960 ms and 120 ms, respectively. That is, for a framing step of 8 ms (i. e., $M = 128$ for a sampling rate of 16 kHz) we set $D = 120$, $V = 15$, and $U = 8$.

To demonstrate the performance of the speech enhancement algorithm, utterances are taken from the TIMIT database [44.38], degraded by various noise types from the Noisex92 database [44.60], and enhanced by the algorithm in Table 44.5. A clean utterance from a female speaker is shown in Fig. 44.12. The speech signal is sampled at 16 kHz and degraded by the various noise types, which include white Gaussian noise, car interior noise, F16 cockpit noise, and babble noise. Figure 44.13 shows the noisy speech signals with SNR of 5 dB. The corresponding enhanced speech signals are shown in Fig. 44.14.

The performance is evaluated by three objective quality measures and informal listening tests. The first quality measure is the segmental SNR (SegSNR), in dB, defined by [44.61]

$$\text{SegSNR} = \frac{1}{T}\sum_{t=0}^{T-1} \mathcal{C}(\text{SNR}_t)\,, \tag{44.95}$$

where T denotes the number of frames in the signal, and

$$\text{SNR}_t = 10\log_{10}\frac{\sum_{n=tM}^{tM+N-1} x^2(n)}{\sum_{n=tM}^{tM+N-1}[x(n)-\hat{x}(n)]^2} \tag{44.96}$$

represents the SNR in the t-th frame. The operator $\mathcal{C}$ confines the SNR at each frame to the perceptually meaningful range between 35 dB and −10 dB ($\mathcal{C}x \triangleq \min[\max(x, -10), 35]$). The operator $\mathcal{C}$ prevents the segmental SNR measure from being biased in either a positive or negative direction due to a few silent or unusually high-SNR frames, which do not contribute

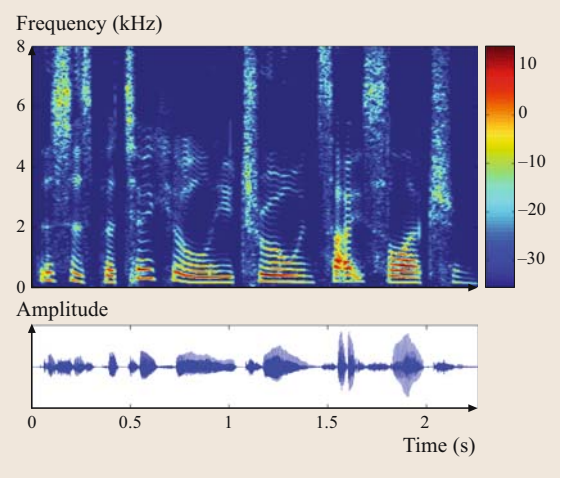


Fig. 44.12 Speech spectrogram and waveform of a clean speech signal: 'This is particularly true in site selection'

significantly to the overall speech quality [44.62, 63]. The second quality measure is log-spectral distortion (LSD), in dB, which is defined by

$$\mathrm{LSD} = \frac{1}{T}\sum_{t=0}^{T-1}\left[\frac{2}{N}\sum_{k=1}^{N/2}\left(\mathcal{L}X_{tk} - \mathcal{L}\hat{X}_{tk}\right)^2\right]^{\frac{1}{2}}, \tag{44.97}$$

where $\mathcal{L}X_{tk} \triangleq \max\{20\log_{10}|X_{tk}|, \delta\}$ is the log spectrum confined to about 50 dB dynamic range (that is, $\delta = \max_{tk}\{20\log_{10}|X_{tk}|\} - 50$). The third quality measure is the perceptual evaluation of speech quality (PESQ) score (ITU-T P.862).

The experimental results for the noisy and enhanced signals are given in the captions of Figs. 44.13 and 44.14. The improvement in SegSNR, reduction in LSD, and increase in PESQ scores are summarized in Table 44.6. Generally, the improvement in SegSNR and reduction in LSD are influenced by the variability of the noise characteristics in time and the initial SegSNR and LSD of the noisy signal. The faster the noise spectrum varies in time, the less reliable the noise spectrum estimator, and consequently the lower the quality gain that can be achieved by the speech enhancement system. Furthermore, the lower the SegSNR, respectively the LSD, for the noisy signal, the higher is the SegSNR improvement, respectively the lower is the LSD reduc-

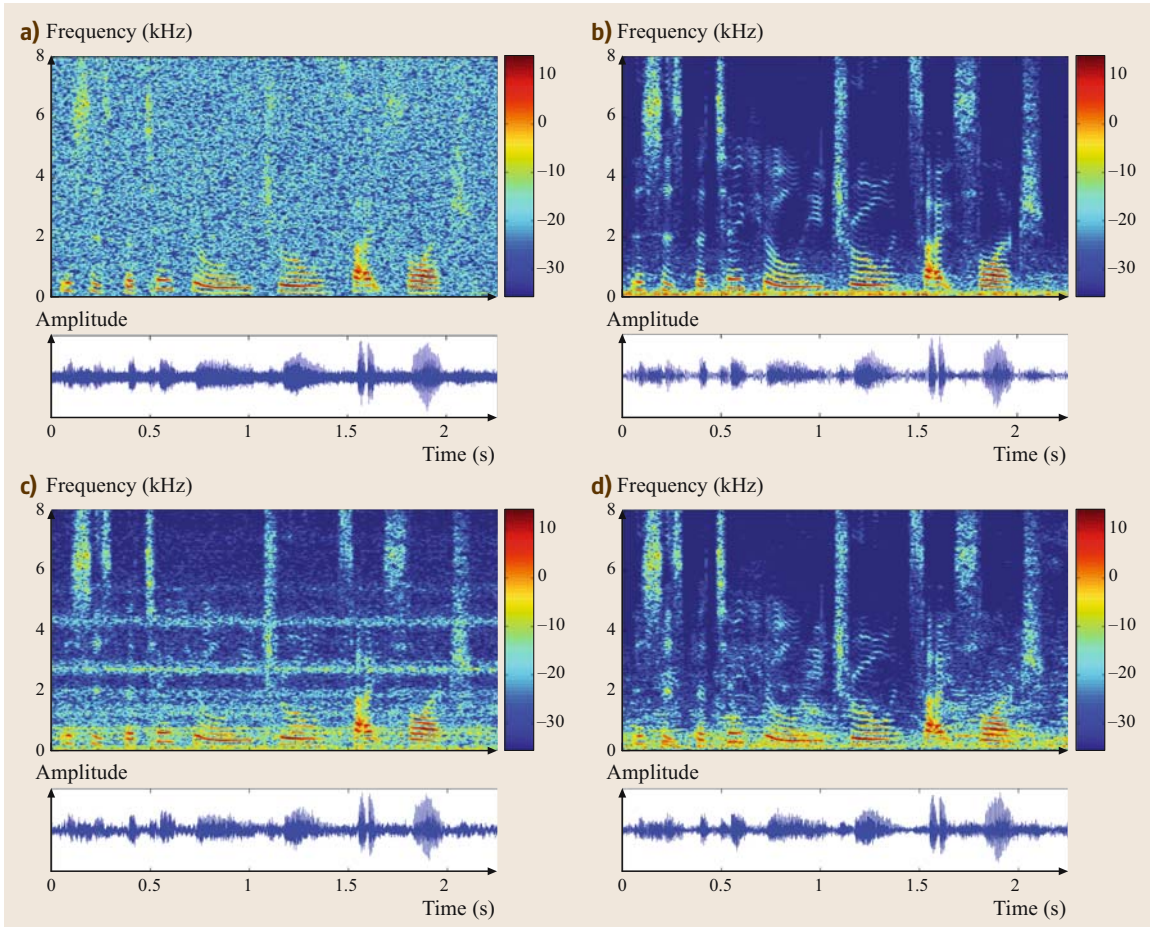


Fig. 44.13a–d Speech spectrograms and waveforms of the speech signal shown in Fig. 44.12 degraded by various noise types with SNR = 5 dB. **(a)** White Gaussian noise (SegSNR = −0.46 dB, LSD = 12.67 dB, PESQ = 1.74); **(b)** car interior noise (SegSNR = 0.30 dB, LSD = 3.48 dB, PESQ = 2.47); **(c)** F16 cockpit noise (SegSNR = −0.33 dB, LSD = 7.99 dB, PESQ = 1.76); **(d)** babble noise (SegSNR = 0.09 dB, LSD = 5.97 dB, PESQ = 1.87)

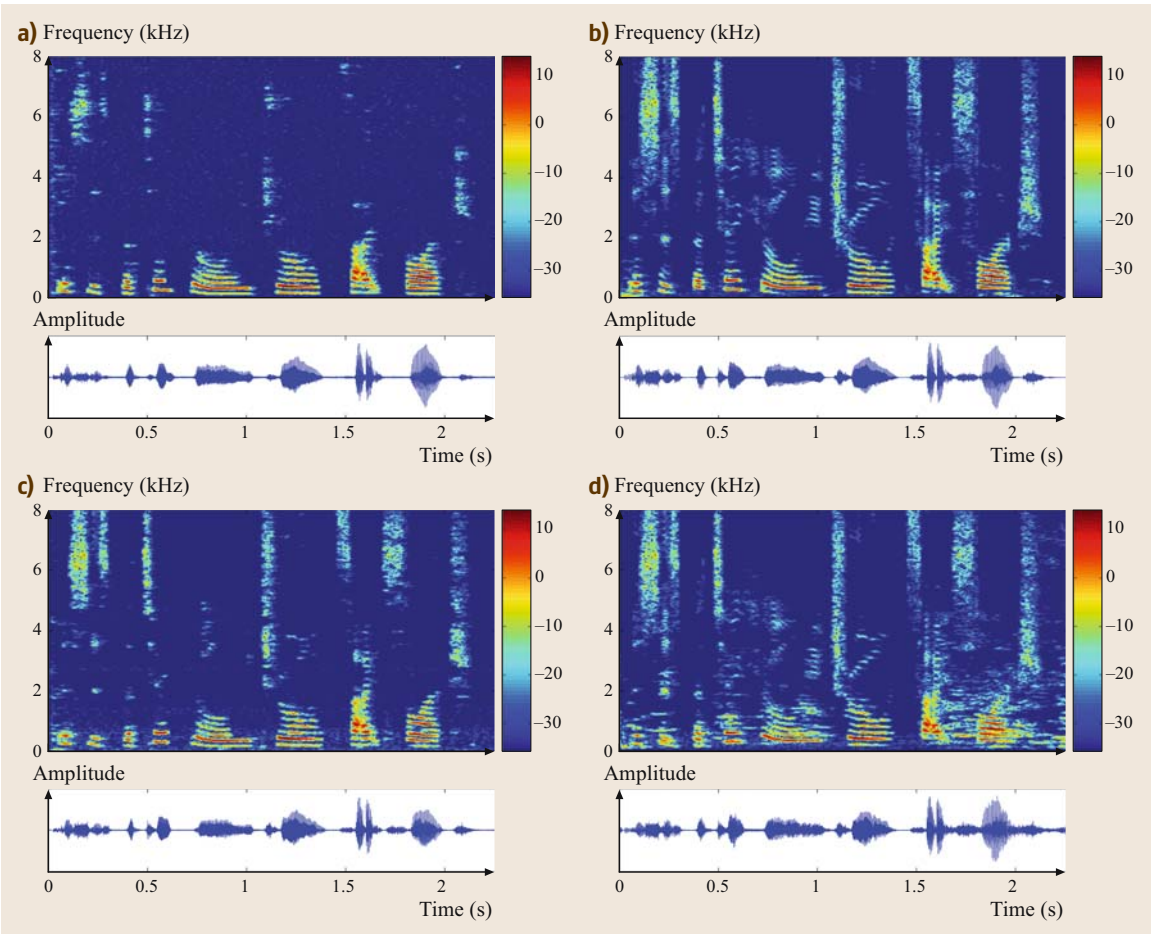


Fig. 44.14a–d Speech spectrograms and waveforms of the signals shown in Fig. 44.13 after enhancement with the algorithm in Table 44.5. **(a)** White Gaussian noise (SegSNR = 5.78 dB, LSD = 5.10 dB, PESQ = 2.34); **(b)** car interior noise (SegSNR = 9.52 dB, LSD = 2.67 dB, PESQ = 3.00); **(c)** F16 cockpit noise (SegSNR = 5.21 dB, LSD = 4.27 dB, PESQ = 2.29); **(d)** babble noise (SegSNR = 4.23 dB, LSD = 4.30 dB, PESQ = 2.13)

tion, that can be achieved by the speech enhancement system.

For car interior noise, most of the noise energy is concentrated in the lower frequencies. Therefore, the noise reduction is large in the low frequencies, and small in the high frequencies. Accordingly, in each frame, the total noise reduction is higher than that obtainable for the case of WGN. Since the SegSNR is mainly affected by the amount of noise reduction in each frame, the improvement in SegSNR is more significant for car interior noise. However, the reduction in LSD for the car interior noise is less substantial, since the initial LSD for the noisy signal is small.

The characteristics of babble noise vary more quickly in time when compared to the other noise

Table 44.6 Segmental SNR improvement, log-spectral distortion reduction and PESQ score improvement for various noise types, obtained by using the speech enhancement algorithm in Table 44.5

Noise type	SegSNR improvement	LSD reduction	PESQ score improvement
WGN	6.24	7.57	0.60
Car	9.22	0.81	0.53
F16	5.54	3.72	0.53
Babble	4.14	1.67	0.26

types. Therefore, the IMCRA noise spectrum estimator is least reliable for babble noise, and the speech enhancement performance is inferior to that achievable in slowly time-varying noise environments. Accordingly, the improvement in SegSNR is smaller for babble noise, when compared with the improvement achieved in more-stationary noise environment with the same initial SegSNR. Similarly, the reduction in LSD, and improvement in PESQ score, are smaller for babble noise, when compared with those achieved in more-stationary noise environment with the same initial LSD, respectively PESQ score, levels.

44.9 Selection of Spectral Enhancement Algorithms

In this section, we discuss some of the fundamental components that constitute a speech spectral enhancement system. Specifically, we address the choice of a statistical model, fidelity criterion, a priori SNR estimator, and noise spectrum estimator.

44.9.1 Choice of a Statistical Model and Fidelity Criterion

The Gaussian model underlies the design of many speech enhancement algorithms, e.g., [44.12, 17, 18, 42, 64–66]. This model is motivated by the central limit theorem, as each Fourier expansion coefficient is a weighted sum of random variables resulting from the random sequence [44.12]. When the span of correlation within the signal is sufficiently short compared to the size of the frames, the probability distribution function of the spectral coefficients asymptotically approaches Gaussian as the frame's size increases. The Gaussian approximation is in the central region of the Gaussian curve near the mean. However, the approximation can be very inaccurate in the tail regions away from the mean [44.67]. *Porter* and *Boll* [44.46] pointed out that a priori speech spectra do not have a Gaussian distribution, but gamma-like distribution. They proposed to compute the optimal estimator directly from the speech data, rather than from a parametric model of the speech statistics.

Martin [44.40] considered a gamma speech model, in which the real and imaginary parts of the clean speech spectral components are modeled as iid gamma random variables. He assumed that distinct spectral components are statistically independent, and derived MMSE estimators for the complex speech spectral coefficients under Gaussian and Laplacian noise modeling. He showed that, under Gaussian noise modeling, the gamma speech model yields a greater improvement in the segmental SNR than the Gaussian speech model. Under Laplacian noise modeling, the gamma speech model results in lower residual musical noise than the Gaussian speech model. *Martin* and *Breithaupt* [44.45] showed that when modeling the real and imaginary parts of the clean speech spectral components as Laplacian random variables, the MMSE estimators for the complex speech spectral coefficients have similar properties to those estimators derived under gamma modeling, but are easier to compute and implement.

Breithaupt and *Martin* [44.68] derived, using the same statistical modeling, MMSE estimators for the magnitude-squared spectral coefficients, and compared their performance to that obtained by using a Gaussian speech model. They showed that improvement in the segmental SNR comes at the expense of additional residual musical noise. *Lotter* and *Vary* [44.69] derived a maximum a posteriori (MAP) estimator for the speech spectral amplitude, based on a Gaussian noise model and a superGaussian speech model. They proposed a parametric pdf for the speech spectral amplitude, which approximates, with a proper choice of the parameters, the gamma and Laplacian densities. Compared with the MMSE spectral amplitude estimator of Ephraim-Malah, the MAP estimator with Laplacian speech modeling demonstrates improved noise reduction.

The Gaussian, gamma, and Laplacian models presented in Sect. 44.3 take into account the time correlation between successive speech spectral components. Spectral components in the STFT domain are assumed to be statistically correlated along the frequency axis, as well as along time trajectories, due to the finite length of the analysis frame in the STFT and the overlap between successive frames [44.15]. Experimental results of speech enhancement performance show [44.16, 43] that the appropriateness of the Gaussian, gamma, and Laplacian speech models are greatly affected by the particular choice of the a priori SNR estimator. When the a priori SNR is estimated by the decision-directed method, the gamma model is more advantageous than the Gaussian model. However, when the a priori SNR is estimated by the noncausal recursive estimation method, the Laplacian speech model yields a higher segmental SNR and a lower LSD than the other speech models,

while the level of residual musical noise is minimal when using a Gaussian speech model. Furthermore, the differences between the Gaussian, gamma, and Laplacian speech models are smaller when using the noncausal a priori estimators than when using the decision-directed method.

It is worthwhile noting that estimators that minimize the MSE distortion of the spectral amplitude or log-spectral amplitude are more suitable for speech enhancement than MMSE estimators. Moreover, it is difficult, or even impossible, to derive analytical expressions for MMSE-LSA estimators under gamma or Laplacian models. Therefore, the MMSE-LSA estimator derived under a Gaussian model is often preferred over the MMSE estimators derived under the other speech models [44.2].

44.9.2 Choice of an A Priori SNR Estimator

Ephraim and *Malah* [44.12, 70] proposed three different methods for the a priori SNR estimation. First, maximum-likelihood estimation, which relies on the assumption that the speech spectral variances are slowly time-varying parameters. This results in musical residual noise, which is annoying and disturbing to the perception of the enhanced signal. Second, decision-directed approach which is particularly useful when combined with the MMSE spectral, or log-spectral, magnitude estimators [44.12, 17, 47]. This results in perceptually colorless residual noise, but is heuristically motivated and its theoretical performance is unknown due to its highly nonlinear nature. Third, maximum a posteriori estimation, which relies on a first-order Markov model for generating a sequence of speech spectral variances. It involves a set of nonlinear equations, which are solved recursively by using the Viterbi algorithm. The computational complexity of the MAP estimator is relatively high, while it does not provide a significant improvement in the enhanced speech quality over the decision-directed estimator [44.70].

The decision-directed approach has become over the last two decades the most acceptable estimation method for the variances of the speech spectral coefficients. However, the parameters of the decision-directed estimator have to be determined by simulations and subjective listening tests for each particular setup of time–frequency transformation and speech enhancement algorithm. Furthermore, since the decision-directed approach is not supported by a statistical model, the parameters are not adapted to the speech components, but are set to specific values in advance. *Ephraim* and *Malah* recognized the limits of their variance estimation methods, and concluded that better speech enhancement performance may be obtained if the variance estimation could be improved [44.12, 70].

The causal estimator for the a priori SNR combines two steps, a *propagation* step and an *update* step, following the rational of Kalman filtering, to predict and update the estimate for the speech spectral variance recursively as new data arrive. The causal a priori SNR estimator is closely related to the decision-directed estimator of Ephraim and Malah. A special case of the causal estimator degenerates to a *decision-directed* estimator with a *time-varying frequency-dependent* weighting factor. The weighting factor is monotonically decreasing as a function of the instantaneous SNR, resulting effectively in a larger weighting factor during speech absence, and a smaller weighting factor during speech presence. This slightly reduces both the musical noise and the signal distortion. Nevertheless, the improvement in speech enhancement performance obtained by using the causal recursive over using the decision-directed method is not substantial. Therefore, if the delay between the enhanced speech and the noisy observation needs to be minimized, the decision-directed method is perhaps preferable due to its computational simplicity. However, in applications where a few-frames delay is tolerable, e.g., digital voice recording, surveillance, and speaker identification, the *noncausal* recursive estimation approach is more advantageous than the decision-directed approach.

The noncausal a priori SNR estimator employs future spectral measurements to predict the spectral variances of clean speech better. A comparison of the causal and noncausal estimators indicates that the differences are primarily noticeable during speech onset. The *causal* a priori SNR estimator, as well as the decision-directed estimator, cannot respond too quickly to an abrupt increase in the instantaneous SNR, since this necessarily implies an increase in the level of musical residual noise. In contrast, the *noncausal* estimator, having a few subsequent spectral measurements at hand, is capable of discriminating between speech onsets and noise irregularities. Experimental results show that the advantages of the noncausal estimator are particularly perceived during onsets of speech and noise only frames. Onsets of speech are better preserved, while a further reduction of musical noise is achieved [44.15, 53]. Furthermore, the differences between the Gaussian, gamma, and Laplacian speech models are smaller when using the noncausal recursive estimation approach than when using the decision-directed method [44.43].

44.9.3 Choice of a Noise Estimator

Traditional noise estimation methods are based on recursive averaging during sections that do not contain speech and holding the estimates during sections which contain speech. However, these methods generally require VADs and the update of the noise estimate is restricted to periods of speech absence. Additionally, VADs are difficult to tune and their reliability severely deteriorates for weak speech components and low input SNR [44.65, 71, 72]. Alternative techniques, based on histograms in the power spectral domain [44.55, 73, 74], are computationally expensive, require much more memory resources, and do not perform well in low-SNR conditions. Furthermore, the signal segments used for building the histograms are typically several hundred milliseconds long, and thus the update rate of the noise estimate is essentially moderate.

A useful noise estimation approach known as the minimum statistics [44.59] tracks the minima values of a smoothed power estimate of the noisy signal, and multiplies the result by a factor that compensates the bias. However, the variance of this noise estimate is about twice as large as the variance of a conventional noise estimator [44.59]. Moreover, this method may occasionally attenuate low-energy phonemes, particularly if the minimum search window is too short [44.75]. These limitations can be overcome, at the price of higher complexity, by adapting the smoothing parameter and the bias compensation factor in time and frequency [44.56].

A computationally efficient minimum tracking scheme is presented in [44.57]. Its main drawbacks are the slow update rate of the noise estimate in the case of a sudden rise in the noise energy level, and its tendency to cancel the signal [44.71]. Other closely related techniques are the *lower-energy envelope tracking* [44.55] and the *quantile-based* [44.76] estimation methods. Rather than picking the minima values of a smoothed periodogram, the noise is estimated based on a temporal quantile of a nonsmoothed periodogram of the noisy signal. Unfortunately, these methods suffer from the high computational complexity associated with the sorting operation, and the extra memory required for keeping past spectral power values.

The IMCRA noise estimator [44.54], presented in Sect. 44.7, combines the robustness of the minimum tracking with the simplicity of recursive averaging. Rather than employing a voice activity detector and restricting the update of the noise estimator to periods of speech absence, the smoothing parameter is adapted in time and frequency according to the speech presence probability. The noise estimate is thereby continuously updated even during weak speech activity. The estimator is controlled by the minima values of a smoothed periodogram of the noisy measurement. It combines conditions on both the instantaneous and local measured power, and provides a soft transition between speech absence and presence. This prevents an occasional increase in the noise estimate during speech activity. Furthermore, carrying out the smoothing and minimum tracking in two iterations allows larger smoothing windows and smaller minimum search windows, while reliably tracking the minima even during strong speech activity. This yields a reduced variance of the minima values and shorter delay when responding to a rising noise power, which eventually improves the tracking capability of the noise estimator. In nonstationary noise environments and under low-SNR conditions, the IMCRA approach is particularly useful [44.54]. It facilitates a lower estimation error, and when integrated into a speech enhancement system, yields improved speech quality and lower residual noise.

44.10 Conclusions

We have described statistical models for speech and noise signals in the STFT domain, and presented estimators for the speech spectral coefficients under speech presence uncertainty. The statistical models take into consideration the time correlation between successive spectral components of the speech signal. The spectral estimators involve estimation of the noise power spectrum, calculation of the speech presence probability, and evaluation of the a priori SNR under speech presence uncertainty. We discussed the behavior of the MMSE-LSA spectral gain function and its advantage for the mechanism that counters the musical noise phenomenon. Local bursts of the a posteriori SNR during noise-only frames are pulled down to the average noise level, thus avoiding local buildup of noise whenever it exceeds its average characteristics. The estimator for the a priori speech presence probability exploits the strong correlation of speech presence in neighboring frequency bins of consecutive frames, which enables further attenuation of noise components while avoid-

ing clipping of speech onsets and misdetection of weak speech tails.

We have presented estimators for the a priori SNR under speech presence uncertainty, and showed that a special case of a causal recursive estimator degenerates to a decision-directed estimator with a time-varying frequency-dependent weighting factor. Furthermore, in applications where a delay of a few short-term frames between the enhanced speech and the noisy observation is tolerable, a noncausal estimation approach may produce less signal distortion and less musical residual noise than a causal estimation approach.

We described the IMCRA approach for the noise power spectrum estimation, and provided a detailed example of a speech enhancement algorithm. We showed that the improvement in SegSNR and reduction in LSD are influenced by the variability of the noise characteristics in time and the initial SegSNR and LSD of the noisy signal. The faster the noise spectrum varies in time, the less reliable the noise spectrum estimator, and consequently the lower the quality gain that can be achieved by the speech enhancement system. Furthermore, the lower the initial quality of the noisy signal, the larger the improvement that can be achieved by the speech enhancement system.

References

44.1 J. Benesty, S. Makino, J. Chen (Eds.): *Speech Enhancement* (Springer, Berlin, Heidelberg 2005)

44.2 Y. Ephraim, I. Cohen: Recent advancements in speech enhancement. In: *The Electrical Engineering Handbook, Circuits, Signals, and Speech and Image Processing*, 3rd edn., ed. by R.C. Dorf (CRC, Boca Raton 2006) pp. 15–12–15–26, Chap. 15

44.3 Y. Ephraim, H. Lev-Ari, W.J.J. Roberts: A brief survey of speech enhancement. In: *The Electronic Handbook*, 2nd edn. (CRC-Press, Boca Raton 2005)

44.4 S.F. Boll: Suppression of acoustic noise in speech using spectral subtraction, IEEE Trans. Acoust. Speech Signal Process. **27**(2), 113–120 (1979)

44.5 J.S. Lim, A.V. Oppenheim: Enhancement and bandwidth compression of noisy speech, Proc. IEEE **67**(12), 1586–1604 (1979)

44.6 M. Berouti, R. Schwartz, J. Makhoul: Enhancement of speech corrupted by acoustic noise, Proc. 4th ICASSP **79**, 208–211 (1979)

44.7 Z. Goh, K.-C. Tan, T.G. Tan: Postprocessing method for suppressing musical noise generated by spectral subtraction, IEEE Trans. Speech Audio Process. **6**(3), 287–292 (1998)

44.8 B.L. Sim, Y.C. Tong, J.S. Chang, C.T. Tan: A parametric formulation of the generalized spectral subtraction method, IEEE Trans. Speech Audio Process. **6**(4), 328–337 (1998)

44.9 H. Gustafsson, S.E. Nordholm, I. Claesson: Spectral subtraction using reduced delay convolution and adaptive averaging, IEEE Trans. Speech Audio Process. **9**(8), 799–807 (2001)

44.10 D.E. Tsoukalas, J.N. Mourjopoulos, G. Kokkinakis: Speech enhancement based on audible noise suppression, IEEE Trans. Speech Audio Process. **5**(6), 497–514 (1997)

44.11 N. Virag: Single channel speech enhancement based on masking properties of the human auditory system, IEEE Trans. Speech Audio Process. **7**(2), 126–137 (1999)

44.12 Y. Ephraim, D. Malah: Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator, IEEE Trans. Acoust. Speech Signal Process. **ASSP-32**(6), 1109–1121 (1984)

44.13 D. Burshtein, S. Gannot: Speech enhancement using a mixture-maximum model, IEEE Trans. Speech Audio Process. **10**(6), 341–351 (2002)

44.14 R. Martin: Speech enhancement based on minimum mean-square error estimation and supergaussian priors, IEEE Trans. Speech Audio Process. **13**(5), 845–856 (2005)

44.15 I. Cohen: Relaxed statistical model for speech enhancement and a priori SNR estimation, IEEE Trans. Speech Audio Process. **13**(5), 870–881 (2005)

44.16 I. Cohen: Speech spectral modeling and enhancement based on autoregressive conditional heteroscedasticity models, Signal Process. **86**(4), 698–709 (2006)

44.17 Y. Ephraim, D. Malah: Speech enhancement using a minimum mean-square error log-spectral amplitude estimator, IEEE Trans. Acoust. Speech Signal Process. **ASSP-33**(2), 443–445 (1985)

44.18 P.J. Wolfe, S.J. Godsill: Efficient alternatives to the Ephraim and Malah suppression rule for audio signal enhancement, Special Issue EURASIP JASP Digital Audio Multim. Commun. **2003**(10), 1043–1051 (2003)

44.19 P.C. Loizou: Speech enhancement based on perceptually motivated bayesian estimators of the magnitude spectrum, IEEE Trans. Speech Audio Process. **13**(5), 857–869 (2005)

44.20 B.H. Juang, L.R. Rabiner: Mixture autoregressive hidden Markov models for speech signals, IEEE Trans. Acoust. Speech Signal Process. **ASSP-33**(6), 1404–1413 (1985)

44.21 Y. Ephraim: Statistical-model-based speech enhancement systems, Proc. IEEE **80**(10), 1526–1555 (1992)

Amazon EX1010 Page 903

44.22 H. Sheikhzadeh, L. Deng: Waveform-based speech recognition using hidden filter models: Parameter selection and sensitivity to power normalization, IEEE Trans. Speech Audio Process. **2**, 80–91 (1994)

44.23 Y. Ephraim, N. Merhav: Hidden Markov processes, IEEE Trans. Inform. Theory **48**(6), 1518–1568 (2002)

44.24 C.J. Wellekens: Explicit time correlations in hidden Markov models for speech recognition, Proc. 12th ICASSP **87**, 384–386 (1987)

44.25 H. Sameti, H. Sheikhzadeh, L. Deng, R.L. Brennan: HMM-based strategies for enhancement of speech signals embedded in nonstationary noise, IEEE Trans. Speech Audio Process. **6**(5), 445–455 (1998)

44.26 L.R. Rabiner, B.-H. Juang: *Fundamentals of Speech Recognition* (Prentice-Hall, Upper Saddle River 1993)

44.27 F. Jelinek: *Statistical Methods for Speech Recognition* (MIT Press, Cambridge 1998)

44.28 Y. Ephraim, H.L.V. Trees: A signal subspace approach for speech enhancement, IEEE Trans. Speech Audio Process. **3**(4), 251–266 (1995)

44.29 F. Asano, S. Hayamizu, T. Yamada, S. Nakamura: Speech enhancement based on the subspace method, IEEE Trans. Speech Audio Process. **8**(5), 497–507 (2000)

44.30 U. Mittal, N. Phamdo: Signal/noise KLT based approach for enhancing speech degraded by colored noise, IEEE Trans. Speech Audio Process. **8**(2), 159–167 (2000)

44.31 Y. Hu, P.C. Loizou: A generalized subspace approach for enhancing speech corrupted by colored noise, IEEE Trans. Speech Audio Process. **11**(4), 334–341 (2003)

44.32 S.H. Jensen, P.C. Hansen, S.D. Hansen, J.A. Sørensen: Reduction of broad-band noise in speech by truncated QSVD, IEEE Trans. Speech Audio Process. **3**(6), 439–448 (1995)

44.33 S. Doclo, M. Moonen: GSVD-based optimal filtering for single and multimicrophone speech enhancement, IEEE Trans. Signal Process. **50**(9), 2230–2244 (2002)

44.34 F. Jabloun, B. Champagne: Incorporating the human hearing properties in the signal subspace approach for speech enhancement, IEEE Trans. Speech Audio Process. **11**(6), 700–708 (2003)

44.35 Y. Hu, P.C. Loizou: A perceptually motivated approach for speech enhancement, IEEE Trans. Speech Audio Process. **11**(5), 457–465 (2003)

44.36 J. Wexler, S. Raz: Discrete Gabor expansions, Speech Process. **21**(3), 207–220 (1990)

44.37 R.E. Crochiere, L.R. Rabiner: *Multirate Digital Signal Processing* (Prentice-Hall, Englewood Cliffs 1983)

44.38 J.S. Garofolo: *Getting Started with the DARPA TIMIT CD-ROM: An Acoustic Phonetic Continuous Speech Database* (NIST, Gaithersburg 1988)

44.39 A. Stuart, J.K. Ord: *Kendall's Advanced Theory of Statistics*, Vol. 1, 6th edn. (Arnold, London 1994)

44.40 R. Martin: Speech enhancement using MMSE short time spectral estimation with Gamma distributed speech priors, Proc. 27th ICASSP **02**, 253–256 (2002)

44.41 I. Cohen: Modeling speech signals in the time-frequency domain using GARCH, Signal Process. **84**(12), 2453–2459 (2004)

44.42 I. Cohen, B. Berdugo: Speech enhancement for non-stationary noise environments, Signal Process. **81**(11), 2403–2418 (2001)

44.43 I. Cohen: Speech enhancement using supergaussian speech models and noncausal a priori SNR estimation, Speech Commun. **47**(3), 336–350 (2005)

44.44 I.S. Gradshteyn, I.M. Ryzhik: *Table of Integrals, Series, and Products*, 4th edn. (Academic Press, New York 1980)

44.45 R. Martin, C. Breithaupt: Speech enhancement in the DFT domain using Laplacian speech priors. In: *Proc. 8th Int. Workshop on Acoustic Echo and Noise Control* (Kyoto, Japan 2003) pp. 87–90

44.46 J. Porter, S. Boll: Optimal estimators for spectral restoration of noisy speech, Proc. ICASSP **84**, 18A.2.1–18A.2.4 (1984)

44.47 O. Cappé: Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor, IEEE Trans. Acoust. Speech Signal Process. **2**(2), 345–349 (1994)

44.48 P. Scalart, J. Vieira-Filho: Speech enhancement based on a priori signal to noise estimation, Proc. 21th ICASSP **96**, 629–632 (1996)

44.49 D. Malah, R.V. Cox, A.J. Accardi: Tracking speech-presence uncertainty to improve speech enhancement in non-stationary noise environments, Proc. 24th ICASSP **99**, 789–792 (1999)

44.50 I. Cohen: On speech enhancement under signal presence uncertainty, Proc. 26th ICASSP **2001**, 167–170 (2001)

44.51 I.Y. Soon, S.N. Koh, C.K. Yeo: Improved noise suppression filter using self-adaptive estimator of probability of speech absence, Signal Process. **75**(2), 151–159 (1999)

44.52 M. Marzinzik: *Noise reduction schemes for digital hearing aids and their use for the hearing impaired*, Ph.D. Thesis (Oldenburg University, Oldenburg 2000)

44.53 I. Cohen: Speech enhancement using a noncausal a priori SNR estimator, IEEE Signal Process. Lett. **11**(9), 725–728 (2004)

44.54 I. Cohen: Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging, IEEE Trans. Speech Audio Process. **11**(5), 466–475 (2003)

44.55 C. Ris, S. Dupont: Assessing local noise level estimation methods: Application to noise robust ASR, Speech Commun. **34**(1–2), 141–158 (2001)

44.56 R. Martin: Noise power spectral density estimation based on optimal smoothing and minimum statis-

Amazon EX1010 Page 904

tics, IEEE Trans. Speech Audio Process. **9**(5), 504–512 (2001)
44.57 G. Doblinger: Computationally efficient speech enhancement by spectral minima tracking in subbands, Proc. 4th Eurospeech **95**, 1513–1516 (1995)
44.58 S. Qian, D. Chen: Discrete Gabor transform, IEEE Trans. Signal Process. **41**(7), 2429–2438 (1993)
44.59 R. Martin: Spectral subtraction based on minimum statistics, Proc. 7th EUSIPCO **94**, 1182–1185 (1994)
44.60 A. Varga, H.J.M. Steeneken: Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems, Speech Commun. **12**(3), 247–251 (1993)
44.61 S.R. Quackenbush, T.P. Barnwell, M.A. Clements: *Objective Measures of Speech Quality* (Prentice-Hall, Englewood Cliffs 1988)
44.62 J.R. Deller, J.H.L. Hansen, J.G. Proakis: *Discrete-Time Processing of Speech Signals*, 2nd edn. (IEEE, New York 2000)
44.63 P.E. Papamichalis: *Practical Approaches to Speech Coding* (Prentice-Hall, Englewood Cliffs 1987)
44.64 A.J. Accardi, R.V. Cox: A modular approach to speech enhancement with an application to speech coding, Proc. 24th ICASSP **99**, 201–204 (1999)
44.65 J. Sohn, N.S. Kim, W. Sung: A statistical model-based voice activity detector, IEEE Signal Process. Lett. **6**(1), 1–3 (1999)
44.66 T. Lotter, C. Benien, P. Vary: Multichannel speech enhancement using bayesian spectral amplitude estimation, Proc. 28th ICASSP **03**, 832–835 (2003)
44.67 J.W.B. Davenport: *Probability and Random Processes: An Introduction for Applied Scientists and Engineers* (McGraw-Hill, New York 1970)
44.68 C. Breithaupt, R. Martin: MMSE estimation of magnitude-squared DFT coefficients with supergaussian priors, Proc. 28th ICASSP **03**, 896–899 (2003)
44.69 T. Lotter, P. Vary: Noise reduction by maximum a posteriori spectral amplitude estimation with supergaussian speech modeling. In: *Proc. 8th Internat. Workshop on Acoustic Echo and Noise Control* (2003) pp. 83–86
44.70 Y. Ephraim, D. Malah: *Signal to Noise Ratio Estimation for Enhancing Speech Using the Viterbi Algorithm*, Tech. Rep. EE PUB 489 (Technion – Israel Institute of Technology, Haifa 1984)
44.71 J. Meyer, K.U. Simmer, K.D. Kammeyer: Comparison of one- and two-channel noise-estimation techniques, Proc. 5th IWAENC **97**, 137–145 (1997)
44.72 B.L. McKinley, G.H. Whipple: Model based speech pause detection, Proc. 22th ICASSP **97**, 1179–1182 (1997)
44.73 R.J. McAulay, M.L. Malpass: Speech enhancement using a soft-decision noise suppression filter, IEEE Trans. Acoust. Speech Signal Process. **ASSP-28**(2), 137–145 (1980)
44.74 H.G. Hirsch, C. Ehrlicher: Noise estimation techniques for robust speech recognition, Proc. 20th ICASSP **95**, 153–156 (1995)
44.75 I. Cohen, B. Berdugo: Speech enhancement for non-stationary noise environments, Signal Process. **81**(11), 2403–2418 (2001)
44.76 V. Stahl, A. Fischer, R. Bippus: Quantile based noise estimation for spectral subtraction and Wiener filtering, Proc. 25th ICASSP **2000**, 1875–1878 (2000)

Amazon EX1010 Page 905

45. Adaptive Echo Cancelation for Voice Signals

M. M. Sondhi

This chapter deals with modern methods of dealing with echoes that may arise during a telephone conversation. Echoes may be generated due to impedance mismatch at various points in a telephone connection, or they may be generated acoustically due to coupling between microphones and loudspeakers placed in the same room or enclosure. If unchecked, such echoes can seriously disrupt a conversation. The device most successful in dealing with such echoes is the adaptive echo canceler. Several million such devices are deployed in the telephone networks around the world. This chapter discusses the principles on which echo cancelers are based, and describes their applications to network telephony and to single- and multichannel teleconferencing.

45.1 **Network Echoes** 904
45.1.1 Network Echo Canceler 905
45.1.2 Adaptation Algorithms 906
45.2 **Single-Channel Acoustic Echo Cancelation** 915
45.2.1 The Subband Canceler 916
45.2.2 RLS for Subband Echo Cancelers 917
45.2.3 The Delayless Subband Structure ... 917
45.2.4 Frequency-Domain Adaptation 918
45.2.5 The Two-Echo-Path Model 919
45.2.6 Variable-Step Algorithm for Acoustic Echo Cancelers 920
45.2.7 Cancelers for Nonlinear Echo Paths 920
45.3 **Multichannel Acoustic Echo Cancelation** .. 921
45.3.1 Nonuniqueness of the Misalignment Vector 923
45.3.2 Solutions for the Nonuniqueness Problem 924
45.4 **Summary** 925
References 926

Echoes are a ubiquitous phenomenon. With rare exceptions, all conversations take place in the presence of echoes. We hear echoes of our speech waves as they are reflected from the floor, walls, and other neighboring objects. If a reflected wave arrives at our ears a very short time after the direct sound, it is perceived not as an echo but as a spectral distortion or reverberation. In general, people prefer some amount of reverberation to a completely anechoic environment. The desirable amount of reverberation depends on the application. (For example, we like more reverberation in a concert hall than in an office.) The situation is very different, however, if a reflected wave arrives a few tens of milliseconds after the direct sound. In this case it is heard as a distinct echo. Such echoes are invariably annoying. Under extreme conditions they can completely disrupt a conversation.

If several reflections arrive in close succession after a long enough delay, the impression is that of a distinct echo with a spectral distortion. It is such, (in general) spectrally distorted, distinct echoes that are the subject of this chapter. Such echoes are of concern in telephone communications because they can be generated *electrically* at points of impedance mismatch along the transmission medium. Such echoes are called *line* or *network* echoes. In a telephone connection between two handsets these are the only type of echoes encountered. In local telephone connections echoes are not a problem because echo levels are low, and if present, they occur after very short delays. However, in a long-distance connection, if the round-trip delay exceeds a few tens of milliseconds, distinct echoes may be heard. The most important source of these echoes is a device called a hybrid, which we will discuss in Sect. 45.1. Hybrids are required whenever a telephone connection uses local networks as well as the long-distance network.

Echoes at hybrids have been a potential source of degradation ever since the advent of long-distance telephony. Many ways of dealing with them have been devised over the past several decades. However, those methods proved to be inadequate for telephone communications via geostationary satellites because of the long round-trip delays – 600 ms or more – encountered on satellite circuits. To deal with echoes arriving with such long delays a new device called an *echo canceler* was developed [45.1–3]. Widespread use of echo cancelers

Amazon EX1010 Page 906

began around 1980 with the arrival of a very large-scale integration (VLSI) implementation [45.4]. The advent of fiber optics obviated satellite delay. However, more recently, with the growing use of speech coding in the telecommunications network – especially for wireless telephony – delay has again become an issue. At present many millions of echo cancelers are in use in telephone networks around the world.

When a telephone connection is between hands-free telephones (speakerphones) or between two conference rooms, a major source of echoes is the acoustic coupling between loudspeakers and microphones via room reverberation. Adaptive cancelation of such echoes has attracted much attention during the past two decades, and is called *acoustic echo cancelation*. We will discuss this problem in Sect. 45.2. Note, however, that although the echoes are generated acoustically, we will discuss their cancelation only in the *electrical* portion of the circuit. We will not discuss the much more-difficult problem of canceling the echoes *acoustically*, i. e., by generating a canceling acoustic field. Strictly speaking, therefore, we should call our topic the cancelation of acoustically generated echoes. However, the term acoustic echo cancelation is firmly entrenched in the literature.

Finally, in recent years the problem of canceling echoes in stereo (or multichannel) communications has received considerable attention. Although stereophonic teleconferencing has not yet been extensively deployed, we believe that in the not too distant future such conferencing will become quite widespread. Therefore it is important to develop echo cancelers for stereophonic conferencing.

In the next three sections we will discuss network echo cancelers, acoustic echo cancelers, and stereo echo cancelers for voice signals. Note, however, that since their initial introduction for voice signals, echo cancelers were proposed for data signals as well, and have found wide application for such signals. However, the implementation and requirements of those cancelers are very different from those of cancelers for voice signals. We will not discuss echo cancelers for data signals in this chapter.

45.1 Network Echoes

As mentioned above, the main source of network echoes is a device called a hybrid. Figure 45.1 shows a highly simplified diagram showing the placement of hybrids in a typical long-distance telephone connection. Each of the telephones shown is connected to a central office by a two-wire line called the subscriber's loop. For a local call the two subscribers are connected to the same central office. So a local call is set up by simply connecting the two subscribers' loops at the central office. If the distance between the two telephones is long enough (usually, longer than about 35 miles) amplification becomes necessary. Therefore a separate path is needed for each direction of transmission. The device that connects the four-wire part of the circuit to the two-wire portion at each end is known as a hybrid. With reference to Fig. 45.1, the purpose of the hybrids is to allow signals from A to go along the path L_1 to B, and to go from B along the path L_2 to A. However, they must prevent signals in path L_1 from returning as an echo along the path L_2 back to A. Similarly, the signal in path L_2 is to be prevented from returning along path L_1 as an echo back to B.

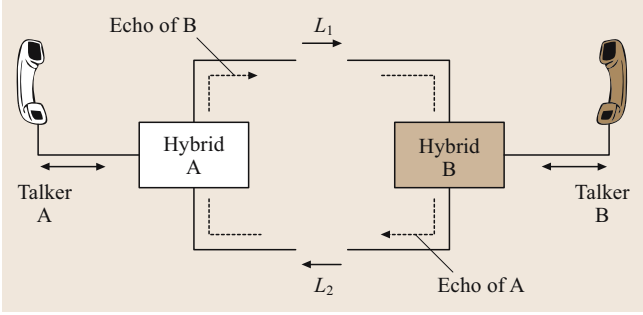


Fig. 45.1 Illustration of a long-distance connection showing local two-wire loops connected through hybrids to a four-wire long-distance network

A hybrid is essentially a Wheatstone bridge network that can achieve the objectives stated above, provided the impedance of the subscriber's loop can be exactly balanced by an impedance located at the hybrid. Unfortunately, this is not possible in practice for several reasons. One reason is that there are far fewer four-wire circuits than there are two-wire circuits. Therefore a hybrid may be connected to any of the subscriber loops served by the central office. By their very nature, subscribers' loops have a wide variety of characteristics – various lengths, type of wire, type of telephone, number of extension phones, etc. Therefore it is impossible to select a single impedance that would balance the

Amazon EX1010 Page 907

impedance of all subscribers' loops. Besides this, the echo path varies to some extent during a conversation, and can change quite significantly if extension phones are connected and/or disconnected. It appears, therefore, that the echo at the hybrid cannot be completely eliminated by merely selecting an appropriate balancing impedance. As a compromise, a nominal impedance is used to balance the bridge. The average attenuation (in the USA) from input to the return-path output of the hybrid is 11 dB with a standard deviation of 3 dB. The attenuation of the echo provided by the echo path is termed the echo return loss (ERL). An ERL of 9–14 dB is not adequate for satisfactory communication on circuits with long delays [45.5, 6].

For controlling such echoes, a device known as an echo suppressor has long been used – one at each end of the telephone connection. Its principle can be explained by referring to Fig. 45.2, which shows the end B of the telephone circuit of Fig. 45.1, with an echo suppressor included. Based on the level of signals in the paths L_1 and L_2, a decision is made as to whether the signal in L_2 is a signal from B or an echo of A's speech. If the decision is the latter, then the circuit L_2 is opened (or a large loss is switched in). A similar switch at the other end prevents B's echo from returning to B. During intervals when the speech signals from both A and B are simultaneously present at the echo suppressor, suppression is inhibited so that the signal returned to A is an echo of A superimposed on the speech from B. Such occurrence of simultaneous speech from both subscribers is whimsically referred to as *double-talk* in the literature.

If A and B would only talk alternately, it would be possible to design a very good decision mechanism. However, in a telephone conversation it is very important to provide the ability to *break in*, i. e., the ability of B to interrupt when A is talking, and vice versa. Without this ability the conversation would be quite unnatural. If the decision mechanism were to behave flawlessly in spite of break-in, the echo suppressor would be a satisfactory form of echo control. However, when break-in is allowed, the decision cannot be perfect. The two signals that have to be distinguished are both speech signals, with more or less the same statistical properties. Essentially the only distinguishing property is the signal level. For this reason, sometimes too high a level of echo is returned, and sometimes the interrupter's speech is clipped. In spite of these difficulties, echo suppressors have been designed that manage to keep such malfunctions at an acceptable level as long as the round-trip delay is not more than about 60–70 ms. Therefore before the introduction of satellite circuits and/or speech coders, echo suppressors provided a quite satisfactory solution to the echo problem.

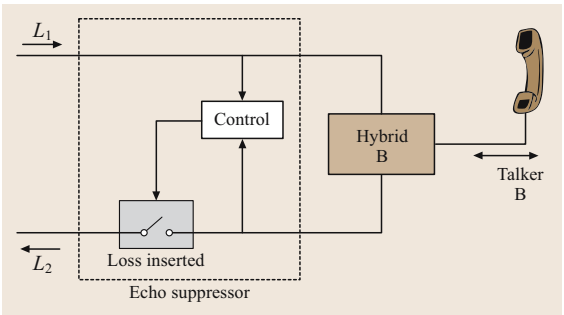


Fig. 45.2 Use of an echo suppressor to remove echo by inserting a switched loss when near-end speech (talker B) is not present

45.1.1 Network Echo Canceler

Echo cancelation is a very different method of echo control that was introduced during the 1960s. Instead of suppressing the echo by introducing a series loss in the circuit, an echo canceler generates a synthetic echo that is a replica of the real echo, and subtracts it from the signal in the echo path. The impetus for the development of echo cancelers came from the fact that echo suppressors failed to provide satisfactory echo control on satellite circuits with their associated long delays. In a circuit via a geostationary communications satellite the round-trip delay of the echo can be about 600 ms. Due to this long delay, the dynamics of a conversation changes, and erroneous switching by the echo suppressors becomes too frequent to be acceptable.

Figure 45.3 shows the same circuit as Fig. 45.2, but with the echo suppressor replaced by an echo

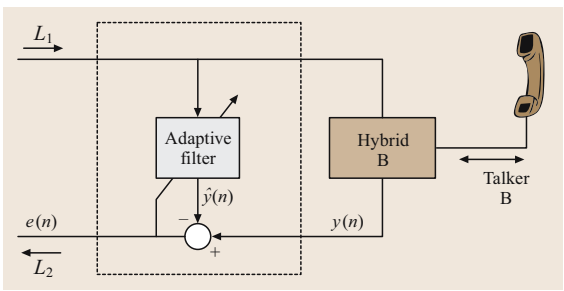


Fig. 45.3 An echo canceler continually removes echo even if the near-end talker is active

canceler. With respect to this echo canceler, the signal from A will be referred to as the far-end signal and the signal from B as the near-end signal. For the canceler located at end A, the roles of the near- and far-end signals are obviously reversed. The function of an echo canceler is to provide additional echo return loss – termed echo return loss enhancement (ERLE) – while allowing uninterrupted communication in both directions.

The box labeled *adaptive filter* in Fig. 45.3 generates a synthetic echo by configuring a filter with a transfer characteristic that matches that of the echo path. For a linear echo path that means matching the impulse response (or, equivalently, the transfer function). Note that the filter has to be *adaptive* in order to cope with the variability of the path due to the reasons mentioned in the previous section. The alternative would be to measure the impulse response periodically by introducing test signals at the input of the echo path. Such intrusions into a telephone conversation are not acceptable. Therefore the filter must adapt to the echo path by using just the existing speech signals. In the next section we will discuss several adaptation algorithms suitable for echo cancelers.

Before turning to the discussion of adaptation algorithms let us mention that a complete echo canceler requires two important features in addition to the adaptation algorithm.

1. During periods of *double-talk* (i. e., periods during which speech from both the near end and the far end are present simultaneously at the canceler) all known adaptation algorithms fail. Therefore an echo canceler must have a double-talk detector that senses this condition. Whenever this condition is detected, the adaptation algorithm is disabled by effectively opening the adaptation path shown in Fig. 45.3. Note that this *does not* interrupt the return path. It only disables adaptation for the (generally quite short) duration of simultaneous talking.
2. The other feature that is required is what is known in the literature as a nonlinear processor. The nonlinearity used is a center clipper. Its function is to set to zero all signals in the return path that are below a certain threshold level. This is necessary because in practice echo cancelers can provide only about 20–25 dB of cancelation. In general a greater degree of cancelation is required, and the center clipper provides that. Since the clipping threshold is very low, the effect of the nonlinear distortion on the near-end signal is not noticeable.

45.1.2 Adaptation Algorithms

The first step in implementing an adaptive filter is to select a structure for the filter such that it can be represented in terms of a finite number of parameters. For linear filters this can be done by modeling the impulse response of the filter as an expansion in terms of any complete set of basis vectors. If $w_l(n)$, $l = 0, 1, 2, \ldots, L-1$, is the set of basis functions, and h_l are the corresponding expansion coefficients, then we assume that the echo $y_e(n)$ is related to the input $x(n)$ by the relation

$$y_e(n) = x(n) * \sum_{l=0}^{L-1} h_l w_l(n) = \sum_{l=0}^{L-1} h_l x_l(n) = \boldsymbol{h}^T \boldsymbol{x}(n) , \quad (45.1)$$

where $*$ indicates convolution $x_l(n) = x(n) * w_l(n)$ and the boldface characters are column vectors of dimension L. The superscript 'T' indicates matrix transpose, and n is the time index. Thus this structure is equivalent to passing the input signal through a bank of L parallel filters with impulse responses $w_l(n)$ and forming a linear combination of the outputs to generate the echo.

The problem thus reduces to the estimation of the expansion coefficient vector $\boldsymbol{h}$. This is done by iteratively adjusting the estimate vector $\hat{\boldsymbol{h}}$ of an adaptive filter so as to drive it towards $\boldsymbol{h}$.

The most popular choice of basis functions is $w_l(n) = \delta(n-l)$, and the resulting filter structure is called a transversal filter (also referred to as a tapped delay line filter or a moving average filter). Its output is just a linear combination of the most recent L samples of the input. For speech signals on the telephone network the sampling frequency $f_s = 8000$ samples per second is used. This is the Nyquist rate for a signal of bandwidth 4 kHz, which is more than adequate for telephone-quality speech whose bandwidth is about 3.4 kHz. At this sampling rate, L is usually chosen to be about 800. For acoustic cancelers, as we shall see, L might have to be chosen to be several thousand.

The adaptation algorithms that we will discuss are strongly dependent on the fact that the output of the adaptive filter is linear in the adapted coefficient vector $\hat{\boldsymbol{h}}$. We will present the adaptation algorithms using the transversal filter as a model. However, except in a few cases, the discussion will be independent of the choice of the component filters w_l. In fact the discussion is valid even if the w_l represent nonlinear filters or filters with infinitely long impulse responses. The actual performance will, of course, depend on the choice of the filter set.

Amazon EX1010 Page 909

The Stochastic Gradient Algorithm

By far the most popular adaptation algorithm used for echo cancelation is the stochastic gradient algorithm, also popularly known as the least-mean-square (LMS) algorithm. This algorithm was initially introduced around 1960 for adaptive switching [45.7]. It was first used for echo cancelation [45.1] and for adaptive antenna arrays [45.8] in the mid 1960s. Since then it has been used in a variety of applications such as noise cancelation, equalization, etc.

Let the signal $y(n)$ returned from the hybrid be given by

$$y(n) = y_e(n) + v(n) , \tag{45.2}$$

where $y_e(n) = \boldsymbol{h}^T \boldsymbol{x}(n)$ is the echo of the far-end signal $x(n)$, and $v(n)$ is the near-end signal of talker B plus any circuit noise. We will assume that the echo path impulse response $\boldsymbol{h}$ has the form given in (45.1). If that is not strictly true, then the term $v(n)$ will also include the modeling error. Let $\hat{\boldsymbol{h}}(n)$ be an estimate of $\boldsymbol{h}$ at time n and $\hat{y}(n) = \hat{\boldsymbol{h}}^T(n)\boldsymbol{x}(n)$ the corresponding estimate of $y(n)$. We wish to update $\hat{\boldsymbol{h}}(n)$ to an improved vector $\hat{\boldsymbol{h}}(n+1)$ by adding an increment $\boldsymbol{\Delta}$. Since $\boldsymbol{h}$ is unknown we must evaluate the goodness of the estimate indirectly. One measure of the performance of $\hat{\boldsymbol{h}}(n)$ is the error

$$e(n) = y(n) - \hat{y}(n) .$$

Since the objective is to make $\hat{\boldsymbol{h}}$ approximate $\boldsymbol{h}$, one might search for the vector that minimizes the expected value of the squared error $e^2(n)$. One way to do this is to move $\hat{\boldsymbol{h}}$ some distance in the direction of the negative gradient of this expected value. Thus we might choose

$$\boldsymbol{\Delta} = -\frac{\mu}{2} \nabla E[e^2(n)] , \tag{45.3}$$

where E denotes mathematical expectation, μ is the step size parameter that controls the rate of change, and ∇ is the gradient with respect to $\hat{\boldsymbol{h}}(n)$. This update corresponds to the update of the LMS algorithm if the *expected value* of e^2 is replaced by the *instantaneous* value. This seems to be a drastic approximation. However, as we shall see, even such a crude estimate of the gradient is adequate, under certain reasonable conditions, to make $\hat{\boldsymbol{h}}(n)$ approach $\boldsymbol{h}$. Making this approximation and evaluating the gradient in (45.3) yields the LMS update

$$\hat{\boldsymbol{h}}(n+1) = \hat{\boldsymbol{h}}(n) + \mu e(n)\boldsymbol{x}(n) . \tag{45.4}$$

Rather than minimizing e^2 one could minimize some other symmetric function of e that has an odd monotonic derivative. That results in a slight generalization of (45.4) to

$$\hat{\boldsymbol{h}}(n+1) = \hat{\boldsymbol{h}}(n) + \mu F[e(n)]\boldsymbol{x}(n) , \tag{45.5}$$

where F is any odd monotonic function of its argument. Two choices of the function F have been found useful in practice. The sign function was found useful in early analog implementations of the echo canceler [45.1] because it replaced analog multipliers by switches. However, it slows down convergence and is therefore no longer used. The other function is the ideal limiter with an adaptive threshold, which has been found useful to improve robustness of the adaptation. We will discuss this later.

Before discussing the convergence properties of the algorithm, let us point out that the update of (45.4) has one undesirable property. Observe that both $\boldsymbol{x}$ and e are proportional to the strength of the far-end signal $x(n)$. Thus the size of the update is proportional to the power of the far-end signal. Since the power of a speech signal has a large dynamic range, it follows that the rate at which the algorithm converges would be highly variable. To avoid this variability, the second term on the right-hand side of (45.4) is normalized by dividing it by the L_2 norm $\| \boldsymbol{x}(n) \|^2 = \boldsymbol{x}^T(n)\boldsymbol{x}(n)$, of the vector $\boldsymbol{x}(n)$. This results in the normalized LMS algorithm (NLMS):

$$\hat{\boldsymbol{h}}(n+1) = \hat{\boldsymbol{h}}(n) + \frac{\mu}{\| \boldsymbol{x}(n) \|^2} \boldsymbol{x}(n)e(n) . \tag{45.6}$$

We will discuss only this normalized form of the LMS algorithm, since the algorithm without normalization is almost never used. (The appropriate normalization for the algorithm of (45.5) has to be worked separately for each chosen function F.)

As a practical matter, a small positive constant δ is added to the denominator in (45.6), to avoid division by zero. So the final form of the NLMS algorithm is

$$\hat{\boldsymbol{h}}(n+1) = \hat{\boldsymbol{h}}(n) + \frac{\mu}{\| \boldsymbol{x}(n) \|^2 + \delta} \boldsymbol{x}(n)e(n) .$$

For the sake of simplicity of notation, we will not show the term δ in the discussion below.

Convergence in the Ideal Case. Consider the ideal case such that

1. the echo path impulse response $\boldsymbol{h}$ is constant, and
2. the error $e(n)$ consists of just the echo of the far-end signal (i. e., the case when the term $v(n)$ is absent in (45.2)).

In this ideal case it is easy to show that the update of (45.6) improves the estimate $\hat{\boldsymbol{h}}$, provided only that $0 <$

$\mu < 2$. To show this let us define the misalignment vector $\boldsymbol{r}(n)$ as the difference between $\boldsymbol{h}$ and $\hat{\boldsymbol{h}}$:

$$\boldsymbol{r}(n) = \boldsymbol{h} - \hat{\boldsymbol{h}}(n) \,. \tag{45.7}$$

In the ideal case

$$e(n) = \boldsymbol{x}^{\mathrm{T}}(n)\boldsymbol{r}(n) \,. \tag{45.8}$$

Subtracting $\boldsymbol{h}$ from both sides of (45.6) and rearranging terms we get

$$\boldsymbol{r}(n+1) = \boldsymbol{r}(n) - \frac{\mu}{\| \boldsymbol{x}(n) \|^2}\boldsymbol{x}(n)e(n) \,. \tag{45.9a}$$

Equation (45.9a) can also be written in an alternative useful form as

$$\boldsymbol{r}(n+1) = \left(\boldsymbol{I} - \mu\frac{\boldsymbol{x}(n)\boldsymbol{x}^{\mathrm{T}}(n)}{\boldsymbol{x}^{\mathrm{T}}(n)\boldsymbol{x}(n)}\right)\boldsymbol{r}(n) \,, \tag{45.9b}$$

where $\boldsymbol{I}$ is the identity matrix.

From (45.9a) it follows that

$$\| \boldsymbol{r}(n+1) \|^2 = \| \boldsymbol{r}(n) \|^2 + \mu(\mu-2)\frac{e^2(n)}{\| \boldsymbol{x}(n) \|^2} \,, \tag{45.10}$$

where we used (45.8) in evaluating the right-hand side. Equation (45.10) shows that, for $0 < \mu < 2$, $\| \boldsymbol{r}(n+1) \|^2 \leq \| \boldsymbol{r}(n) \|^2$, where the equality holds if and only if $e(n) = 0$. Thus as long as there is an uncanceled echo, the length of the misalignment vector decreases, i. e., $\hat{\boldsymbol{h}}$ moves closer to $\boldsymbol{h}$. This conclusion can also be drawn directly from (45.9b) when it is recognized that $(\boldsymbol{x}(n)\boldsymbol{x}^{\mathrm{T}}(n)/ \| \boldsymbol{x}(n) \|^2)\boldsymbol{r}(n)$ is an orthogonal projection of $\boldsymbol{r}(n)$ on $\boldsymbol{x}(n)$. The largest decrease in the length of $\boldsymbol{r}(n)$ occurs for $\mu = 1$. In practice, however, μ is usually set at about 0.3–0.5 to prevent instability due to the perturbations that were neglected in deriving (45.9a) and (45.9b).

Equation (45.10) can be used to show that $e^2(n) \to 0$ as $n \to \infty$. Note that summing the two sides of the equation for $0 \leq n \leq N-1$ and rearranging terms gives

$$\frac{\mu(2-\mu)}{\| \boldsymbol{x}(n) \|^2}\sum_{n=0}^{N-1} e^2(n) = \| \boldsymbol{r}(0) \|^2 - \| \boldsymbol{r}(N) \|^2 \,. \tag{45.11}$$

In view of (45.10), for $0 < \mu < 2$ the right-hand side of (45.11) is bounded, which implies that $e^2(N) \to 0$.

It is not enough, however, to make just the error go to zero. We would in fact like the misalignment vector $\boldsymbol{r}(n)$ to go to zero so that, once the echo canceler has converged, the echo should be canceled for all subsequent inputs. However, without further restrictions on the signal $x(n)$ (45.10) does not allow us to conclude that the misalignment $\boldsymbol{r}(n) \to 0$. This is because $e(n) = 0$ does not imply that $\boldsymbol{r}(n) = 0$, but only that $\boldsymbol{r}(n)$ is orthogonal to $\boldsymbol{x}(n)$. Intuitively speaking, therefore, convergence of $\boldsymbol{r}(n)$ requires that the time-varying vector $\boldsymbol{x}(n)$ is not confined to a subspace of dimension less than L for too long, i. e., $\boldsymbol{x}(n)$ should evolve so as to cover the entire L-dimensional space. This condition is referred to as a mixing condition [45.9] or *persistent excitation* [45.10, pp. 748–751].

Even if $\boldsymbol{x}(n)$ satisfies the mixing condition it is quite difficult to derive accurate estimates of the *rate* of convergence of $\boldsymbol{r}(n)$. Even if $x(n)$ is a member of a stationary ergodic process it is difficult to estimate the expected rate of convergence. Taking the expectation of the two sides of (45.10) gets us nowhere because the right-hand side depends on $\boldsymbol{r}$ itself. However, if μ is sufficiently small, it is reasonable to assume that $\boldsymbol{r}$ changes slowly and the expectation over the $\boldsymbol{x}$ ensemble can be taken assuming $\boldsymbol{r}$ to be quasiconstant [45.1]. This effectively assumes that $\boldsymbol{x}$ and $\boldsymbol{r}$ are statistically independent. Under this independence assumption some useful bounds can be derived for convergence rate when the far-end signal $x(n)$ is a stationary random process. Note that $e^2(n) = \boldsymbol{r}^{\mathrm{T}}(n)\boldsymbol{x}(n)\boldsymbol{x}^{\mathrm{T}}(n)\boldsymbol{r}(n)$. Therefore with the independence assumption, taking the expectation of (45.10) gives

$$\begin{aligned} &E[\| \boldsymbol{r}(n+1) \|^2] \\ &= E[\| \boldsymbol{r}(n) \|^2] + \mu(\mu-2) \\ &\quad \times E\left[\boldsymbol{r}^{\mathrm{T}}(n)E_x\left[\frac{\boldsymbol{x}(n)\boldsymbol{x}^{\mathrm{T}}(n)}{\boldsymbol{x}^{\mathrm{T}}(n)\boldsymbol{x}(n)}\right]\boldsymbol{r}(n)\right] \,, \end{aligned} \tag{45.12}$$

where E_x is the expectation over the x ensemble. For a stationary process one may make the approximation that for reasonably large L, $\boldsymbol{x}^{\mathrm{T}}(n)\boldsymbol{x}(n) \approx L\sigma^2$ where σ^2 is the variance of the x process. With this approximation (45.12) becomes

$$\begin{aligned} &E[\| \boldsymbol{r}(n+1) \|^2] \\ &= E[\| \boldsymbol{r}(n) \|^2] + \frac{\mu(\mu-2)}{L}E\left[\boldsymbol{r}^{\mathrm{T}}(n)\boldsymbol{R}_{xx}\boldsymbol{r}(n)\right] \,, \end{aligned} \tag{45.13}$$

where $\boldsymbol{R}_{xx}$ is the normalized correlation matrix of $\boldsymbol{x}$. Since the second term on the right-hand side of (45.13) can be bounded in terms of the largest and smallest

Amazon EX1010 Page 911

eigenvalues of the matrix $\boldsymbol{R}_{xx}$, we get

$$\left[1-\frac{\mu(2-\mu)\lambda_{\max}}{L}\right]^n \le \left[\frac{E(\| \boldsymbol{r}(n) \|^2)}{\| \boldsymbol{r}(0) \|^2}\right] \le \left[1-\frac{\mu(2-\mu)\lambda_{\min}}{L}\right]^n . \tag{45.14a}$$

For the range of values $0<\mu<2$, (45.14a) provides geometrically decreasing upper and lower bounds for $E[\| \boldsymbol{r}(n) \|^2]$. With the approximation $(1-x/L)^L \approx \mathrm{e}^{-x}$, the bounds given by (45.14a) are seen to be essentially exponential:

$$\exp\left[-\mu(2-\mu)\frac{n\lambda_{\max}}{L}\right] \le \left[\frac{E(\| \boldsymbol{r}(n) \|^2)}{\| \boldsymbol{r}(0) \|^2}\right] \le \exp\left[-\mu(2-\mu)\frac{n\lambda_{\min}}{L}\right] . \tag{45.14b}$$

In practice the bounds provided by (45.14a, 14b) are surprisingly good. There is little in the literature concerning convergence rates without the independence assumption; only some very loose bounds may be found in [45.9] and [45.11].

Convergence in the Nonideal Case. In practice the convergence process is much more complicated than that discussed above. First of all, the speech signal is a highly nonstationary signal with ill-defined statistical properties. Therefore the bounds on convergence derived above can be used only as a rough guide in designing a canceler. Besides that, there is additive noise and the echo path is not stationary. Very little can be said quantitatively about the effects of these perturbations on the convergence process. Some bounds can be derived [45.9] that show that, in the presence of such perturbations, the algorithm drives the vector $\boldsymbol{r}$ to within a sphere around the origin whose radius is proportional to the root-mean-squared value of the noise and to the rate of change of the impulse response.

The most serious perturbation occurs during periods of *double-talk*, i. e., during intervals when the speech from both the near-end and the far-end speakers is present simultaneously at the echo canceler. The near-end signal is an interference to the adaptation. If the echo canceler has converged to a small misalignment, the near-end signal can be very large compared to the uncanceled echo. In such a situation the adaptation algorithm will quickly misalign the canceler. As mentioned in Sect. 45.1.2, the only known way to deal with this problem is to detect periods of double-talk and disable adaptation during those intervals. The most commonly used double-talk detector is the so-called Geigel algorithm [45.12] that declares double-talk whenever

$$|y(n)| > \alpha \max_{n-L+1\le m\le n} |x(m)| , \tag{45.15}$$

where α is a threshold that is determined by the minimum echo return loss expected from the hybrid (usually set at $\alpha=0.5$, corresponding to a 6 dB hybrid loss). If double-talk is declared, the step-size parameter μ is set to 0. However, the signal e, which now contains the near-end speech along with the residual echo, is returned to the far end. Thus no interruption in the conversation takes place. Whenever double-talk is declared adaptation is interrupted for at least a specified minimum interval so as to avoid very rapid switching of the detector.

This simple algorithm works quite well for network echoes, where minimum hybrid return loss is generally well defined. The selection of α is not so straightforward for acoustic echo cancelers where the echo return loss may even be negative (i. e., the echo may be stronger than the far-end signal). We will discuss strategies employed for those cancelers in Sect. 45.2.5 when we discuss acoustic echo cancelers.

Another Derivation of the NLMS Algorithm. Although the NLMS algorithm is traditionally derived as a gradient algorithm, there is another interesting way of deriving the NLMS update that admits of other generalizations. To show that derivation, let us define an a posteriori error $\varepsilon(n)=y(n)-\hat{\boldsymbol{h}}^{\mathrm{T}}(n+1)\boldsymbol{x}(n)$, i. e., the error that would result if one used the updated estimate of $\hat{\boldsymbol{h}}$ with the current vector $\boldsymbol{x}$.

Obviously, unless all the components of $\boldsymbol{x}(n)$ are zero, it is always possible to find a vector $\hat{\boldsymbol{h}}(n+1)$ in a subspace of dimension $L-1$ such that $\varepsilon(n)=0$ or, more generally, such that $\varepsilon(n)=(1-\mu)e(n)$. If just any arbitrary vector is chosen in that subspace, clearly one learns nothing about the true echo path response $\boldsymbol{h}$. However, with $\hat{\boldsymbol{h}}(n+1)=\boldsymbol{h}(n)+\boldsymbol{\Delta}$, let us find the update $\boldsymbol{\Delta}$ with the smallest norm $\| \boldsymbol{\Delta} \|^2$ that makes $\varepsilon(n)=(1-\mu)e(n)$. Thus let us minimize $(1/2)\| \boldsymbol{\Delta} \|^2$ subject to the constraint $\varepsilon(n)=(1-\mu)e(n)$, or, equivalently the constraint

$$\boldsymbol{\Delta}^{\mathrm{T}}\boldsymbol{x}(n)=\mu e(n) . \tag{45.16}$$

The constrained minimization is equivalent to minimizing

$$\left(\frac{1}{2}\right) \| \boldsymbol{\Delta} \|^2 + \lambda[\boldsymbol{\Delta}^{\mathrm{T}}\boldsymbol{x}(n) - \mu e(n)] \,, \tag{45.17}$$

where λ is a Lagrange multiplier to be determined so as to satisfy the constraint (45.16). Setting the derivative of the expression (45.17) with respect to $\boldsymbol{\Delta}$ to zero gives

$$\boldsymbol{\Delta} = \lambda \boldsymbol{x}(n) \,.$$

Substituting this into (45.16) gives $\lambda = \frac{\mu e(n)}{[\boldsymbol{x}^{\mathrm{T}}(n)\boldsymbol{x}(n)]}$. Hence

$$\boldsymbol{\Delta} = \frac{\mu}{\| \boldsymbol{x}(n) \|^2} \boldsymbol{x}(n) e(n) \,, \tag{45.18}$$

which is precisely the NLMS update given in (45.6).

The PNLMS Algorithm

Until recently, the NLMS algorithm has been the exclusive choice for network echo cancelers. A new algorithm called the proportionate NLMS (PNLMS) algorithm has been introduced recently [45.13]. It significantly improves convergence rate if the echo path impulse response is *sparse*, i. e., if only a small subset of the L components of $\boldsymbol{h}$ are active, and the rest are zero. [In order to study the effectiveness of algorithms for sparse impulse responses, we need to define sparseness quantitatively. This can be done in several ways. For example, for an impulse response of length L, one may define sparseness S as $S = (L - \sum_{i=0}^{L-1} |h_i|/h_{\max})/(L-1)$. Reference [45.14] gives other possible definitions.] This is precisely the property that makes it useful for network cancelers. As mentioned in Sect. 45.1.2, the length $L$of the transversal filter $\hat{\boldsymbol{h}}$ is usually chosen to be quite large – several hundred taps. This is necessary in order to accommodate the range of impulse responses encountered in the network. However, any particular impulse response is active only at a small (but unknown) subset – a few tens – of the taps.

To motivate the derivation of the PNLMS algorithm, we note that the bounds of (45.14a, 14b) show that the rate at which the NLMS algorithm converges becomes slower as the number of adapted coefficients L increases. Therefore, if it were possible to identify and adapt only the active taps the convergence rate could be significantly improved for sparse echo paths. The PNLMS algorithm does the identification using the following intuitive reasoning. As the NLMS algorithm starts adapting, the coefficients for the inactive taps will stay around zero, while those corresponding to the active taps will build up in magnitude. Therefore the PNLMS algorithm makes the step-size parameter μ proportional to the current estimate of each coefficient. This is accomplished by use of a diagonal matrix $\boldsymbol{G}(n)$, whose diagonal elements $g_{ll}(n)$ are *essentially* the magnitudes of the components of the current vector $\hat{\boldsymbol{h}}(n)$, i. e., $g_{ll}(n) = |\hat{h}_l(n)|, l = 1, 2, \ldots, L$.

In terms of the matrix $\boldsymbol{G}(n)$, the PNLMS update is defined as

$$\hat{\boldsymbol{h}}(n+1) = \hat{\boldsymbol{h}}(n) + \frac{\mu}{\boldsymbol{x}^{\mathrm{T}}(n)\boldsymbol{G}(n)\boldsymbol{x}(n)} \boldsymbol{G}(n)\boldsymbol{x}(n)e(n) \,. \tag{45.19}$$

Since $\boldsymbol{G}(n)$ is diagonal, it follows that (45.19) effectively makes the step-size parameter for each component of $\hat{\boldsymbol{h}}(n)$ proportional to the magnitude of that component. If $\boldsymbol{G}(n)$ is chosen to be the identity matrix, of course, the algorithm reduces to the NLMS algorithm.

As in the case of the NLMS algorithm, a small constant δ is added to the denominator to avoid division by zero. Also, as a practical matter, the diagonal entries of $\boldsymbol{G}(n)$ must be modified somewhat from those given above, to avoid some singular behavior. Note, for instance, that if at any instant n a diagonal entry of $\boldsymbol{G}(n)$ becomes zero the corresponding coefficient will stop adapting. Indeed, if the vector $\hat{\boldsymbol{h}}(n) = 0$, (as is usually the case initially, at $n = 0$) no adaptation would take place. Also, components that are very small should still be adapted at some minimum rate. Finally, the average value of the diagonal elements should be held constant in order to be able to select a value of μ for stable operation. If the average value is set to unity, then μ can be chosen such that $0 < \mu < 2$ as for NLMS. To take these factors into account, the diagonal elements of $\boldsymbol{G}(n)$ are modified as

$$\begin{aligned}
\gamma_{\min} &= \rho \max\left\{\delta_p, |\hat{h}_1(n)|, |\hat{h}_2(n)|, \ldots, |\hat{h}_L(n)|\right\} \,, \\
\gamma_l(n) &= \max\left\{\gamma_{\min}, |\hat{h}_l(n)|\right\} \,, \quad 1 \le l \le L \,, \\
g_{ll}(n) &= \frac{\gamma_l(n)}{\frac{1}{L}\sum_{l=1}^{L} \gamma_l(n)} \,.
\end{aligned} \tag{45.20}$$

The parameter δ_p, typically chosen to be 0.01, prevents stalling if the vector $\hat{\boldsymbol{h}}(n) = 0$. The parameter $\gamma_{\min}$ provides a minimum adaptation rate for each coefficient. The parameter ρ (typically chosen to be about $5/L$) sets the value of $\gamma_{\min}$ relative to the largest adapted coefficient. Finally, the third line of (45.20) makes the average value of the diagonal elements equal to unity.

The PNLMS algorithm has been extensively tested and shown to be significantly better than NLMS for network cancelers. And, as can be seen, it requires only

Amazon EX1010 Page 913

a small increase in the computational requirements compared to NLMS ($3L$ multiplies compared to $2L$). It has been incorporated in recently developed network echo cancelers.

It is interesting to note that, although the PNLMS algorithm has been used successfully on telephone networks, no satisfactory proof of its convergence is known at present. Let us show the difficulty that arises in an attempt at a convergence proof.

Consider only the ideal noiseless situation with a perfectly constant echo path. As in the case of the NLMS algorithm, subtract both sides of (45.19) from the true response vector $\boldsymbol{h}$ and rewrite the equation in terms of the misalignment vector

$$\boldsymbol{r}(n+1) = \boldsymbol{r}(n) - \frac{\mu}{\boldsymbol{x}^{\mathrm{T}}(n)\boldsymbol{G}(n)\boldsymbol{x}(n)}\boldsymbol{G}(n)\boldsymbol{x}(n)\boldsymbol{x}^{\mathrm{T}}(n)\boldsymbol{r}(n)\,. \tag{45.21}$$

We will assume that $\boldsymbol{G}(n)$ is symmetric and positive definite. [The specification of (45.20) assures that. The following discussion is valid for any positive definite symmetric matrix.] Then we can define the positive-definite symmetric square-root matrix $\boldsymbol{G}^{1/2}(n)$, such that $\boldsymbol{G}^{1/2}(n)\boldsymbol{G}^{1/2}(n) = \boldsymbol{G}(n)$.

In terms of $\boldsymbol{G}^{1/2}(n)$ let us define the vectors $\boldsymbol{s}(n) = \boldsymbol{G}^{-1/2}(n)\boldsymbol{r}(n)$, $\boldsymbol{s}(n+1) = \boldsymbol{G}^{-1/2}(n+1)\boldsymbol{r}(n+1)$, and $\boldsymbol{u}(n) = \boldsymbol{G}^{1/2}(n)\boldsymbol{x}(n)$. Substituting these into (45.20) and rearranging terms we get

$$\boldsymbol{G}^{-1/2}(n)\boldsymbol{G}^{-1/2}(n+1)\boldsymbol{s}(n+1) = \boldsymbol{s}(n) - \frac{\mu}{\boldsymbol{u}^{\mathrm{T}}(n)\boldsymbol{u}(n)}\boldsymbol{u}(n)\boldsymbol{u}^{\mathrm{T}}(n)\boldsymbol{s}(n)\,. \tag{45.22}$$

If the matrix $\boldsymbol{G}$ were a constant independent of n, then the product of the two matrices on the left-hand side would be an identity and (45.22) would be identical to the NLMS update of (45.9b) with $\boldsymbol{x}$ replaced by $\boldsymbol{u}$ and $\boldsymbol{r}$ replaced by $\boldsymbol{s}$. In this case the discussion of the convergence properties of NLMS would carry over unchanged to the convergence of $\boldsymbol{s}(n)$, except that the covariance matrix of $\boldsymbol{x}$ would be replaced by the covariance matrix of $\boldsymbol{u}$, $\boldsymbol{R}_{uu} = \boldsymbol{G}^{1/2}\boldsymbol{R}_{xx}\boldsymbol{G}^{1/2}$. We could therefore conclude that $\| \boldsymbol{s}(n) \| \to 0$, and hence $\| \boldsymbol{r}(n) \| \to 0$ (since $\boldsymbol{G}^{1/2}$ is positive definite). However, since $\boldsymbol{G}$ is not constant, we cannot draw this conclusion. If the parameter μ is small, one can argue that $\boldsymbol{G}^{-1/2}(n)\boldsymbol{G}^{1/2}(n+1) \approx \boldsymbol{I}$, in which case convergence can again be inferred. It is also possible to get some exponential bounds on the convergence process (as for NLMS) in terms of the eigenvalues of $\boldsymbol{G}^{-1/2}(n)\boldsymbol{G}^{1/2}(n+1)$. However there does not appear to be any simple way to relate those eigenvalues to the properties of the far-end signal $x(n)$. Clearly, the convergence properties of the PNLMS algorithm are not yet well understood.

In spite of this lack of understanding of its convergence properties, intuitive reasoning has been used to provide an improvement of the PNLMS algorithm. The main drawback of the PNLMS algorithm is that although it performs much better than the NLMS algorithm for sparse impulse responses, it does not perform well (indeed it is even worse than NLMS) for impulse responses that are not sparse. In the telephone network nonsparse impulse responses are rare but they do occur, so it is desirable to be able to deal with both sparse and nonsparse responses. One simple solution (dubbed PNLMS++) is to alternate PNLMS and NLMS updates. A much better solution is the improved PNLMS (IPNLMS) algorithm proposed in [45.15]. This algorithm uses an update that is a compromise between the NLMS and the PNLMS updates. The update is accomplished by replacing the diagonal matrix $\boldsymbol{G}$ defined in (45.20) by a diagonal matrix $\boldsymbol{K}$ whose diagonal elements k_{ll} are obtained by replacing the corresponding elements g_{ll} of $\boldsymbol{G}$ as

$$k_{ll}(n) = \frac{1-\alpha}{2L} + (1+\alpha)\frac{|\hat{h}_l(n)|}{2\sum_{l=1}^{L}|\hat{h}_l(n)|}\,, \tag{45.23}$$

where α is a parameter such that $-1 \le \alpha < 1$. For $\alpha = -1$, k_{ll} is constant independent of l, so IPNLMS is identical to NLMS. For α close to 1, k_{ll} is essentially proportional to $|h_l(n)|$, so IPNLMS behaves like PNLMS. In [45.15] it is shown that, for α close to 0 or -0.5, IPNLMS always behaves better than both NLMS and PNLMS.

We close the discussion of the PNLMS algorithm by deriving it via a constrained minimization quite similar to the one used in deriving the NLMS algorithm. Again, let $\hat{\boldsymbol{h}}(n+1) = \boldsymbol{h}(n) + \boldsymbol{\Delta}$, and require that the a posteriori error be related to $e(n)$ as $\varepsilon(n) = (1-\mu)e(n)$, which is the same as requiring that

$$\boldsymbol{\Delta}^{\mathrm{T}}\boldsymbol{x}(n) = \mu e(n)\,.$$

Again we find the update with the minimum norm that satisfies this constraint. Only instead of the L_2 norm we choose the norm $\boldsymbol{\Delta}^{\mathrm{T}}\boldsymbol{M}^{-1}\boldsymbol{\Delta}$, where $\boldsymbol{M}$ is some, as yet unspecified, symmetric positive definite matrix. Thus we need to minimize

$$\frac{1}{2}\boldsymbol{\Delta}^{\mathrm{T}}\boldsymbol{M}^{-1}\boldsymbol{\Delta} + \lambda[\boldsymbol{\Delta}^{\mathrm{T}}\boldsymbol{x}(n) - \mu e(n)]\,.$$

Setting the derivative of this expression with respect to $\boldsymbol{\Delta}$ equal to zero gives

$$\boldsymbol{\Delta} = \lambda \boldsymbol{M}\boldsymbol{x}(n) \, .$$

Substituting this in the constraint equation gives

$$\lambda = \frac{\mu e(n)}{\boldsymbol{x}^{\mathrm{T}}(n)\boldsymbol{M}\boldsymbol{x}(n)} \, ,$$

and hence

$$\boldsymbol{\Delta} = \frac{\mu}{\boldsymbol{x}^{\mathrm{T}}(n)\boldsymbol{M}\boldsymbol{x}(n)} \boldsymbol{M}\boldsymbol{x}(n)e(n) \, .$$

If $\boldsymbol{M}$ is chosen to be the matrix $\boldsymbol{G}(n)$ as defined by (45.20) we get the PNLMS update. If $\boldsymbol{M}$ is chosen to be the matrix $\boldsymbol{K}$ defined by (45.23) we get the IPNLMS update.

More generally, choosing different specifications for the matrix $\boldsymbol{M}$ yields different updates. For instance $\boldsymbol{M} = \boldsymbol{G}^2(n)$ is a reasonable choice. That would measure the norm of the vector of relative increments (i. e., the vector with components $(\Delta_l/\hat{h}_l)$. Reference [45.14] considers other possibilities for the matrix $\boldsymbol{M}$ that are suitable for sparse impulse responses (e.g., the exponential gradient algorithm). To date, however, no choice has been found that provides any useful improvement over IPNLMS for network echo cancelation.

Let us note in passing that the form of the PNLMS update, as well as our derivation of it, has a similarity to the natural gradient update [45.16]. The natural gradient update just takes a step in the direction of $\boldsymbol{\Delta}$. This is equivalent to setting $\lambda = \mu e(n)$. Thus PNLMS is just a normalized form of the natural gradient update, and has been discussed from that point of view in [45.17].

Robust LMS and PNLMS

As mentioned in the Section on the ideal case, a double-talk detector is essential to prevent the LMS algorithm from diverging in the presence of double-talk. However, a double-talk detector [e.g., the Geigel detector of (45.15)] cannot turn off adaptation instantaneously, and even a very short interval of undetected double-talk can significantly increase the misalignment. A slight modification of the LMS algorithm can make it quite robust to such a short-lived disturbance [45.18]. The modification consists of replacing $e(n)$ in the NLMS update equation by $F[e(n)]$ as shown in (45.5). Likewise, for the PNLMS algorithm the same change is made in its update (45.19). From the discussion following (45.5) it would appear that the choice of a fixed function F in general slows down convergence and has therefore not been found to be useful. However, robustness can be achieved if the function is adapted. The form of the function is chosen such that it can be adjusted depending on the presence or absence of the disturbance. As noted in [45.18], $F(e)$ must be chosen such that $\lim_{|e|\to\infty} |F(e)| < \infty$, i. e., F has to be a saturating nonlinearity. A simple and convenient choice is $F(e) = \mathrm{sign}(e)sH\,(|e|/s)$, where H is a hard limiter, i. e.,

$$H\left(\frac{|e|}{s}\right) = \frac{|e|}{s} \, , \quad \frac{|e|}{s} \le k$$
$$= k \, , \quad \frac{|e|}{s} > k \, . \tag{45.24}$$

In (45.24), s is a positive scale factor whose effect depends upon the manner in which it is adapted. Initially, we would like s to be large so that H is in the linear region. Then $F(e) = e$ and the updates are just the normal NLMS updates. As the canceler converges we would like s to follow the level of the uncanceled echo to lower values ending with a value somewhat larger than the expected level of the circuit noise at the near end. During a short burst of double-talk (for the short interval required for the double-talk detector to act) we would like s to stay small and let the temporarily large value of e drive H into the saturation region, thus limiting the convergence rate. A simple adaptation of s that achieves this objective is given by

$$s(n+1) = \lambda_{\mathrm{s}} s(n) + (1-\lambda_{\mathrm{s}})\frac{1}{\beta} s(n) H\left(\frac{|e(n)|}{s(n)}\right) \, , \tag{45.25}$$

where λ_{s} is an exponential forgetting factor and β is a constant with a value of about 0.7.

The initial value $s(0)$ is set to be somewhat larger than the expected level of speech signals on the telephone network, so that initially H is in the linear range. As long as H stays in the linear range, $s(n)$ is $1/\beta$ times the average value of $|e(n)|$ in the immediate past. The forgetting factor λ_{s} gives an averaging interval of roughly $1/(1-\lambda_{\mathrm{s}})$ samples, so λ_{s} is chosen at some value between 0.99 and 1. As the canceler converges, the scale factor decreases from its initial value to a value slightly larger than the level of the circuit noise. At the initiation of double-talk, for a duration of roughly $1/(1-\lambda_{\mathrm{s}})$ samples, the scale factor stays low, the nonlinearity H saturates, and the rate of divergence is reduced. This gives time for the double-talk detector to act and prevent further divergence. This robustness does not come for free. If the error increases suddenly due to a sudden change in the echo path, the canceler will not track the change right away. However, as the scale factor estimator adapts to a larger value, the convergence rate

Amazon EX1010 Page 915

accelerates. There is thus a trade-off between immunity to double-talk and the rate of recovery from a sudden change of echo path.

Other Algorithms

For network echo cancelation, as far as we know, no other algorithms have found use.

For acoustic echo cancelation some other algorithms have been found useful.

An algorithm known as the two-path model was proposed in the 1970s, and has some interesting properties. However, since economy of computational requirements was the overriding criterion at the time, and the improvement it afforded for network applications was not significant, it did not gain favor. Recently there has been a revived interest in it for use in acoustic echo cancelation. Another algorithm that has been useful for acoustic echo cancelation is the subband canceler, as well as frequency-domain and multidelay filters. These algorithms turn out to be useful for the acoustic case because of the much longer impulse responses encountered in that application. We will discuss these in Sect. 45.2.

Adaptive filtering, however, has an extensive literature [45.10]. Several other algorithms selected from that literature have been proposed for echo cancelation, but have not yet found wide application. We will mention some of them briefly here, but will not discuss them in any detail.

Another Variable-Step Algorithm. PNLMS is a variable-step algorithm in that it assigns different adaptation step sizes to the different coefficients of the estimated response vector $\hat{\boldsymbol{h}}(n)$. It provides fast convergence for sparse echo paths. However, there is one undesirable property of PNLMS that deserves attention. Suppose the algorithm has converged to a small misalignment. According to the algorithm, the coefficient $\hat{h}_j(n)$ will be assigned a step size essentially proportional to the magnitude $|h_j|$ of the corresponding coefficient of the echo path response. This means that the step size will be large for the large coefficients, even though they might have converged to their correct values. Obviously, this is not a desirable assignment when the coefficients have converged. In that case all coefficients should have small step sizes to keep the residual misalignment low. Ideally one would like the step size to be proportional to $|\hat{h}_j(n) - h_j|$. This, of course, is not possible because h_j is not known. However, an algorithm described in [45.19] assigns step sizes in a way that makes them smaller as the estimated coefficient comes close to its target. The algorithm is based on the simple observation that, when a coefficient has converged close to its target, the successive updates of that coefficient will no longer push it consistently in one direction. So for each coefficient, the sign of the update is monitored over D consecutive samples in the immediate past. If the number of sign changes exceeds a specified number N_1, the step size is decreased; if it is less than a specified number N_2, the step size is increased. The parameters D, N_1, and N_2, and the range over which the step size is allowed to vary, are determined empirically. This algorithm was proposed as a general adaptive filter algorithm. As far as we know, it has not been applied to echo cancelation. Intuitively speaking, it seems to have the right properties for assigning step sizes near convergence. In the initial phase, however, the step assignment according to PNLMS should give faster convergence. Some combination of the two should yield better performance than that of either one by itself.

The Block Least-Squares Algorithm. In the section on the stochastic gradient algorithm we noted that $\hat{\boldsymbol{h}}$ can be estimated by minimizing the expected value of the squared error $e^2(n)$. However, for simplicity, we took the instantaneous value of the squared error as an estimate of its expected value. An improvement in performance can be expected if better estimates of the expected error are used. One such estimate is the average, or arithmetic mean, of $e^2(m)$ over a range of values of the index m. At time index n, let us average over the range $n-M+1 \le m \le n$. Ignoring the normalizing constant M, we see that $\hat{\boldsymbol{h}}(n)$ may be estimated as the vector $\hat{\boldsymbol{h}}$ that minimizes the quantity

$$
\begin{aligned}
J(\hat{\boldsymbol{h}}) &= \sum_{m=n-M+1}^{n} e^2(m) \\
&= \sum_{m=n-M+1}^{n} [y(m) - \hat{\boldsymbol{h}}^{\mathrm{T}} \boldsymbol{x}(m)]^2 \\
&= \hat{\boldsymbol{h}}^{\mathrm{T}} \sum_{m=n-M+1}^{n} \boldsymbol{x}(m)\boldsymbol{x}^{\mathrm{T}}(m)\hat{\boldsymbol{h}} - 2\hat{\boldsymbol{h}}^{\mathrm{T}}\boldsymbol{w}(n) \\
&\quad + \sum_{m=n-M+1}^{n} y^2(m) \\
&= \hat{\boldsymbol{h}}^{\mathrm{T}} \boldsymbol{X}(n)\boldsymbol{X}^{\mathrm{T}}(n)\hat{\boldsymbol{h}} - 2\hat{\boldsymbol{h}}^{\mathrm{T}}\boldsymbol{w}(n) + \sum_{m=n-M+1}^{n} y^2(m)\,.
\end{aligned}
\tag{45.26}
$$

In the last line of (45.26) $\boldsymbol{X}(n)$ is an $L \times M$ matrix whose m-th column is $\boldsymbol{x}(n-M+m)$ for $1 \le m \le M$, and the

vector $\boldsymbol{w}(n)$ is given by $\boldsymbol{w}(n) = \sum_{m=n-M+1}^{n} y(m)\boldsymbol{x}(m)$. Setting the gradient of J with respect to $\hat{\boldsymbol{h}}$ equal to zero shows that the minimum is obtained as the solution to

$$\boldsymbol{X}(n)\boldsymbol{X}^{\mathrm{T}}(n)\hat{\boldsymbol{h}}(n) = \boldsymbol{w}(n)\,. \tag{45.27}$$

For the solution to be unique, the matrix $\boldsymbol{X}(n)\boldsymbol{X}^{\mathrm{T}}(n)$ must be of full rank. If the matrix is not of full rank (either because $M < L$, or because the number of independent rows is $< L$), a pseudoinverse can be used to yield a minimum norm solution.

The solution of (45.27) requires the inversion of an $L \times L$ matrix, which would normally require $O(L^3)$ operations. However, due to the special Toeplitz structure of $\boldsymbol{X}$, the inversion can be done with approximately $12L^2$ operations [45.20].

The solution depends on the time index n as well as on the averaging interval M. Given the nonstationary nature of the problem, $\hat{\boldsymbol{h}}(n)$ should be updated as often as the available computational resources allow. Note, however, that even if the inversion is done once every block, the per-sample computations are approximately $12L$. That is still several times more than the computations required for NLMS.

The Recursive Least-Squares Algorithm. In the block least-squares algorithm, the estimate of $\hat{\boldsymbol{h}}(n)$ is obtained once per block. The recursive least-squares (RLS) algorithm allows an update of the algorithm every sample. In this algorithm, instead of approximating the expected value of $e^2(n)$ by the average over a block, it is approximated by a weighted sum of the squared error at all past samples. With an exponential weighting, one minimizes

$$J(\hat{\boldsymbol{h}}) = \sum_{m=-\infty}^{n} \lambda^{n-m} e^2(m)\,, \tag{45.28}$$

where the *forgetting factor* λ is chosen in the range $0 < \lambda < 1$. The solution proceeds as for the case of the block algorithm, except that equation to be solved is

$$\boldsymbol{R}(n)\hat{\boldsymbol{h}}(n) = \boldsymbol{w}(n)\,, \tag{45.29}$$

with $\boldsymbol{R}(n) = \sum_{m=-\infty 1}^{n} \lambda^{n-m}\boldsymbol{x}(m)\boldsymbol{x}^{\mathrm{T}}(m)$ and $\boldsymbol{w}(n) = \sum_{m=-\infty}^{n} y(m)\boldsymbol{x}(m)$. From these definitions the following recursions are seen to be valid:

$$\boldsymbol{R}(n) = \lambda\boldsymbol{R}(n-1) + \boldsymbol{x}(n)\boldsymbol{x}^{\mathrm{T}}(n) \tag{45.30}$$

and

$$\boldsymbol{w}(n) = \lambda\boldsymbol{w}(n-1) + y(n)\boldsymbol{x}(n)\,. \tag{45.31}$$

Because of the recursion (45.30), the matrix inversion lemma [45.9, p. 480] can be used to compute $\boldsymbol{R}^{-1}(n)$ efficiently by updating $\boldsymbol{R}^{-1}(n-1)$. Thus the estimate of $\hat{\boldsymbol{h}}(n)$ can be obtained recursively from $\hat{\boldsymbol{h}}(n-1)$. The procedure is straightforward, but a bit cumbersome. The details may be found in [45.10, Chap. 13]. The computational requirement of the recursion is of order $O(L^2)$ per iteration. By making use of the shift property of the transversal filter the computational requirements can be reduced to as low as $7L$ per iteration The algorithms that achieve this low rate are the fast recursive least-squares (FRLS) and fast transversal filter (FTF) algorithms [45.21–23]. Their drawback is that they are rather numerically unstable, and have to be periodically reset.

The RLS algorithm generally gives much faster convergence than NLMS. However, its tracking ability is not necessarily better [45.24].

Affine Projection Algorithms. The affine projection algorithm (APA) is a generalization of the NLMS algorithm. It was first introduced in 1984 [45.25]. By definition, an affine projection is the combination of a projection and a translation. Let us write the NLMS update of (45.6) in a slightly different form using (45.7) and (45.8):

$$\begin{aligned}\hat{\boldsymbol{h}}(n+1) &= \hat{\boldsymbol{h}}(n) + \boldsymbol{\Delta} \\ \boldsymbol{\Delta} &= -\mu\boldsymbol{x}(n)[\boldsymbol{x}^{\mathrm{T}}(n)\boldsymbol{x}(n)]^{-1} \\ &\quad \times \boldsymbol{x}^{\mathrm{T}}(n)[\hat{\boldsymbol{h}}(n) - \boldsymbol{h}(n)]\,.\end{aligned} \tag{45.32}$$

The matrix $\boldsymbol{x}(n)[\boldsymbol{x}^{\mathrm{T}}(n)\boldsymbol{x}(n)]^{-1}\boldsymbol{x}^{\mathrm{T}}(n)$ is a projection operator that orthogonally projects $[\hat{\boldsymbol{h}}(n) - \boldsymbol{h}(n)]$ on $\boldsymbol{x}(n)$. Therefore $\boldsymbol{x}(n)[\boldsymbol{x}^{\mathrm{T}}(n)\boldsymbol{x}(n)]^{-1}\boldsymbol{x}^{\mathrm{T}}(n)[\hat{\boldsymbol{h}}(n) - \boldsymbol{h}(n)]$ is an affine projection of $\hat{\boldsymbol{h}}(n)$ on $\boldsymbol{x}(n)$. Thus the NLMS update is obtained by subtracting from $\hat{\boldsymbol{h}}(n)$, μ times an affine projection of $\hat{\boldsymbol{h}}(n)$ on $\boldsymbol{x}(n)$. Instead of subtracting the projection of $[\hat{\boldsymbol{h}}(n) - \boldsymbol{h}(n)]$ on a single vector, the APA subtracts the projection on the subspace spanned by the past M vectors $\boldsymbol{x}(m)$, $n - M + 1 \le m \le n$. This is accomplished by simply replacing $\boldsymbol{x}(n)$ in (45.32) by the $L \times M$ matrix whose columns are these M vectors. That is in fact the matrix $\boldsymbol{X}(n)$ defined just before (45.27). Thus the APA is obtained by replacing $\boldsymbol{x}(n)$ by $\boldsymbol{X}(n)$ in (45.32), i. e.,

$$\begin{aligned}\hat{\boldsymbol{h}}(n+1) &= \hat{\boldsymbol{h}}(n) + \boldsymbol{\Delta} \\ \boldsymbol{\Delta} &= -\mu\boldsymbol{X}(n)[\boldsymbol{X}^{\mathrm{T}}(n)\boldsymbol{X}(n)]^{-1} \\ &\quad \times \boldsymbol{X}^{\mathrm{T}}(n)[\hat{\boldsymbol{h}}(n) - \boldsymbol{h}(n)]\,.\end{aligned} \tag{45.33}$$

If the columns of $\boldsymbol{X}(n)$ are independent, then as M increases they span a larger subspace of the L-dimensional

Amazon EX1010 Page 917

space of the $\boldsymbol{x}(n)$ and $\hat{\boldsymbol{h}}(n)$ vectors. Therefore, in general, the convergence rate increases significantly as M increases.

The price paid for the faster convergence is higher computational cost. The APA requires $2LM + KM^2$ multiplications per iteration, where K can be as low as 7 [45.26]. If $M \ll L$ then the algorithm is essentially M times as costly as NLMS. If the update is computed only once every M samples the algorithm is called the partial-rank algorithm (PRA), which was introduced originally in [45.27] for adaptive beam forming. The cost per sample of the PRA is comparable to that of NLMS, but its convergence is slower than that of the APA. With highly colored signals such as speech the convergence rate of PRA is more or less the same as that of APA. The main disadvantage of PRA compared to APA is that its computations are very unevenly spread out over an M-sample cycle. To overcome this disadvantage a fast affine projection (FAP) algorithm that requires $2L + 30M$ computations per iteration has been developed [45.26, 28]. One disadvantage of the FAP algorithm is that, although the current error signal is saved, the current estimate of $\hat{\boldsymbol{h}}$ is not saved. What is saved instead is a vector that depends on both the current estimate of $\hat{\boldsymbol{h}}$ as well as the current matrix $\boldsymbol{X}$. For network echo cancelers, this can lead to undesirable performance. Note that during periods of double-talk and during silence at the far end, adaptation is inhibited. Since network echo paths usually change very slowly, NLMS needs to reconverge only to a slightly changed echo path. An algorithm that does not save the estimate of the echo path must re-converge from scratch. For acoustic echo cancelers this might not be such a detriment because the echo path is highly variable, and hence NLMS would also have to reconverge to a significantly changed echo path.

We turn next to acoustic echo cancelers.

45.2 Single-Channel Acoustic Echo Cancelation

Acoustic echoes arise in hands-free telephony and in teleconferencing due to the coupling between the loudspeaker and the microphone via the transfer characteristics of the room in which they are located. A typical configuration is illustrated in Fig. 45.4. In that figure we have shown one end of a teleconference where a four-wire connection is set up between the two conference rooms. If the connection is a two-wire connection then hybrids must be included in the circuit. We neglect them for the present discussion.

In principle the problem of acoustic echoes is not fundamentally different from the problem of network echoes. Instead of the echo being generated due to impedance mismatch, it is now generated due to the acoustic coupling between the loudspeaker and the microphone. Other than that the problem in the two cases is identical. However, the techniques required to deal with acoustic echoes are significantly different from those required for network echoes. The main difference is that the properties of the echo paths in the two situations are very different [45.29, 30].

One main difference is that the room impulse response is much longer than the response of a network connection. Recall that in a network connection the impulse response of the echo path is usually about 500–600 taps long (at the standard sampling frequency of 8000 samples per second), of which only about 50–60 are active taps. In contrast, a room response even for a small office can be significant over an interval of 300–500 ms, corresponding to 2500–4000 taps, all of which are active.

A second difference is that a room impulse response is much more time varying than a network impulse response. The transmission characteristics of the network vary very slowly (except occasional rapid changes due, for example, to someone picking up or hanging up an extension phone). In contrast, the impulse response of a room is constantly changing. Note that the impulse re-

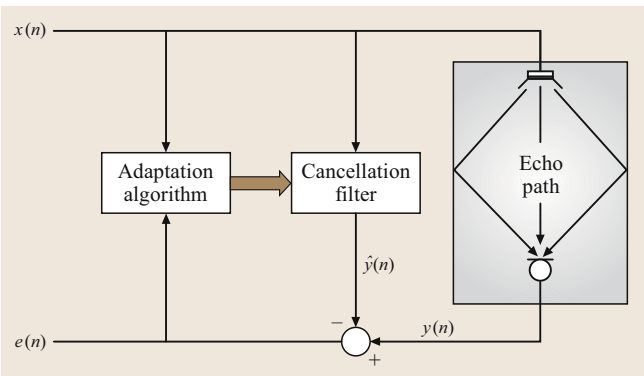


Fig. 45.4 An acoustic echo canceler is used to cancel echoes that arise from coupling between a loudspeaker and microphone

sponse of a room is made up of reflections from the walls, ceiling, floor, and all objects and people in the room. As people move around, the impulse response changes. Opening or closing of doors or windows changes the echo path. Even changes in the speed of sound due to temperature changes can produce a significant amount of misalignment. (A temperature change of 5.5 °C produces roughly a 1% change in the speed of sound. This corresponds to a change of about one sampling interval, i. e., 125 μs, in travel time for an acoustic path of about 4 m.)

Thus we note that acoustic echo cancelers require more computing power due to the increased length of the impulse response to be identified and also due to the faster convergence required to track the faster time variation. We will discuss some of the techniques used to deal with these problems.

45.2.1 The Subband Canceler

The technique used for dealing with the increased length of the impulse response is to utilize subband echo cancelers. The structure of such a canceler is shown in Fig. 45.5. It was first proposed in 1984, independently in [45.31] and [45.32] and further developed in [45.33]. The idea is to use a bank of contiguous bandpass filters with impulse responses $w_0, w_1, w_2, \ldots, w_{K-1}$ that span the frequency range of the speech signal. Rather than taking a linear combination of the outputs of these filters as the model for the echo (as discussed in Sect. 45.1.2), the output of each bandpass filter is used as an input to a transversal filter. The coefficients of each transversal filter are adapted using an NLMS algorithm so that it cancels the echo in its subband. (The symbol $\Rightarrow$ is used in Fig. 45.5 to indicate a bundle of paths, one for each subband filter.) Thus the structure is equivalent to M echo cancelers – one for each subband. The outputs of subband filters, after cancelation of the respective echoes, are summed to give the final signal returned to the far end.

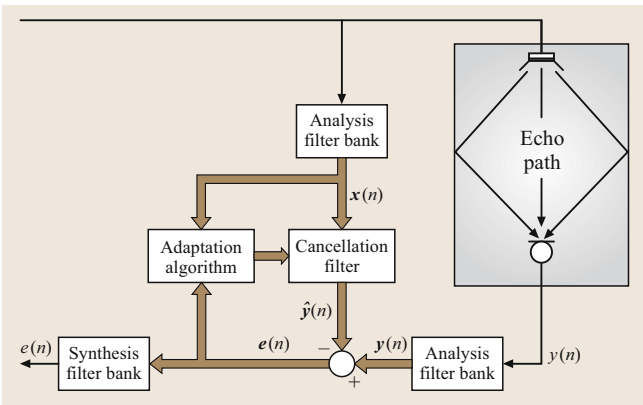


Fig. 45.5 Block diagram of a subband echo canceler showing analysis and synthesis filter banks, subband cancelation filter, and adaptation algorithm

The following simple calculation shows that the number of computations per second required for this structure is $1/M$ times that required for a single NLMS algorithm for the entire impulse response.

Let the room response have duration T_H seconds. Let L be the number of samples representing this impulse response. Let the sampling interval for the full-band speech signal be T_s seconds. (As mentioned above, the normal sampling frequency f_s on the telephone network is 8000 samples per second, which gives $T_s = 125\,\mu s$.) Assume that the impulse response when bandlimited to each subband has the same duration T_H as the full-band impulse response. If M subbands are used, the minimum required sampling frequency for each subband signal is f_s/M. Hence each subband impulse response has L/M coefficients. Each of these has to be adapted once in MT_s seconds. Thus each subband requires $L/(M^2T_s)$ adaptations per second. For all the M subbands combined, the total number of adaptations is therefore $L/(MT_s)$ per second, which is M times smaller than the L/T_s adaptations per second required for a single full-band canceler.

In this derivation we have neglected the overhead required for the analysis and synthesis filters shown in Fig. 45.5. If this overhead is taken into account, the subband structure provides computational savings only if T_H is longer than about 90 ms. Thus for network cancelers the subband structure is not very attractive. However, for room responses it provides a significant reduction in computational requirements.

Clearly, the reduction in computational complexity derived above depends on sampling the subband signals at a rate $1/M$ that of the full-band signal. This requires the use of multirate filters described in [45.33, 34]. As shown in [45.33], exact cancelation can occur only if the filters are free of aliasing. In general, less aliasing requires longer analysis and synthesis filters, with consequent additional transmission delay. Design of these filters is thus a compromise between computational complexity, transmission delay, and the degree of aliasing. These aspects of the design of subband filters are discussed in some detail in [45.35].

It should be noted that the sampling frequency of f_s/M for the subband filters is the minimum sampling

rate (also called critical sampling). Sampling at this critical frequency makes the control of aliasing very difficult. In practice, therefore, the sampling frequency is chosen to be f_s/R with R somewhat smaller than M. With this change, the computations are reduced by the somewhat smaller factor R^2/M. The reduction factor approaches M as R approaches the critical sampling rate M.

The performance of a given design depends on the interplay of two sources of error: the aliasing introduced by the analysis filters and the error due to the finite length of the transversal filters in each subband. If very short transversal filters are used, the truncation error dominates. Increasing the length of the transversal filters allows smaller misalignment. However, once the aliasing error becomes the controlling factor, further increase in length does not afford any significant improvement.

The subband structure has several advantages. The main advantage, as mentioned earlier, is the very significant reduction in computational complexity. This reduction in computational complexity can be exploited in several ways: hardware can be saved to reduce cost, the overall system bandwidth or the duration of the impulse response to be modeled can be increased, or more-complex adaptation algorithms can be employed. An example of the use of a more-complex algorithm is given in Sect. 45.2.2. Another advantage is that the structure of the subband canceler is well suited to parallel processing. A third advantage is that it provides an improved convergence rate. The reason for this is that the speech spectrum has a much smaller dynamic range within any one of the subbands. Therefore, the eigenvalues of the correlation matrix for the signal within each subband have smaller spread. Some small eigenvalues are introduced at the edges of each subband, due to the transition band of the filter response. These small eigenvalues can in fact *increase* the eigenvalues spread. However, the analysis and synthesis filters reduce the error signal at those frequencies, so the effect of these eigenvalues on the overall misalignment is small. Initially, therefore, the subband canceler converges faster than a full-band canceler. The small eigenvalues do cause slow asymptotic convergence [45.36]. One possible solution to this problem is to make the analysis filter bandwidth slightly larger than the pass band of the synthesis filter. This was proposed in [45.36] and implemented in [45.37]. One final advantage of the subband structure is in the design of the nonlinear processor that is used to eliminate the residual echo. The center clipper can now be placed in each subband. This produces much less distortion of the near-end signal at the same clipping level. Alternatively, it allows one to increase the threshold for the same overall distortion of the near-end signal, thus allowing a less-expensive canceler with a larger misalignment.

45.2.2 RLS for Subband Echo Cancelers

If economy of computation is not the overriding consideration, each subband canceler can be adapted by algorithms other than NLMS. The most ambitious such modification for acoustic echo cancelation to date is the algorithm proposed in [45.38], which uses the subband structure but instead of the NLMS algorithm it uses the FRLS algorithm. As mentioned, FRLS tends to be numerically unstable, and has to be frequently restarted. That introduces frequent undesirable interruptions. However, in the subband structure, it is possible to restart the algorithm in one band at a time, in some sequence that periodically visits all the subbands. When there are a large number of subbands, the slight short-lived misalignment in any one band is perceptually unnoticed.

45.2.3 The Delayless Subband Structure

The only *disadvantage* of the subband approach is the extra delay introduced by the analysis and synthesis filters. This delay depends on the design of the analysis and synthesis filters, but can be 20 ms or more (double that for the round-trip delay). For many applications

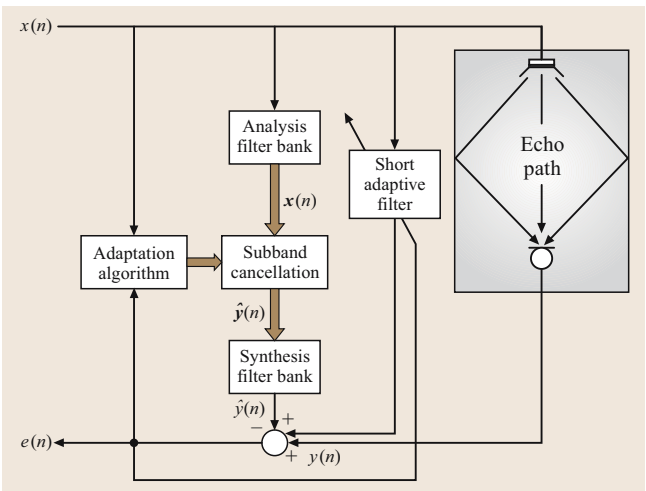


Fig. 45.6 Delayless subband adaptive filter. An auxiliary short wide-band adaptive filter is needed to fill the gap due to the delay in the analysis and synthesis filters

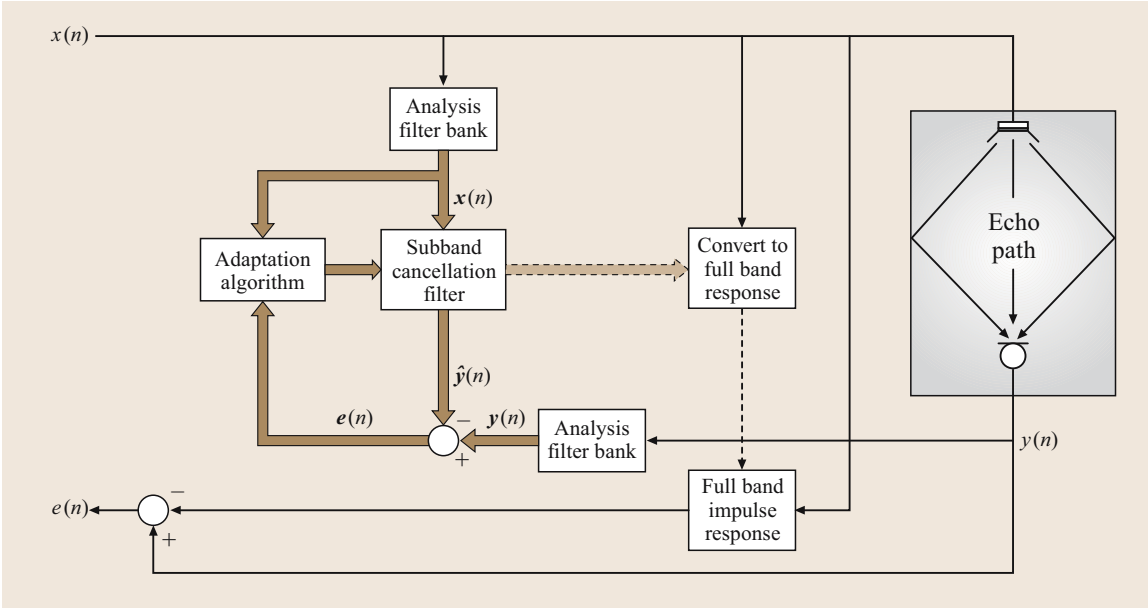


Fig. 45.7 The delayless subband structure of [45.39] that eliminates signal path delay by transforming weights of the canceling subband filter to the equivalent wide-band filter

this extra delay is undesirable. Two methods have been suggested to avoid this delay.

One proposal [45.40] eliminates the delay by moving the analysis and synthesis filters from the signal return path to the cancelation filter path as shown in Fig. 45.6. When this is done, the canceler cannot model the initial part of the echo impulse response due to the delay introduced in the model path. Therefore, a short full-band canceler is added in parallel to model the initial portion of the echo path response. This structure has two drawbacks. Delay in a feedback loop is always problematic, especially when the impulse response is time-varying. Also, the extra computations needed for the auxiliary full-band canceler, reduces the computational advantage of the subband structure.

The second proposal [45.39] is to use the subband structure for adaptation as above but transform the weights to the equivalent single full-band transversal filter to do the actual cancelation of the echo in the return signal path. This structure is shown in Fig. 45.7. It eliminates the delay in the return path without introducing delay in the feedback loop and without requiring an auxiliary canceler. It does require additional computations for the derivation of the full-band filter, and for the extra full-band convolution. The latter can be done quite efficiently by using a well-known technique: transform the two signals to be convolved to the frequency domain using a fast Fourier transform (FFT), multiply the two transformed signals, and compute the inverse FFT of the product to return to the time domain.

45.2.4 Frequency-Domain Adaptation

Another technique that is used to deal with the long impulse responses of rooms is frequency-domain adaptation. A block diagram of a typical implementation is shown in Fig. 45.8. It uses block processing in which one block of input data is processed at a time and produces one block of output. Referring to Fig. 45.8, the Fourier transform $\tilde{\boldsymbol{x}}$ of a block of samples of the far-end signal $x(n)$ is computed via the fast Fourier transform (FFT). The components of the adapted coefficient vector $\tilde{\boldsymbol{h}}$ now multiply the corresponding components of $\tilde{\boldsymbol{x}}$ to produce the Fourier transform of the output of the filter. (The inverse FFT of $\tilde{\boldsymbol{h}}$ is the coefficient vector of the corresponding transversal filter.) The inverse FFT of the filter output gives the block of time-domain samples of the estimated response. The error between this block and the block of samples of the echo path output y, gives the block of error samples. Its Fourier transform controls the vector $\tilde{\boldsymbol{h}}$ to drive the error towards zero. The coefficient vector $\tilde{\boldsymbol{h}}$ is adapted once per block. The main advantage of this technique is that, because of the use of the FFT, the number of computations per

Amazon EX1010 Page 921

sample is significantly reduced compared to the NLMS algorithm.

The first such proposal was presented in [45.41]. However, in that proposal circular convolution was used rather than the linear convolution required for modeling the echo response. (Circular convolution is convolution of signals obtained by periodic extension of the signals within a block.) Due to this, the adaptive filter does not converge to the true transversal solution [45.42]. How this drawback can be removed through the use of five FFTs per block is shown in [45.42]. A further improvement is presented in [45.43], where the algorithm requires only three FFTs per block. In these papers in order to realize linear convolution, the transversal filter is extended with a block of zero coefficients, and a two-to-one overlap between successive input blocks is used. The FFT is computed over a two-block window.

In references [45.41,42] and [45.43] the block length used is equal to the length L of the transversal filter. For large L, since the FFT of a window of $2L$ samples requires $2L\log_2(L)$ computations, the number of computations per block is essentially $CL\log_2(L)$, where the constant of proportionality C depends on the particular implementation, but is close to 10. Hence the number of computations per sample is $C\log_2(L)$. The NLMS algorithm requires $2L$ computations per sample, which is substantially greater.

Besides the advantage of reduction in the computational requirements, the frequency-domain approach has two additional advantages. For signals like speech, the components of the Fourier transform tend to be more uncorrelated than successive samples in the transversal filter. Also, the step sizes for the coefficients can be adjusted individually to reduce the disparity in the convergence rates of the components due to the large spread in the eigenvalues of the correlation matrix of the speech signal. One simple way of doing this is given in [45.43].

These good features are offset by two important disadvantages. First, the block processing introduces a delay of L samples, which for a room response is intolerably long. And second, since the filter coefficients are held constant during each block, the ability of the adaptive filter to track changes in the room response is reduced.

The problem of long delay can be alleviated by a proposal in [45.44] called multidelay filter (MDF) adaptation. By sectioning the transversal filter into K nonoverlapping segments the delay is reduced by a factor of K. A generalized MDF is presented in [45.45]. It allows the overlap between successive blocks to be greater than 2 to 1, thus allowing more-frequent adaptation and hence better tracking ability. A unified treatment of these ideas is presented in [45.46, Chap. 8].

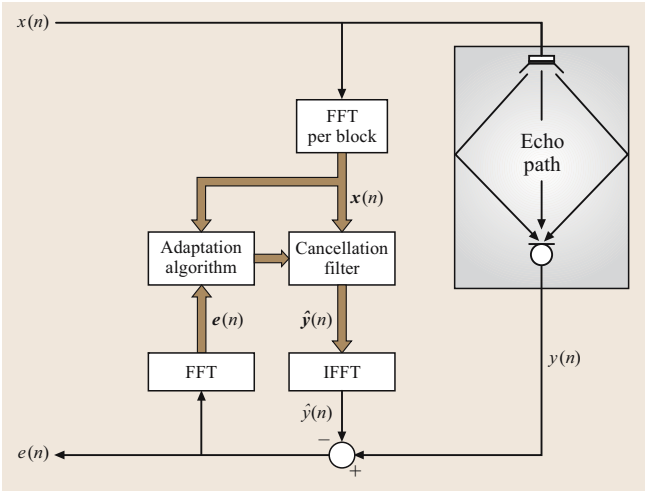


Fig. 45.8 Frequency-domain adaptation. The input is processed in blocks. One output block is generated for each input block

45.2.5 The Two-Echo-Path Model

As mentioned in the section on other algorithms, the two-path model [45.47] was proposed in the 1970s but did not find application for network echo cancelation. It has attracted renewed interest for acoustic echo cancelation.

The basic idea of the two-path model is very simple. Observe that in the implementation of the echo canceler as shown in Fig. 45.3 a single filter serves the dual purpose of adapting itself to the echo path and of canceling the echo in the signal returned to the far end. This provides computational economy but is not necessary. The two functions can be performed by separate filters. This simple observation leads to the idea of the

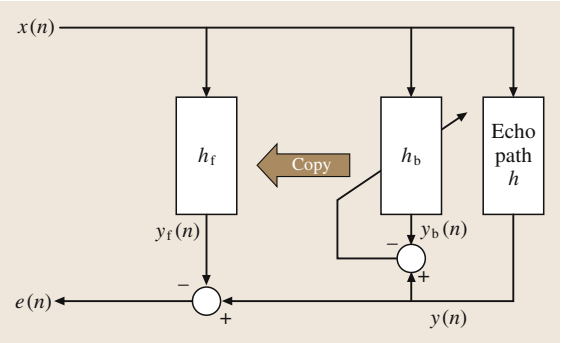


Fig. 45.9 Two-path model of echo canceler

Amazon EX1010 Page 922

two path model whose structure is shown in Fig. 45.9. A *background* filter adapts as in a conventional echo canceler. A *foreground* canceler does the actual cancelation. Its coefficients are updated by copying from the background canceler whenever a decision logic declares that the background canceler is performing better than the current foreground canceler.

The main reason this method has found favor for acoustic echo cancelation is the fact that it deals very effectively with double-talk in this application. Unlike network echoes, for which the hybrid provides a minimum guaranteed echo return loss, acoustically generated echoes can produce a signal at the microphone whose amplitude is as large as, or often larger than that of the far-end signal. This makes it impossible to set the threshold for the Geigel double-talk detector. Other methods are needed to disable adaptation during double-talk. Several methods have been proposed recently [45.46, Chap. 6]. However, with the two-path model a double-talk detector is not necessary. Any reasonable logic will decide that the background canceler is worse than the foreground canceler when double-talk sets in. Therefore, the foreground canceler is not updated until the double-talk ceases and the background canceler has adapted to a better estimate.

Referring to Fig. 45.9, we define $\hat{\boldsymbol{h}}(n)$ and $\hat{\boldsymbol{h}}_\mathrm{b}(n)$ as the current estimated impulse responses of the foreground and background cancelers, respectively, and $e(n)$ and $e_\mathrm{b}(n)$ the corresponding errors. The decision to update the foreground canceler is, in general, a function of $e(n)$, $e_\mathrm{b}(n)$, the far-end signal $x(n)$, and the echo path signal $y(n)$. The success of this method depends crucially upon the choice of that function. The algorithm for this decision described in [45.47] is not suitable for acoustic echo cancelers. The main reason is that it relies on the hybrid to provide a minimum echo return loss. The algorithm proposed recently in [45.48] has been shown to work very well. It essentially monitors the echo return loss enhancement (ERLE) of the two paths averaged by a first-order integrator with a time constant of about 150 ms. The coefficients of $\hat{\boldsymbol{h}}_\mathrm{b}(n)$ are copied into $\hat{\boldsymbol{h}}(n)$ whenever the average ERLE of the background canceler is better than that of the foreground canceler.

45.2.6 Variable-Step Algorithm for Acoustic Echo Cancelers

During our discussion of the convergence of the PNLMS algorithm, we had mentioned the possibility of using a fixed (i. e., constant over time) matrix $\boldsymbol{G}$ to assign different step sizes to different components of the adaptive filter response vector $\hat{\boldsymbol{h}}$. Indeed, convergence of the PNLMS algorithm was proved only for this special case. This possibility of using a fixed $\boldsymbol{G}$ was already mentioned as early as the 1960s [45.1]. The idea was that the step size for the coefficient $\hat{h}_j$ should be made proportional to the expected value of its target h_j over the ensemble of echo responses on the telephone network. On average, therefore, all coefficients would then require the same number of steps to reach their respective targets. The ensemble of impulse responses on the network turned out to be too varied (especially due to the wide range of initial delays) to make this idea useful. The situation is different in the case of acoustic echo cancelers used for teleconferencing. For any given room the echo response has a fairly well-defined envelope, which is in general exponential, with a decay constant that depends on the reverberation time of the room. The envelope changes when people move around or if the loudspeaker and microphone positions are changed. However, a compromise exponential envelope can be estimated from several measurements in the room. It is therefore reasonable to make the step sizes decay exponentially with coefficient number, with the decay constant of the exponential chosen on the basis of the reverberation time of the room. This idea was presented in [45.49] where it was shown that for a room of the size of a typical office, the technique can make the initial convergence of the NLMS algorithm two to three times faster than with a fixed step size.

45.2.7 Cancelers for Nonlinear Echo Paths

Throughout the discussion so far, we have assumed that the echo path is linear and can therefore be characterized by an impulse response. The problem of echo cancelation for a nonlinear path was considered as early as 1971 [45.50].The proposal was to use the Volterra kernel expansion of the output $y(n)$ of the nonlinear echo path, when its input is the far-end signal $x(n)$. The Volterra expansion has the form

$$y(n) = \sum_{l=0}^{L-1} h_l x(n-l) + \sum_{l_1=0}^{L-1} \sum_{l_2>l_1}^{L-1} h_{l_1 l_2} x(n-l_1) x(n-l_2) + \cdots , \tag{45.34}$$

where the series continues with third- and higher-order terms. The adaptation algorithm then has to estimate

the coefficients $h_l, h_{l_1 l_2}, \ldots$ Note that $y(n)$ depends linearly on these coefficients. Hence the stochastic gradient algorithm can again be used to estimate them.

The major drawback of this method is that the number of coefficients to be adapted becomes enormously large even if third- and higher-order terms are neglected. Fortunately, the telephone network is quite well represented by a linear system; hence the use of Volterra kernels was never required.

For acoustic echo cancelation there has been a revival of some interest in the Volterra kernel expansion. The reason for this interest is that, in certain applications such as echo cancelation in cars, inexpensive loudspeakers are used, and to get sufficient volume they are driven into their nonlinear region. The echo path, which includes the loudspeaker, thus becomes nonlinear. (It may be argued that one could replace the signal $x(n)$ in the adaptation algorithm by the signal picked up by a microphone close to the loudspeaker. However, in general this is a bad idea. The signal from such a microphone would include the echo as well as any near-end speech. Adaptation with such a signal would be problematic, especially in a small space such as a car, where these extraneous disturbing signals can be quite large.)

Recently, the idea of using the Volterra kernel expansion has been proposed to deal with this problem [45.51]. If the Volterra expansion of (45.34) were to be used, the number of coefficients to be estimated would, of course, be prohibitively large. However, one can make use of the fact that the echo path consists of a zero-memory nonlinearity (or a nonlinearity with a very short memory) cascaded with a long linear impulse response. When this structure is taken into account, the number of coefficients can be reduced to a manageable size [45.51].

This concludes our discussion of single-channel acoustic echo cancelation. We turn next to the most modern application of echo cancelers – multichannel acoustic echo cancelers.

45.3 Multichannel Acoustic Echo Cancelation

The most common use of acoustic echo cancelers today is in single-channel communication. Recently, however, there has been a growing interest in stereophonic (or, more generally, multichannel) echo cancelation. The need for such cancelers arises in several applications where the echo picked up by a microphone is due to two or more different, but correlated signals.

The primary motivation for developing stereophonic cancelers is the desire for high-quality teleconferencing between a group of people in one room with another group at a different location. At present such teleconferencing is almost exclusively monophonic. However, it is our belief that in the future many, if not all, teleconferences will be stereophonic. A stereophonic system has several advantages when compared to a monophonic system. With a stereo system each group gets a much better feeling of *presence* at the remote location. Imagine, for instance, half of a circular table around which several people are seated. The table is placed adjacent to a large video screen on which the remote conference room is projected, with several people seated at the *other* half of the table, thereby creating the impression of a circular table with people seated around it. If a monophonic system is used for such a teleconference it would be very difficult to associate voices at the remote location with their respective talkers. With a multichannel system (or at a minimum a stereo system) the task is much easier because sounds can be made to appear to come from the directions of their sources. Not only does a multichannel system provide a much more natural interaction, it also improves intelligibility, especially when two or more people happen to be talking simultaneously. This is due to the well-known *cocktail party* effect – the ability of humans to pay attention to one speaker at a cocktail party, even when several other conversations are taking place simultaneously. The mechanisms by which humans are able to do this are not well understood. However, the ability to identify the directions from which sounds are received is a prerequisite for the cocktail party effect. If one listens to a monophonic recording of a cocktail party, it is very difficult to pick out one conversation.

Besides such teleconferencing, there are several other potential applications for stereophonic echo cancelers. One example is a so called desktop teleconference in which the conferees are all at different locations. Each conferee has a desktop with pictures, or videos of all the other conferees displayed on the screen. As in the conference described in the previous paragraph, the interaction between the conferees is improved if a participant's voice appears to come from the direction of his or her picture (or video). A minimum of two loudspeakers and one microphone are needed at each location to realize this. Another application might arise if one uses

a microphone in a car to communicate with, say, a speech recognizer. For a natural interaction, one would like to be able to do so while the stereo system in the car is in use. This requires a stereo canceler to cancel the signals from the stereo system. Finally, multichannel cancelers are needed for interactive games involving multichannel sound.

At first glance, stereo echo cancelation appears to be just a minor modification of single-channel cancelation. However going from a single channel to two channels produces a basic change in the requirements of an echo canceler. Extension to more than two channels does not introduce any new problems. Hence we will discuss only stereo cancelers here. The extension of the ideas to a system with three or more channels is fairly straightforward.

In order to understand the fundamental problem of stereo acoustic echo cancelation [45.52], let us first describe the configuration of a stereo teleconference. Figure 45.10 shows a schematic of such a conference. As before, we will consider echo cancelation for one end of the circuit (room B), it being understood that analogous configurations are used for echo cancelation in room A. For a discussion of echo cancelation in room B, we will refer to room A as the transmitting room and room B as the receiving room. Each room has two loudspeakers and two microphones. The microphones in the transmitting room pick up signals $x_1(n)$ and $x_2(n)$. These two signals are transmitted to loudspeakers in the receiving room. Each microphone in the receiving room picks up an echo of *both* signals, linearly combined via the acoustic echo paths to the microphone from the two loudspeakers. To keep the discussion simple we will consider the cancelation of the echo picked up by only one of the microphones, as shown in Fig. 45.10, and denote the impulse responses from the loudspeakers to that microphone by $h_1(n)$ and $h_2(n)$. A similar discussion applies to cancelation of the echo at the other microphone, with $h_1(n)$ and $h_2(n)$ replaced by the responses appropriate to that microphone. (For convenience, whenever we refer to the impulse response of an acoustic path we will assume that it includes the responses of the loudspeaker and microphone in that path.)

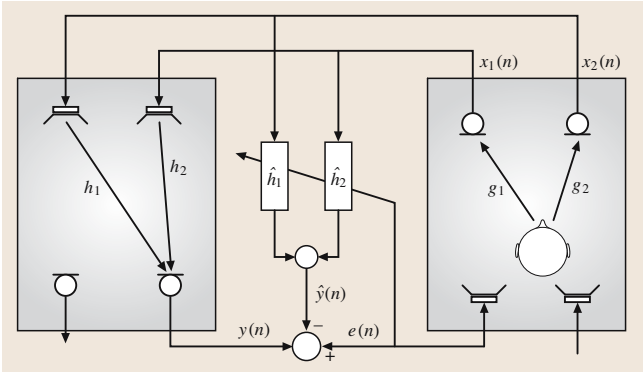


Fig. 45.10 Schematic diagram of stereophonic echo cancelation, showing the use of adaptive filters $\hat{h}_1$ and $\hat{h}_2$ to cancel the echo $y(n)$ arising from the echo paths h_1 and h_2 from the two loudspeakers on the left to one of the microphones

Neglecting ambient noise and signals generated in the receiving room, the signal picked up by the microphone is

$$y(n) = \boldsymbol{h}_1^{\mathrm{T}}\boldsymbol{x}_1(n) + \boldsymbol{h}_2^{\mathrm{T}}\boldsymbol{x}_2(n)\,, \tag{45.35}$$

where we have defined the vectors $\boldsymbol{h}_1$, $\boldsymbol{h}_2$, $\boldsymbol{x}_1$, and $\boldsymbol{x}_2$ analogously to the impulse response and far-end signal vectors described in the single-channel case. In principle the vectors $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ can have different dimensions. However, without loss of generality we will assume that they have an equal number of components, L, each. The vectors $\boldsymbol{x}_1(n)$ and $\boldsymbol{x}_2(n)$ are then also of the same dimension L. As a straightforward generalization of the single-channel case we now implement two adaptive filters whose outputs are added together to generate a replica of the echo. Denoting the current estimated impulse response vectors of the two adaptive filters as $\hat{\boldsymbol{h}}_1(n)$ and $\hat{\boldsymbol{h}}_2(n)$, we get

$$\hat{y}(n) = \hat{\boldsymbol{h}}_1^{\mathrm{T}}(n)\boldsymbol{x}_1(n) + \hat{\boldsymbol{h}}_2^{\mathrm{T}}(n)\boldsymbol{x}_2(n)\,, \tag{45.36}$$

and the error signal

$$e(n) = y(n) - \hat{y}(n) = \boldsymbol{r}_1^{\mathrm{T}}(n)\boldsymbol{x}_1(n) + \boldsymbol{r}_2^{\mathrm{T}}(n)\boldsymbol{x}_2(n)\,, \tag{45.37}$$

where $\boldsymbol{r}_1(n)$ and $\boldsymbol{r}_2(n)$ are the current misalignment vectors for the two channels. Let us define the *stacked* vector $\boldsymbol{x}(n)$ as the concatenation of $\boldsymbol{x}_1(n)$ and $\boldsymbol{x}_2(n)$ (i. e., $\boldsymbol{x}(n) = [\boldsymbol{x}_1^{\mathrm{T}}(n)\boldsymbol{x}_2^{\mathrm{T}}(n)]^{\mathrm{T}}$). Similarly define the stacked vectors $\boldsymbol{h}$, $\hat{\boldsymbol{h}}(n)$, and $\boldsymbol{r}(n)$. In terms of these vectors the expressions for $y(n)$, $\hat{y}(n)$, and $e(n)$ are exactly the same as the ones in the section on the stochastic gradient algorithm. Therefore one would expect to be able to use the NLMS algorithm of (45.6) in the ideal case to drive the misalignment vector $\boldsymbol{r}(n) \to 0$, i. e., $\hat{\boldsymbol{h}}_1(n) \to \boldsymbol{h}_1$ and $\hat{\boldsymbol{h}}_2(n) \to \boldsymbol{h}_2$. This turns out not to be true. In the ideal case, the NLMS algorithm does drive the error to zero. However, the misalignment does not necessarily converge to zero. This is because, as shown in the next subsection, the condition of zero error can be achieved by infinitely many nonzero vectors $\boldsymbol{r}(n)$. Thus, although

Amazon EX1010 Page 925

$\boldsymbol{r}(n) = 0$ implies $e(n) = 0$, the converse is not true. The condition $e(n) = 0$ only implies a relationship between the impulse responses of the receiving room and the transmission room. This nonuniqueness is the basic difference between single- and two-channel cancelers. It arises when only one source is active in the transmitting room, and is due to the fact that in such a case the signals $x_1(n)$ and $x_2(n)$ are highly correlated.

45.3.1 Nonuniqueness of the Misalignment Vector

The nonuniqueness of the misalignment vector is a property of the stereo configuration, and is independent of the particular adaptation algorithm used. Thus, ignoring for the moment the question of *how* convergence is achieved, let us just assume that the error has been driven to be identically equal to zero. We will show that the setting of the adaptive filters that achieves this is not unique.

Let us consider the nonuniqueness problem in terms of the signals and impulse responses as time series. An equivalent vector–matrix treatment is, of course, possible, but is somewhat more cumbersome.

The error $e(n)$ can be written

$$e(n) = r_1(n) * x_1(n) + r_2(n) * x_2(n) \,, \tag{45.38}$$

where, as before, $*$ denotes convolution, and $r_1(n)$ and $r_2(n)$ are the misalignments in the two channels. The nonuniqueness problem arises when a single source is active in the transmitting room, as shown in Fig. 45.10. In such a case, the source signal $s(n)$ is convolved with the two transmitting room impulse responses $g_1(n)$ and $g_2(n)$ to generate the two signals $x_1(n)$ and $x_2(n)$. That is $x_1(n) = s(n) * g_1(n)$ and $x_2(n) = s(n) * g_2(n)$. The simplest way to show the nonuniqueness is to note that, since convolution is commutable,

$$g_2(n) * x_1(n) = g_2(n) * g_1(n) * s(n) = g_1(n) * x_2(n) \,. \tag{45.39}$$

In view of (45.39), the expression for the error in (45.38) can be rewritten

$$\begin{aligned} e(n) = {} & [r_1(n) + \alpha g_2(n)] * x_1(n) \\ & + [r_2(n) - \alpha g_1(n)] * x_2(n) \,, \end{aligned} \tag{45.40}$$

where α is an arbitrary constant. Clearly the settings of the adaptive filters $\hat{h}_1(n)$ and $\hat{h}_2(n)$ that achieve zero error are not unique.

Another way to show the nonuniqueness is to rewrite the error in (45.38) as

$$e(n) = [r_1(n) * g_1(n) + r_2(n) * g_2(n)] * s(n) \,. \tag{45.41}$$

For the error to be identically zero for all $s(n)$, the expression in square brackets must be zero, i. e.,

$$r_1(n) * g_1(n) + r_2(n) * g_2(n) = 0 \,. \tag{45.42}$$

In the z-transform domain (45.42) is

$$R_1(z)G_1(z) + R_2(z)G_2(z) = 0 \,, \tag{45.43}$$

where the upper-case quantities denote the z-transforms of the corresponding lower-case quantities. Clearly, (45.42) and (45.43) *do not* imply that the misalignment in each channel is zero. Only the expression involving the two misalignments has to be zero.

The problem with stereo cancelers is evident from (45.43) [or equivalently from (45.42)]. Even if the receiving room impulse responses are constant, any change in G_1 and/or G_2 requires a readjustment of $r_1(n)$ and $r_2(n)$, i. e., of the adaptive filters $\hat{h}_1(n)$ and $\hat{h}_2(n)$. Thus the adaptation algorithm must track not only the variations in the receiving room responses, but also the variations in the transmitting room responses. Given the long impulse responses in a room, it is barely possible to track the changes in just the receiving room alone. To track changes in *both* rooms is quite difficult. Furthermore, the transmitting room responses are particularly difficult to track. If, for instance, one talker stops and another starts speaking the responses $g_1(n)$ and $g_2(n)$ change quite abruptly and by large amounts. The difficult challenge therefore, is to devise an algorithm whose convergence (as in the case of a single-channel canceler) is unaffected by changes in the transmitting room.

Before we discuss currently available solutions to the nonuniqueness problem, let us make two observations concerning stereo cancelation. First, the nonuniqueness problem essentially disappears if two or more statistically independent and spatially separated sources are simultaneously active in the transmitting room. This is because R_1 and R_2 would have to satisfy an equation of the type of (45.43) for each additional independent source, with impulse responses appropriate to that source. Except in some highly improbable situation these equations will be linearly independent. Therefore the only possible solution would be $R_1(z) = R_2(z) = 0$. However, the fact that the solution is unique when two or more sources are active is not of much use for a teleconference because most of the time only one person speaks at a time.

Amazon EX1010 Page 926

The second observation is that all room responses have tails extending beyond the length L modeled by the adaptive filter. Therefore, theoretically there is the possibility that the frequency response of the transmitting room impulse responses varies more rapidly with frequency than is possible for the adaptive filter response to replicate. This too would tend to force the misalignment to zero in order to satisfy (45.43). However, in any practical situation this fact is also of little consequence because, in any usable implementation of an echo canceler, L has to be large enough that the tails are quite negligible. We will mention the effect of these tails a bit more quantitatively in the following section.

45.3.2 Solutions for the Nonuniqueness Problem

Examination of (45.38) suggests a possible solution, first proposed in [45.52]. If the signals $x_1(n)$ and $x_2(n)$ were uncorrelated they would not be able to compensate for each other. The minimum of the error (zero in the ideal case) would then be obtained only when the two misalignments $r_1(n)$ and $r_2(n)$ were both zero.

Thus, one could try decorrelating the two-channel signals. Of course, $x_1(n)$ and $x_2(n)$ cannot be completely decorrelated because then the stereo effect itself would be lost. The challenge, therefore, is to decorrelate them by an amount adequate to make the adaptive algorithm converge, yet small enough to be perceptually negligible.

Several attempts have been made to exploit this idea. One simple proposal is to add uncorrelated noise signals to $x_1(n)$ and $x_2(n)$. Adding white Gaussian noise does improve convergence [45.52] but it needs to be at a level that degrades the perceptual quality of the signals. Adding so-called Schroeder noise is still not satisfactory. [Schroeder noise for a signal $x(n)$ is given by $\varepsilon\gamma(n)x(n)$, where $\gamma(n)$ are independent identically distributed random variables that take on the values ± 1 with equal probability, and the constant ε controls the level of the noise. This noise has the property that it is white and has a time-varying variance that is proportional to the local variance of $x(n)$. Therefore it is perceptually less disturbing than white noise of constant variance.]

Another method of decorrelation is to introduce a time-varying filter in one of the channels, say $x_1(n)$. One suggestion [45.53] is to periodically introduce a one-sample delay. To get a rough idea of the effect of a one-sample delay, let us make the approximation $x_1(n-1) \approx x_1(n) - Tx_1'(n)$, where T is the sampling interval and x_1' denotes the derivative of x_1. With this approximation it is seen that the proposal is essentially equivalent to periodically adding a noise proportional to the derivative of the signal. Another suggestion of this general nature is [45.54] to insert an all pass filter that introduces a frequency-dependent group delay whose magnitude is randomly varying in time. Again, because of the small amount of delay introduced, this is equivalent to adding a frequency shaped noise.

A very effective way of dealing with the nonuniqueness was proposed in [45.55]. The proposal is to *distort* the signals $x_1(n)$ and $x_2(n)$ by passing each through a zero-memory nonlinearity. The rationale for this rather drastic proposal is as follows. Consider the RLS algorithm. The optimal filter is obtained as the solution of (45.29). For the present discussion the recursive update of the estimate is not of interest. So to keep the notation simple we will omit the time index n on all vectors and matrices. The important point is that, if there is a single active source in the transmitting room, and the transmitting room impulse responses are of length $\leq L$, then the correlation matrix $\boldsymbol{R}$ is readily shown to be singular. This is a direct consequence of the linear relation between the signals $x_1(n)$ and $x_2(n)$ given in (45.39). Since room responses are infinitely long, as noted at the end of the last section, $\boldsymbol{R}$ is in principle not singular but highly ill-conditioned.

For the two-channel case the matrix $\boldsymbol{R}$ has the structure

$$\boldsymbol{R} = \begin{pmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{R}_{21} & \boldsymbol{R}_{22} \end{pmatrix} , \tag{45.44}$$

where the submatrices $\boldsymbol{R}_{pq} = \sum_{m=-\infty}^{n} \lambda^{n-m} \boldsymbol{x}_p(m) \boldsymbol{x}_q^{\mathrm{T}}(m)$, for $p, q = 1, 2$. Asymptotically, for large L these submatrices become circulant matrices. Hence it follows that

$$\boldsymbol{R}_{pq} = \boldsymbol{F}^{-1} \boldsymbol{S}_{pq} \boldsymbol{F} , \quad p, q = 1, 2 , \tag{45.45}$$

where $\boldsymbol{F}$ is the discrete Fourier transform matrix, and $\boldsymbol{S}_{pq}$ is a diagonal matrix whose diagonal elements $S_{pq}(f)$, $f = 0, \ldots, L-1$, constitute the discrete Fourier transform of the first row of $\boldsymbol{R}_{pq}$. Thus S_{11} and S_{22} are the auto spectra and S_{12} and S_{21} the cross spectra of $x_1(n)$ and $x_2(n)$.

This fact can be used to show that the eigenvalues of $\boldsymbol{R}$ are lower bounded by $1 - |\gamma(f)|^2$, where $\gamma(f)$ is the coherence function at frequency f, defined as

$$\gamma(f) = \frac{S_{12}(f)}{\sqrt{S_{11}(f) S_{22}(f)}} . \tag{45.46}$$

If $|\gamma(f)|$ becomes equal to 1 at any of the discrete frequencies, the matrix becomes singular.

Amazon EX1010 Page 927

The aim of decorrelating the signals $x_1(n)$ and $x_2(n)$ is to reduce $|\gamma(f)|$ at all f. Additive uncorrelated noise, as well as the time-varying all-pass filters, decreases the coherence function. However, as pointed out earlier the amount of additive noise needed for proper convergence is unacceptably high. What about the possibility of reducing the coherence by some *fixed transformation* of the signals? A linear transformation will not do because the coherence function is invariant under linear transformations. Hence the proposal is to try some nonlinear transformation.

We have given the above derivation in some detail because it turns out that a very simple zero-memory nonlinearity works rather well. That nonlinear transformation [45.55] is obtained by adding to the signal a full-wave rectified version of itself, scaled by a constant α that controls the amount of nonlinearity. The nonlinearity can be written

$$\tilde{x}_i(n) = x_i(n) + \alpha|x_i(n)| \,, \quad i = 1, 2 \,. \tag{45.47}$$

To deal with certain pathological situations (almost never encountered in practice) α is replaced by $-\alpha$ in one of the channels.

Such a nonlinearity allows convergence of the misalignment. Yet, somewhat surprisingly, for speech signals it is hardly perceptible for α even as large as 0.2. The reason we surmise is that voiced portions of a speech signal are rich in harmonics of the fundamental frequency; hence the distortion products are hidden under existing harmonics. Several other transformations have been tried [45.56] but have not shown any improvement over the simple nonlinearity shown in (45.47).

Although this nonlinear transformation provides a good solution to the nonuniqueness problem for teleconferencing, it does not provide a completely satisfactory solution for music signals. For complex musical signals consisting of several instruments playing together, the distortion is almost imperceptible, as in the case of speech signals. However, for certain music signals the nonlinearity introduces an unacceptable degradation. The signal from a flute, for instance, is almost a pure sinusoid. The distortion products for such a signal are quite perceptible.

Even after the introduction of the nonlinearity of (45.47) the reduction in coherence is fairly small in several regions of the spectrum. Hence for good performance a rapidly converging algorithm like RLS is still required.

An improved use of nonlinear distortion is presented in [45.56]. The distortion used is the one shown in (45.47), except that in one of the channels α is multiplied by a time-varying parameter σ. The magnitude of σ is unity, but its sign is flipped at every positive-going zero crossing of the signal. The signals $\alpha|x_1(n)|$ and $\sigma\alpha|x_2(n)|$ are shown to be uncorrelated with each other and with the signals $x_1(n)$ and $x_2(n)$. However, the main modification introduced is that *four* adaptive filters are used to model the echo instead of the usual two. Two of these filters, $\hat{h}_1$ and $\hat{h}_2$, have the signals $x_1(n)$ and $x_2(n)$ as inputs, and the other two filters, $\overline{h}_1$ and $\overline{h}_2$, have the signals $\alpha|x_1(n)|$ and $\sigma\alpha|x_2(n)|$, respectively, as inputs. The sum of the outputs of all four adaptive filters is subtracted from the echo to give the error signal that controls all four adaptive filters. It has been shown that the filters $\overline{h}_1$ and $\overline{h}_2$ adapt quite rapidly because their inputs are uncorrelated with each other as well as with $x_1(n)$ and $x_2(n)$. The actual cancelation is done by two additional filters whose coefficients are copied from the filters $\overline{h}_1$ and $\overline{h}_2$ in a manner similar to that used in the two-echo-path model described in Sect. 45.2.5. The adaptation is claimed to be rapid enough so that the NLMS algorithm is adequate. This advantage is offset by the fact that one now needs two more adaptive filters and two canceling filters.

45.4 Summary

This chapter dealt with the problem of echoes that arise during a telephone conversation. The traditional method for dealing with echoes on telephone circuits was a device known as an echo suppressor. It was shown that when echoes arrive with a long delay, e.g., in satellite communication, echo suppressors do not provide a satisfactory solution. During the 1960s a new device, called an echo *canceler*, was developed to deal with this problem. This chapter described the operation of such a device in great detail. Also discussed were more-recent applications of echo cancelers to single- and multichannel teleconferencing.

Amazon EX1010 Page 928

References

45.1 M.M. Sondhi: An adaptive echo canceler, Bell Syst. Tech. J. **46**, 497–511 (1967)

45.2 M.M. Sondhi: Echo Canceller, U.S. Patent 3499999 (1970), (filed Oct. 31, 1966).

45.3 J.L. Kelly, B.F. Logan: Self-adjust echo suppressor, U.S. Patent 3500000 (1970), (filed Oct. 31, 1966).

45.4 D.L. Duttweiler, Y.S. Chen: A single chip VLSI echo canceller, Bell Syst. Tech. J. **59**, 149–160 (1980)

45.5 R.R. Reisz, E.T. Klemmer: Subjective evaluation of delay and echo suppressors in telephone communications, Bell Syst. Tech. J. **42**, 2919–2943 (1963)

45.6 G. Williams, L.S. Moye: Subjective evaluation of unsuppressed echo in simulated long-delay telephone communications, Proc. Inst. Electr. Elect. **118**, 401–408 (1971)

45.7 B. Widrow, M.E. Hoff Jr.: Adaptive switching circuits, IRE Wescon Conf. Rec. (Institute of Radio Engineers, New York 1960) pp. 96–104, Part 4

45.8 S.P. Applebaum: Adaptive arrays, IEEE Trans. Antennas Propagation **24**, 573–598 (1976)

45.9 M.M. Sondhi, D. Mitra: New results on the performance of a well-known class of adaptive filters, Proc. IEEE **64**, 1583–1597 (1976)

45.10 S. Haykin: *Adaptive Filter Theory* (Prentice-Hall, Englewood Cliffs 1996) pp. 748–751

45.11 A. Weiss, D. Mitra: Digital adaptive filters: Conditions for convergence, rates of convergence, and effects of noise and errors arising from the implementation, IEEE Trans. Inform. Theory **25**, 637–652 (1979)

45.12 D.L. Duttweiler: A twelve-channel digital echo canceler, IEEE Trans. Commun. **26**, 647–653 (1978)

45.13 D.L. Duttweiler: Proportionate normalized least-mean-squares adaptation in echo cancelers, IEEE Trans. Speech Audio Process. **8**, 508–518 (2000)

45.14 J. Benesty, Y. Huang, J. Chen, P.A. Naylor: Adaptive algorithms for the identification of sparse impulse responses. In: *Topics in Acoustic Echo and Noise Control*, ed. by E. Haensler, G. Schmidt (Springer, Berlin, Heidelberg 2006)

45.15 J. Benesty, S.L. Gay: An improved PNLMS algorithm, Proc. IEEE ICASSP **2002**, 1881–1884 (2002)

45.16 S. Amari: Natural gradient works efficiently in learning, Neural Comput. **10**, 251–276 (1998)

45.17 S.L. Gay, S.C. Douglas: Normalized natural gradient adaptive filtering for sparse and nonsparse systems, Proc. IEEE ICASSP **2002**, 1405–1408 (2002)

45.18 T. Gaensler, S.L. Gay, M.M. Sondhi, J. Benesty: Double-talk robust fast converging algorithms for network echo cancellation, IEEE Trans. Speech Audio Process. **8**, 656–663 (2000)

45.19 R.W. Harris, D.M. Chabries, F.A. Bishop: A variable step (VS) adaptive filter algorithm, IEEE Trans. Acoust. Speech Signal Process. **34**, 309–316 (1986)

45.20 S. Marple: Efficient least-squares FIR system identification, IEEE Trans. Acoust. Speech Signal Process. **29**, 62–73 (1981)

45.21 L. Ljung, M. Morf, D.D. Falconer: Fast calculation of gain matrices for recursive estimation schemes, Int. J. Control **27**, 1–19 (1978)

45.22 D.D. Falconer, L. Ljung: Application of fast Kalman estimation to adaptive equalization, IEEE Trans. Commun. **26**, 1439–1446 (1978)

45.23 J. Cioffi, T. Kailath: Fast, recursive-least-squares transversal filters for adaptive filtering, IEEE Trans. Acoust. Speech Signal Process. **34**, 304–337 (1984)

45.24 E. Eweda: Comparison of RLS, LMS, and sign algorithms for tracking randomly time-varying channels, IEEE Trans. Signal Process. **42**, 2937–2944 (1994)

45.25 K. Ozeki, T. Umeda: An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties, Electron. Commun. Jpn. **67-A**, 19–27 (1984)

45.26 S.L. Gay: The fast affine projection algorithm. In: *Acoustic Signal Processing for Telecommunication*, ed. by S.L. Gay, J. Benesty (Kluwer Academic, Boston 2000), Chap. 2

45.27 S.G. Kratzer, D.R. Morgan: The partial-rank algorithm for adaptive beamforming, Proc. SPIE Int. Soc. Optic. Eng. **564**, 9–14 (1985)

45.28 S.L. Gay, S. Tavathia: The fast affine projection algorithm, Proc. IEEE ICASSP **1995**, 3023–3026 (1995)

45.29 E. Haensler: The hands-free telephone problem – an annotated bibliography, Signal Process. **27**, 259–271 (1992)

45.30 M.M. Sondhi, W. Kellermann: Adaptive echo cancellation for speech. In: *Advances in Speech Signal Processing*, ed. by S. Furui, M.M. Sondhi (Marcel Dekker, New York 1992), Chap. 11

45.31 I. Furukawa: A design of canceller of broad band acoustic echo, Int. Teleconf. Symp. (IEEE, New York 1984) pp. 232–239

45.32 W. Kellermann: Kompensation akustischer Echos in Frequenzteil-Bändern. In: *Proc. Aachener Kolloquium*, ed. by L. Butzer (RWTH Aachen, Aachen 1984) pp. 322–325, (in German)

45.33 W. Kellermann: Analysis and design of multirate systems for cancellation of acoustical echoes, Proc. IEEE ICASSP **1998**, 2570–2573 (1998)

45.34 P.P. Vaidyanathan: *Multirate Systems and Filter Banks* (Prentice-Hall, Englewood Cliffs 1992)

45.35 M. Vetterli: A theory of multirate filter banks, IEEE Trans. Acoust. Speech Signal Process. **35**, 356–372 (1987)

45.36 D.R. Morgan: Slow asymptotic convergence of LMS acoustic echo cancelers, IEEE Trans. Speech Audio Process. **3**, 126–136 (1995)

Amazon EX1010 Page 929

45.37 P.L. DeLeon, D.M. Etter: Experimental results with increased bandwidth analysis filters in over sampled sub-band acoustic echo cancelers, IEEE Signal Process. Lett. **2**, 1–3 (1995)

45.38 B. Hatty: Recursive least squares algorithms using multirate systems for cancellation of acoustical echoes, Proc. IEEE ICASSP **90**, 1145–1148 (1990)

45.39 D.R. Morgan, J. Thi: A delayless sub-band adaptive filter, IEEE Trans. Signal Process. **43**, 1819–1830 (1995)

45.40 J. Chen, H. Bes, J. Vandewalle, P. Janssens: A new structure for sub-band acoustic echo canceler, Proc. IEEE ICASSP **1988**, 2574–2577 (1988)

45.41 M. Dentino, J. McCool, B. Widrow: Adaptive filtering in the frequency domain, Proc. IEEE **66**, 1658–1659 (1978)

45.42 E.R. Ferrara Jr.: Fast implementation of LMS adaptive filter, IEEE Trans. Acoust. Speech Signal Process. **28**, 474–475 (1980)

45.43 D. Mansour, A.H. Gray Jr.: Unconstrained frequency-domain adaptive filter, IEEE Trans. Acoust. Speech Signal Process. **30**, 726–734 (1982)

45.44 J.-S. Soo, K.K. Pang: Multidelay block frequency domain adaptive filter, IEEE Trans. Acoust. Speech Signal Process. **38**, 373–376 (1990)

45.45 E. Moulines, O. Ait Amrane, Y. Grenier: The generalized multidelay adaptive filter: structure and convergence analysis, IEEE Trans. Signal Process. **43**, 14–28 (1995)

45.46 J. Benesty, T. Gaensler, D.R. Morgan, M.M. Sondhi, S.L. Gay: *Advances in Network and Acoustic Echo Cancellation* (Springer, Berlin, Heidelberg 2001)

45.47 K. Ochiai, T. Araseki, T. Ogihara: Echo canceler with two echo path models, IEEE Trans. Commun. **COM-25**(6), 589–595 (1977)

45.48 E.J. Diethorn: Improved decision logic for two-path echo cancelers. In: *Proc. IEEE Workshop on Acoustic Echo and Noise Control*, ed. by E. Haensler, P. Dreiseitel (IEEE, New York 2001)

45.49 S. Makino, Y. Kaneda, N. Koizumi: Exponentially weighted step size NLMS adaptive filter based on the statistics of a room impulse response, IEEE Trans. Speech Audio Process. **1**, 101–108 (1993)

45.50 E.J. Thomas: Some considerations on the application of Volterra representation of nonlinear networks to adaptive echo cancellers, Bell Syst. Tech. J. **50**, 2797–2805 (1971)

45.51 A. Stenger, L. Trautmann, R. Rabenstein: Nonlinear acoustic echo cancellation with 2nd order adaptive Volterra filters, Proc. IEEE ICASSP **1999**, 877–880 (1999)

45.52 M.M. Sondhi, D.R. Morgan, J.L. Hall: Stereophonic acoustic echo cancellation – an overview of the fundamental problem, IEEE Signal Process. Lett. **2**, 148–151 (1995)

45.53 Y. Joncour, A. Sugiyama: A stereo echo canceller with pre-processing for correct echo path identification, Proc. IEEE ICASSP **1998**, 3677–3680 (1998)

45.54 M. Ali: Stereophonic echo cancellation system using time-varying all-pass filtering for signal decorrelation, Proc. IEEE ICASSP **1998**, 3689–3692 (1998)

45.55 J. Benesty, D.R. Morgan, M.M. Sondhi: A better understanding and an improved solution to the specific problems of stereophonic acoustic echo cancellation, IEEE Trans. Speech Audio Process. **6**, 156–165 (1998)

45.56 S. Shimauchi, Y. Haneda, S. Makino, Y. Kaneda: New configuration for a stereo echo canceller with nonlinear pre-processing, Proc. IEEE ICASSP **1998**, 3685–3688 (1998)

Amazon EX1010 Page 930

46. Dereverberation

Y. Huang, J. Benesty, J. Chen

Room reverberation is one of the two major causes (the other is background noise) of speech degradation and there has been an increasing need for speech dereverberation in various speech processing and communication applications. Although it has been studied for decades, speech dereverberation remains a challenging problem from both a theoretical and practical perspective. This chapter provides a methodical overview of speech dereverberation algorithms. They are classified into three categories: source model-based speech enhancement algorithms for dereverberation, separation of speech and reverberation via homomorphic transformation, and speech dereverberation by channel inversion and equalization. We outline the basic ideas behind these approaches, explain the assumptions they make, and discuss their characteristics as well as performance.

46.1 **Background and Overview** 929
46.1.1 Why Speech Dereverberation? 929
46.1.2 Room Acoustics and Reverberation Evaluation 930
46.1.3 Classification of Speech Dereverberation Methods 930
46.2 **Signal Model and Problem Formulation** .. 931
46.3 **Source Model-Based Speech Dereverberation** 932
46.3.1 Speech Models 932
46.3.2 LP Residual Enhancement Methods 933
46.3.3 Harmonic Filtering 934
46.3.4 Speech Dereverberation Using Probabilistic Models 935
46.4 **Separation of Speech and Reverberation via Homomorphic Transformation** 936
46.4.1 Cepstral Liftering 936
46.4.2 Cepstral Mean Subtraction and High-Pass Filtering of Cepstral Frame Coefficients 936
46.5 **Channel Inversion and Equalization** 937
46.5.1 Single-Channel Systems 937
46.5.2 Multichannel Systems 938
46.6 **Summary** 941
References 942

46.1 Background and Overview

46.1.1 Why Speech Dereverberation?

Speech is probably the most natural and efficient form of human–human communication and arguably for human–machine interaction. But voice in rooms is subject to degradation caused by acoustic reverberation in addition to ambient noise.

Reverberation is the collection of reflected sounds from the surfaces in an enclosure. It adds *warmth* to sound, which is essential for music, and helps people better orient themselves in the listening environment. However, room reverberation leads to temporal and spectral smearing, which would distort both the envelope and fine structure of a speech signal. As a result, speech becomes difficult to be understood in the presence of room reverberation, especially for hearing-impaired and elderly people [46.1] and for automatic speech recognition systems [46.2, 3]. This gives rise to a strong need for effective speech dereverberation algorithms.

A speech dereverberation algorithm employing either one or multiple microphones is intended to suppress or compensate the effect of room reverberation such that the original clean speech can be faithfully recovered or its intelligibility is improved. Since neither the source speech signal nor the acoustic channel impulse responses are known a priori, speech dereverberation is a difficult, blind problem. This practical importance and theoretical challenge have inspired great amounts of research into this area in the last four decades. While significant progresses have been made, it is still far away from

Amazon EX1010 Page 931

claiming victory in the battle against room reverberation. This chapter provides an overview of the state of the art of research into speech dereverberation problem.

46.1.2 Room Acoustics and Reverberation Evaluation

Before we present various speech dereverberation methods, it is important to get an idea as to how reverberation impairs speech quality and how we can quantitatively describe the level of reverberation that proportionally indicates the amount of reduction in speech intelligibility.

Many psychoacoustic studies have reported that human perceptions of reflected sounds with short and long delays are different. It was clearly explained in [46.4] that, for long delays, a clutter of separate echoes is heard; but if the delay is short, the ear cannot distinguish the time difference between the direct and reverberant sounds. As such, a room impulse response can be broken into three parts: direct-path response, early and late reverberations, as illustrated in Fig. 46.1 with a sample acoustic impulse response measured in the varechoic chamber at Bell Labs. The cutoff time delay k_c with respect to direct sound between early and late reverberations is about 50 ms for speech and 80 ms for music.

Due to the so-called *precedence effect* [46.5], early reverberations tend to perceptually reinforce the direct sound and are therefore considered harmless to speech intelligibility. However, the frequency response of early reverberations is rarely flat such that the speech spectrum is distorted and a change in speech quality is perceived. This procedure is known as *coloration*.

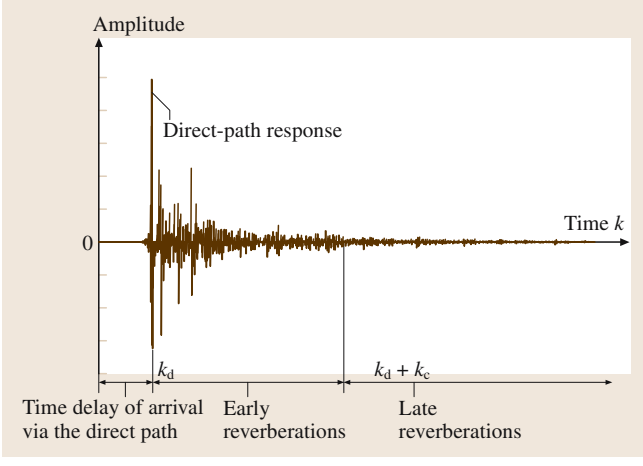


Fig. 46.1 Illustration of a sample acoustic impulse response measured in the varechoic chamber at Bell Labs

The impact of late reverberations on speech intelligibility can be interpreted from two perspectives. First, when examining the time sequence of a clean speech signal, one observes that the difference in level between a peak and its adjoining valley is evident. But after late reverberations add energy in the valleys between peaks, the difference becomes blurred and speech intelligibility gets worse. Second, reverberant sounds with long delays can stretch out to the following syllables or even next word and mask them as interference, which causes difficulty in understanding the meaning that those syllables and words convey. The degree of difficulty varies, depends on how long reverberant sounds persist and how strong they are relatively to the direct sound.

It is then easy to understand why the level of reverberation is typically measured by the reverberation time, which was introduced by *Sabine* in [46.6] and is defined as the length of time that it takes the reverberation to decay 60 dB from the level of the original sound. For measuring (or more precisely, estimating) the reverberation time of an acoustic impulse response, *Schroeder*'s backward integration [46.7] is probably the most widely used method; a general guideline was suggested in [46.8] as to how the measured impulse response can be accurately truncated at the knee where the main decay slope intersects the noise floor. The reverberation time for a typical room in a home or office ranges from 100 ms (slightly reverberant) to 600 ms (highly reverberant).

Another reverberation measure as an objective predicator of speech intelligibility is the early to late energy ratio (ELER) [46.9], which is defined in dB as follows:

$$\mathrm{ELER} \triangleq 10\log_{10}\left[\frac{\sum_{k=k_d}^{k_d+k_c-1} h^2(k)}{\sum_{k=k_d+k_c}^{k_d+L_h-1} h^2(k)}\right], \qquad (46.1)$$

where $h(k)$ is a discrete-time acoustic impulse response, k_d is the time delay of arrival via the direct path, k_c corresponds to the cutoff time delay separating early and late reverberations, and L_h is the length of the impulse response. This ratio is also known as the *clarity index*.

46.1.3 Classification of Speech Dereverberation Methods

There are different ways to classify various existing speech dereverberation algorithms, depending on what one is interested in. With the number of microphones as the feature to focus on, speech dereverberation can be classified into the classes of single- and multichan-

Amazon EX1010 Page 932

nel methods. But in this chapter, we care more about the underlying principles and therefore choose to classify speech dereverberation methods into the following three groups: source model-based speech dereverberation, separation of speech and reverberation via homomorphic transformation, and speech dereverberation by channel inversion and equalization. This classification is reflected by the structure of this chapter.

Source model-based speech dereverberation methods compute an estimate of the clean speech by using our a priori knowledge about the structure of clean speech signals and how they are distorted by room reverberation. Typical algorithms in this group include linear-predication (LP) residual enhancement methods [46.10–13], harmonic filtering [46.14], and speech dereverberation using probabilistic models [46.15]. They employ either speech production model or other statistical speech models. In Sect. 46.3, we will discuss these algorithms.

The reverberation process in the time domain is mathematically depicted as a convolution of a clean speech sequence with an unknown room impulse response. In the second group of methods, speech dereverberation is carried out from the perspective of signal separation in the light of cepstral analysis, which is a homomorphic transformation (see [46.16, Chap. 12] and the references therein for a more complete description of homomorphic systems and cepstral analysis). Since signals that are convolved in the time domain have complex cepstra that are additively combined, speech and reverberation can be more feasibly separated in the cepstral domain. There are two major techniques in this group: cepstral liftering [46.17–20], and cepstral mean subtraction/high-pass filtering of cepstral frame coefficients [46.21–23], which will be studied in Sect. 46.4.

Speech dereverberation by channel inversion and equalization seeks to estimate an inverse filter of the acoustic impulse response. This is a commonly used deconvolution approach. But the main problem is that the acoustic impulse response must be known, or blindly estimated, for successful dereverberation. Since there is a significant difference in difficulty and complexity of channel estimation and inversion for single- and multichannel systems, we further divide this class of speech dereverberation methods into the techniques for single- and multichannel systems. For a single-channel acoustic system, we have direct inverse, minimum mean square error (MMSE), and least-squares algorithms. For a multichannel acoustic system, we can use either beamforming [46.24], matched filtering [46.25], or multichannel inverse theorem (MINT) method [46.26]. All these algorithms will be developed in Sect. 46.5.

46.2 Signal Model and Problem Formulation

In a reverberant environment, the signal that a single microphone captures is the convolution of the source speech signal and the corresponding channel impulse response plus additive noise:

$$x(k) = s(k) * h + b(k) , \tag{46.2}$$

where $x(k)$, $s(k)$, and $b(k)$ are the microphone output, source speech signal, and background noise at time k, respectively, h is the channel impulse response, and $*$ denotes linear convolution. In acoustic signal processing, the channel impulse response is usually modeled as an finite impulse response (FIR) filter [46.27]. Moreover, since room acoustics change slower than speech, it is usually assumed that the channel impulse response is time invariant (at least short-time time invariant) for speech dereverberation. As a result, for ease of presentation and discussion, h in (46.2) is not considered as a function of time. Putting (46.2) into vector/matrix form, we get

$$x(k) = \boldsymbol{h}^{\mathrm{T}} \boldsymbol{s}(k) + b(k) , \tag{46.3}$$

where

$$\boldsymbol{h} = [\, h(0) \;\; h(1) \;\; \cdots \;\; h(L_h - 1) \,]^{\mathrm{T}} ,$$
$$\boldsymbol{s}(k) = [\, s(k) \;\; s(k-1) \;\; \cdots \;\; s(k - L_h + 1) \,]^{\mathrm{T}} ,$$

and L_h is the length of h.

When multiple microphones are available, we have a single-input multiple-output (SIMO) system and, similar to (46.3), each microphone output is given by

$$x_n(k) = \boldsymbol{h}_n^{\mathrm{T}} \boldsymbol{s}(k) + b_n(k) , \quad n = 1, 2, \cdots , N , \tag{46.4}$$

where N is the number of microphones, $x_n(k)$, $\boldsymbol{h}_n$, and $b_n(k)$ with respect to microphone n are similarly defined as in the case of single microphone.

The task of speech dereverberation is to recover $s(k)$, given only the observed microphone signal $x(k)$ or signals $x_n(k)$ $(n = 1, 2, \cdots , N)$, without assuming any a priori knowledge regarding h or h_n, respectively.

46.3 Source Model-Based Speech Dereverberation

Signal modeling is based on our knowledge about the signal source and has been found very useful for various signal processing and communication systems. In particular, source models are important to signal enhancement. A good signal model provides a concise and insightful description of how the signal is produced. Consequently, when applying the model, we enhance the signal in a smaller space of the model parameters such that a practical and effective system can be implemented in a very efficient manner. Following this line of ideas, we will study source model-based speech dereverberation methods first in this section.

46.3.1 Speech Models

Speech Production Model

Speech is produced by our articulation systems, which can be abstracted by a source-vocal-tract model. The traditional framework for speech analysis is centered around the speech production model first proposed by Homer *Dudley* at Bell Laboratories in the 1930s [46.28]. In this model, as depicted in Fig. 46.2, speech is produced by passing an excitation through a time-varying all-pole filter. The coefficients of the all-pole filter hinge on the shape of the vocal tract for the specific sound being generated. The excitation signal $e(k)$ is either a pulse train for voiced speech or random noise for unvoiced sound. The speech signal is then written as

$$s(k) = \sum_{p=1}^{P} a_{\mathrm{p}} s(k-p) + e(k) \,, \tag{46.5}$$

where P and the a_{p} are the order and the coefficients of the all-pole filter, respectively. By analyzing $s(k)$ with linear prediction [46.29,30], we can determine a_{p} and the excitation $e(k)$, which is also called LP residual signal.

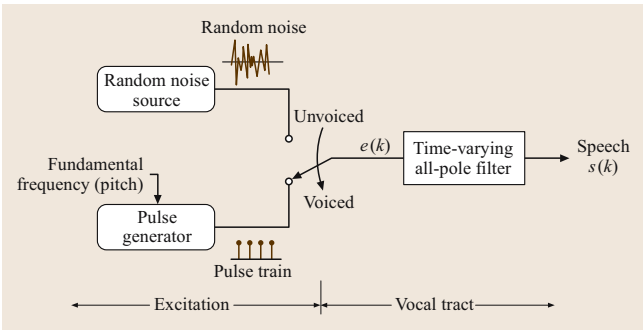


Fig. 46.2 The speech production model

Statistical Speech Models

Speech is an intrinsically random process. Therefore, in addition to describing a speech signal in a deterministic manner, we can alternatively choose to characterize speech with a statistical model. Nowadays, the most successful statistical speech model is the hidden Markov model (HMM). Its merits were recognized as the statistical pattern matching based on Bayes' theorem becomes a dominant approach to speech recognition. HMM keenly captures two levels of variation: one due to the uncertainty in the realization of a particular sound and the other due to the uncertainty in the sequential changes from one sound to another [46.31]. But HMM is too complex a model to be employed in a real-time speech dereverberation system.

For speech dereverberation, a simplified statistical speech model such as that proposed in [46.15] is desired. This is a probabilistic speech production model in which the joint probability density function (PDF) of a frame of clean speech samples $\boldsymbol{s}$, the LP feature vector $\boldsymbol{\theta} \triangleq [a_1, a_2, \cdots, a_P, \sigma_{\mathrm{e}}]^{\mathrm{T}}$, and the index c of a cluster of speech frame is expressed using the chain rule as follows:

$$p(\boldsymbol{s}, \boldsymbol{\theta}, c) = p(\boldsymbol{s}|\boldsymbol{\theta}, c)\, p(\boldsymbol{\theta}|c) p(c) \,. \tag{46.6}$$

Note that here we omit the frame index for ease of presentation. Moreover, the excitation is assumed to be a zero-mean Gaussian random process with variance σ_{e}^2 and then the conditional PDF $p(\boldsymbol{s}|\boldsymbol{\theta}, c)$ is calculated using the speech production model (46.5) as follows:

$$p(\boldsymbol{s}|\boldsymbol{\theta}, c) = p(e = x * a_{\mathrm{LPE}}) \sim \mathcal{N}\left(0, \sigma_{\mathrm{e}}^2\right) , \tag{46.7}$$

where a_{LPE} is the linear prediction error filter whose transfer function is given by

$$A_{\mathrm{LPE}}(z) = 1 - \sum_{p=1}^{P} a_{\mathrm{p}} z^{-p} \,,$$

and $\mathcal{N}(0, \sigma_{\mathrm{e}}^2)$ denotes a Gaussian distribution with mean 0 and variance σ_{e}^2. The LP coefficients and excitation are considered independent such that

$$p(\boldsymbol{\theta}|c) = p(a_1, a_2, \cdots, a_P|c) \cdot p(\sigma_{\mathrm{e}}|c) \,. \tag{46.8}$$

It is further asserted that the LP coefficients are Gaussian distributed and the nonnegative random variable σ_{e} is Gamma distributed. The parameters of the two distributions depend on the cluster. They are estimated by training and so is the a priori distribution $p(c)$. In [46.15], the cluster size was 256.

Amazon EX1010 Page 934

Finally, the distribution of a clean speech frame is obtained:

$$p(\boldsymbol{s}) = \sum_{c} \left[\int_{\boldsymbol{\theta}} p(\boldsymbol{s}, \boldsymbol{\theta}, c)\, \mathrm{d}\boldsymbol{\theta} \right] . \qquad (46.9)$$

For simplicity, the integration over all possible feature vectors ($\boldsymbol{\theta}$) and the summation over all speech clusters (c) are omitted in [46.15], which leads to

$$p(\boldsymbol{s}) \approx p(\boldsymbol{s}, \boldsymbol{\theta}, c) = p(\boldsymbol{s}|\boldsymbol{\theta}, c)p(\boldsymbol{\theta}|c)p(c) . \qquad (46.10)$$

46.3.2 LP Residual Enhancement Methods

According to the speech production model, speech is produced as an all-pole filter is excited by a pulse train or noise-like signal. Since an acoustic channel is usually modeled as an FIR filter whose transfer function consists of only zeros for speech dereverberation and many other acoustic signal processing problems, it was assumed that room reverberation would introduce only zeros into the microphone signals and, as a result, would primarily affect only the nature of the speech excitation sequence, having little impact on the all-pole filter [46.10]. Therefore, speech dereverberation can be accomplished by leaving the LP coefficients untouched and processing only the speech excitation signal obtained as the residual following LP analysis, as illustrated in Fig. 46.3.

LP residual of voiced speech is a well-structured pulse train, which would be smeared when room reverberation is present. Therefore reverberation effect can be clearly identified and readily eliminated in these periods of time. There have been a number of LP residual enhancement methods in the literature of speech dereverberation, which are based on this idea. In the following, we will review these methods in the order of their publication date:

Wavelet Transform Domain LP Residual Enhancement [46.10]

Brandstein and *Griebel* observed that a wavelet is the derivative of a smoothing function, which leads to the fact that a wavelet transform will contain local extrema at large changes in the examined signal. Therefore, they argued that impulses in the residual signal of voiced speech will manifest as extrema are detected. The desired wavelet extrema that correspond to the impulses in the clean LP residual are estimated via a clustering method and finally the LP residual sequence is enhanced.

Although the significance of the wavelet extrema in the unvoiced speech segments is not as clearly understood, the authors reported that this wavelet transform

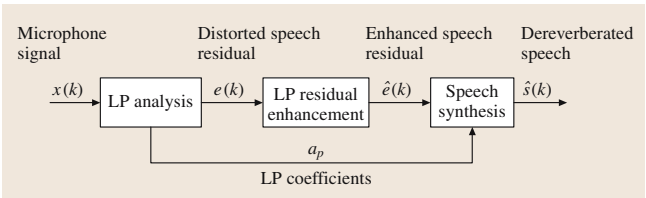


Fig. 46.3 Block diagram of an LP residual enhancement method for speech dereverberation

domain LP residual enhancement method is also effective in reconstructing a noise-like residual signal.

LP Residual Enhancement via Kurtosis Maximization [46.13]

Speech is a super-Gaussian random process whose kurtosis is greater than 3. [Kurtosis (Pearson kurtosis) is the fourth central moment divided by the fourth power of the standard deviation. Kurtosis measures the degree of peakedness of a distribution. The Gaussian distribution has a moderate tail and its kurtosis is 3. A super-Gaussian distribution has a high peak around its mean and a small tail, leading to a large kurtosis greater than 3. On the contrary, a sub-Gaussian distribution with a flat-topped curve and a long tail has a small kurtosis less than 3.] In the presence of room reverberation, a microphone signal is the mixture of the source speech signal and its delayed/attenuated copies. According to the central limit theorem, a mixture of independent non-Gaussian random variables is more like a Gaussian-distributed signal. Therefore, clean speech has a larger kurtosis compared to its reverberant observations. *Gillespie* et al. in [46.13] proposed to use kurtosis as a reverberation metric and developed an LP residual enhancement algorithm via kurtosis maximization.

LP Residual Enhancement via Hilbert Envelope Weighting [46.12]

Yegnanarayana et al. suggested to use the Hilbert envelope for LP residual reconstruction. For an LP residual signal $e(k)$, its Hilbert envelope is computed as

$$e_{\mathrm{HE}}(k) = \sqrt{e^2(k) + e_{\mathrm{H}}^2(k)} , \qquad (46.11)$$

where $e_{\mathrm{H}}(k)$ denotes the Hilbert transform of $e(k)$. The Hilbert transform is obtained by switching the real and imaginary parts of the discrete Fourier transform (DFT) of $e(k)$ and then taking the inverse DFT. The feature that the Hilbert envelope has a large amplitude at the instant of strong excitation makes it a good indicator of glottal closure, where an excitation pulse takes place. Therefore, applying the Hilbert envelope to an LP residual as

a weighting function has the effect of emphasizing the pulse train structure for voiced speech, which leads to an enhanced LP residual signal with less reverberation.

When there are multiple microphones, the Hilbert envelopes of the LP residuals from all microphone outputs can be *coherently* added as follows

$$e_{\mathrm{ca},\,1}(k) = \sqrt{\sum_{n=1}^{N} e_{\mathrm{HE}}^2(k+\tau_{1n})}\,, \qquad (46.12)$$

where $e_{\mathrm{ca},\,1}(k)$ is the coherent average of the Hilbert envelopes with respect to microphone 1, and τ_{1n} is the relative time difference of arrival between microphone 1 and n. This average can further suppress large amplitude spikes in the Hilbert envelope caused by reverberation and therefore can improve the performance of speech dereverberation.

Discussion

Speech dereverberation by LP residual enhancement is motivated by the assumption that room reverberation would have very little impact on the estimated vocal-tract responses but distort only the residual signal of LP analysis. Although intuitively making some sense, the assumption has not yet been justified in practice and, on the contrary, it disagrees somewhat with our experience. As shown by a comparison of a clean speech signal and its reverberant microphone output in Fig. 46.4, their vocal-tract responses estimated by LP analysis are significantly different: even the first two formants have been altered by room reverberation most of the time.

However, the inaccuracy of the assumption does not lead to the imprudent conclusion that algorithms based on LP residual enhancement are futile or ineffective for speech dereverberation. From a signal processing perspective, LP analysis essentially prewhitens the microphone outputs. Its impact on the signal will be compensated by the inverse LP filter at the end. Therefore the key point to success for this class of methods is whether room reverberation could be much more easily offset by processing the prewhitened LP residual signals. The LP enhancement algorithms presented in this section can remove strong reverberations at best. They are quite helpless in combating weak reverberations in the tail of an acoustic impulse response, which unfortunately are also harmful to speech intelligibility. In addition, speech artifacts are usually introduced, which makes the dereverberated speech sound unnatural. Therefore, LP residual enhancement-based speech dereverberation methods can only achieve moderate successes in relatively low-reverberant environments.

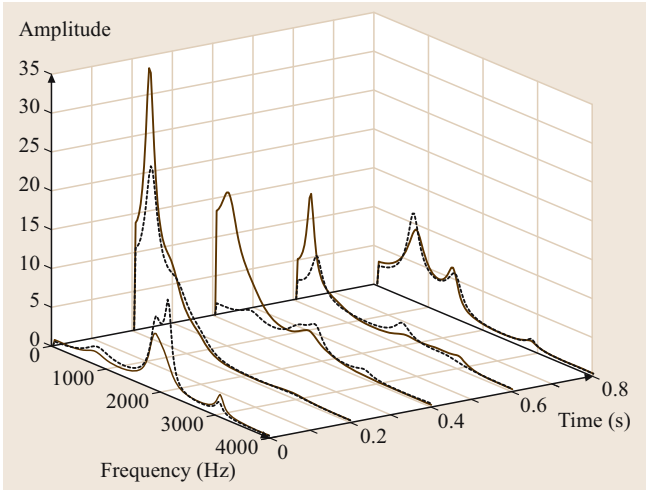


Fig. 46.4 Comparison of the all-pole filter responses for a clean speech signal (*solid line*) and its microphone output (*dotted line*) obtained by convolving the clean speech with an acoustic impulse response measured in a reverberant environment with an approximately 580 ms T_{60} reverberation time in the absence of noise. The sampling rate is 8 kHz and the two signals have been properly realigned prior to LP analysis. The order of the all-pole filter is $P = 10$. The frame size is 10 ms and the analysis window is 30 ms long with 2/3 overlapping. The results of only five frames are plotted here

46.3.3 Harmonic Filtering

Harmonic structure is an important attribute of speech, particularly voiced speech whose excitation is produced by the vibrating vocal cord. In the frequency domain, the spectrum of a voiced speech signal is composed of a fundamental frequency F_0 (i. e., pitch) and a series of harmonics. As long as the harmonics are precise multiples of the fundamental, the voice sounds clear and pleasant. But when nonharmonic components are added as room reverberation possibly does, varying degrees of roughness, harshness, or hoarseness will be perceived.

Motivated by this psychoacoustic fact, *Nakatani* et al. proposed in [46.14] to conduct speech dereverberation by suppressing the nonharmonic components and reconstructing the damaged harmonic structures. In such an algorithm, it is critical to estimate the fundamental frequency of a reverberant speech signal accurately, which itself is not an easy problem, as discussed in detail in [46.32]. After the F_0 is determined, the harmonic components are extracted from the speech spectrum to form an enhanced speech spectrum (for voiced speech segments only). By comparing the en-

Amazon EX1010 Page 936

hanced and corresponding reverberant speech spectrum, a dereverberation filter is calculated. A mean dereverberation filter is obtained by averaging the dereverberation filters of different voiced speech segments. Finally, applying the mean dereverberation filter to the entire reverberant speech sequence yields the dereverberated signal. It is suggested to repeat this procedure once more to obtain better performance.

Discussion

Harmonic filtering is a perception-based speech enhancement method for dereverberation. It eliminates the nonharmonic components, which contain both speech and reverberation even in the periods of voiced speech. Consequently, artifacts can be perceived in the enhanced signal. In addition, the dereverberation filter is estimated from voiced speech segments. Its performance for unvoiced speech segments cannot be theoretically analyzed and predicted. Moreover, the harmonic filter method ignores the reverberation effect on the harmonic components, which certainly prevents perfect dereverberation even in a presumably ideal case. Finally, this is an offline technique and cannot be implemented in real time. This makes it less attractive for many speech processing and communication applications.

46.3.4 Speech Dereverberation Using Probabilistic Models

LP residual enhancement and harmonic filtering methods are all based on the *deterministic* speech production model. In [46.15], *Attias* et al. attempted to use the *probabilistic* models described previously in Sect. 46.3.1 for speech dereverberation.

Given a frame of the microphone output $\boldsymbol{x}$, speech dereverberation is carried out by estimating a frame of speech signal $\hat{\boldsymbol{s}}$ that gives the largest posterior probability $p(\boldsymbol{s}|\boldsymbol{x})$:

$$\hat{\boldsymbol{s}} = \arg\max_{\boldsymbol{s}} p(\boldsymbol{s}|\boldsymbol{x}) \,. \tag{46.13}$$

Since

$$p(\boldsymbol{s}|\boldsymbol{x}) = \frac{p(\boldsymbol{x},\boldsymbol{s})}{p(\boldsymbol{x})} \,, \tag{46.14}$$

maximizing $p(\boldsymbol{s}|\boldsymbol{x})$ is equivalent to maximizing the joint distribution $p(\boldsymbol{x},\boldsymbol{s})$. By using the chain rule and incorporating the probabilistic clean speech model (46.10), we have

$$p(\boldsymbol{x},\boldsymbol{s}) = p(\boldsymbol{x}|\boldsymbol{s})p(\boldsymbol{s}) \approx p(\boldsymbol{x}|\boldsymbol{s})p(\boldsymbol{s}|\boldsymbol{\theta},c)p(\boldsymbol{\theta}|c)p(c) \,. \tag{46.15}$$

The a priori distribution $p(\boldsymbol{x}|\boldsymbol{s})$ is computed by using the signal model given by (46.2) and assuming the knowledge of the distribution of the additive noise:

$$p(\boldsymbol{x}|\boldsymbol{s}) = p(b = x - h * s) \,. \tag{46.16}$$

If the additive noise is Gaussian distributed with zero mean and variance σ_{b}^2, then

$$p(x - h * s) \sim \mathcal{N}\left(0, \sigma_{\mathrm{b}}^2\right) \,. \tag{46.17}$$

Clearly, the joint distribution $p(\boldsymbol{x},\boldsymbol{s})$ depends on the speech signal's characteristics $(\boldsymbol{s},\boldsymbol{\theta},c)$ and the acoustic channel conditions $(h,\sigma_{\mathrm{b}}^2)$. While $\boldsymbol{s}$ is the target, the other variables also need to be estimated. In [46.15], the iterative expectation-maximization (EM) algorithm was employed.

In the E-step, the algorithm computes the expected value of the clean speech using the current estimate of the acoustic channel conditions and the microphone signal

$$\hat{\boldsymbol{s}}(\kappa+1) = E\left\{\boldsymbol{s}\,\middle|\,\boldsymbol{x},\hat{h}(\kappa),\hat{\sigma}_{\mathrm{b}}^2(\kappa)\right\} \,, \tag{46.18}$$

where $E\{\cdot\}$ denotes mathematical expectation and κ is the iteration step index. In the M-step, the algorithm uses the data from the E-step, as if it were clean speech, to determine a maximum-likelihood estimate of the acoustic channel conditions:

$$\begin{aligned}&\left[\hat{h}(\kappa+1),\hat{\sigma}_{\mathrm{b}}^2(\kappa+1)\right]\\ &= \arg\max_{h,\sigma_{\mathrm{b}}^2} p\left[\hat{\boldsymbol{s}}(\kappa+1)|\boldsymbol{x},h,\sigma_{\mathrm{b}}^2\right] \,.\end{aligned} \tag{46.19}$$

The EM algorithm iterates (46.18) and (46.19) until convergence.

Discussion

While the algorithm presented in this subsection looks sophisticated, its essence is not difficult to be comprehended: as previously explained, speech dereverberation is challenging since it is a blind problem where neither the clean speech nor the channel impulse response is known; if an estimated speech distribution (which can be achieved by training) is provided, then we can process the microphone signals such that the enhanced signal matches the speech distribution well. The E-step performs essentially speech recognition in which the knowledge of the acoustic channel is assumed (the channel impulse response estimated in the previous step is used). Using the recognized speech, the algorithm then tries to estimate the acoustic channel impulse response in the M-step, which is a typical system identification problem where a reference signal is available. After

Amazon EX1010 Page 937

convergence, not only has speech been dereverberated, but also an estimate of the acoustic channel impulse response can be generated.

Although the idea appears technically sound, its effectiveness for speech dereverberation has not yet been justified with simulations. Speech recognition improvement in terms of accuracy for a common recognizer processing speech signals with and without dereverberation (one of the key performance measures) was unfortunately not reported. Statistical model-based speech dereverberation relies largely on the accuracy of the model, but one needs to trade off model accuracy for intractability. For example, the number of speech clusters has to be moderate. It is unexpected that such an algorithm can work well without careful tuning, which is similar to speech recognition systems.

The statistical speech model-based dereverberation method produces an estimate of the acoustic channel impulse response as a byproduct. From the perspective of blind system identification, this algorithm actually uses higher-order statistics (HOS) to estimate the channel impulse responses. But computing reliable HOS needs a great amount of data samples, which together with intensive computational complexity, makes the algorithm difficult to implement in real-time systems.

46.4 Separation of Speech and Reverberation via Homomorphic Transformation

Homomorphic transformation is a well-known signal processing method, which transforms nonadditively combined signals into a vector space in which the transformed signals are additively mixed [46.16]. To process signals that have been combined by convolution, cepstrum analysis is the most widely used homomorphic transformation. For a signal $x(k)$, its complex cepstrum (subscript 'c') is defined as

$$x_c(k) = \mathrm{IFFT}\{\ln[\mathrm{FFT}\{x(k)\}]\} , \qquad (46.20)$$

where $\mathrm{FFT}\{\cdot\}$ and $\mathrm{IFFT}\{\cdot\}$ are the fast Fourier and inverse fast Fourier transforms, respectively. If $x(k) = h * s(k)$, then in the cepstral domain we have

$$x_c(k) = h_c + s_c(k) , \qquad (46.21)$$

and we can separate $s_c(k)$ and h_c by means of linear filtering. The cepstrum transform was found to be very useful for deconvolution and has also been applied to speech dereverberation, particularly in robust speech recognition systems.

46.4.1 Cepstral Liftering

In order to separate the complex cepstra of a clean speech signal and an acoustic channel impulse response, we need to look for a property by which the two components can be differentiated. It is believed from observations that speech's cepstrum only has *low-time* components concentrated around the cepstral origin, whereas a typical acoustic channel impulse response is characterized by pulses with rapid ripples located far away from the cepstral origin. Therefore, an estimate of the speech cepstrum $\hat{s}_c(k)$ can be obtained by *low-time* liftering the microphone signal's cepstrum. (Low-time liftering refers to the operation that keeps low-time while suppressing high-time cepstral components. Low-time liftering in the cepstral domain is analogous to low-pass filtering in the frequency domain.)

Cepstral liftering relies on empirical evidence and uses a number of heuristics to determine the proper cutoff time for low-time liftering [46.17, 18]. Common problems with this kind of speech dereverberation techniques include inconsistent performance and the introduction of new distortion to the speech signal. Therefore, it has little use in real-world speech dereverberation systems for speech communication. The ideal of cepstral liftering (or log-spectral filtering) was also proposed for speech dereverberation in early speech recognition systems, include bandpass liftering [46.19] and cepstral weighting [46.20]. However, their effectiveness is also limited.

46.4.2 Cepstral Mean Subtraction and High-Pass Filtering of Cepstral Frame Coefficients

While cepstral liftering is an intraframe technique for speech dereverberation, interframe features of speech and acoustic room impulse responses can also be exploited to separate speech from reverberation. Since speech usually varies at a faster rate than acoustic room impulse responses, the channel impulse response is reasonably considered as stationary over short times. Then

taking the average of (46.21) over a proper duration of time yields

$$\bar{x}_c = h_c + \bar{s}_c \,, \tag{46.22}$$

where $\bar{x}_c$ and $\bar{s}_c$ are the short-time means of $x_c(k)$ and $s_c(k)$, respectively. In cepstral mean subtraction (CMS) [46.21], the short-time mean of speech cepstrum is assumed to be zero, which leads to $\bar{x}_c = h_c$. Therefore the dereverberated speech cepstrum will be given as

$$\hat{s}_c(k) = x_c(k) - \bar{x}_c \,. \tag{46.23}$$

Subtracting the short-time cepstral mean removes the direct-current (DC) component while keeping their quickly varying counterparts. This idea can easily be generalized to high-pass filtering of cepstral frame coefficients, which is known as the relative spectral processing (RASTA) method [46.22].

Both the CMS and RASTA methods assume that the speech cepstrum has zero mean, which is not precisely so in practice. Observing this drawback, Rahim and Juang proposed the signal bias removal (SBR) algorithm in which speech cepstrum vectors are classified into a number of clusters and each cluster has a different speech cepstral mean estimate [46.23]. Compared to the CMS and RASTA, the SBR algorithm is more computationally intensive and is hence less popular in robust speech recognition systems combating room reverberation and channel distortion.

All these cepstrum domain algorithms were proposed from the speech recognition research community. While various levels of speech dereverberation effect were reported for speech recognizers, they are not very effective for speech communication where the human auditory system is the target.

46.5 Channel Inversion and Equalization

Although it is debatable whether it is necessary in practice, perfect speech dereverberation is definitely attractive and can be achieved probably only by traditional channel inversion and equalization schemes. In this section, we will review existing channel inversion algorithms for speech dereverberation. Techniques for single- and multichannel systems will be separately developed. We focus merely on linear channel equalizers, particularly their stability, causality, and robustness.

46.5.1 Single-Channel Systems

Direct Inverse

Among all existing channel inversion methods, the most straightforward is the direct inverse method. This method assumes that the acoustic channel impulse response is known or has already been estimated. Then, as shown in Fig. 46.5a, the equalizer filter is determined by inverting the channel transfer function $H(z)$, which is the z-transform of the channel impulse response:

$$G(z) = \frac{1}{H(z)} \,. \tag{46.24}$$

In practice, the inverse filter g needs to be stable and causal. It is well known that the poles of a stable, causal, and rational system must be inside the unit circle in the z-plane. As a result, a stable, causal system has a stable and causal inverse only if both its poles and zeros are inside the unit circle. Such a system is commonly referred to as a *minimum-phase* system [46.16]. Although many systems are minimum phase, room acoustic impulse responses are unfortunately almost never minimum phase [46.33]. Consequently, while a stable inverse filter can still be found by using an all-pass filter, the inverse filter will be infinite impulse response (IIR), which is noncausal and has a long delay. In addition, inverting a transfer function is sensitive to estimation errors in the channel impulse response, particularly at those frequencies where the channel transfer function has a small

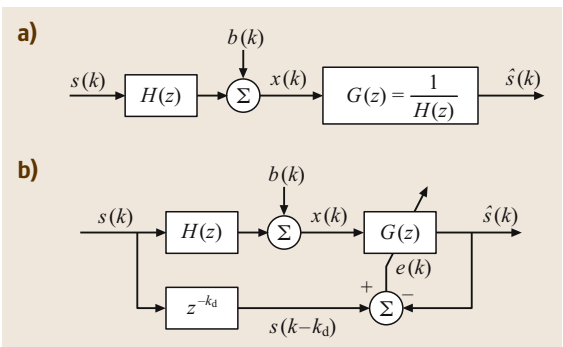


Fig. 46.5a,b Illustration of two well-known channel equalization approaches to speech dereverberation for a single-channel system: **(a)** direct inverse (or zero forcing) and **(b)** minimum mean square error (or least-squares) methods

amplitude. These drawbacks make direct-inverse equalizers impracticable for real-time speech dereverberation systems.

Minimum Mean-Square Error (MMSE) and Least-Squares (LS) Methods

If a reference source signal rather than an estimate of the acoustic channel impulse response is available, we can directly apply a linear equalizer to the microphone signal and adjust the equalizer coefficients such that the output can be as close to the reference as possible, as shown in Fig. 46.5b. The error signal is defined as

$$e(k) \triangleq s(k-d) - \hat{s}(k) = s(k-d) - g * x(k) \,, \quad (46.25)$$

where d is the decision delay for the equalizer. Then the equalization filter g is determined as the one that either minimizes the mean square error or yields the least squares of the error signal:

$$\hat{g}_{\mathrm{MMSE}} = \arg\min_{g} E\{e^2(k)\} \,, \quad (46.26)$$

$$\hat{g}_{\mathrm{LS}} = \arg\min_{g} \sum_{k=0}^{K-1} e^2(k) \,, \quad (46.27)$$

where K is the number of observed data samples. This is a typical problem in estimation theory. The solution can be found with well-known adaptive or recursive algorithms.

For minimum-phase single-channel systems, it can be shown that the MMSE/LS equalizer is the same as the direct-inverse or zero-forcing equalizer. But for non-minimum-phase acoustic systems, the MMSE/LS method essentially equalizes the channel by inverting only those components whose zeros are inside the unit circle [46.26]. In addition, it is clear that, for the MMSE/LS equalizer, a reference signal needs to be accessible. However, although the MMSE/LS method has these limitations, it is quite useful in practice and has been successfully applied for many speech dereverberation systems.

46.5.2 Multichannel Systems

While not much can be done about channel inversion when only one microphone is available, the use of multiple microphones opens up a wealth of new opportunities with speech dereverberation via channel inversion and equalization.

Beamforming

Beamforming is one of the oldest microphone array technique for speech acquisition and enhancement.

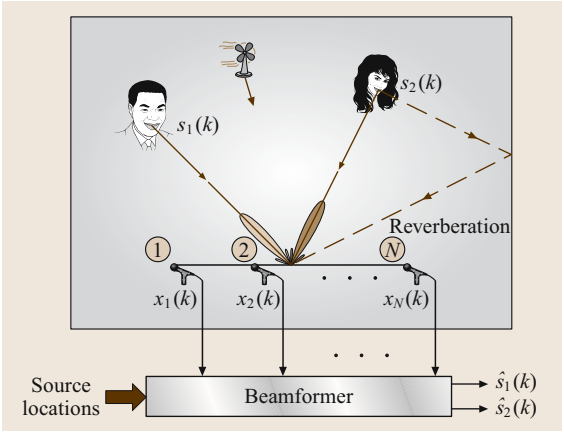


Fig. 46.6 Illustration of spatial filtering with a microphone array beamformer for speech acquisition and enhancement in an enclosed environment

A beamformer is a spatial filter that operates on the outputs of an array of microphones in order to enhance the signal coming from one direction while suppressing acoustic waves propagating from other directions, as illustrated in Fig. 46.6. While microphone array beamforming was primarily motivated to separate sound sources at different locations, it can effectively reduce the effect of room reverberation in the acquired speech signal.

Beamforming technique has two major types: delay-and-sum and filter-and-sum beamformers, as shown in Fig. 46.7. For a delay-and-sum beamformer, the estimate of a targeted speech signal is then given by

$$\hat{s}(k) = \sum_{n=1}^{N} g_n \cdot x_n(k - \tau_n) \,, \quad (46.28)$$

where the g_n are scalars and the delays τ_n are chosen such that the signals coming from the targeted source are aligned in phase in the output. A delay-and-sum beamformer is simple, but not very powerful. On one hand, it does not put nulls in the directions of the noise and interference sources whose positions are already known. On the other hand, the response of a delay-and-sum beamformer to an incoming signal depends not only on the signal's incident angle but also on the signal's spectrum. A delay-and-sum beamformer is essentially a narrowband beamformer, which does not have a uniform beam pattern over a wide frequency span. Consequently, if we use a delay-and-sum beamformer to capture speech that is a typical broadband signal, reverberations coming from a direction different from the beamformer's

Amazon EX1010 Page 940

look direction will not be equally attenuated over its entire spectrum. This *spectral tilt* results in a disturbing artifact in the array output.

In a filter-and-sum beamformer, as the name suggests, each microphone signal is first processed by a linear filter and then they are all summed to produce a dereverberated speech signal

$$\hat{s}(k) = \sum_{n=1}^{N} g_n * x_n(k) \,, \tag{46.29}$$

where g_n is the filter impulse response of length L_g for the n-th channel. Filter-and-sum beamformers are much more flexible and work well with wideband speech signals. The determination of the beamformer filters can be either data independent or data driven. In the former case, the filter coefficients and the beam pattern are relatively fixed. The look direction is steered to the direction of a specified target. In the latter case, the power of the beamformer output is minimized under the constraint that the signals coming from the look direction are aligned in phase or are distortionless, depending on how much is known about the acoustic channel from the target to the array of microphones. The linearly constrained minimum-variance (LCMV) algorithm developed by *Frost* [46.34] is the most studied and most influential algorithm of this kind. The generalized sidelobe canceler (GSC), developed by *Griffiths* and *Jim* [46.35], is rather an interesting way to implement the Frost's LCMV algorithm. The GSC algorithm transforms the LCMV beamforming from a constrained optimization problem into an unconstrained form. While they are essentially the same, the GSC has some implementation advantages, particularly for adaptive operation in time-varying acoustic environments (see [46.36] for a more-detailed discussion on the implementation advantages of the GSC algorithm). One main drawback of the GSC beamformer is that the cancelation of the target signals could occur in reverberant acoustic environments. Many robust variants of the GSC have been proposed to combat this problem [46.37–40].

Beamforming is a promising technique for speech dereverberation, but its performance is limited by a couple of factors in practice. Other than the issue of wideband speech signals, the problem of how to deal with near-field sound sources in reverberant acoustic environments still lacks of satisfactory solutions and even a decent framework for analysis. (Beamforming algorithms make a common assumption that the targeted sound source is in the far field of the array such that the wavefronts impinging on the array can be regarded as plane waves.)

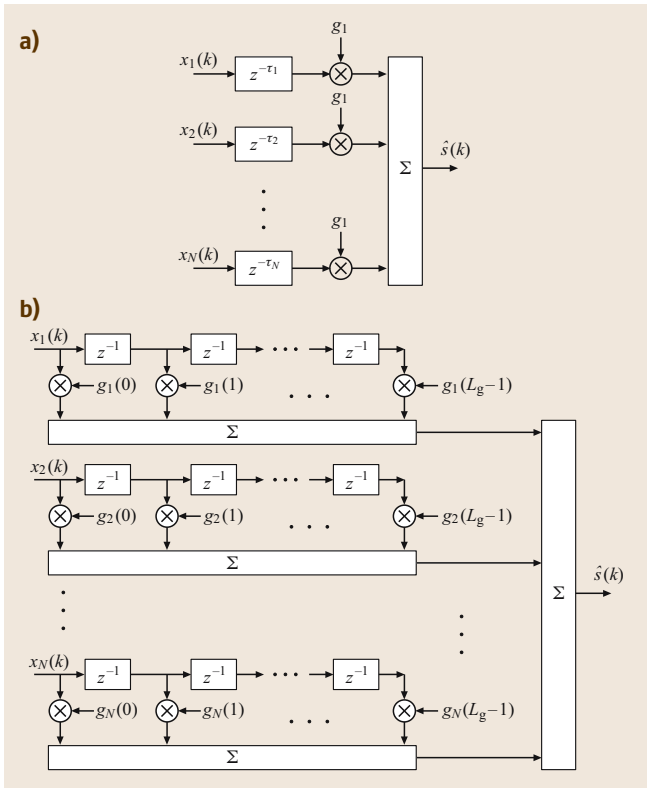


Fig. 46.7a,b The structure of **(a)** a delay-and-sum beamformer and **(b)** a filter-and-sum beamformer

Matched Filtering

In [46.25], *Flanagan* et al. proposed a matched filtering algorithm for speech acquisition and dereverberation as an alternative to beamforming. As opposed to knowing only the position of the speech source with a beamformer, the matched filtering algorithm assumes the availability of channel impulse responses from the targeted speech source to all microphones. Rather than finding inverse filters for those impulse responses, the algorithm convolved each microphone output with the time-reversed source to microphone impulse response, i. e., the matched filter. Then, without taking additive noise into account, the dereverberated speech signal is found as

$$\hat{s}(k) = \sum_{n=1}^{N} x_n(k) * h_n^{\mathrm{MF}} = s(k) * \left(\sum_{n=1}^{N} h_n * h_n^{\mathrm{MF}} \right) \,, \tag{46.30}$$

where

$$h_n^{\mathrm{MF}}(l) = \begin{cases} h_n(-l)\,, & -L_h < l \le 0 \\ 0\,, & \text{otherwise} \end{cases} \tag{46.31}$$

is the matched filter for the nth channel. It was shown in [46.25] that the matched-filtering algorithm is theoretically superior to the delay-and-sum beamformer. The gain in signal-to-reverberation ratio no longer depends on the number of reflections in the impulse response, but using full-length matched filters has a serious problem in practice. The convolution of h_n with h_n^{MF} is equivalent to the autocorrelation function of h_n, which has long tails on both the positive and negative sides. As a result, pre-echo (an attenuated replica of a signal that arrives before the actual signal) will be heard in the output. Pre-echo degrades the quality of the dereverberated speech signal. To overcome this problem, a truncated matched filter was suggested in [46.41]. It can reduce but cannot eliminate the pro-echo.

Matched filtering is a computationally efficient way to invert an acoustic channel when its impulse response is known, but it is by no means ideal. It can only marginally improve the signal-to-reverberation ratio and has little conducive impact on reverberation time. Its superiority over the delay-and-sum beamformer is not always observed and is not practically evident.

MINT Method

For a SIMO system, let us consider the polynomials $G_n(z)$ $(n = 1, 2, \cdots, N)$ and the following equation:

$$\begin{aligned} \hat{S}(z) &= \sum_{n=1}^{N} G_n(z) X_n(z) \\ &= \left[\sum_{n=1}^{N} H_n(z) G_n(z)\right] S(z) + \sum_{n=1}^{N} G_n(z) B_n(z)\,. \end{aligned} \tag{46.32}$$

The polynomials $G_n(z)$ should be found in such a way that $\hat{S}(z) = S(z)$ in the absence of noise by using the Bezout theorem, which is mathematically expressed as

$$\begin{aligned} &\gcd[H_1(z), H_2(z), \cdots, H_N(z)] = 1 \\ &\Leftrightarrow \exists G_1(z), G_2(z), \cdots, G_N(z): \\ &\sum_{n=1}^{N} H_n(z) G_n(z) = 1\,, \end{aligned} \tag{46.33}$$

where $\gcd[\cdot]$ denotes the greatest common divisor of the polynomials involved. In other words, as long as the channel impulse responses h_n are co-prime (even though they may not be minimum phase), i. e., the SIMO system is irreducible, there exists a group of g filters to completely remove the reverberations and perfectly recover the source signals. The idea of using the Bezout theorem to equalize a SIMO system was first proposed in [46.26] in the context of room acoustics, where the principle is more widely referred to as the MINT theory.

If the channels of the SIMO system share common zeros, i. e.,

$$C(z) = \gcd[H_1(z), H_2(z), \cdots, H_N(z)] \neq 1\,, \tag{46.34}$$

then we have

$$H_n(z) = C(z) H_n'(z)\,, \quad n = 1, 2, \cdots, N\,, \tag{46.35}$$

and the polynomials $G_n(z)$ can be found such that

$$\sum_{n=1}^{N} H_n'(z) G_n(z) = 1\,. \tag{46.36}$$

In this case, (46.32) becomes

$$\hat{S}(z) = C(z) S(z) + \sum_{n=1}^{N} G_n(z) B_n(z)\,. \tag{46.37}$$

We see that by using the Bezout theorem, the *reducible* SIMO system can be equalized up to the polynomial $C(z)$. So when there are common zeros, the MINT equalizer can only partially suppress the reverberations. For more complete cancelation of the room reverberations, we have to combat the effect of $C(z)$ using either the direct inverse or MMSE/LS methods, which depends on whether $C(z)$ is a minimum-phase filter.

To find the MINT equalization filters, we write the Bezout equation (46.33) in the time domain as

$$\boldsymbol{H}^{\mathrm{c}} \boldsymbol{g} = \sum_{n=1}^{N} \boldsymbol{H}_n^{\mathrm{c}} \boldsymbol{g}_n = \boldsymbol{e}_1\,, \tag{46.38}$$

where

$$\begin{aligned} \boldsymbol{H}^{\mathrm{c}} &= \left[\boldsymbol{H}_1^{\mathrm{c}} \;\; \boldsymbol{H}_2^{\mathrm{c}} \;\; \cdots \;\; \boldsymbol{H}_N^{\mathrm{c}}\right], \\ \boldsymbol{g} &= \left[\boldsymbol{g}_1^{\mathrm{T}} \;\; \boldsymbol{g}_2^{\mathrm{T}} \;\; \cdots \;\; \boldsymbol{g}_N^{\mathrm{T}}\right]^{\mathrm{T}}, \\ \boldsymbol{g}_n &= \left[g_n(0) \;\; g_n(1) \;\; \cdots \;\; g_n(L_g - 1)\right]^{\mathrm{T}}, \end{aligned}$$

Amazon EX1010 Page 942

L_g is the length of the FIR filter g_n,

$$\boldsymbol{H}_n^{\mathrm{c}} = \begin{pmatrix} h_n(0) & 0 & \cdots & 0 \\ h_n(1) & h_n(0) & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ h_n(L_h-1) & \cdots & \cdots & \vdots \\ 0 & h_n(L_h-1) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & h_n(L_h-1) \end{pmatrix}$$

is an $(L_h+L_g-1)\times L_g$ convolution matrix (superscript c), and

$$\boldsymbol{e}_1 = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}^{\mathrm{T}}$$

is an $(L_h+L_g-1)\times 1$ vector.

In order to have a unique solution for (46.38), L_g must be chosen in such a way that $\boldsymbol{H}^{\mathrm{c}}$ is a square matrix. In this case, we have:

$$L_g = \frac{L_h-1}{N-1} \,. \tag{46.39}$$

However, this may not be practical since $(L_h-1)/(N-1)$ is not necessarily always an integer. Therefore, a larger L_g is usually chosen and we solve (46.38) for $\boldsymbol{g}$ in the least-squares sense as

$$\hat{\boldsymbol{g}}_{\mathrm{MINT}} = \boldsymbol{H}^{\mathrm{c}\dagger}\boldsymbol{e}_1 \,, \tag{46.40}$$

where

$$\boldsymbol{H}^{\mathrm{c}\dagger} = \left(\boldsymbol{H}^{\mathrm{c}\mathrm{T}}\boldsymbol{H}^{\mathrm{c}}\right)^{-1}\boldsymbol{H}^{\mathrm{c}\mathrm{T}}$$

is the pseudo-inverse of the matrix $\boldsymbol{H}^{\mathrm{c}}$.

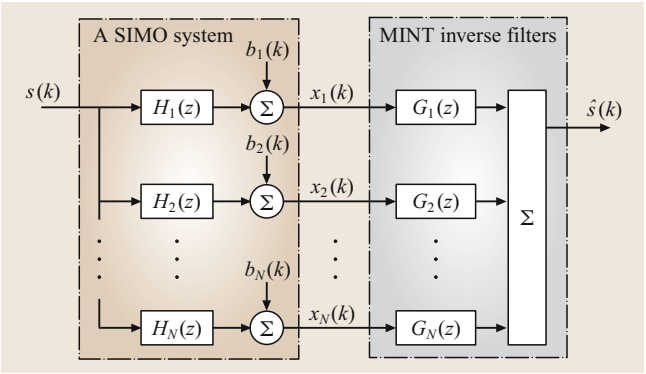


Fig. 46.8 Illustration of the MINT approaches to speech dereverberation for a single-input multiple-output acoustic system

If a decision delay d is taken into account, then the equalization filters turn out to be

$$\hat{\boldsymbol{g}}_{\mathrm{MINT}} = \boldsymbol{H}^{\mathrm{c}\dagger}\boldsymbol{e}_d \,, \tag{46.41}$$

where

$$\boldsymbol{e}_d = \left(\underbrace{0 \cdots 0}_{d} \; 1 \; \underbrace{0 \cdots 0}_{L_h+L_g-d-2}\right)^{\mathrm{T}} .$$

MINT equalization is an appealing approach to speech dereverberation. It can perfectly remove the effect of room reverberation even though acoustic impulse responses are not minimum phase. But in practice, the MINT method is found to be very sensitive to even small errors in the estimated channel impulse responses. Therefore, it is only useful when background noise is weak or well controlled.

46.6 Summary

In this chapter we have attempted to present the state of the art of speech dereverberation techniques. We have explained how room reverberation smears clean speech signals and hence impairs their intelligibility. The difficulty of solving this problem in practice has been elucidated. We have introduced a new classification of speech dereverberation algorithms, according to which well-known and recent developments fall into the following three categories: source model-based speech enhancement algorithms for dereverberation, separation of speech and reverberation via homomorphic transformation, and speech dereverberation by channel inversion and equalization. We have chosen to focus on the ideas behind such a rather large number of approaches for both single-channel and multichannel systems; hence we have avoided tedious derivations of some of the algorithms. We have discussed in details the merits and drawbacks of these speech dereverberation algorithms, trying to shed light on the performance that we believe can be expected when they are implemented in real-world systems.

Amazon EX1010 Page 943

References

46.1 A.K. Nábělek: Communication in noisy and reverberant environments. In: *Acoustical Factors Affecting Hearing Aid Performance*, 2nd edn., ed. by G.A. Studebaker, I. Hochberg (Allyn Bacon, Needham Height 1993)

46.2 S. Sandhu, O. Ghitza: A comparative study of MEL cepstra and EIH for phone classification under adverse conditions, Proc. IEEE ICASSP **1**, 409–412 (1995)

46.3 B.E.D. Kingsbury, N. Morgan: Recognizing reverberant speech with RASTA-PLP, Proc. IEEE ICASSP **2**, 1259–1262 (1997)

46.4 D.A. Berkley, J.B. Allen: Normal listening in typical rooms – the physical and psychophysical correlates of reverberation. In: *Acoustical Factors Affecting Hearing Aid Performance*, 2nd edn., ed. by G.A. Studebaker, I. Hochberg (Allyn Bacon, Needham Height 1993)

46.5 R.Y. Litovsky, H.S. Colburn, W.A. Yost, S.J. Guzman: The precedence effect, J. Acoust. Soc. Am. **106**, 1633–1654 (1999)

46.6 W.C. Sabine: *Collected Papers on Acoustics* (Harvard Univ. Press, Cambridge 1922)

46.7 M.R. Schroeder: New method of measuring reverberation time, J. Acoust. Soc. Am. **37**, 409–412 (1965)

46.8 D.R. Morgan: A parametric error analysis of the backward integration method for reverberation time estimation, J. Acoust. Soc. Am. **101**, 2686–2693 (1997)

46.9 H. Kuttruff: *Room Acoustics*, 4th edn. (Spon, New York 2000)

46.10 M.S. Brandstein, S.M. Griebel: Nonlinear, model-based microphone array speech enhancement. In: *Acoustic Signal Processing for Telecommunication*, ed. by S.L. Gay, J. Benesty (Kluwer Academic, Boston 2000) pp. 261–279, Chap. 12

46.11 B. Yegnanarayana, P.S. Murthy: Enhancement of reverberant speech using LP residual signal, IEEE Trans. Speech Audio Process. **8**, 267–281 (2000)

46.12 B. Yegnanarayana, S.R. Mahadeva Prasanna, K. Sreenivasa Rao: Speech enhancement using excitation source information, Proc. IEEE ICASSP **1**, 541–544 (2002)

46.13 B.W. Gillespie, H.S. Malvar, D.A.F. Florencio: Speech dereveration via maximum-kurtosis subband adaptive filtering, Proc. IEEE ICASSP **6**, 3701–3704 (2001)

46.14 T. Nakatani, M. Miyoshi, K. Kinoshita: Single-microphone blind dereverberation. In: *Speech Enhancement*, ed. by J. Benesty, S. Makino, J. Chen (Springer, Berlin, Heidelberg 2005) pp. 247–270, Chap. 11

46.15 H. Attias, J.C. Platt, A. Acero, L. Deng: Speech denoising and dereverberation using probabilistic models. In: *Advances in Neural Information Processing Systems*, Vol. 13, ed. by S. Thrun, L. Saul, B. Schölkopf (MIT Press, Cambridge 2001) pp. 758–764

46.16 A.V. Oppenheim, R.W. Schafer: *Discrete-Time Signal Processing* (Prentice-Hall, Englewood Cliffs 1989)

46.17 D. Bees, M. Blostein, P. Kabal: Reverberant speech enhancement using cepstral processing, Proc. IEEE ICASSP **2**, 977–980 (1991)

46.18 M. Tohyama, R.H. Lyon, T. Koike: Source waveform recovery in a reverberant space by cepstrum dereverberation, Proc. IEEE ICASSP **1**, 157–160 (1993)

46.19 B.-H. Juang, L.R. Rabiner, J.G. Wilpon: On the use of bandpass liftering in speech recognition, Proc. IEEE ICASSP **1**, 765–768 (1986)

46.20 Y. Tohkura: A weighted cepstral distance measure for speech recognition, IEEE Trans. Acoust. Speech Signal Process. **ASSP-35**(10), 1414–1422 (1987)

46.21 B.S. Atal: Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification, J. Acoust. Soc. Am. **55**(6), 1304–1312 (1974)

46.22 H. Hermansky, N. Morgan: RASTA processing of speech, IEEE Trans. Speech Audio Process. **2**(4), 578–589 (1994)

46.23 M.G. Rahim, B.-H. Juang: Signal bias removal by maximum likelihood estimation for robust telephone speech recognition, IEEE Trans. Speech Audio Process. **4**(1), 19–30 (1996)

46.24 J.L. Flanagan, J.D. Johnston, R. Zahn, G.W. Elko: Computer steered imcrophone arrays for sound transduction in large rooms, J. Acoust. Soc. Am. **78**(5), 1508–1518 (1985)

46.25 J.L. Flanagan, A.C. Surendran, E.E. Jan: Spatially selective sound capture for speech and audio processing, Speech Commun. **13**, 207–222 (1993)

46.26 M. Miyoshi, Y. Kaneda: Inverse filtering of room acoustics, IEEE Trans. Acoust. Speech Signal Process. **36**, 145–152 (1988)

46.27 Y. Huang, J. Benesty, J. Chen: *Acoustic MIMO Signal Processing* (Springer, Berlin, Heidelberg 2006)

46.28 H. Dudley: The vocoder, Bell Labs Rec. **17**, 122–126 (1939)

46.29 B.S. Atal, M.R. Schroeder: Predictive coding of speech, Proc. AFCRL/IEEE Conf. Speech Communication and Processing (1967) pp. 360–361

46.30 B.S. Atal, M.R. Schroeder: Adaptive predictive coding of speech, Bell Syst. Tech. J. **49**(8), 1973–1986 (1970)

46.31 B.-H. Juang, M.M. Sondhi, L.R. Rabiner: Digital speech processing. In: *Encyclopedia of Physical Science and Technology*, Vol. 4, 3rd edn., ed. by R.A. Meyers (Academic, Orlando 2000) pp. 485–500

46.32 W.J. Hess: Pitch and voicing determination. In: *Advances in Speech Signal Processing*, ed. by S. Fu-

Amazon EX1010 Page 944

rui, M.M. Sondhi (Marcel Dekker, New York 1992) pp. 3–48
46.33 S.T. Neely, J.B. Allen: Invertibility of a room impulse response, J. Acoust. Soc. Am. **68**, 165–169 (1979)
46.34 O.L. Frost: An algorithm for linearly constrained adaptive array processing, Proc. IEEE **60**, 926–935 (1972)
46.35 L.J. Griffiths, C.W. Jim: An alternative approach to linearly constrained adaptive beamforming, IEEE Trans. Antennas Propagat. **AP-30**, 27–34 (1982)
46.36 B.D. Van Veen, K.M. Buckley: Beamforming: a versatile approach to spatial filtering, IEEE ASSP Mag. **5**, 4–24 (1988)
46.37 S. Affes, S. Gazor, Y. Grenier: Robust adaptive beamforming via LMS-like target tracking, Proc. IEEE ICASSP **4**, 269–272 (1994)
46.38 O. Hoshuyama, A. Sugiyama, A. Hirano: A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filter, IEEE Trans. Signal Process. **47**, 2677–2684 (1999)
46.39 S. Gannot, D. Burshtein, E. Weinstein: Signal enhancement using beamforming and nonstationarity with applications to speech, IEEE Trans. Signal Process. **49**, 1614–1626 (2001)
46.40 W. Herbordt, W. Kellermann: Adaptive beamforming for audio signal acquisition. In: *Adaptive Signal Processing: Applications to Real-World Problems*, ed. by J. Benesty, Y. Huang (Springer, Berlin, Heidelberg 2003) pp. 155–194, Chap. 6
46.41 D. Rabinkin, R. Renomeron, J.L. Flanagan, D.F. Macomber: Optimal truncation time for matched filter array processing, Proc. IEEE ICASSP **6**, 3629–3632 (1998)

Amazon EX1010 Page 945

47. Adaptive Beamforming and Postfiltering

S. Gannot, I. Cohen

In this chapter, we explore many of the basic concepts of array processing with an emphasis on adaptive beamforming for speech enhancement applications. We begin in Sect. 47.1 by formulating the problem of microphone array in a noisy and reverberant environment. In Sect. 47.2, we derive the frequency-domain linearly constrained minimum-variance (LCMV) beamformer, and its generalized sidelobe canceller (GSC) variant. The GSC components are explored in Sect. 47.3, and several commonly used special cases of these blocks are presented. As the GSC structure necessitates an estimation of the speech related acoustical transfer functions (ATFs), several alternative system identification methods are addressed in Sect. 47.4. Beamformers often suffer from sensitivity to signal mismatch. We analyze this phenomenon in Sect. 47.5 and explore several cures to this problem. Although the GSC beamformer yields a significant improvement in speech quality, when the noise field is spatially incoherent or diffuse, the noise reduction is insufficient and additional postfiltering is normally required. In Sect. 47.6, we present multimicrophone postfilters, based on either minimum mean-squared error (MMSE) or log-spectral amplitude estimate criteria. An interesting relation between the GSC and the Wiener filter is derived in this Section as well. In Sect. 47.7, we analyze the performance of the transfer-function GSC (TF-GSC), and in Sect. 47.8 demonstrate the advantage of multichannel postfiltering over single-channel postfiltering in nonstationary noise conditions.

47.1 **Problem Formulation** 947
47.2 **Adaptive Beamforming** 948
47.2.1 Frequency-Domain Frost Algorithm 948
47.2.2 Frequency-Domain Generalized Sidelobe Canceller 950
47.2.3 Time-Domain Generalized Sidelobe Canceller 952
47.3 **Fixed Beamformer and Blocking Matrix** .. 953
47.3.1 Using Acoustical Transfer Functions 953
47.3.2 Using Delay-Only Filters 954
47.3.3 Using Relative Transfer Functions .. 954
47.4 **Identification of the Acoustical Transfer Function** 955
47.4.1 Signal Subspace Method 955
47.4.2 Time Difference of Arrival 956
47.4.3 Relative Transfer Function Estimation 956
47.5 **Robustness and Distortion Weighting** 960
47.6 **Multichannel Postfiltering** 962
47.6.1 MMSE Postfiltering 963
47.6.2 Log-Spectral Amplitude Postfiltering 964
47.7 **Performance Analysis** 967
47.7.1 The Power Spectral Density of the Beamformer Output 967
47.7.2 Signal Distortion 968
47.7.3 Stationary Noise Reduction 968
47.8 **Experimental Results** 972
47.9 **Summary** 972
47.A **Appendix: Derivation of the Expected Noise Reduction for a Coherent Noise Field** 973
47.B **Appendix: Equivalence Between Maximum SNR and LCMV Beamformers** 974
References 975

Amazon EX1010 Page 946

Over the last four decades, array processing has become a well-established discipline, see e.g., [47.1–14]. In the mid 1980s, array processing and beamforming methods were adopted by the speech community to deal with data received by microphone arrays. Since then, beamforming techniques for microphone arrays have been used in many applications, such as speaker separation, speaker localization, speech dereverberation, acoustic echo cancellation, and speech enhancement.

Adaptive beamforming for speech signals requires particular consideration of problems that are specific to speech signals and to the acoustic environment. The speech signal is wide-band, highly nonstationary, and has a very wide dynamic range. An acoustic enclosure is usually modeled as a filter with very long impulse response due to multiple reflections from the room walls. In a typical office, the length of the filters may reach several thousand taps. Furthermore, the impulse response is often time varying due to speaker and objects movements.

The term beamforming refers to the design of a spatiotemporal filter. Broadband arrays comprise a set of filters, applied to each received microphone signal, followed by a summation operation. The main objective of the beamformer is to extract a desired signal, impinging on the array from a specific position, out of noisy measurements thereof. Usually, the interference signals occupy the same frequency band as the desired signal, rendering temporal-only filtering useless. The simplest structure is the delay-and-sum beamformer, which first compensates for the relative delay between distinct microphone signals and then sums the steered signal to form a single output. This beamformer, which is still widely used, can be very effective in mitigating noncoherent, i. e., spatially white, noise sources, provided that the number of microphones is relatively high. However, if the noise source is coherent, the noise reduction (NR) is strongly dependent on the direction of arrival of the noise signal. Consequently, the performance of the delay and sum beamformer in reverberant environments is often insufficient. *Jan* and *Flanagan* [47.15, 16] and *Rabinkin* et al. [47.17] extended the delay and sum concept by introducing the filter-and-sum beamformer. This structure, designed for multipath environments, namely reverberant enclosures, replaces the simpler delay compensator with a matched filter.

The array beam pattern can generally be designed to have a specified response. This can be done by properly setting the values of the multichannel filters' weights. However, the application of data-independent design methods is very limited in dynamic acoustical environments. Statistically optimal beamformers are designed based on the statistical properties of the desired and interference signals. In general, they aim at enhancing the desired signal, while rejecting the interference signal. Several criteria can be applied in the design of the beamformer, e.g., maximum signal-to-noise ratio (MSNR), minimum mean-squared error (MMSE), and linearly constrained minimum variance (LCMV). A summary of several design criteria can be found in [47.5, 7].

Beamforming methods use the signals' statistics (at least second-order statistics), which is usually not available and should be estimated from the data. Moreover, the acoustical environment is time varying, due to talker and objects movements, and abrupt changes in the noise characteristics (e.g., passing cars). Hence, adaptation mechanisms are required. An adaptive counterpart of each of the prespecified design criteria can be derived. Early contributions to the field of adaptive beamformers design can be attributed to *Sondhi* and *Elko* [47.18], to *Kaneda* and *Ohga* [47.19], and to *Van Compernolle* [47.20]. *Kellermann* [47.21] addressed the problem of joint echo cancellation and NR by incorporating echo cancellers into the beamformer design. *Nordholm* et al. [47.22, 23] used microphone arrays in a car environment, and designed a beamformer employing calibration signals to enhance the obtained performance. *Martin* [47.24] analyzed beamforming techniques for small microphone arrays. Many other applications of microphone arrays such as hearing aids, blind source separation (BSS), and dereverberation are addressed elsewhere in this handbook.

The minimization of the mean-squared error (MSE) in the context of array processing leads to the well-known multichannel Wiener filter [47.25]. *Doclo* and *Moonen* [47.26–28] proposed an efficient implementation of the Wiener filter based on the generalized singular-value decomposition (GSVD) of the microphone data matrix. This method yields an optimal estimation (in the MMSE sense) of the desired signal component of one of the microphone signals. The authors further proposed efficient schemes for recursive update of the GSVD. An optional, adaptive noise cancellation postfiltering stage is proposed as well. In that scheme, in addition to the optimal estimation of the desired speech signal, an optimal noise channel is also estimated. This estimated noise component can be used as a reference noise signal (similar to the one used in [47.25]), to further enhance the speech signal. *Spriet* et al. [47.29] proposed a subband implementation of the GSVD-based scheme, and *Rombouts* and

Amazon EX1010 Page 947

Moonen [47.30, 31] proposed to apply the efficient QR decomposition to the problem at hand.

In many adaptive array schemes the acoustical transfer-function (ATF) relating the speech source and the microphone should be known in advance, or at least estimated from the received data (note that in case of delay-only propagation, the acoustical transfer function reduces to a steering vector, consisting of phase-only components.) In contrast, the multichannel Wiener filter is uniquely based on estimates of the second-order statistics of the recorded noisy signal and the noise signal (estimated during noise-only segments), and does not make any a priori assumptions about the signal model. Unfortunately, as pointed by *Chen* et al. [47.32], the Wiener filter, which is optimal in the MMSE sense, cannot guarantee undistorted speech signal at its output. This drawback can however be mitigated by modifying the MMSE criterion to control the amount of imposed speech distortion. A method that employs this modification is presented in [47.33, 34]. It is also shown that the ATFs information (only a simple delay-only case is presented in the contributions) can be incorporated into the Wiener filter scheme (called the spatially preprocessed Wiener filter), resulting in an improved performance. The Wiener filter and its application to speech enhancement is addressed in a separate chapter of this handbook (6; 43).

In this chapter, we concentrate on a different adaptive structure based on the LCMV criterion. The LCMV beamformer, proposed by *Frost* [47.35], is aiming at minimizing the output power under linear constraints on the response of the array towards the desired speech signal. Frost proposed an adaptive scheme, which is based on a constrained least-mean-square (LMS)-type adaptation (for the LMS algorithm please refer to [47.25]). To avoid this constrained adaptation, *Griffiths* and *Jim* [47.36] proposed the GSC structure, which separates the output power minimization and the application of the constraint. The GSC structure is based on the assumption that the different sensors receive a delayed version of the desired signal, and therefore we refer to it as the delay generalized sidelobe canceller (D-GSC). The GSC structure was rederived in the frequency domain, and extended to deal with, the more-complicated general ATFs case by *Affes* and *Grenier* [47.37] and later by *Gannot* et al. [47.38]. This frequency-domain version, which takes into account the reverberant nature of the enclosure, was nicknamed the transfer-function generalized sidelobe canceller (TF-GSC). The GSC comprises three blocks: a fixed beamformer (FBF), which aligns the desired signal components, a blocking matrix (BM), which blocks the desired speech components resulting in reference noise signals, and a multichannel adaptive noise canceller (ANC), which eliminates noise components that leak through the sidelobes of the FBF.

Nordholm and *Leung* [47.39] analyze the limits of the obtainable NR of the GSC in an isotropic noise field. *Bitzer* et al. address the problem in [47.40, 41] and [47.42]. In [47.40], the authors derive an expression for the NR as a function of the noise field and evaluate the degradation as a function of the reverberation time (T_{60}). The special two-microphone case is treated in [47.41]. The additional NR due to the ANC branch of the GSC, implemented by a closed-form Wiener filter rather than the adaptive Widrow least-mean-square (LMS) procedure, is presented in [47.42]. The frequency-band nested subarrays structure is presented and its NR is theoretically analyzed by *Marro* et al. [47.43]. A more-complex dual GSC structure employing calibration signals was suggested and analyzed by *Nordholm* et al. [47.44]. *Huarng* and *Yeh* [47.45] addressed the distortion issue by evaluating the desired signal leakage into the reference noise branch of the GSC structure. However, the delay-only ATFs assumption is imposed and the expected degradation due to pointing errors alone is evaluated. The performance degradation due to constraining the Wiener filters to an finite impulse response (FIR) structure is demonstrated by *Nordholm* et al. in [47.46]. The resulting performance limits of the GSC structure strongly depend on the cross-correlation between the sensors' signals induced by the noise field, as shown in the above references and by *Cox* [47.47].

47.1 Problem Formulation

Consider an array of M sensors in a noisy and reverberant environment. The received signals generally include three components. The first is a desired speech signal, the second is some stationary interference signal, and the third is some nonstationary (transient) noise component. Our goal is to reconstruct the speech component from the received signals. Let $s(t)$ denote the desired source signal, let $a_m(t)$ represent the room impulse response (RIR) of the m-th sensor to the desired source, and let $n_m(t)$ denote the noise component at the m-th sensor.

The observed signal at the m-th sensor ($m = 1, \ldots, M$) is given by

$$\begin{aligned} z_m(t) &= a_m(t) * s(t) + n_m(t) \\ &= a_m(t) * s(t) + n_m^{\mathrm{s}}(t) + n_m^{\mathrm{t}}(t)\,, \end{aligned} \tag{47.1}$$

where $n_m^{\mathrm{s}}(t)$ and $n_m^{\mathrm{t}}(t)$ represent the stationary and nonstationary noise components at the m-th sensor, respectively, and $*$ denotes convolution. We assume that both noise components may comprise coherent (directional) noise component and diffused noise component.

The observed signals are divided in time into overlapping frames by the application of a window function and analyzed using the short-time Fourier transform (STFT). Assuming time-invariant transfer functions, we have in the time–frequency domain

$$\begin{aligned} Z_m(k,\ell) &\approx A_m(k)S(k,\ell) + N_m(k,\ell) \\ &\approx A_m(k)S(k,\ell) + N_m^{\mathrm{s}}(k,\ell) + N_m^{\mathrm{t}}(k,\ell)\,, \end{aligned} \tag{47.2}$$

where ℓ is the frame index and $k = 1, 2, \ldots, K$ represents the frequency bin index. (The equality in (47.2) is only justified for segments which are longer than the RIR length. Since RIRs tend to be very long, the conditions allowing for this representation to hold cannot be exactly met. We assume, however, that the STFT relation is a reasonable approximation.) $Z_m(k,\ell)$, $S(k,\ell)$, $N_m(k,\ell)$, $N_m^{\mathrm{s}}(k,\ell)$, and $N_m^{\mathrm{t}}(k,\ell)$ are the STFT of the respective signals. $A_m(k)$ is the ATF relating the speech source with the m-th sensor. The vector formulation of the equation set (47.2) is

$$\begin{aligned} \boldsymbol{Z}(k,\ell) &= \boldsymbol{A}(k)S(k,\ell) + \boldsymbol{N}(k,\ell) \\ &= \boldsymbol{A}(k)S(k,\ell) + \boldsymbol{N}_{\mathrm{s}}(k,\ell) + \boldsymbol{N}_{\mathrm{t}}(k,\ell)\,, \end{aligned} \tag{47.3}$$

where

$$\begin{aligned} \boldsymbol{Z}(k,\ell) &= \big(Z_1(k,\ell)\;\; Z_2(k,\ell)\;\; \cdots\;\; Z_M(k,\ell)\big)^{\mathrm{T}}\,, \\ \boldsymbol{A}(k) &= \big(A_1(k)\;\; A_2(k)\;\; \cdots\;\; A_M(k)\big)^{\mathrm{T}}\,, \\ \boldsymbol{N}(k,\ell) &= \big(N_1(k,\ell)\;\; N_2(k,\ell)\;\; \cdots\;\; N_M(k,\ell)\big)^{\mathrm{T}}\,, \\ \boldsymbol{N}_{\mathrm{s}}(k,\ell) &= \big(N_1^{\mathrm{s}}(k,\ell)\;\; N_2^{\mathrm{s}}(k,\ell)\;\; \cdots\;\; N_M^{\mathrm{s}}(k,\ell)\big)^{\mathrm{T}}\,, \\ \boldsymbol{N}_{\mathrm{t}}(k,\ell) &= \big(N_1^{\mathrm{t}}(k,\ell)\;\; N_2^{\mathrm{t}}(k,\ell)\;\; \cdots\;\; N_M^{\mathrm{t}}(k,\ell)\big)^{\mathrm{T}}. \end{aligned}$$

47.2 Adaptive Beamforming

Frost [47.35] proposed a beamformer that relies on the assumption that the ATFs between the desired source and the array of sensors can be uniquely determined by gain and delay values. In this section, we follow Frost's approach in the STFT domain and derive a beamforming algorithm for the arbitrary ATF case. We first obtain a closed form of the LCMV beamformer, and subsequently derive an adaptive solution. The outcome is a constrained LMS-type algorithm. We proceed, following the seminal work of *Griffiths* and *Jim* [47.36], with the formulation of an unconstrained adaptive solution namely, the transfer-function generalized sidelobe canceller (TF-GSC). We initially assume that the ATFs are known. Later, in Sect. 47.4, we present several alternatives for estimating the ATFs.

47.2.1 Frequency-Domain Frost Algorithm

Optimal Solution

Let $W_m^*(k,\ell)\,;\, m = 1, \ldots, M$ denote a set of M filters, and define

$$\boldsymbol{W}^{\mathrm{H}}(k,\ell) = \big(W_1^*(k,\ell)\;\; W_2^*(k,\ell)\;\; \cdots\;\; W_M^*(k,\ell)\big)\,,$$

where the superscript H denotes conjugation transpose. A filter-and-sum beamformer, depicted in Fig. 47.1, is realized by filtering each sensor signal by $W_m^*(k,\ell)$ and summing the outputs,

$$\begin{aligned} Y(k,\ell) &= \boldsymbol{W}^{\mathrm{H}}(k,\ell)\boldsymbol{Z}(k,\ell) \\ &= \boldsymbol{W}^{\mathrm{H}}(k,\ell)\boldsymbol{A}(k)S(k,\ell) + \boldsymbol{W}^{\mathrm{H}}(k,\ell)\boldsymbol{N}_{\mathrm{s}}(k,\ell) \\ &\quad + \boldsymbol{W}^{\mathrm{H}}(k,\ell)\boldsymbol{N}_{\mathrm{t}}(k,\ell) \\ &\triangleq Y_{\mathrm{s}}(k,\ell) + Y_{n,\mathrm{s}}(k,\ell) + Y_{n,\mathrm{t}}(k,\ell)\,, \end{aligned} \tag{47.4}$$

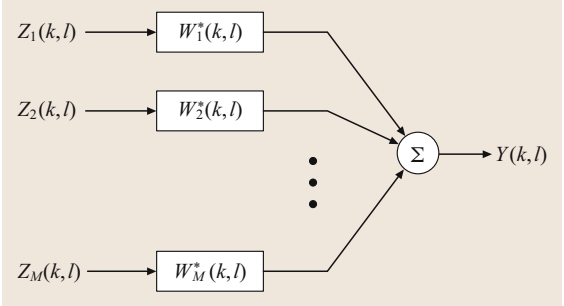


Fig. 47.1 Filter-and-sum beamformer

where $Y_{\mathrm{s}}(k,\ell)$ is the signal component and $Y_{n,\mathrm{s}}(k,\ell)$ and $Y_{n,\mathrm{t}}(k,\ell)$ are the stationary and nonstationary noise components, respectively. The output power of the beamformer is given by

$$\begin{aligned} &E\{Y(k,\ell)Y^*(k,\ell)\} \\ &= E\{\boldsymbol{W}^{\mathrm{H}}(k,\ell)\boldsymbol{Z}(k,\ell)\boldsymbol{Z}^{\mathrm{H}}(k,\ell)\boldsymbol{W}(k,\ell)\} \\ &= \boldsymbol{W}^{\mathrm{H}}(k,\ell)\boldsymbol{\phi}_{\boldsymbol{ZZ}}(k,\ell)\boldsymbol{W}(k,\ell)\,, \end{aligned}$$

where $\boldsymbol{\phi}_{\boldsymbol{ZZ}}(k,\ell) \triangleq E\{\boldsymbol{Z}(k,\ell)\boldsymbol{Z}^{\mathrm{H}}(k,\ell)\}$ is the power spectral density (PSD) matrix of the received signals. We want to minimize the output power subject to the following constraint on $Y_{\mathrm{s}}(k,\ell)$:

$$\begin{aligned} Y_{\mathrm{s}}(k,\ell) &= \boldsymbol{W}^{\mathrm{H}}(k,\ell)\boldsymbol{A}(k)S(k,\ell) \\ &= \mathcal{F}^*(k,\ell)S(k,\ell)\,, \end{aligned}$$

where $\mathcal{F}(k,\ell)$ is some prespecified filter, usually a simple delay. Without loss of generality we assume hereinafter that $\mathcal{F}(k,\ell)=1$. Hence, the minimization problem can be stated as

$$\begin{aligned} &\min_{\boldsymbol{W}} \left\{\boldsymbol{W}^{\mathrm{H}}(k,\ell)\boldsymbol{\phi}_{\boldsymbol{ZZ}}(k,\ell)\boldsymbol{W}(k,\ell)\right\} \\ &\text{subject to} \quad \boldsymbol{W}^{\mathrm{H}}(k,\ell)\boldsymbol{A}(k)=1\,. \end{aligned} \tag{47.5}$$

The minimization problem (47.5) is demonstrated in Fig. 47.2. The point where the equipower contours are tangent to the constraint plane is the optimum vector of beamforming filters. The perpendicular $\boldsymbol{F}(k)$ from the origin to the constraint plane will be calculated in the next section.

To solve (47.5) we first define the complex Lagrangian,

$$\begin{aligned} \mathcal{L}(\boldsymbol{W}) &= \boldsymbol{W}^{\mathrm{H}}(k,\ell)\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)\boldsymbol{W}(k,\ell) \\ &\quad + \lambda[\boldsymbol{W}^{\mathrm{H}}(k,\ell)\boldsymbol{A}(k)-1] \\ &\quad + \lambda^*[\boldsymbol{A}^{\mathrm{H}}(k,\ell)\boldsymbol{W}(k)-1]\,, \end{aligned} \tag{47.6}$$

where λ is a Lagrange multiplier. Setting the derivative with respect to $\boldsymbol{W}^*$ to 0 [47.48] yields

$$\nabla_{\boldsymbol{W}^*}\mathcal{L}(W)\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)\boldsymbol{W}(k,\ell)+\lambda\boldsymbol{A}(k)=0.$$

Now, recalling the constraint in (47.5), we obtain the LCMV optimal filter

$$\boldsymbol{W}^{\mathrm{LCMV}}(k,\ell) = \frac{\boldsymbol{\Phi}_{\boldsymbol{ZZ}}^{-1}(k,\ell)\boldsymbol{A}(k)}{\boldsymbol{A}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{ZZ}}^{-1}(k,\ell)\boldsymbol{A}(k)}\,. \tag{47.7}$$

This closed-form solution is difficult to implement, and is not suitable for time-varying environments. Therefore we often have to resort to an adaptive solution, which is derived in the sequel.

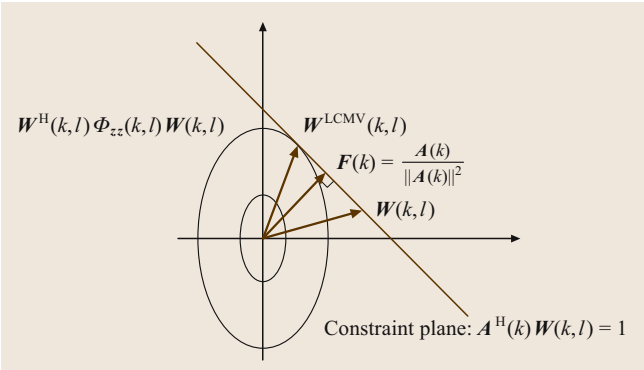


Fig. 47.2 Constrained minimization

It is interesting to show the equivalence between the LCMV solution (47.7) and the MSNR beamformer [47.7], which is obtained from

$$\max_{\boldsymbol{W}} \frac{|\boldsymbol{W}^{\mathrm{H}}(k,\ell)\boldsymbol{A}(k)|^2}{\boldsymbol{W}^{\mathrm{H}}(k,\ell)\boldsymbol{\Phi}_{\boldsymbol{NN}}(k,\ell)\boldsymbol{W}(k,\ell)}\,. \tag{47.8}$$

The well-known solution to (47.8) is the (colored-noise) matched filter

$$\boldsymbol{W}(k,\ell) \propto \boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}(k,\ell)\boldsymbol{A}(k)\,.$$

If the array response is constrained to fulfil $\boldsymbol{W}^{\mathrm{H}}(k,\ell)\boldsymbol{A}(k)=1$, i. e., no distortion in the desired direction, we have

$$\boldsymbol{W}^{\mathrm{MSNR}}(k,\ell) = \frac{\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}(k,\ell)\boldsymbol{A}(k)}{\boldsymbol{A}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}(k,\ell)\boldsymbol{A}(k)}\,. \tag{47.9}$$

Using (47.3) it can be verified that

$$\begin{aligned} \boldsymbol{\Phi}_{\boldsymbol{ZZ}} &= \phi_{\mathrm{ss}}(k,\ell)\boldsymbol{A}(k)\boldsymbol{A}^{\mathrm{H}}(k) \\ &\quad + \boldsymbol{\Phi}_{\boldsymbol{N}_{\mathrm{s}}\boldsymbol{N}_{\mathrm{s}}}(k,\ell) + \boldsymbol{\Phi}_{\boldsymbol{N}_{\mathrm{t}}\boldsymbol{N}_{\mathrm{t}}}(k,\ell) \\ &= \phi_{\mathrm{ss}}(k,\ell)\boldsymbol{A}(k)\boldsymbol{A}^{\mathrm{H}}(k) + \boldsymbol{\Phi}_{\boldsymbol{NN}}(k,\ell)\,, \end{aligned} \tag{47.10}$$

where $\boldsymbol{\Phi}_{\boldsymbol{NN}}(k,\ell) \triangleq \boldsymbol{\Phi}_{\boldsymbol{N}_{\mathrm{s}}\boldsymbol{N}_{\mathrm{s}}}(k,\ell)+\boldsymbol{\Phi}_{\boldsymbol{N}_{\mathrm{t}}\boldsymbol{N}_{\mathrm{t}}}(k,\ell)$, the overall noise PSD matrix. Using the matrix inversion lemma, it is shown in Appendix 47.B that

$$\boldsymbol{W}^{\mathrm{LCMV}}(k,\ell) = \frac{\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}(k,\ell)\boldsymbol{A}(k)}{\boldsymbol{A}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}(k,\ell)\boldsymbol{A}(k)}\,. \tag{47.11}$$

This solution is identical to the solution of the MSNR beamformer.

While both methods are shown to be equal, provided that the ATFs $\boldsymbol{A}(k)$ are known, their behavior in the case of unknown ATFs is different. Analysis of these differences is given by *Cox* [47.47].

Amazon EX1010 Page 950

Note also that, due to the nonstationary noise component, the term $\boldsymbol{\Phi}_{\boldsymbol{NN}}(k,\ell)$ depends on the frame index. This time dependence is one of the major factors leading to performance degradation in beamforming. We address this problem by introducing the multichannel postfilter in Sect. 47.6.

Adaptive Solution

Consider the following steepest-descent adaptive algorithm:

$$\begin{aligned}&\boldsymbol{W}(k,\ell+1)\\&=\boldsymbol{W}(k,\ell)-\mu\nabla_{\boldsymbol{W}^*}\mathcal{L}(k,\ell)\\&=\boldsymbol{W}(k,\ell)-\mu\left[\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)\boldsymbol{W}(k,\ell)+\lambda\boldsymbol{A}(k)\right]\,.\end{aligned}$$

Imposing the look-direction constraint on $\boldsymbol{W}(\ell+1,k)$ yields

$$\begin{aligned}1&=\boldsymbol{A}^{\mathrm{H}}(k)\boldsymbol{W}(k,\ell+1)\\&=\boldsymbol{A}^{\mathrm{H}}(k)\boldsymbol{W}(k,\ell)-\mu\boldsymbol{A}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)\boldsymbol{W}(k,\ell)\\&\quad-\mu\boldsymbol{A}^{\mathrm{H}}(k)\boldsymbol{A}(k)\lambda\,.\end{aligned}$$

Solving for the Lagrange multiplier and applying further rearrangement of terms yields:

$$\begin{aligned}&\boldsymbol{W}(k,\ell+1)\\&=P(k)\boldsymbol{W}(k,\ell)-\mu P(k)\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)\boldsymbol{W}(k,\ell)+\boldsymbol{F}(k)\,,\end{aligned}\tag{47.12}$$

where

$$P(k)\triangleq\mathbf{I}-\frac{\boldsymbol{A}(k)\boldsymbol{A}^{\mathrm{H}}(k)}{\|\boldsymbol{A}(k)\|^2}\tag{47.13}$$

and

$$\boldsymbol{F}(k)\triangleq\frac{\boldsymbol{A}(k)}{\|\boldsymbol{A}(k)\|^2}.\tag{47.14}$$

Further simplification can be obtained by replacing $\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)$ by its instantaneous estimator, $\boldsymbol{Z}(k,\ell)\boldsymbol{Z}^{\mathrm{H}}(k,\ell)$, and recalling (47.4). We finally obtain,

$$\begin{aligned}&\boldsymbol{W}(k,\ell+1)\\&=P(k)[\boldsymbol{W}(k,\ell)-\mu\boldsymbol{Z}(k,\ell)Y^*(k,\ell)]+\boldsymbol{F}(k)\,.\end{aligned}$$

The entire algorithm is summarized in Table 47.1.

Table 47.1 Frequency-domain Frost algorithm

$\boldsymbol{W}(\ell=0,k)=\boldsymbol{F}(k)$
$\boldsymbol{W}(t+1,k)=P(k)\left[\boldsymbol{W}(k,\ell)-\mu\boldsymbol{Z}(k,\ell)Y^*(k,\ell)\right]+\boldsymbol{F}(k)$
$\ell=0,1,\ldots$
[$P(k)$ and $\boldsymbol{F}(k)$ are defined by (47.13) and (47.14)].

47.2.2 Frequency-Domain Generalized Sidelobe Canceller

In [47.36], *Griffiths* and *Jim* considered the case where each ATF is a delay element (with gain). They obtained an unconstrained adaptive enhancement algorithm, using the same linear constraint imposed by *Frost* [47.35]. The unconstrained algorithm is more tractable, reliable, and computationally more efficient in comparison with its constrained counterpart. In the adaptive solution Section, we obtained an adaptive algorithm for the case where each ATF is represented by an arbitrary linear time-invariant system. We now repeat the arguments of Griffiths and Jim for the arbitrary ATFs case, and derive an unconstrained adaptive enhancement algorithm. A detailed description can be found in [47.38].

Derivation

Consider the null space of $\boldsymbol{A}(k)$, defined by

$$\mathcal{N}(k)\triangleq\{\boldsymbol{W}\,|\,\boldsymbol{A}^{\mathrm{H}}(k)\boldsymbol{W}=0\}\,.$$

The constraint hyperplane,

$$\Lambda(k)\triangleq\{\boldsymbol{W}\,|\,\boldsymbol{A}^{\mathrm{H}}(k)\boldsymbol{W}=1\}$$

is parallel to $\mathcal{N}(k)$. In addition, let

$$\mathcal{R}(k)\triangleq\{\kappa\boldsymbol{A}(k)\,|\,\text{for any real }\kappa\}$$

be the column space. By the fundamental theorem of linear algebra [47.49], $\mathcal{R}(k)\perp\mathcal{N}(k)$. In particular, $\boldsymbol{F}(k)$ is perpendicular to $\mathcal{N}(k)$, since $\boldsymbol{F}(k)=\frac{1}{\|\boldsymbol{A}(k)\|^2}\boldsymbol{A}(k)\in\mathcal{R}(k)$. Furthermore,

$$\boldsymbol{A}^{\mathrm{H}}(k)\boldsymbol{F}(k)=\boldsymbol{A}^{\mathrm{H}}(k)\boldsymbol{A}(k)(\boldsymbol{A}^{\mathrm{H}}(k)\boldsymbol{A}(k))^{-1}=1\,.$$

Thus, $\boldsymbol{F}(k)\in\Lambda(k)$ and $\boldsymbol{F}(k)\perp\Lambda(k)$. Hence, $\boldsymbol{F}(k)$ is the perpendicular from the origin to the constraint hyperplane, $\Lambda(k)$. The matrix $P(k)$, defined in (47.13), is the projection matrix to the null space of $\boldsymbol{A}(k)$, $\mathcal{N}(k)$.

A vector in linear space can be uniquely split into a sum of two vectors in mutually orthogonal subspaces [47.49]. Hence,

$$\boldsymbol{W}(k,\ell)=\boldsymbol{W}_0(k,\ell)-\boldsymbol{V}(k,\ell)\,,\tag{47.15}$$

where $\boldsymbol{W}_0(k,\ell)\in\mathcal{R}(k)$ and $-\boldsymbol{V}(k,\ell)\in\mathcal{N}(k)$. By the definition of $\mathcal{N}(k)$,

$$\boldsymbol{V}(k,\ell)=\mathcal{H}(k)\boldsymbol{G}(k,\ell)\,,\tag{47.16}$$

where $\mathcal{H}(k)$ is a matrix such that its columns span the null space of $\boldsymbol{A}(k)$, i. e.,

$$\boldsymbol{A}^{\mathrm{H}}(k)\mathcal{H}(k)=0\,,\qquad\operatorname{rank}\{\mathcal{H}(k)\}\le M-1\,,\tag{47.17}$$

Amazon EX1010 Page 951

where $\mathcal{H}(k)$ is usually called a BM (blocking matrix). The outputs of the BM will be denoted, for reasons that will be clear in the sequel, noise reference signals $\boldsymbol{U}(k,\ell)$, defined as

$$\boldsymbol{U}(k,\ell) = \mathcal{H}^{\mathrm{H}}(k)\boldsymbol{Z}(k,\ell)\,, \tag{47.18}$$

where

$$\boldsymbol{U}(k,\ell) = \big(U_2(k,\ell)\;\; U_3(k,\ell)\;\; \ldots\;\; U_M(k,\ell)\big)^{\mathrm{T}}\,.$$

The vector $\boldsymbol{G}(k,\ell)$ is a rank$\{\mathcal{H}(k)\}\times 1$ vector of adjustable filters. We assume hereinafter that rank$\{\mathcal{H}(k)\} = M-1$. Hence, the set of filters is defined as

$$\boldsymbol{G}(k,\ell) = \big(G_2(k,\ell)\;\; G_3(k,\ell)\;\; \ldots\;\; G_M(k,\ell)\big)^{\mathrm{T}}\,. \tag{47.19}$$

By the geometrical interpretation of Frost's algorithm,

$$\boldsymbol{W}_0(k,\ell) = \boldsymbol{F}(k) = \frac{\boldsymbol{A}(k)}{\|\boldsymbol{A}(k)\|^2} \tag{47.20}$$

(Recall that $\boldsymbol{F}(k)$ is the perpendicular from the origin to the constraint hyperplane, $\Lambda(k)$.) Now, using (47.4), (47.15), and (47.16) we obtain

$$Y(k,\ell) = Y_{\mathrm{FBF}}(k,\ell) - Y_{\mathrm{ANC}}(k,\ell)\,, \tag{47.21}$$

where

$$\begin{aligned} Y_{\mathrm{FBF}}(k,\ell) &= \boldsymbol{W}_0^{\mathrm{H}}(k,\ell)\boldsymbol{Z}(k,\ell) \\ Y_{\mathrm{ANC}}(k,\ell) &= \boldsymbol{G}^{\mathrm{H}}(k,\ell)\mathcal{H}^{\mathrm{H}}(k)\boldsymbol{Z}(k,\ell)\,. \end{aligned} \tag{47.22}$$

The output of the constrained beamformer is a difference of two terms, both operating on the input signal $\boldsymbol{Z}(k,\ell)$. The first term, $Y_{\mathrm{FBF}}(k,\ell)$, utilizes only fixed components (which depend on the ATFs), so it can be viewed as a FBF. The FBF coherently sums the desired speech components, while in general it destructively sums the noise components. Hence, it is expected that the signal-to-noise ratio (SNR) at the FBF output will be higher than the input SNR. However, this result cannot be guaranteed. We will elaborate on this issue while discussing the performance analysis in Sect. 47.7.

We now examine the second term $Y_{\mathrm{ANC}}(k,\ell)$. Note that

$$\begin{aligned} \boldsymbol{U}(k,\ell) &= \mathcal{H}^{\mathrm{H}}(k)\boldsymbol{Z}(k,\ell) \\ &= \mathcal{H}^{\mathrm{H}}(k)[\boldsymbol{A}(k)S(k,\ell) \\ &\quad + \boldsymbol{N}_{\mathrm{s}}(k,\ell) + \boldsymbol{N}_{\mathrm{t}}(k,\ell)] \\ &= \mathcal{H}^{\mathrm{H}}(k)[\boldsymbol{N}_{\mathrm{s}}(k,\ell) + \boldsymbol{N}_{\mathrm{t}}(k,\ell)]\,. \end{aligned} \tag{47.23}$$

The last transition is due to (47.17). It is worth mentioning that, when a perfect BM is applied, $\boldsymbol{U}(k,\ell)$ indeed contains only noise components. In general, however, $\mathcal{H}^{\mathrm{H}}(k)\boldsymbol{A}(k,\ell) \neq 0$, hence desired speech components may leak into the noise reference signals. If the speech component is indeed completely eliminated (blocked) by $\mathcal{H}(k)$, $Y_{\mathrm{ANC}}(k,\ell)$ becomes a pure noise term. The residual noise term in $Y_{\mathrm{FBF}}(k,\ell)$ can then be reduced by properly adjusting the filters $\boldsymbol{G}(k,\ell)$, using the minimum output power criterion. This minimization problem is in fact the classical multichannel noise cancellation problem. An adaptive LMS solution to the problem was proposed by *Widrow* [47.25].

To summarize, the beamformer is comprised of three parts. An FBF $\boldsymbol{W}_0$, which aligns the desired signal components, a BM $\mathcal{H}(k)$, which blocks the desired speech components resulting in the reference noise signals $\boldsymbol{U}(k,\ell)$, and a multichannel ANC $\boldsymbol{G}(k,\ell)$, which eliminates the stationary noise that leaks through the sidelobes of the FBF.

Noise Canceller Adaptation

The reference noise signals are emphasized by the ANC and subtracted from the output of the FBF, yielding

$$Y(k,\ell) = \big[\boldsymbol{W}_0^{\mathrm{H}}(k,\ell) - \boldsymbol{G}^{\mathrm{H}}(k,\ell)\mathcal{H}^{\mathrm{H}}(k)\big]\boldsymbol{Z}(k,\ell)\,. \tag{47.24}$$

Let three hypotheses $H_{0\mathrm{s}}$, $H_{0\mathrm{t}}$, and H_1 indicate, respectively, the absence of transients, the presence of an interfering transient, and the presence of a desired source transient at the beamformer output. The optimal solution for the filters $\boldsymbol{G}(k,\ell)$ is obtained by minimizing the power of the beamformer output during the stationary noise frames (i. e., when $H_{0\mathrm{s}}$ is true) [47.2]. We note, however, that no adaptation should be carried out during abrupt changes in the characteristics of the noise signal (e.g., a passing car). When the noise source position is constant and the noise statistics is slowly varying, the ANC filters can track the changes.

Let $\boldsymbol{\Phi}_{\boldsymbol{N}_{\mathrm{s}}\boldsymbol{N}_{\mathrm{s}}}(k,\ell) = E\{\boldsymbol{N}_{\mathrm{s}}(k,\ell)\boldsymbol{N}_{\mathrm{s}}^{\mathrm{H}}(k,\ell)\}$ denote the PSD matrix of the input stationary noise. Then, the power of the stationary noise at the beamformer output is minimized by solving the unconstrained optimization problem:

$$\begin{aligned} \min_{\boldsymbol{G}}\{&[\boldsymbol{W}_0(k,\ell) - \mathcal{H}(k,\ell)\boldsymbol{G}(k,\ell)]^{\mathrm{H}} \\ &\times \boldsymbol{\Phi}_{\boldsymbol{N}_{\mathrm{s}}\boldsymbol{N}_{\mathrm{s}}}(k,\ell)[\boldsymbol{W}_0(k,\ell) - \mathcal{H}(k,\ell)\boldsymbol{G}(k,\ell)]\}\,. \end{aligned} \tag{47.25}$$

A multichannel Wiener solution is given by (see also [47.42, 46])

$$\begin{aligned} \boldsymbol{G}(k,\ell) = &\big[\mathcal{H}^{\mathrm{H}}(k,\ell)\boldsymbol{\Phi}_{\boldsymbol{N}_{\mathrm{s}}\boldsymbol{N}_{\mathrm{s}}}(k,\ell)\mathcal{H}(k)\big]^{-1} \\ &\times \mathcal{H}^{\mathrm{H}}(k,\ell)\boldsymbol{\Phi}_{\boldsymbol{N}_{\mathrm{s}}\boldsymbol{N}_{\mathrm{s}}}(k,\ell)\boldsymbol{W}_0(k,\ell)\,. \end{aligned} \tag{47.26}$$

In practice, this optimization problem is solved by using the normalized LMS algorithm [47.2]:

$$\boldsymbol{G}(k,\ell+1) = \begin{cases} \boldsymbol{G}(k,\ell) + \frac{\mu_g}{P_{\text{est}}(k,\ell)}\boldsymbol{U}(k,\ell)Y^*(k,\ell) & H_{0s} \text{ true}, \\ \boldsymbol{G}(k,\ell)\,, & \text{otherwise}, \end{cases} \tag{47.27}$$

where

$$P_{\text{est}}(k,\ell) = \alpha_p P_{\text{est}}(k,\ell-1) + (1-\alpha_p)\|\boldsymbol{U}(k,\ell)\|^2 \tag{47.28}$$

represents the power of the noise reference signals, μ_g is a step size that regulates the convergence rate, and α_p is a smoothing parameter in the PSD estimation process.

To allow for the use of the STFT, we further assume that the ANC filters $\boldsymbol{g}_m$ have a time-varying finite impulse response (FIR) structure:

$$\boldsymbol{g}_m^{\mathrm{T}}(t) = \left(g_{m,-K_L}(t) \;\; \ldots \;\; g_{m,K_R}(t)\right). \tag{47.29}$$

Note, that the impulse responses are taken to be non-causal, to allow for relative delays between the FBF and the ANC branches.

In order to fulfill the FIR structure constraint in (47.29), the filters update is now given by

$$\tilde{\boldsymbol{G}}(\ell+1,k) = \boldsymbol{G}(k,\ell) + \mu\frac{\boldsymbol{U}(k,\ell)Y^*(k,\ell)}{P_{\text{est}}(k,\ell)}\,,$$
$$\boldsymbol{G}(\ell+1,k) \xleftarrow{\text{FIR}} \tilde{\boldsymbol{G}}(\ell+1,k)\,. \tag{47.30}$$

The operator $\xleftarrow{\text{FIR}}$ includes the following three stages, applied per filter: transformation of $\tilde{G}_m(\ell+1,k)$ to the time domain, truncation of the resulting impulse response to the interval $[-K_L, K_R]$ (i. e., imposing the FIR constraint), and transformation back to the frequency domain. The various filtering operations involved in the algorithm (multiplications in the transform domain) are realized using the overlap-and-save method [47.50, 51].

The resulting algorithm is merely an extension of the original Griffiths and Jim algorithm for the arbitrary ATF case. Figure 47.3 depicts a block diagram of the algorithm. The steps involved in the computation are summarized in Table 47.2. The matched beamformer $\boldsymbol{W}_0(k)$ and the BM $\mathcal{H}(k)$ are assumed to be known at this stage.

47.2.3 Time-Domain Generalized Sidelobe Canceller

The most commonly used GSC structure is the classical time-domain counterpart of the algorithm, proposed

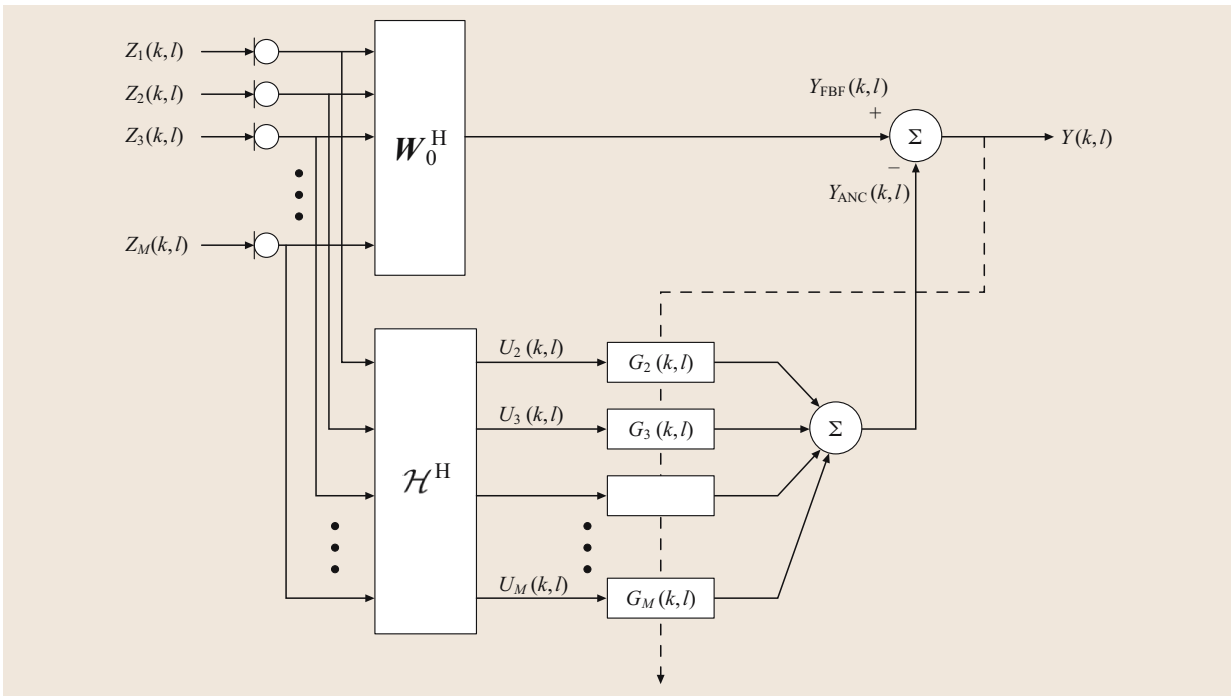


Fig. 47.3 Linearly constrained adaptive beamformer

Table 47.2 Frequency-domain GSC algorithm

1) Fixed beamformer:
$Y_{\mathrm{FBF}}(k,\ell) = \boldsymbol{W}_0^{\mathrm{H}}(k)\boldsymbol{Z}(k,\ell)$
2) Noise reference signals:
$\boldsymbol{U}(k,\ell) = \mathcal{H}^{\mathrm{H}}(k)\boldsymbol{Z}(k,\ell)$
3) Output signal:
$Y(k,\ell) = Y_{\mathrm{FBF}}(k,\ell) - \boldsymbol{G}^{\mathrm{H}}(k,\ell)\boldsymbol{U}(k,\ell)$
4) Filters update
$\tilde{\boldsymbol{G}}(t+1,k) = \boldsymbol{G}(k,\ell) + \mu_{\mathrm{g}} \frac{\boldsymbol{U}(k,\ell)Y^*(k,\ell)}{P_{\mathrm{est}}(k,\ell)}$
$\boldsymbol{G}(\ell+1,k) \overset{\mathrm{FIR}}{\longleftarrow} \tilde{\boldsymbol{G}}(\ell+1,k)\,,$
where
$P_{\mathrm{est}}(k,\ell) = \alpha_{\mathrm{p}} P_{\mathrm{est}}(\ell-1,k) + (1-\alpha_{\mathrm{p}}) \sum_m \lvert Z_m(k,\ell)\rvert^2$
5) Keep only nonaliased samples

by *Griffiths* and *Jim* [47.36]. For completeness of the exposition, we present now the time-domain algorithm.

Assuming that the array is steered towards the desired speech signal (refer to steering-related issues in Sect. 47.4), the FBF is given by

$$y_{\mathrm{FBF}}(t) = \sum_{m=1}^{M} z_m(t)\,,$$

which is the simple delay-and-sum beamformer. Under the same delay-only steered array assumptions, it is evident that

$$u_m(t) = z_m(t) - z_1(t);\ m = 2, \ldots, M\,,$$

are noise-only signals and that the desired speech component is cancelled out.

The filters $\boldsymbol{g}_m$ are updated in the time domain. The error signal (which is also the output of the enhancement algorithm) is given by,

$$y(t) = y_{\mathrm{FBF}}(t) - \sum_{m=2}^{M} \sum_{i=-K_L}^{K_R} g_{m,i}(t) u_m(t-i)\,. \tag{47.31}$$

Define, for $m = 2, \ldots, M$:

$$\begin{aligned}&\boldsymbol{u}_m^{\mathrm{T}}(t)\\ &= \big(u_m(t+K_L) \ \cdots \ u_m(t) \ \cdots \ u_m(t-K_R)\big)\,.\end{aligned}$$

Then, the adaptive normalized multichannel LMS solution is given by

$$\begin{aligned}&\boldsymbol{g}_m(t+1)\\ &= \boldsymbol{g}_m(t) + \frac{\mu}{p_{\mathrm{est}}(t)} \boldsymbol{u}_m(t) y(t);\ m = 2, \ldots, M\,,\end{aligned}$$

where

$$p_{\mathrm{est}}(t) = \sum_{m=1}^{M} \lVert \boldsymbol{u}_m(t)\rVert^2\,. \tag{47.32}$$

47.3 Fixed Beamformer and Blocking Matrix

In the previous section, we derived the generalized sidelobe canceller and showed that it includes a fixed beamformer, given in (47.20), a blocking matrix, given in (47.17), and a multichannel ANC, given in (47.26). Note that knowledge of the ATFs $\boldsymbol{A}(k)$ (assumed to be slowly time variant) suffices to determine both the FBF and BM. In this section we present three methods for determination of the fixed beamformer and the blocking matrix.

47.3.1 Using Acoustical Transfer Functions

A typical RIR is depicted in Fig. 47.4. It can be seen that the impulse response can get very long (several thousand taps), which makes the estimation task quite cumbersome. This impulse response was generated by the image method [47.52] proposed by Allen and Berkley (The authors thank E. A. P. Habets from TU Eindhoven, The Netherlands, for providing an efficient implementation

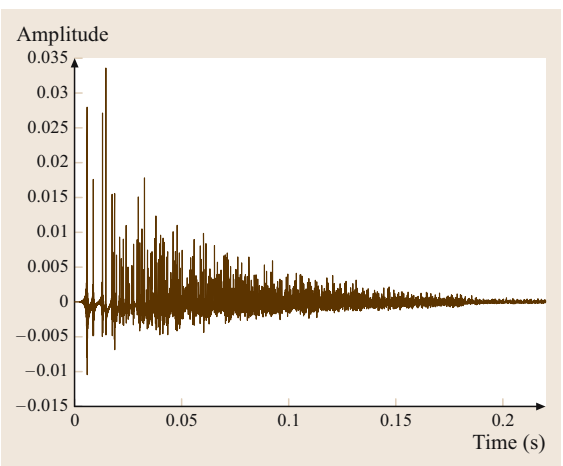


Fig. 47.4 A typical room impulse response with a reverberation time of 0.4 s

of the image method.). By this method, the RIR is generated by simulating multiple reflections of the sound source from the room walls. Two distinct segments of the impulse response can be observed. The first consists of the direct propagation path with a few early, distinguishable reflections. The second segment consists of overlapping random arrivals, with an exponentially decaying envelope. This segment is usually referred to as the tail of the RIR. Using this model we can estimate the various blocks of the GSC.

Assuming that $\boldsymbol{A}(k)$ is known, we have by (47.22), (47.20), and (47.3)

$$Y_{\mathrm{FBF}}(k,\ell) = S(k,\ell) + \frac{\boldsymbol{A}^{\mathrm{H}}(k)}{\|\boldsymbol{A}(k)\|^2}[\boldsymbol{N}_{\mathrm{s}}(k,\ell)+\boldsymbol{N}_{\mathrm{t}}(k,\ell)]\,. \quad (47.33)$$

The first term on the right-hand side is the signal term, and the second is the noise term. The FBF in this case is hence a matched filter-and-sum beamformer (see also [47.16, 17]).

Considering the blocking matrix, there are many alternatives for blocking the desired speech signal in the reference channels. One alternative is calculation of

$$U_m(k,\ell) = A_m(k)Z_{m-1}(k,\ell) - A_{m-1}(k)Z_m(k,\ell)$$

for $m = 2,\ldots,M$. Any other combination of the microphone signals is applicable.

47.3.2 Using Delay-Only Filters

The simplest and yet the most widely used model for the ATF is a delay-only model. Arbitrarily defining the first microphone as the reference microphone we have

$$\boldsymbol{A}(k) = \left(1 \;\; \mathrm{e}^{-\mathrm{i}\frac{2\pi}{K}\tau_2} \;\; \mathrm{e}^{-\mathrm{i}\frac{2\pi}{K}\tau_3} \;\; \ldots \;\; \mathrm{e}^{-\mathrm{i}\frac{2\pi}{K}\tau_M}\right),$$

where $\tau_2,\ldots,\tau_M$ are the relative delays between each microphone and the reference microphone.

In the delay-only case, the FBF simplifies to the delay-and-sum beamformer, given by

$$\boldsymbol{W}_0(k) = \left(1 \;\; \mathrm{e}^{\mathrm{i}\frac{2\pi}{K}\tau_2} \;\; \mathrm{e}^{\mathrm{i}\frac{2\pi}{K}\tau_3} \;\; \ldots \;\; \mathrm{e}^{\mathrm{i}\frac{2\pi}{K}\tau_M}\right).$$

To avoid noncausal delays, a fixed amount of delay can be introduced.

It can easily be verified that the matrix

$$\mathcal{H}(k) = \begin{pmatrix} -\mathrm{e}^{\mathrm{i}\frac{2\pi}{K}\tau_2} & -\mathrm{e}^{\mathrm{i}\frac{2\pi}{K}\tau_2} & \ldots & -\mathrm{e}^{\mathrm{i}\frac{2\pi}{K}\tau_2} \\ 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ & \ldots & \ddots & \\ 0 & 0 & \ldots & 1 \end{pmatrix} \quad (47.34)$$

is a proper BM under the assumption of delay-only impulse response. This BM, originally proposed Griffiths and *Jim* [47.36], performs delay compensation and subtraction. It can be regarded as steering $M-1$ null beams towards the desired speech signal.

47.3.3 Using Relative Transfer Functions

We have shown, on the one hand, that the RIR can be very long and hence difficult to estimate. On the other hand, the use of delay-only model suffers from severe undermodeling problems. A good compromise is the use of the relative transfer function (RTF) between sensors. Define the RTFs as the ratio

$$\tilde{\boldsymbol{A}}^{\mathrm{T}}(k) \triangleq \left(1 \;\; \frac{A_2(k)}{A_1(k)} \;\; \frac{A_3(k)}{A_1(k)} \;\; \ldots \;\; \frac{A_M(k)}{A_1(k)}\right) = \frac{\boldsymbol{A}^{\mathrm{T}}(k)}{A_1(k)}\,. \quad (47.35)$$

Note that the ATF may have zeros outside the unit circle, as it is not necessarily a minimum-phase system. Thus to ensure stability of the RTFs we allow for noncausal systems. Therefore, we model the impulse response of the m-th ratio as

$$\tilde{\boldsymbol{a}}_m^{\mathrm{T}}(t) = \left(\tilde{a}_{m,-q_L}(t) \;\; \ldots \;\; \tilde{a}_{m,q_R}(t)\right). \quad (47.36)$$

It was experimentally shown that RTFs are usually much shorter than the corresponding ATFs [47.38], hence the FIR assumption may be justified.

Replacing in (47.20) the actual ATFs by the RTFs, the FBF becomes

$$\boldsymbol{W}_0(k) = \frac{\tilde{\boldsymbol{A}}(k)}{\|\tilde{\boldsymbol{A}}(k)\|^2}\,. \quad (47.37)$$

By (47.22) and (47.3) we then have

$$Y_{\mathrm{FBF}}(k,\ell) = A_1(k)S(k,\ell) + \frac{\tilde{\boldsymbol{A}}^{\mathrm{H}}(k)}{\|\tilde{\boldsymbol{A}}(k)\|^2}[\boldsymbol{N}_{\mathrm{s}}(k,\ell)+\boldsymbol{N}_{\mathrm{t}}(k,\ell)]\,. \quad (47.38)$$

Thus, when $\boldsymbol{W}_0(k,\ell)$ is given by (47.37), the signal term of $Y_{\mathrm{FBF}}(k,\ell)$ is the desired signal distorted only by the first ATF, $A_1(k)$. Note, however, that all the sensor outputs are summed coherently.

It can be easily verified that the use of the following BM suffices for completely eliminating the desired speech signal, provided that the RTFs are correctly modeled and estimated:

$$\mathcal{H}(k)=\begin{pmatrix} -\tilde{A}_2^*(k) & -\tilde{A}_3^*(k) & \dots & -\tilde{A}_M^*(k) \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ & \dots & \ddots & \\ 0 & 0 & \dots & 1 \end{pmatrix} . \quad (47.39)$$

For this choice of the BM, the reference signals are given by,

$$U_m(k,\ell) = Z_m(k,\ell) - \tilde{A}_m(k) Z_1(k,\ell); \quad m = 2, 3, \dots, M . \quad (47.40)$$

47.4 Identification of the Acoustical Transfer Function

The specific choice of the ATFs model governs the applicable estimation method.

47.4.1 Signal Subspace Method

Affes and *Grenier* [47.37] prove that the identification of source-to-array impulse responses is possible by subspace tracking. Assume that the approximation in (47.3) is valid, i. e., the multichannel correlation matrix of the received signals is given by

$$\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell) = \phi_{\mathrm{ss}}(k,\ell)\boldsymbol{A}(k)\boldsymbol{A}^{\mathrm{H}}(k) + \boldsymbol{\Phi}_{\boldsymbol{NN}}(k,\ell) .$$

Assume also that the noise signal is spatially white, i. e., $\boldsymbol{\Phi}_{\boldsymbol{NN}}(k,\ell) = \sigma_n^2(k,\ell)\mathbf{I}$ (an extension for spatially nonwhite noise is described in [47.53]). The eigenvalues of the received signals correlation matrix are given by

$$\begin{aligned} &\lambda_l = \sigma_n^2(k,\ell) \quad l = 1, \dots, M-1 \\ &\lambda_M > \sigma_n^2(k,\ell) \ \text{ otherwise} . \end{aligned} \quad (47.41)$$

From the matrix structure we conclude that the most dominant eigenvector of $\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)$ is given by $\boldsymbol{A}(k)$ (up to a scale factor). Hence, using the eigenvalue decomposition of $\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)$, we can estimate the desired signal ATFs. In the case of spatially nonwhite noise signals the generalized eigenvalue decomposition (GEVD), using the known noise correlation matrix $\boldsymbol{\Phi}_{\boldsymbol{NN}}(k,\ell)$, can be applied instead [47.53].

Yang [47.54] proved that finding the most dominant eigenvectors is equivalent to minimizing a quadratic cost function. For the following derivation it is more convenient to deal with normalized terms. Let

$$\begin{aligned} \boldsymbol{Z}(k,\ell) &= \boldsymbol{A}(k)S(k,\ell) + \boldsymbol{N}(k,\ell) \\ &= \frac{\boldsymbol{A}(k)}{\|\boldsymbol{A}(k)\|}\|\boldsymbol{A}(k)\|S(k,\ell) + \boldsymbol{N}(k,\ell) \\ &\triangleq \bar{\boldsymbol{A}}(k)\bar{S}(k,\ell) + \boldsymbol{N}(k,\ell) , \end{aligned} \quad (47.42)$$

where $\bar{\boldsymbol{A}}(k) \triangleq \boldsymbol{A}(k)/\|\boldsymbol{A}(k)\|$ is the normalized ATFs vector, namely $\bar{\boldsymbol{A}}^{\mathrm{H}}(k)\bar{\boldsymbol{A}}(k) = 1$, and $\bar{S}(k,\ell) \triangleq \|\boldsymbol{A}(k)\| S(k,\ell)$ is the normalized desired speech signal.

When only the single most dominant eigenvector is required (as in the discussed case), *Affes* and *Grenier* [47.37] showed that Yang's criterion simplifies to the following minimization,

$$E\{\|[\mathbf{I} - \bar{\boldsymbol{A}}(k)\bar{\boldsymbol{A}}^{\mathrm{H}}(k)]\boldsymbol{Z}(k,\ell)\|^2\} . \quad (47.43)$$

Similar to the approximation used in the derivation of LMS algorithm, the received signal correlation matrix is approximated by its instantaneous value $\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell) \approx \boldsymbol{Z}(k,\ell)\boldsymbol{Z}^{\mathrm{H}}(k,\ell)$. Furthermore, approximating $\hat{\bar{\boldsymbol{A}}}^{\mathrm{H}}(k,\ell)\hat{\bar{\boldsymbol{A}}}(k,\ell) \approx 1$, where $\hat{\bar{\boldsymbol{A}}}(k,\ell)$ is an estimate of $\bar{\boldsymbol{A}}(k)$ at the current time instant, the following sequential procedure can be derived:

$$\begin{aligned} &\hat{\bar{\boldsymbol{A}}}(k,\ell+1) \\ &= \hat{\bar{\boldsymbol{A}}}(k,\ell) + \mu_{\mathrm{a}}(k,\ell)[\boldsymbol{Z}(k,\ell) \\ &\quad - \hat{\bar{\boldsymbol{A}}}(k,\ell)\hat{\bar{\boldsymbol{A}}}^{\mathrm{H}}(k,\ell)\boldsymbol{Z}(k,\ell)][\hat{\bar{\boldsymbol{A}}}^{\mathrm{H}}(k,\ell)\boldsymbol{Z}(k,\ell)]^* . \end{aligned} \quad (47.44)$$

Define $\bar{Y}_{\mathrm{FBF}}(k,\ell) \triangleq \bar{\boldsymbol{A}}^{\mathrm{H}}(k,\ell)\boldsymbol{Z}(k,\ell)$, and observe that the desired signal component at $\bar{Y}_{\mathrm{FBF}}(k,\ell)$ is $\bar{S}(k,\ell)$. Then,

$$\begin{aligned} &\hat{\bar{\boldsymbol{A}}}(k,\ell+1) \\ &= \hat{\bar{\boldsymbol{A}}}(k,\ell) + \mu_{\mathrm{a}}(k,\ell)[\boldsymbol{Z}(k,\ell) \\ &\quad - \hat{\bar{\boldsymbol{A}}}(k,\ell)\bar{Y}_{\mathrm{FBF}}(k,\ell)]\bar{Y}^*_{\mathrm{FBF}}(k,\ell) . \end{aligned} \quad (47.45)$$

Now, it is easy to verify that

$$\boldsymbol{Z}(k,\ell) - \hat{\bar{\boldsymbol{A}}}(k,\ell)\bar{Y}_{\mathrm{FBF}}(k,\ell)$$

are noise-only signals, provided that the estimate $\hat{\bar{\boldsymbol{A}}}(k,\ell)$ converges to the true normalized ATFs vector $\bar{\boldsymbol{A}}(k)$.

Hence, it can serve as a BM output, namely $\boldsymbol{U}(k,\ell)$. (There is a slight difference between the proposed BM and the conventional BM, as the number of the components in $\boldsymbol{U}(k,\ell)$ in the current scheme is M rather than $M-1$ as with the latter.) Collecting all terms we finally have

$$\hat{\tilde{\boldsymbol{A}}}(k,\ell+1) = \hat{\tilde{\boldsymbol{A}}}(k,\ell)+\mu_{\mathrm{a}}(k,\ell)\boldsymbol{U}(k,\ell)\bar{Y}^{*}_{\mathrm{FBF}}(k,\ell)\,. \tag{47.46}$$

Note that this procedure yields an estimate of the normalized ATFs rather than the ATFs themselves. *Affes* and *Grenier* [47.37] argue that $\|\boldsymbol{A}(k)\|$ is invariant to small talker movements and could be estimated in advance.

As a final remark, we would like to point out that the estimation procedure in (47.46) has many similarities to the method of *Hoshuyama* et al. [47.55] for robust design of the BM, and with the decorrelation criterion presented by *Weinstein* et al. [47.56] and further adapted to the GSC structure by *Gannot* [47.57]. We will elaborate on this issue in Sect. 47.5 when discussing the robust beamformers.

47.4.2 Time Difference of Arrival

When a delay-only steering is applied, an estimation of the time difference of arrival between the microphones suffices to model the entire impulse response. It should be noted however, that this procedure usually undermodels the RIR and is not sufficient for the problem at hand. Many algorithms were proposed for estimation the time difference of arrival (TDOA) [47.58, 59]. A survey of state-of-the-art methods for TDOA estimation can be found in [47.60]. This topic is beyond the scope of this chapter.

47.4.3 Relative Transfer Function Estimation

We present two methods for RTF estimation. The first is based on speech nonstationarity [47.38] , and the second employs the speech presence probability [47.61, 62].

Using Signal Nonstationarity

In this section, we review the system identification technique proposed by *Shalvi* and *Weinstein* [47.63] and later used by *Gannot* et al. [47.38] in the context of microphone arrays. This method relies on the assumptions that the background noise signal is stationary, that the desired signal $s(t)$ is nonstationary, and that the support of the relative impulse response between the sensors is finite and slowly time varying. (Note that the relative impulse response between the sensors is generally of infinite length, since it represents the ratio of ATFs. However, in real environments, the energy of the relative impulse response often decays much faster than the corresponding ATF [47.38]. Therefore, the finite-support assumption is practically not very restrictive.)

Rearranging terms in (47.40) we have

$$Z_m(k,\ell) = \tilde{A}_m(k)Z_1(k,\ell)+U_m(k,\ell)\,. \tag{47.47}$$

We assume that the RTFs are slowly changing in time compared to the time variations of the desired signal. We further assume that the statistics of the noise signal is slowly changing compared to the statistics of the desired signal. Consider some analysis interval during which the ATFs are assumed to be time invariant and the noise signal is assumed to be stationary. We divide that analysis interval into frames. Consider the i-th frame. By (47.47) we have

$$\boldsymbol{\Phi}^{(i)}_{z_m z_1}(k) = \tilde{A}_m(k)\boldsymbol{\Phi}^{(i)}_{z_1 z_1}(k)+\boldsymbol{\Phi}_{u_m z_1}(k)\,, \quad i=1,\ldots,I\,, \tag{47.48}$$

where I is the number of frames, $\boldsymbol{\Phi}^{(i)}_{z_i z_j}(k)$ is the cross-PSD between z_i and z_j at frequency bin k during the i-th frame, and $\boldsymbol{\Phi}_{u_m z_1}(k)$ is the cross-PSD between u_m and z_1 at frequency bin k, which is independent of the frame index due to the noise stationarity. Now, equations (47.2) and (47.40) imply that, when the signal is present,

$$U_m(k,\ell) = N^{\mathrm{s}}_m(k,\ell)-\tilde{A}_m(k)N^{\mathrm{s}}_1(k,\ell) \tag{47.49}$$

$$Z_1(k,\ell) = A_1(k)S(k,\ell)+N^{\mathrm{s}}_1(k,\ell)\,. \tag{47.50}$$

Let $\hat{\boldsymbol{\Phi}}^{(i)}_{z_1 z_1}(k)$, $\hat{\boldsymbol{\Phi}}^{(i)}_{z_m z_1}(k)$, and $\hat{\boldsymbol{\Phi}}^{(i)}_{u_m z_1}(k)$ be estimates of $\boldsymbol{\Phi}^{(i)}_{z_1 z_1}(k)$, $\boldsymbol{\Phi}^{(i)}_{z_m z_1}(k)$, and $\boldsymbol{\Phi}_{u_m z_1}(k)$, respectively. The estimates are obtained by replacing expectations with averages. Note that (47.48) also holds for the estimated values. Let $\varepsilon^{(i)}_m(k) = \hat{\boldsymbol{\Phi}}^{(i)}_{u_m z_1}(k)-\boldsymbol{\Phi}_{u_m z_1}(k)$ denote the estimation error of the cross-PSD between z_1 and u_m in the i-th frame. We then obtain,

$$\hat{\boldsymbol{\Phi}}^{(i)}_{z_m z_1}(k) = \tilde{A}_m(k)\hat{\boldsymbol{\Phi}}^{(i)}_{z_1 z_1}(k)+\boldsymbol{\Phi}_{u_m z_1}(k) +\varepsilon^{(i)}_m(k), \qquad i=1,\ldots,I\,. \tag{47.51}$$

If the noise reference signals $U_m(k,\ell)$, $m=2,\ldots,M$ were uncorrelated with $Z_1(k,\ell)$, then the standard system identification estimate, $\tilde{A}_m(k)=\hat{\boldsymbol{\Phi}}_{z_m z_1}(k)/\hat{\boldsymbol{\Phi}}_{z_1 z_1}(k)$, could be used to obtain an unbiased estimate of $\tilde{A}_m(k)$. Unfortunately, by (47.49) and (47.50), $U_m(k,\ell)$ and $Z_1(k,\ell)$ are in general correlated. Hence in [47.63] it is

Amazon EX1010 Page 957

proposed to obtain an unbiased estimate of $\tilde{A}_m(k)$ by applying the least-squares (LS) procedure to the following set of overdetermined equations

$$\begin{aligned}\boldsymbol{x} &\triangleq \begin{pmatrix} \hat{\boldsymbol{\Phi}}^{(1)}_{z_m z_1}(k) \\ \hat{\boldsymbol{\Phi}}^{(2)}_{z_m z_1}(k) \\ \vdots \\ \hat{\boldsymbol{\Phi}}^{(K)}_{z_m z_1}(k) \end{pmatrix} \\ &= \begin{pmatrix} \hat{\boldsymbol{\Phi}}^{(1)}_{z_1 z_1}(k) & 1 \\ \hat{\boldsymbol{\Phi}}^{(2)}_{z_1 z_1}(k) & 1 \\ \vdots & \\ \hat{\boldsymbol{\Phi}}^{(K)}_{z_1 z_1}(k) & 1 \end{pmatrix} \begin{pmatrix} \tilde{A}_m(k) \\ \boldsymbol{\Phi}_{u_m z_1}(k) \end{pmatrix} + \begin{pmatrix} \varepsilon^{(1)}_m(k) \\ \varepsilon^{(2)}_m(k) \\ \vdots \\ \varepsilon^{(K)}_m(k) \end{pmatrix} \\ &\triangleq \boldsymbol{G}\,\boldsymbol{\theta} + \boldsymbol{\epsilon}\,, \end{aligned} \tag{47.52}$$

where a separate set of equations is used for each $m = 2, \ldots, M$. The weighted least-squares (WLS) estimate of $\boldsymbol{\theta}$ is obtained by

$$\begin{aligned}\begin{pmatrix} \hat{\tilde{A}}_m(k) \\ \hat{\boldsymbol{\Phi}}_{u_m z_1}(k) \end{pmatrix} &= \hat{\boldsymbol{\theta}} \\ &= \arg\min_{\boldsymbol{\theta}} (\boldsymbol{x} - \boldsymbol{G}\,\boldsymbol{\theta})^{\mathrm{H}} \boldsymbol{W} (\boldsymbol{x} - \boldsymbol{G}\,\boldsymbol{\theta}) \\ &= (\boldsymbol{G}^{\mathrm{H}} \boldsymbol{W} \boldsymbol{G})^{-1} \boldsymbol{G}^{\mathrm{H}} \boldsymbol{W} \boldsymbol{x}\,, \end{aligned} \tag{47.53}$$

where $\boldsymbol{W}$ is a positive Hermitian weighting matrix, and $\boldsymbol{G}^{\mathrm{H}}\boldsymbol{W}\boldsymbol{G}$ is required to be invertible.

Shalvi and Weinstein suggested two alternative weighting matrices. One alternative is given by

$$W_{i\,j} = \begin{cases} T_i\,, & i = j \\ 0\,, & i \neq j \end{cases}\,, \tag{47.54}$$

where T_i is the length of subinterval i, so that longer intervals have higher weights. In this case, (47.53) reduces to

$$\hat{\tilde{A}}(k) = \frac{\langle \hat{\phi}_{z_m z_1}(k)\, \hat{\phi}_{z_1 z_1}(k) \rangle - \langle \hat{\phi}_{z_m z_1}(k) \rangle \langle \hat{\phi}_{z_1 z_1}(k) \rangle}{\langle \hat{\phi}^2_{z_1 z_1}(k) \rangle - \langle \hat{\phi}_{z_1 z_1}(k) \rangle^2} \tag{47.55}$$

with the average operation defined by

$$\langle \varphi(k) \rangle \triangleq \frac{\sum_{i=1}^{I} T_i \varphi^{(i)}(k)}{\sum_{i=1}^{I} T_i}\,. \tag{47.56}$$

Another alternative for $\boldsymbol{W}$, that minimizes the covariance of $\hat{\boldsymbol{\theta}}$, is given by

$$W_{i\,j} = \frac{B}{\hat{\phi}_{u_m u_m}(k)} \begin{cases} T_i / \hat{\phi}^{(i)}_{z_1 z_1}(k)\,, & i = j \\ 0\,, & i \neq j \end{cases}\,, \tag{47.57}$$

where B is related to the bandwidth of the window used for the cross-PSD estimation [47.63]. With this choice of the weighting function, (47.53) yields

$$\begin{aligned}\hat{\tilde{A}}(k) = &\frac{\langle 1/\hat{\phi}_{z_1 z_1}(k) \rangle \langle \hat{\phi}_{z_m z_1}(k) \rangle}{\langle \hat{\phi}_{z_1 z_1}(k) \rangle \langle 1/\hat{\phi}_{z_1 z_1}(k) \rangle - 1} \\ &- \frac{\langle \hat{\phi}_{z_m z_1}(k)\,/\,\hat{\phi}_{z_1 z_1}(k) \rangle}{\langle \hat{\phi}_{z_1 z_1}(k) \rangle \langle 1/\hat{\phi}_{z_1 z_1}(k) \rangle - 1} \end{aligned} \tag{47.58}$$

and the variance of $\hat{\tilde{A}}(k)$ is given by

$$\mathrm{var}\{\hat{\tilde{A}}(k)\} = \frac{1}{BT} \frac{\phi_{u_m u_m}(k) \langle 1/\phi_{z_1 z_1}(k) \rangle}{\langle \phi_{z_1 z_1}(k) \rangle \langle 1/\phi_{z_1 z_1}(k) \rangle - 1}\,, \tag{47.59}$$

where $T \triangleq \sum_{i=1}^{I} T_i$ is the total observation interval. Special attention should be given to choosing the frame length. On the one hand, it should be longer than the correlation length of $z_m(t)$, which must be longer than the length of the filter $a_m(t)$. On the other hand, it should be short enough for the filter time invariance and the noise quasistationarity assumptions to hold.

A major limitation of the WLS optimization in (47.53) is that both the identification of $\tilde{\boldsymbol{A}}(k)$ and the estimation of the cross-PSD $\phi_{u_m z_1}(k)$ are carried out using the same weight matrix $\boldsymbol{W}$. That is, each subinterval i is given the same weight, whether we are trying to find an estimate for $\tilde{\boldsymbol{A}}(k)$ or for $\phi_{u_m z_1}(k)$. However, subintervals with higher SNR values are of greater importance when estimating $\tilde{A}(k)$, whereas the opposite is true when estimating $\phi_{u_m z_1}(k)$. Consequently, the optimization criterion in (47.53) consists of two conflicting requirements: one is minimizing the error variance of $\tilde{\boldsymbol{A}}(k)$, which pulls the weight up to higher values on higher SNR subintervals. The other requirement is minimizing the error variance of $\phi_{u_m z_1}(k)$, which rather implies *smaller* weights on higher SNR subintervals. For instance, suppose we obtain observations on a relatively long low-SNR interval of length T_0, and on a relatively short high-SNR interval of length T_1 ($T_1 \ll T_0$). Then, the variance of $\tilde{\boldsymbol{A}}(k)$ in (47.59) is inversely proportional to the relative length of the high-SNR interval, $T_1/(T_0 + T_1)$. That is, including in the observation interval additional segments that do not contain speech (i. e., increasing T_0) increases the variance of $\tilde{\boldsymbol{A}}(k)$. This unnatural

consequence is a result of the desire to minimize the variance of $\phi_{u_m z_1}(k)$ by using larger weights on the segments that do not contain speech, while increasing the weights on such subintervals degrades the estimate for $\tilde{A}(k)$.

Another major limitation of RTF identification using nonstationarity is that the interfering signals are required to be stationary during the entire observation interval. The observation interval should include a certain number of subintervals that contain the desired signal, such that $\phi_{z_1 z_1}(k)$ is sufficiently nonstationary for all k. Unfortunately, if the desired signal is speech, the presence of the desired signal in the observed signals may be sparse in some frequency bands. This entails a very long observation interval, thus constraining the interfering signals to be stationary over long intervals. Furthermore, the RTF $\tilde{\boldsymbol{A}}(k)$ is assumed to be constant during the observation interval. Hence, very long observation intervals also restrict the capability of the system identification technique to track varying $\tilde{\boldsymbol{A}}(k)$ (e.g., tracking moving talkers in reverberant environments).

Using Speech Presence Probability

In this section, we present a system identification approach that is adapted to speech signals. Specifically, the presence of the desired speech signal in the time–frequency domain is uncertain, and the speech presence probability is utilized to separate the tasks of system identification and cross-PSD estimation. An estimate for $\tilde{\boldsymbol{A}}(k)$ is derived based on subintervals that contain speech, while subintervals that do not contain speech are of more significance when estimating the components of $\phi_{u_m z_1}(k)$.

Let the observed signals be divided in time into overlapping frames by the application of a window function and analyzed using the STFT. Under the same considerations leading to the estimation procedure based on speech nonstationarity (and based on the assumption that the RTFs can be modelled by short filters), (47.48) is still valid. Now using (47.48)-(47.50) and the fact that the desired signal $s(t)$ is uncorrelated with the interfering signals $n_m^{\mathrm{s}}(t)$; $m = 1, 2, \ldots, M$, we have

$$\begin{aligned}\phi_{z_m z_1}(k,\ell) &= \tilde{A}_m(k)|A_1|^2(k)\phi_{\mathrm{s\,s}}(k,\ell) \\ &\quad + \phi_{n_m^{\mathrm{s}} n_1^{\mathrm{s}}}(k,\ell)\,. \end{aligned} \tag{47.60}$$

Writing this equation in terms of the PSD estimates, we obtain for $m = 2, 3, \ldots, M$

$$\begin{aligned}\hat{\phi}_{z_m z_1}(k,\ell) &= \tilde{A}_m(k)|\hat{A}_1|^2(k)\hat{\phi}_{\mathrm{s\,s}}(k,\ell) \\ &\quad + \hat{\phi}_{n_m^{\mathrm{s}} n_1^{\mathrm{s}}}(k,\ell) + \varepsilon_m(k,\ell) \\ &= \tilde{A}_m(k)\hat{\phi}_{\tilde{\mathrm{s}}\,\tilde{\mathrm{s}}}(k,\ell) + \hat{\phi}_{n_m^{\mathrm{s}} n_1^{\mathrm{s}}}(k,\ell) + \varepsilon_m(k,\ell)\,, \end{aligned} \tag{47.61}$$

where $\varepsilon_m(k,\ell)$ denotes an estimation error and $\hat{\phi}_{\tilde{\mathrm{s}}\,\tilde{\mathrm{s}}}(k,\ell) = |\hat{A}_1|^2(k)\hat{\phi}_{\mathrm{s\,s}}(k,\ell)$ represents the PSD of the speech signal component in microphone 1. This gives us L equations, which may be written in a matrix form as

$$\begin{aligned}\hat{\boldsymbol{\psi}}_m(k) &\triangleq \begin{pmatrix} \hat{\phi}_{z_m z_1}(k,1) - \hat{\phi}_{n_m^{\mathrm{s}} n_1^{\mathrm{s}}}(k,1) \\ \hat{\phi}_{z_m z_1}(k,2) - \hat{\phi}_{n_m^{\mathrm{s}} n_1^{\mathrm{s}}}(k,2) \\ \vdots \\ \hat{\phi}_{z_m z_1}(k,L) - \hat{\phi}_{n_m^{\mathrm{s}} n_1^{\mathrm{s}}}(k,L) \end{pmatrix} \\ &= \begin{pmatrix} \hat{\phi}_{\tilde{\mathrm{s}}\,\tilde{\mathrm{s}}}(k,1) \\ \hat{\phi}_{\tilde{\mathrm{s}}\,\tilde{\mathrm{s}}}(k,2) \\ \vdots \\ \hat{\phi}_{\tilde{\mathrm{s}}\,\tilde{\mathrm{s}}}(k,L) \end{pmatrix} \tilde{A}_m(k) + \begin{pmatrix} \varepsilon_m(k,1) \\ \varepsilon_m(k,2) \\ \vdots \\ \varepsilon_m(k,L) \end{pmatrix} \\ &\triangleq \hat{\boldsymbol{\phi}}_{\tilde{\mathrm{s}}\,\tilde{\mathrm{s}}}(k)\,\tilde{A}_m(k) + \boldsymbol{\varepsilon}_m(k)\,. \end{aligned} \tag{47.62}$$

Since the RTF $\tilde{\boldsymbol{A}}_m(k)$ represents the coupling between the primary and reference sensors with respect to the *desired* source signal, the optimization criterion for the identification of $\tilde{A}_m(k)$ has to take into account only short-time frames which contain desired signal components. Specifically, let $I(k,\ell)$ denote an indicator function for the signal presence [$I(k,\ell) = 1$ if $\phi_{\tilde{\mathrm{s}}\,\tilde{\mathrm{s}}}(k,\ell) \neq 0$, i. e., during H_1, and $I(k,\ell) = 0$ otherwise], and let $\mathbf{I}(k)$ represent a diagonal matrix with the elements $[I(k,1), I(k,2), \ldots, I(k,L)]$ on its diagonal. Then the WLS estimate of $\tilde{A}(k)$ is obtained by

$$\begin{aligned}\hat{\tilde{A}}_m &= \arg\min_{\tilde{A}_m} \left\{ [\mathbf{I}\boldsymbol{\varepsilon}_m]^{\mathrm{H}} \mathbf{W} [\mathbf{I}\boldsymbol{\varepsilon}_m] \right\} \\ &= \arg\min_{\tilde{A}_m} \Big\{ [\hat{\boldsymbol{\psi}}_m - \hat{\boldsymbol{\phi}}_{\tilde{\mathrm{s}}\tilde{\mathrm{s}}}\tilde{A}_m]^{\mathrm{H}} \\ &\quad \mathbf{I}\mathbf{W}\mathbf{I}[\hat{\boldsymbol{\psi}}_m - \hat{\boldsymbol{\phi}}_{\tilde{\mathrm{s}}\tilde{\mathrm{s}}}\tilde{A}_m] \Big\} \\ &= [\hat{\boldsymbol{\phi}}_{\tilde{\mathrm{s}}\tilde{\mathrm{s}}}^{\mathrm{T}} \mathbf{I}\mathbf{W}\mathbf{I}\hat{\boldsymbol{\phi}}_{\tilde{\mathrm{s}}\tilde{\mathrm{s}}}]^{-1} \hat{\boldsymbol{\phi}}_{\tilde{\mathrm{s}}\tilde{\mathrm{s}}}^{\mathrm{T}} \mathbf{I}\mathbf{W}\mathbf{I}\hat{\boldsymbol{\psi}}_m\,, \end{aligned} \tag{47.63}$$

where the argument k has been omitted for notational simplicity. Recognizing the product $\mathbf{I}\,\mathbf{W}\,\mathbf{I}$ as the equivalent weight matrix, the variance of $\hat{\tilde{A}}$ is given by ([47.64] p. 405)

$$\begin{aligned}\mathrm{var}\{\hat{\tilde{A}}_m\} &= \left(\boldsymbol{\phi}_{\tilde{\mathrm{s}}\,\tilde{\mathrm{s}}}^{\mathrm{T}} \mathbf{I}\mathbf{W}\mathbf{I}\boldsymbol{\phi}_{\tilde{\mathrm{s}}\,\tilde{\mathrm{s}}}\right)^{-1} \boldsymbol{\phi}_{\tilde{\mathrm{s}}\,\tilde{\mathrm{s}}}^{\mathrm{T}} \mathbf{I}\mathbf{W}\mathbf{I}\,\mathrm{cov}(\boldsymbol{\varepsilon}_m) \\ &\quad \times \mathbf{I}\mathbf{W}\mathbf{I}\boldsymbol{\phi}_{\tilde{\mathrm{s}}\,\tilde{\mathrm{s}}} \left(\boldsymbol{\phi}_{\tilde{\mathrm{s}}\,\tilde{\mathrm{s}}}^{\mathrm{T}} \mathbf{I}\mathbf{W}\mathbf{I}\boldsymbol{\phi}_{\tilde{\mathrm{s}}\,\tilde{\mathrm{s}}}\right)^{-1}\,, \end{aligned} \tag{47.64}$$

where $\mathrm{cov}(\boldsymbol{\varepsilon}_m)$ is the covariance matrix of $\boldsymbol{\varepsilon}_m$. The matrix $\mathbf{W}$ that minimizes the variance of $\hat{\tilde{A}}$ therefore satisfies ([47.64] prop. 8.2.4)

$$\mathbf{I}\mathbf{W}\mathbf{I} = \mathbf{I}\,[\mathrm{cov}(\boldsymbol{\varepsilon}_m)]^{-1}\,\mathbf{I}\,. \tag{47.65}$$

Amazon EX1010 Page 959

This choice of $\mathbf{W}$ yields an asymptotically unbiased estimator

$$\hat{A} = \left\{\hat{\boldsymbol{\phi}}_{\tilde{s}\tilde{s}}{}^{\mathrm{T}}\mathbf{I}[\mathrm{cov}(\boldsymbol{\varepsilon}_m)]^{-1}\mathbf{I}\hat{\boldsymbol{\phi}}_{\tilde{s}\tilde{s}}\right\}^{-1} \times \hat{\boldsymbol{\phi}}_{\tilde{s}\tilde{s}}{}^{\mathrm{T}}\mathbf{I}[\mathrm{cov}(\boldsymbol{\varepsilon}_m)]^{-1}\mathbf{I}\hat{\boldsymbol{\psi}}_m \,, \tag{47.66}$$

which is known as the *minimum variance* or Gauss–Markov estimator. Substituting (47.65) into (47.64), we obtain the variance of the resulting estimator

$$\mathrm{var}\{\hat{A}\} = \left\{\boldsymbol{\phi}_{\tilde{s}\tilde{s}}{}^{\mathrm{T}}\mathbf{I}[\mathrm{cov}(\boldsymbol{\varepsilon}_m)]^{-1}\mathbf{I}\,\boldsymbol{\phi}_{\tilde{s}\tilde{s}}\right\}^{-1} . \tag{47.67}$$

The elements of $\mathrm{cov}(\boldsymbol{\varepsilon}_m)$ are asymptotically given by (see [47.61])

$$\mathrm{cov}[\varepsilon_m(k,\ell), \varepsilon_m(k,\ell')] = \begin{cases} \phi_{\tilde{s}\tilde{s}}(k,\ell)\,\phi_{u_m u_m}(k,\ell) \,, & \text{if } \ell = \ell' \,, \\ 0 \,, & \text{otherwise.} \end{cases} \tag{47.68}$$

Under the assumption that hypothesis $H_{0\mathrm{t}}$ is false, i. e., the noise is stationary, $\phi_{u_m u_m}(k,\ell)$ is independent of the frame index ℓ (in practice, it suffices that the statistics of the interfering signals is slowly changing compared with the statistics of the desired signal). Denoting by $\langle\cdot\rangle_\ell$ an average operation over the frame index ℓ

$$\langle\varphi(k,\ell)\rangle_\ell \triangleq \frac{1}{L}\sum_{\ell=1}^{L}\varphi(k,\ell) \,, \tag{47.69}$$

and substituting (47.68) into (47.66) and (47.67) we obtain

$$\hat{\tilde{A}}_m(k) = \frac{\langle I(k,\ell)[\hat{\phi}_{z_m z_1}(k,\ell) - \hat{\phi}_{n_m^{\mathrm{s}} n_1^{\mathrm{s}}}(k,\ell)]\rangle_\ell}{\langle I(k,\ell)\,\hat{\phi}_{\tilde{s}\tilde{s}}(k,\ell)\rangle_\ell} \,, \tag{47.70}$$

$$\mathrm{var}\{\hat{\tilde{A}}_m(k)\}\frac{\phi_{u_m u_m}(k)}{L\,\langle I(k,\ell)\,\phi_{\tilde{s}\tilde{s}}(k,\ell)\rangle_\ell} \,. \tag{47.71}$$

Note that, for a given frequency-bin index k, only frames that contain speech $[I(k,\ell) \neq 0]$ influence the values of $\hat{\tilde{A}}_m(k)$ and $\mathrm{var}\{\hat{\tilde{A}}_m(k)\}$. In contrast to the nonstationarity method, including in the observation interval additional segments that do not contain speech does not increase the variance of $\hat{\tilde{A}}_m(k)$ for any k. However, the proposed identification approach requires an estimate for $I(k,\ell)$, i. e. identifying which time–frequency bins (k,ℓ) contain the desired signal. In practice, the speech presence probability $p(k,\ell)$ can be estimated from the beamformer output (see Sect. 47.6), and an estimate for the indicator function is obtained by

$$\hat{I}(k,\ell) = \begin{cases} 1 \,, & \text{if } p(k,\ell) \geq p_0 \,, \\ 0 \,, & \text{otherwise,} \end{cases} \tag{47.72}$$

where p_0 $(0 \leq p_0 < 1)$ is a predetermined threshold. Note, also that the output of the beamformer consists of the filtered version of the desired speech $\tilde{s}(t)$, when the RTFs is used in the FBF branch. The parameter p_0 controls the trade-off between the detection and false alarm probabilities, which are defined by $P_{\mathrm{D}} \triangleq \mathcal{P}\{p(k,\ell) \geq p_0 \,|\, I(k,\ell) = 1\}$ and $P_{\mathrm{FA}} \triangleq \mathcal{P}\{p(k,\ell) \geq p_0 \,|\, I(k,\ell) = 0\}$. A smaller value of p_0 increases the detection probability and allows for more short-time frames to be involved in the estimation of $\tilde{\boldsymbol{A}}(k)$. However, a smaller value of p_0 also increases the false alarm probability, which may cause a mismodification of $\hat{\tilde{\boldsymbol{A}}}(k)$ due to frames that do not contain desired speech components.

The identification algorithm based on speech presence probability requires estimates for $\phi_{z_m z_1}(k,\ell)$, $\phi_{\tilde{s}\tilde{s}}(k,\ell)$, and $\phi_{n_m^{\mathrm{s}} n_1^{\mathrm{s}}}(k,\ell)$. An estimate for $\phi_{z_m z_1}(k,\ell)$ is obtained by applying a first-order recursive smoothing to the cross-periodogram of the observed signals, $Z_m(k,\ell)\,Z_1^*(k,\ell)$. Specifically,

$$\hat{\phi}_{z_m z_1}(k,\ell) = \alpha_{\mathrm{s}}\,\hat{\phi}_{z_m z_1}(k,\ell-1) + (1-\alpha_{\mathrm{s}})Z_m(k,\ell)\,Z_1^*(k,\ell) \,, \tag{47.73}$$

where the smoothing parameter α_{s} $(0 \leq \alpha_{\mathrm{s}} < 1)$ determines the equivalent number of cross-periodograms that are averaged, $N_\ell \approx (1+\alpha_{\mathrm{s}})/(1-\alpha_{\mathrm{s}})$. Typically, speech periodograms are recursively smoothed with an equivalent rectangular window of $T_{\mathrm{s}} = 0.2$ s long, which represents a good compromise between smoothing the noise and tracking the speech spectral variations [47.65]. Therefore, for a sampling rate of 8 kHz, an STFT window length of 256 samples and a frame update step of 128 samples, we use $\alpha_{\mathrm{s}} = (T_{\mathrm{s}} \cdot 8000/128 - 1)/(T_{\mathrm{s}} \cdot 8000/128 + 1) \approx 0.85$.

To obtain an estimate for the PSD of the desired signal, we can use the output of the multichannel postfilter discussed in Sect. 47.6, during speech presence, i. e., H_1 is true.

$$\hat{\phi}_{\tilde{s}\tilde{s}}(k,\ell) = \alpha_{\mathrm{s}}\,\hat{\phi}_{\tilde{s}\tilde{s}}(k,\ell-1) + (1-\alpha_{\mathrm{s}})I(k,\ell)\,|Y(k,\ell)|^2 \,. \tag{47.74}$$

The cross-PSD of the interfering signals, $n_m^{\mathrm{s}}(t)$ and $n_1^{\mathrm{s}}(t)$, is estimated by using the minima-controlled

recursive averaging (MCRA) approach [47.66, 67]. Specifically, past spectral cross-power values of the noisy observed signals are recursively averaged with a time-varying frequency-dependent smoothing parameter during periods for which H_{0s} is true

$$\hat{\phi}_{n_m^s n_1^s}(k,\ell) = \tilde{\alpha}_u(k,\ell)\,\hat{\phi}_{n_m^s n_1^s}(k,\ell-1) + \beta\,[1-\tilde{\alpha}_u(k,\ell)]\,Z_m(k,\ell)\,Z_1^*(k,\ell)\,, \tag{47.75}$$

where $\tilde{\alpha}_u(k,\ell)$ is the smoothing parameter ($0 < \tilde{\alpha}_u(k,\ell) \le 1$), and β ($\beta \ge 1$) is a factor that compensates the bias when the desired signal is absent [47.67]. The smoothing parameter is determined by the signal presence probability, $p(k,\ell)$, and a constant α_u ($0 < \alpha_u < 1$) that represents its minimal value

$$\tilde{\alpha}_u(k,\ell) = \alpha_u + (1-\alpha_u)p(k,\ell)\,. \tag{47.76}$$

The value of $\tilde{\alpha}_u$ is close to 1 when the desired signal is present to prevent the noise cross-PSD estimate from increasing as a result of signal components. It decreases linearly with the probability of signal presence to allow a faster update of the noise estimate. The value of α_u compromises between the tracking rate (response rate to abrupt changes in the noise statistics) and the variance of the noise estimate. Typically, in the case of high levels of nonstationary noise, a good compromise is obtained by $\alpha_u = 0.85$ [47.67]. Substituting these spectral estimates into (47.70) we obtain an estimate for $\tilde{A}_m(k)$.

47.5 Robustness and Distortion Weighting

Beamformers often suffer from sensitivity to signal mismatch. The GSC in particular suffers from two basic problems. First, nonideal FBF can lead to noncoherent filter-and-sum operation. *Doclo* and *Moonen* [47.27] and *Nordholm* et al. [47.68] use spatial and frequency-domain constraints to improve the robustness of beamformers. The second problem, which is the concern of this survey, is the leakage phenomenon, caused by imperfect BM. If the desired speech leaks into the noise reference signals $\boldsymbol{U}(k,\ell)$ the noise canceller filters will subtract speech components from the FBF output, causing self-cancellation of the desired speech, and hence a severe distortion. Note that, even when the ANC filters are adapted during noise-only periods, the self-cancellation is unavoidable. The goal of this section is to present several concepts for increasing the robustness of the GSC structure and reducing its sensitivity to signal mismatch.

Cox et al. [47.7] presented a thorough analysis of array sensitivity. The array output SNR is evidently given by

$$\mathrm{SNR}_{out}(k,\ell) = \frac{\phi_{ss}(k,\ell)\boldsymbol{W}^H(k)\boldsymbol{A}(k)\boldsymbol{A}^H(k)\boldsymbol{W}(k)}{\boldsymbol{W}^H(k)\boldsymbol{\Phi}_{\boldsymbol{NN}}(k,\ell)\boldsymbol{W}(k)}\,.$$

Now assume that the signal's ATFs are different from the ATFs used for designing the LCMV beamformer, i. e., $\breve{\boldsymbol{A}}(k) = \boldsymbol{A} + \boldsymbol{\epsilon}(k)$. Assume also that the spatial correlation matrix of the perturbation $\boldsymbol{\epsilon}(k)$ is given by $E\{\boldsymbol{\epsilon}(k)\boldsymbol{\epsilon}^H(k)\} = \sigma_\epsilon^2\mathbf{I}$, where $\mathbf{I}$ is the identity matrix of dimensions $M \times M$. Namely, we assume that the perturbation components are uncorrelated. Hence the expected output SNR is given by

$$E\{\mathrm{SNR}_{out}(k,\ell)\} = \frac{\phi_{ss}(k,\ell)\boldsymbol{W}^H(k)\big(\boldsymbol{A}(k)\boldsymbol{A}^H(k) + \sigma_\epsilon^2\mathbf{I}\big)\boldsymbol{W}(k)}{\boldsymbol{W}^H(k)\boldsymbol{\Phi}_{\boldsymbol{NN}}(k,\ell)\boldsymbol{W}(k)}\,. \tag{47.77}$$

Define the sensitivity of the array to the ATFs perturbation as $J(k,\ell)$

$$J(k,\ell) \triangleq \frac{\frac{\partial}{\partial\sigma_\epsilon^2}E\{\mathrm{SNR}_{out}(k,\ell)\}}{E\{\mathrm{SNR}_{out}(k,\ell)\}|_{\sigma_\epsilon^2=0}} = \frac{\boldsymbol{W}^H(k)\boldsymbol{W}(k)}{\boldsymbol{W}^H(k)\boldsymbol{A}(k)\boldsymbol{A}^H(k)\boldsymbol{W}(k)}\,. \tag{47.78}$$

The resulting expression is the reciprocal of the white-noise gain of the array. Using the array constraint $\boldsymbol{W}^H(k)\boldsymbol{A}(k) = 1$ we finally obtain the following expression for the sensitivity of the array to the ATFs perturbation,

$$J(k,\ell) = \boldsymbol{W}^H(k)\boldsymbol{W}(k)\,. \tag{47.79}$$

Specifically, the array sensitivity is equal to the norm of the beamformer weights. Hence, reducing the sensitivity of the array is equivalent to constraining the norm of the array filter coefficients. Due to (47.15) the array filters can be decomposed into two orthogonal filters, $\boldsymbol{W}(k) = \boldsymbol{W}_0(k) - \boldsymbol{V}(k) = \boldsymbol{W}_0(k) - \mathcal{H}(k)\boldsymbol{G}(k,\ell)$. It is therefore sufficient to constrain the adaptive filter norm, namely $\boldsymbol{G}^H(k,\ell)\boldsymbol{G}(k,\ell) = \|\boldsymbol{G}(k,\ell)\|^2 \le \Omega(k,\ell)$, where $\Omega(k,\ell)$ is a prespecified norm. The GSC structure is

modified to fulfil the norm constraint, as follows:

$$\tilde{\boldsymbol{G}}'(k) = \boldsymbol{G}(k,\ell) + \mu \frac{\boldsymbol{U}(k,\ell)Y^*(k,\ell)}{P_{\text{est}}(k,\ell)}\,, \tag{47.80}$$

$$\tilde{\boldsymbol{G}}(\ell+1,k) = \begin{cases} \tilde{\boldsymbol{G}}'(k) & \|\tilde{\boldsymbol{G}}'(k)\|^2 \le \Omega(k,\ell+1) \\ \frac{\sqrt{\Omega(k,\ell+1)}}{\|\tilde{\boldsymbol{G}}'(k)\|}\tilde{\boldsymbol{G}}'(k) & \text{otherwise.} \end{cases} \tag{47.81}$$

Finally, the conventional FIR constraint is imposed on the norm-constrained filters

$$\boldsymbol{G}(\ell+1,k) \xleftarrow{\text{FIR}} \tilde{\boldsymbol{G}}(\ell+1,k)\,.$$

Based on this concept, *Hoshuyama* et al. [47.55, 69] proposed several methods for addressing the robustness issue, concentrating on the self-cancellation phenomenon, caused by the leakage of the desired speech signal to the BM outputs $\boldsymbol{U}(k,\ell)$. This phenomenon is emphasized in reverberant environments, in the case where the BM only compensates for the relative delay [as in (47.34)]. In general there are two ways to mitigate this leakage problem. First, an improved spatial filtering can be incorporated into the design of the BM. *Claesson* and *Nordholm* [47.22] proposed to apply spatial high-pass filter to cancel out all signals within a specified frequency and angular range. *Huarng* and *Yeh* [47.45] analyzed the leakage phenomenon and applied a derivative constraint on the array response, yielding wider tolerance to pointing errors.

A second cure for the leakage problems involves applying constraints on the ANC filters. *Hoshuyama* et al. [47.55] proposed several structures combining modifications for both the BM and the ANC blocks. The conventional delay-compensation BM is replaced by an adaptive BM based on signal cancellers. Two constraining strategies may be applied to the involved filters. The first strategy uses norm-constraint, and the second uses the leaky LMS adaptation scheme. *Haykin* [47.1] proved that both strategies are equivalent. The modified BM outputs, for $m = 1, \ldots, M$, are given by

$$U_m(k,\ell) = Z_m(k,\ell) - H_m^*(k,\ell)Y_{\text{FBF}}(k,\ell)\,, \tag{47.82}$$

where $H_m(k,\ell)$ are updated as to minimize the power of $U_m(k,\ell)$, by cancelling all desired speech components. Whenever, the SNR in $Y_{\text{FBF}}(k,\ell)$ is sufficiently high, the blocking ability of the structure is improved.

Gannot [47.57] showed that this equation, in conjunction with the expression for the beamformer output

$$Y(k,\ell) = Y_{\text{FBF}}(k,\ell) - \boldsymbol{G}^{\text{H}}(k,\ell)\boldsymbol{U}(k,\ell)\,,$$

is closely related to the decorrelation criterion proposed by *Weinstein* et al. [47.56]. A different decorrelation based structure was later proposed by *Fancourt* and *Parra* [47.70].

Two alternative schemes are proposed for adapting the filters $H_m(k,\ell)$. (Originally, *Hoshuyama* et al. [47.55] stated their formulation in the time domain using the original GSC structure. Here we state the frequency-domain counterpart of the proposed algorithm. The first to propose frequency domain implementation of Hoshuyama's concepts were *Herbordt* and *Kellermann* [47.71, 72].) The first scheme is the leaky LMS,

$$H_m(k,\ell+1) = (1-\delta)H_m(k,\ell) + \frac{\mu_{\text{h}}}{|Y_{\text{FBF}}(k,\ell)|^2}U_m(k,\ell)Y_{\text{FBF}}^*(k,\ell) \tag{47.83}$$

for $m = 1, \ldots, M$. The regular FIR constraint, omitted for the clarity of the exposition, is then applied. The second scheme constrains the filter coefficients to a predefined mask, yielding for $m = 1, \ldots, M$:

$$H'_m(k,\ell+1) = H_m(k,\ell) + \frac{\mu_{\text{h}}}{|Y_{\text{FBF}}(k,\ell)|^2}U_m(k,\ell)Y_{\text{FBF}}^*(k,\ell) \tag{47.84}$$

and

$$H(\ell+1,k) = \begin{cases} \phi_{\text{low}}(k,\ell+1) & H'_m(k,\ell+1) \ge \phi_{\text{low}}(k,\ell+1)\,, \\ \phi_{\text{up}}(k,\ell+1) & H'_m(k,\ell+1) \le \phi_{\text{up}}(k,\ell+1)\,, \\ H'_m(k,\ell+1) & \text{otherwise.} \end{cases} \tag{47.85}$$

The ANC filter is either adapted by the leaky LMS algorithm or the norm-constrained adaptation mechanism proposed by *Cox* (see (47.81)). As a concluding remark summarizing Hoshuyama's methods, we draw the reader attention to the resemblance of the proposed adaptation of the BM filters and the subspace tracking procedure presented by *Affes* and *Grenier* depicted in (47.46).

Spriet et al. [47.33] adopted a different approach to mitigating the leakage problem, by modifying the adaptation criterion for the ANC filters, $\boldsymbol{G}(k,\ell)$. The minimization criterion in (47.25) is altered to deal with the leakage problem. Let, $Y_{\text{FBF}}^{\text{s}}(k,\ell)$ be the speech component at the FBF output. Let $\boldsymbol{U}^{\text{s}}(k,\ell)$ and $\boldsymbol{U}^{\text{n}}(k,\ell)$ be the speech and noise (without distinction between stationary and transient noise signals) components in the

reference signals, respectively. Then, the filters $\boldsymbol{G}(k,\ell)$ minimize the following expression

$$E\{\|Y^{\mathrm{s}}_{\mathrm{FBF}}(k,\ell)-\boldsymbol{G}^{\mathrm{H}}(k,\ell)(\boldsymbol{U}^{\mathrm{s}}(k,\ell)+\boldsymbol{U}^{\mathrm{n}}(k,\ell))\|^2\}\,.$$

Since the speech and noise signals are uncorrelated, the above expression can be restated as

$$E\{\|\boldsymbol{G}^{\mathrm{H}}(k,\ell)\boldsymbol{U}^{\mathrm{n}}(k,\ell)\|^2\} + E\{\|Y^{\mathrm{s}}_{\mathrm{FBF}}(k,\ell)-\boldsymbol{G}^{\mathrm{H}}(k,\ell)\boldsymbol{U}^{\mathrm{s}}(k,\ell)\|^2\}\,. \quad (47.86)$$

Note, that the first term is related to the noise signal and the second term to the speech distortion. Hence, the Wiener filter design criterion can be easily generalized [47.26] to allow for a trade-off between speech distortion and NR, by incorporating a weighting factor $\mu\in[0,\infty)$. The resulting criterion is then given by

$$\mu E\{\|\boldsymbol{G}^{\mathrm{H}}(k,\ell)\boldsymbol{U}^{\mathrm{n}}(k,\ell)\|^2\} + E\{\|Y^{\mathrm{s}}_{\mathrm{FBF}}(k,\ell)-\boldsymbol{G}^{\mathrm{H}}(k,\ell)\boldsymbol{U}^{\mathrm{s}}(k,\ell)\|^2\}\,. \quad (47.87)$$

It is easily verified that the corresponding minimizer is

$$\boldsymbol{G}(k,\ell)=\left(\frac{1}{\mu}\boldsymbol{\Phi}_{\boldsymbol{U}^{\mathrm{s}}\boldsymbol{U}^{\mathrm{s}}}+\boldsymbol{\Phi}_{\boldsymbol{U}^{\mathrm{n}}\boldsymbol{U}^{\mathrm{n}}}\right)^{-1}\boldsymbol{\phi}_{\boldsymbol{U}Y^{\mathrm{s}}_{\mathrm{FBF}}}\,, \quad (47.88)$$

where

$$\boldsymbol{\Phi}_{\boldsymbol{U}^{\mathrm{s}}\boldsymbol{U}^{\mathrm{s}}}=E\{\boldsymbol{U}^{\mathrm{s}}(k,\ell)(\boldsymbol{U}^{\mathrm{s}}(k,\ell))^{\mathrm{H}}\}\,,$$
$$\boldsymbol{\Phi}_{\boldsymbol{U}^{\mathrm{n}}\boldsymbol{U}^{\mathrm{n}}}=E\{\boldsymbol{U}^{\mathrm{n}}(k,\ell)(\boldsymbol{U}^{\mathrm{n}}(k,\ell))^{\mathrm{H}}\}\,,$$
$$\boldsymbol{\phi}_{\boldsymbol{U}Y^{\mathrm{s}}_{\mathrm{FBF}}}=E\{\boldsymbol{U}^{\mathrm{s}}(k,\ell)(Y^{\mathrm{s}}_{\mathrm{FBF}}(k,\ell))^{*}\}\,.$$

Using the reference signals definitions we have

$$\boldsymbol{\Phi}_{\boldsymbol{U}^{\mathrm{s}}\boldsymbol{U}^{\mathrm{s}}}=\mathcal{H}^{\mathrm{H}}(k,\ell)\boldsymbol{\Phi}_{\boldsymbol{SS}}(k,\ell)\mathcal{H}(k,\ell)\,,$$
$$\boldsymbol{\Phi}_{\boldsymbol{U}^{\mathrm{n}}\boldsymbol{U}^{\mathrm{n}}}=\mathcal{H}^{\mathrm{H}}(k,\ell)\boldsymbol{\Phi}_{\boldsymbol{NN}}(k,\ell)\mathcal{H}(k,\ell)\,,$$
$$\boldsymbol{\phi}_{\boldsymbol{U}^{\mathrm{s}}Y^{\mathrm{s}}_{\mathrm{FBF}}}=\mathcal{H}^{\mathrm{H}}(k,\ell)\boldsymbol{\Phi}_{\boldsymbol{SS}}(k,\ell)\boldsymbol{W}_0(k,\ell)\,. \quad (47.89)$$

Since $\boldsymbol{\Phi}_{\boldsymbol{SS}}(k,\ell)$ is not available it can be evaluated using

$$\boldsymbol{\Phi}_{\boldsymbol{SS}}(k,\ell)=\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)-\boldsymbol{\Phi}_{\boldsymbol{NN}}(k,\ell)$$

and $\boldsymbol{\Phi}_{\boldsymbol{NN}}(k,\ell)$ is estimated while the speech signal is absent. *Doclo* and *Moonen* [47.73] prove that the output SNR after NR with the above speech distortion weighted multichannel Wiener filter (SDW-MWF) is always larger than or equal to the input SNR, for any filter length, and for any value of the trade-off parameter μ between NR and speech distortion.

This solution for the ANC filters constitutes the speech distortion regularized generalized sidelobe canceller (SDR-GSC) structure. *Spriet* et al. [47.33] further proposed to incorporate a single-channel postfilter, which compensates for the distortion imposed by the structure in case of speech leakage into the reference signals. Further discussion of this structure is beyond the scope of this survey. In [47.74], the authors propose a stochastic gradient-based implementation of their criterion. The robustness of both the multichannel Wiener filter and the GSC structures are analyzed by *Spriet* et al. [47.75] in the context of hearing-aid application.

Improving the robustness of the BM is an ongoing research topic. An interesting direction was taken by *Low* et al. [47.76]. The authors propose to incorporate concepts adopted from the BSS discipline to improve the separation of the speech and noise signals and hence reducing the amount of leakage of the desired signal into the reference noise signals.

47.6 Multichannel Postfiltering

Postfiltering methods for multimicrophone speech enhancement algorithms have recently attracted an increased interest. It is well known that beamforming methods yield a significant improvement in speech quality [47.9]. However, when the noise field is spatially incoherent or diffuse, the NR is insufficient [47.77] and additional postfiltering is normally required [47.78]. Furthermore, as nonstationary noise cannot, in general, be distinguished from speech signals, a significant performance degradation is expected in nonstationary noise environment.

Most multimicrophone speech enhancement methods consist of a multichannel part (either delay and sum beamformer or GSC [47.36]) followed by a postfilter, which is based on Wiener filtering (sometimes in conjunction with spectral subtraction). Numerous articles have been published on the subject, e.g., [47.79–87] to mention just a few.

In general, the postfilters can be divided into two groups. The first is a single-channel postfilter which acts as a single-microphone speech enhancement algorithm on the beamformer output. Multichannel postfilters, on the other hand, explicitly use the spatial information, extracted by the GSC structure, to gain better distinction between the speech signal and the transient noise.

47.6.1 MMSE Postfiltering

Simmer et al. [47.78] address the general problem of the single-channel postfilter. They first derive the multichannel Wiener filter for estimating the speech signal, $S(k,\ell)$ from the microphone signals $\boldsymbol{Z}(k,\ell)$ given in (47.3). Then, they show that the Wiener filter can be factorized into a multiplication of the LCMV (Frost) beamformer given in (47.7) and a single-channel Wiener filter that depends on the output speech and noise signals.

To show this, we will start with the multichannel Wiener filter, which is given by

$$\boldsymbol{W}^{\text{Wiener}}(k,\ell) = \boldsymbol{\Phi}_{\boldsymbol{ZZ}}^{-1}(k,\ell)\boldsymbol{\Phi}_{\boldsymbol{ZS}}(k,\ell)\,, \tag{47.90}$$

where $\boldsymbol{\Phi}_{\boldsymbol{ZZ}}$ is given in (47.10) and

$$\begin{aligned}\boldsymbol{\Phi}_{\boldsymbol{ZS}}(k,\ell) &= E\{\boldsymbol{Z}(k,\ell)S^*(k,\ell)\}\\ &= \boldsymbol{A}(k)\phi_{\text{ss}}(k,\ell)\,.\end{aligned} \tag{47.91}$$

Hence,

$$\begin{aligned}&\boldsymbol{W}^{\text{Wiener}}(k,\ell)\\ &= \left(\phi_{\text{ss}}(k,\ell)\boldsymbol{A}(k)\boldsymbol{A}^{\text{H}}(k) + \boldsymbol{\Phi}_{\boldsymbol{NN}}(k,\ell)\right)^{-1}\\ &\quad\times\phi_{\text{ss}}(k,\ell)\boldsymbol{A}(k)\,.\end{aligned} \tag{47.92}$$

Omitting, for the clarity of the exposition, the explicit time- and frequency-domain dependence and using the matrix inversion lemma yields

$$\begin{aligned}&\boldsymbol{W}^{\text{Wiener}}(k,\ell)\\ &= \left(\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1} - \frac{\phi_{\text{ss}}\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}\boldsymbol{A}\boldsymbol{A}^{\text{H}}\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}}{1+\phi_{\text{ss}}\boldsymbol{A}^{\text{H}}\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}\boldsymbol{A}}\right)\phi_{\text{ss}}\boldsymbol{A}\\ &= \left(1 - \frac{\phi_{\text{ss}}\boldsymbol{A}\boldsymbol{A}^{\text{H}}\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}}{1+\phi_{\text{ss}}\boldsymbol{A}^{\text{H}}\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}\boldsymbol{A}}\right)\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}\phi_{\text{ss}}\boldsymbol{A}\\ &= \left(\frac{\phi_{\text{ss}}}{1+\phi_{\text{ss}}\boldsymbol{A}^{\text{H}}\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}\boldsymbol{A}}\right)\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}\boldsymbol{A}\\ &= \left(\frac{\phi_{\text{ss}}}{\phi_{\text{ss}}+(\boldsymbol{A}^{\text{H}}\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}\boldsymbol{A})^{-1}}\right)\frac{\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}\boldsymbol{A}}{\boldsymbol{A}^{\text{H}}\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}\boldsymbol{A}}\,.\end{aligned} \tag{47.93}$$

(The derivation here is a slight modification of the results introduced in [47.78].) The reader can easily identify the second multiplier as the MSNR beamformer (47.8) which was shown to be equivalent to the LCMV beamformer (47.11).

We turn now to analyzing the first term in the multiplicative expression. The PSD of desired signal component at the output of the LCMV beamformer is given by

$$\begin{aligned}&\phi_{Y_{\text{s}}Y_{\text{s}}}(k,\ell)\\ &= \phi_{\text{ss}}(\boldsymbol{W}^{\text{LCMV}})^{\text{H}}\boldsymbol{A}\boldsymbol{A}^{\text{H}}\boldsymbol{W}^{\text{LCMV}}\\ &= \phi_{\text{ss}}\left(\frac{\boldsymbol{A}^{\text{H}}\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}\boldsymbol{A}}{\boldsymbol{A}^{\text{H}}\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}\boldsymbol{A}}\right)^2 = \phi_{\text{ss}}\,.\end{aligned} \tag{47.94}$$

As expected, the LCMV is a distortionless beamformer. The noise component at the beamformer output is given by,

$$\begin{aligned}&\phi_{Y_nY_n}(k,\ell)\\ &= (\boldsymbol{W}^{\text{LCMV}})^{\text{H}}\boldsymbol{\Phi}_{\boldsymbol{NN}}\boldsymbol{W}^{\text{LCMV}} = \frac{\boldsymbol{A}^{\text{H}}\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}\boldsymbol{A}}{\left(\boldsymbol{A}^{\text{H}}\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}\boldsymbol{A}\right)^2}\\ &= \frac{1}{\boldsymbol{A}^{\text{H}}\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}\boldsymbol{A}}\,.\end{aligned} \tag{47.95}$$

Using (47.94) and (47.95), the first term in the multichannel Wiener filter can be rewritten as

$$\begin{aligned}&\frac{\phi_{\text{ss}}}{\phi_{\text{ss}}+\left(\boldsymbol{A}^{\text{H}}\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}\boldsymbol{A}\right)^{-1}}\\ &= \frac{\phi_{Y_{\text{s}}Y_{\text{s}}}(k,\ell)}{\phi_{Y_nY_n}(k,\ell)+\phi_{Y_{\text{s}}Y_{\text{s}}}(k,\ell)}\,,\end{aligned} \tag{47.96}$$

which is evidently recognized as the single-channel Wiener filter applied to the LCMV beamformer output. It was therefore proven that the multichannel Wiener filter can be factorized into a product of the LCMV beamformer and a single-channel Wiener postfilter applied to the beamformer output,

$$\begin{aligned}&\boldsymbol{W}^{\text{Wiener}}(k,\ell)\\ &= \underbrace{\frac{\phi_{Y_{\text{s}}Y_{\text{s}}}(k,\ell)}{\phi_{Y_nY_n}(k,\ell)+\phi_{Y_{\text{s}}Y_{\text{s}}}(k,\ell)}}_{\text{Wiener postfilter}}\\ &\quad\times\underbrace{\frac{\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}(k,\ell)\boldsymbol{A}(k)}{\boldsymbol{A}^{\text{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{NN}}^{-1}(k,\ell)\boldsymbol{A}(k)}}_{\text{LCMV beamformer}}\,.\end{aligned} \tag{47.97}$$

Several algorithms have been proposed for designing the postfilter; all differ in their treatment of the single-channel Wiener postfilter estimation. *Zelinski* [47.79] was probably the first to apply a postfilter to the output of a microphone array (delay and sum beamformer in his formulation). Zelinski proposed the following Wiener

Amazon EX1010 Page 964

filter estimation

$$\boldsymbol{W}^{\text{Zelinski}}(k,\ell) = \frac{\frac{2}{M(M-1)}\sum_{i=1}^{M-1}\sum_{j=i+1}^{M}\text{Re}[Z_i(k,\ell)Z_j^*(k,\ell)]}{\frac{1}{M}\sum_{i=1}^{M}|Z_i(k,\ell)|^2}\,. \tag{47.98}$$

Later, postfiltering was incorporated into the Griffiths and Jim GSC beamformer by *Bitzer* et al. [47.86, 87]. The authors proposed to use two postfilters in succession. The first is applied to the FBF branch, and the second to the GSC output. In directional noise source and in the low-frequency band of a diffused noise field, correlation between the noise components at each sensor exists. While the first postfilter is useless in this case, the latter suppresses the noise.

Simmer and *Wasiljeff* [47.88] showed that Zelinski's postfilter has two disadvantages. First, only minor SNR improvement can be expected in frequencies, for which the coherence function between the received noise signals is high (i. e., coherent noise sources). Second, for frequencies with a low coherence function, the noise PSD is overestimated by a factor M (the number of microphones). They propose to mitigate the second disadvantage by slightly modifying the noise estimation, to obtain an estimate of the noise PSD at the output of the beamformer, rather than at its input.

In diffused noise field the noise coherence function tends to be high in lower frequencies, whereas in higher-frequency bands it tends to be low. The cutoff frequency depends on the distance between microphones (see further discussion in Sect. 47.7). This property is the cause for the first drawback of Zelinski's postfilter. It was therefore proposed by *Fischer* and *Kammeyer* [47.83] to split the beamformer into three nonoverlapping subarrays (with different inter-microphone distances) for which the noise coherence function is kept low. To avoid grating lobes, bandpass filters with corresponding cutoff frequencies, are applied to the beamformer output. *Marro* et al. [47.43] improved this concept and further modified the Wiener postfilter estimation. A comprehensive survey of these postfiltering methods can be found in [47.78]. *McCowan* and *Bourlard* [47.89, 90] develop a more-general expression of the postfilter estimation, based on an assumed knowledge of the complex coherence function of the noise field. This general expression can be used to construct a more-appropriate postfilter in a variety of different noise fields.

47.6.2 Log-Spectral Amplitude Postfiltering

A major drawback of single-channel postfiltering techniques is that highly nonstationary noise components are not addressed. The time variation of the interfering signals is assumed to be sufficiently slow, such that the postfilter can track and adapt to the changes in the noise statistics. Unfortunately, transient interferences are often much too brief and abrupt for the conventional tracking methods.

Transient Beam-to-Reference Ratio

Generally, the TF-GSC output comprises three components: a nonstationary desired source component, a pseudostationary noise component, and a transient interference. Our objective is to determine which category a given time–frequency bin belongs to, based on the beamformer output and the reference signals.

Recall the three hypotheses H_{0s}, H_{0t}, and H_1 that indicate, respectively, the absence of transients, the presence of an interfering transient, and the presence of a desired source transient at the beamformer output (the pseudostationary interference is present in any case). Then, if transients have not been detected at the beamformer output and the reference signals, we can accept the H_{0s} hypothesis. If a transient is detected at the beamformer output but not at the reference signals, the transient is likely a source component and therefore we determine that H_1 is true. On the contrary, a transient that is detected at one of the reference signals but not at the beamformer output is likely an interfering component, which implies that H_{0t} is true. If a transient is simultaneously detected at the beamformer output and at one of the reference signals, a further test is required, which involves the ratio between the transient power at beamformer output and the transient power at the reference signals. The discussion here is partly based on [47.77]. A real-time version of the method that incorporates adaptive estimation of the ATFs is introduced in [47.91].

Let $\mathcal{S}$ be a smoothing operator in the power-spectral domain,

$$\mathcal{S}Y(k,\ell) = \alpha_s \cdot \mathcal{S}Y(k,\ell-1) + (1-\alpha_s)\sum_{i=-w}^{w} b_i |Y(k-i,\ell)|^2\,, \tag{47.99}$$

where α_s $(0 \le \alpha_s \le 1)$ is a forgetting factor for the smoothing in time, and b is a normalized window function $(\sum_{i=-w}^{w} b_i = 1)$ that determines the order of

Amazon EX1010 Page 965

smoothing in frequency. Let $\mathcal{M}$ denote an estimator for the PSD of the background pseudostationary noise, derived using the MCRA approach [47.66,67]. The decision rules for detecting transients at the TF-GSC output and reference signals are

$$\Lambda_Y(k,\ell) \triangleq \mathcal{S}Y(k,\ell)/\,\mathcal{M}Y(k,\ell) > \Lambda_0\,, \tag{47.100}$$

$$\Lambda_{\boldsymbol{U}}(k,\ell) \triangleq \max_{2\le i\le M}\left\{\frac{\mathcal{S}U_i(k,\ell)}{\mathcal{M}U_i(k,\ell)}\right\} > \Lambda_1\,, \tag{47.101}$$

respectively, where Λ_Y and Λ_U denote measures of the local nonstationarities, and Λ_0 and Λ_1 are the corresponding threshold values for detecting transients [47.92]. The transient beam-to-reference ratio (TBRR) is defined by the ratio between the transient power of the beamformer output and the transient power of the strongest reference signal

$$\Omega(k,\ell) = \frac{\mathcal{S}Y(k,\ell) - \mathcal{M}Y(k,\ell)}{\max\limits_{2\le i\le M}\{\mathcal{S}U_i(k,\ell) - \mathcal{M}U_i(k,\ell)\}}\,. \tag{47.102}$$

Transient signal components are relatively strong at the beamformer output, whereas transient noise components are relatively strong at one of the reference signals. Hence, we expect $\Omega(k,\ell)$ to be large for signal transients, and small for noise transients. Assuming there exist thresholds $\Omega_{\text{high}}(k)$ and $\Omega_{\text{low}}(k)$ such that

$$\Omega(k,\ell)|_{H_{0t}} \le \Omega_{\text{low}}(k) \le \Omega_{\text{high}}(k) \le \Omega(k,\ell)|_{H_1} \tag{47.103}$$

the decision rule for differentiating desired signal components from the transient interference components is

$$\begin{aligned} &H_{0t}: \gamma_{\text{s}}(k,\ell) \le 1 \text{ or } \Omega(k,\ell) \le \Omega_{\text{low}}(k)\,,\\ &H_1: \gamma_{\text{s}}(k,\ell) \ge \gamma_0 \text{ and } \Omega(k,\ell) \ge \Omega_{\text{high}}(k)\,,\\ &H_{\text{r}}: \text{otherwise}, \end{aligned} \tag{47.104}$$

where

$$\gamma_{\text{s}}(k,\ell) \triangleq \frac{|Y(k,\ell)|^2}{\mathcal{M}Y(k,\ell)} \tag{47.105}$$

represents the a posteriori SNR at the beamformer output with respect to the pseudostationary noise, γ_0 denotes a constant satisfying $\mathcal{P}\,(\gamma_{\text{s}}(k,\ell) \ge \gamma_0 \mid H_{0\text{s}}) < \epsilon$ for a certain significance level ϵ, and H_{r} designates a *reject* option where the conditional error of making a decision between $H_{0\text{t}}$ and H_1 is high.

Figure 47.5 summarizes a block diagram for the hypothesis testing. The hypothesis testing is carried out in the time–frequency plane for each frame and frequency bin. $H_{0\text{s}}$ is accepted when transients have neither been detected at the beamformer output nor at the reference signals. If a transient is detected at the beamformer output but not at the reference signals, we accept H_1. On the other hand, if a transient is detected at one of the reference signals but not at the beamformer output, we accept $H_{0\text{t}}$. If a transient is detected simultaneously at the beamformer output and at one of the reference signals, we compute the TBRR $\Omega(k,\ell)$ and the a posteriori SNR at the beamformer output with respect to the pseudostationary noise $\gamma_{\text{s}}(k,\ell)$ and decide on the hypothesis according to (47.104).

Log-Spectral Amplitude Estimation

We address now the problem of estimating the time-varying PSD of the TF-GSC output noise component, and present the multichannel postfiltering technique. Figure 47.6 describes a block diagram of the multichannel postfiltering. Following the hypothesis testing, an estimate $\hat{q}(k,\ell)$ for the a priori signal absence probability is produced. Subsequently, we derive an estimate $p(k,\ell) \triangleq \mathcal{P}(H_1 \mid Y, \boldsymbol{U})$ for the signal presence probability, and an estimate $\hat{\lambda}_{\text{d}}(k,\ell)$ for the noise PSD.

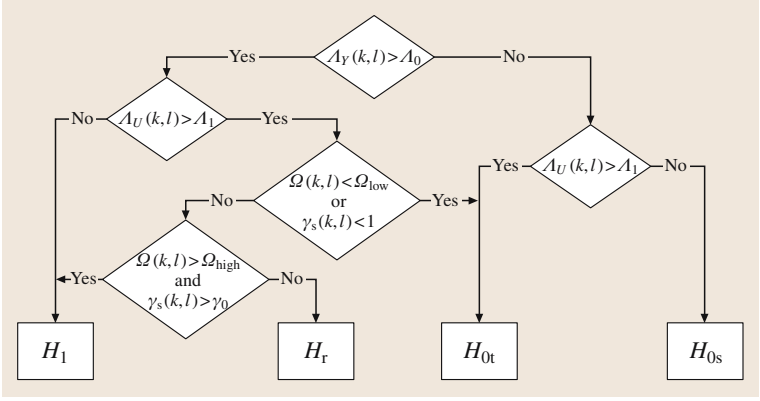


Fig. 47.5 Block diagram for hypothesis testing

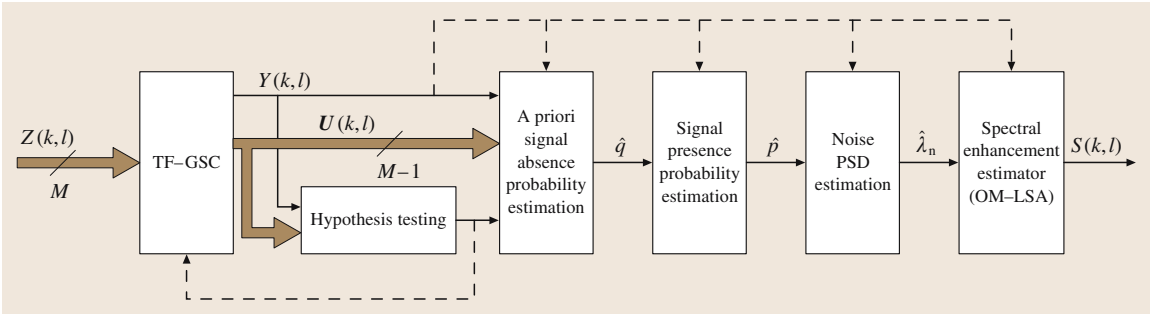


Fig. 47.6 Block diagram of multichannel postfiltering

Finally, spectral enhancement of the beamformer output is achieved by applying the optimally-modified log-spectral amplitude (OM-LSA) gain function [47.93], which minimizes the MSE of the log-spectral amplitude under signal presence uncertainty.

Based on a Gaussian statistical model [47.94], the signal presence probability is given by

$$p(k,\ell) = \left\{1+\frac{q(k,\ell)}{1-q(k,\ell)}[1+\xi(k,\ell)]\exp[-\upsilon(k,\ell)]\right\}^{-1}, \quad (47.106)$$

where $\xi(k,\ell) \triangleq \lambda_s(k,\ell)/\lambda_n(k,\ell)$ is the a priori SNR, $\lambda_s(k,\ell)$ is the desired signal PSD at the beamformer output $\lambda_n(k,\ell)$ is the noise PSD at the beamformer output, $\upsilon(k,\ell) \triangleq \gamma(k,\ell)\,\xi(k,\ell)/[1+\xi(k,\ell)]$, and $\gamma(k,\ell) \triangleq |Y(k,\ell)|^2/\lambda_n(k,\ell)$ is the a posteriori SNR. The a priori signal absence probability $\hat{q}(k,\ell)$ is set to 1 if the signal absence hypotheses (H_{0s} or H_{0t}) are accepted, and is set to 0 if the signal presence hypothesis (H_1) is accepted. In the case of the reject hypothesis H_r, a soft signal detection is accomplished by letting $\hat{q}(k,\ell)$ be inversely proportional to $\Omega(k,\ell)$ and $\gamma_s(k,\ell)$:

$$\hat{q}(k,\ell) = \max\left\{\frac{\gamma_0-\gamma_s(k,\ell)}{\gamma_0-1}, \frac{\Omega_{high}-\Omega(k,\ell)}{\Omega_{high}-\Omega_{low}}\right\}. \quad (47.107)$$

The a priori SNR is estimated by [47.93]

$$\hat{\xi}(k,\ell) = \alpha K_{H_1}^2(k,\ell-1)\gamma(k,\ell-1) + (1-\alpha)\max\{\gamma(k,\ell)-1, 0\}, \quad (47.108)$$

where α is a weighting factor that controls the trade-off between NR and signal distortion, and

$$K_{H_1}(k,\ell) \triangleq \frac{\xi(k,\ell)}{1+\xi(k,\ell)}\exp\left(\frac{1}{2}\int_{\upsilon(k,\ell)}^{\infty}\frac{e^{-t}}{t}\,dt\right) \quad (47.109)$$

is the spectral gain function of the log-spectral amplitude (LSA) estimator when signal is surely present [47.95]. An estimate for noise PSD is obtained by recursively averaging past spectral power values of the noisy measurement, using a time-varying frequency-dependent smoothing parameter. The recursive averaging is given by

$$\hat{\lambda}_n(k,\ell+1) = \tilde{\alpha}_n(k,\ell)\hat{\lambda}_n(k,\ell)+\beta[1-\tilde{\alpha}_n(k,\ell)]|Y(k,\ell)|^2, \quad (47.110)$$

where the smoothing parameter $\tilde{\alpha}_n(k,\ell)$ is determined by the signal presence probability $p(k,\ell)$,

$$\tilde{\alpha}_n(k,\ell) \triangleq \alpha_n+(1-\alpha_n)\,p(k,\ell), \quad (47.111)$$

and β is a factor that compensates the bias when signal is absent. The constant α_n ($0<\alpha_n<1$) represents the minimal smoothing parameter value. The smoothing parameter is close to 1 when signal is present, to prevent an increase in the noise estimate as a result of signal components. It decreases when the probability of signal presence decreases, to allow a fast update of the noise estimate.

Table 47.3 Values of the parameters used in the implementation of the log-spectral amplitude postfiltering for a sampling rate of 8 kHz

Normalized LMS:	$\alpha_p = 0.9$	$\mu_h = 0.05$
ATF identification:	$N = 10$	$R = 10$
Hypothesis testing:	$\alpha_s = 0.9$	$\gamma_0 = 4.6$
	$\Lambda_0 = 1.67$	$\Lambda_1 = 1.81$
	$\Omega_{low} = 1$	$\Omega_{high} = 3$
	$b = (0.25\ \ 0.5\ \ 0.25)$	
Noise PSD estimation:	$\alpha_n = 0.85$	$\beta = 1.47$
Spectral enhancement:	$\alpha = 0.92$	
	$K_{min} = -20\,dB$	

Amazon EX1010 Page 967

The estimate of the clean signal STFT is finally given by

$$\hat{S}(k,\ell) = K(k,\ell)Y(k,\ell)\,, \tag{47.112}$$

where

$$K(k,\ell) = \left\{K_{H_1}(k,\ell)\right\}^{p(k,\ell)} K_{\min}^{1-p(k,\ell)} \tag{47.113}$$

is the OM-LSA gain function and $K_{\min}$ denotes a lower bound constraint for the gain when signal is absent. The implementation of the integrated TF-GSC and multichannel postfiltering algorithm is summarized in Fig. 47.6.

Typical values of the respective parameters, for a sampling rate of 8 kHz, are given in Table 47.3. The STFT and its inverse are implemented with biorthogonal Hamming windows of 256 samples length (32 ms) and 64 samples frame update step (75% overlapping windows).

47.7 Performance Analysis

The use of actual signals (such as noisy speech recordings in room environment) demonstrates the ability of the TF-GSC algorithm to reduce the noise while maintaining the desired signal spectral content (Sect. 47.8). However, it is also beneficial to perform analytical evaluation of the expected performance, especially for determining the performance limits. While the D-GSC [47.36] is widely analyzed in the literature, the more-realistic arbitrary ATFs scenario, is only superficially treated. In more-complex environments such as reverberating room this assumption is not valid, and may result in severe degradation in the performance. Furthermore, most references address the NR obtained by the algorithm but do not present any measure of distortion imposed on the desired signal, even in the simple delay-only ATFs. In this section we will analyze both the NR of the TF-GSC structure (while using the RTFs rather than the ATFs themselves) and the distortion it imposes. For a thorough performance evaluation of the multichannel postfilter (for the two-channel case) please refer to [47.96].

47.7.1 The Power Spectral Density of the Beamformer Output

Using (47.24) and (47.37), the algorithm's output is given by

$$Y(k,\ell) = \frac{\hat{\tilde{\boldsymbol{A}}}^{\mathrm{H}}(k)}{\|\hat{\tilde{\boldsymbol{A}}}(k)\|^2}\boldsymbol{Z}(k,\ell) - \boldsymbol{G}^{\mathrm{H}}(k,\ell)\widehat{\mathcal{H}}^{\mathrm{H}}(k)\boldsymbol{Z}(k,\ell)\,,$$

where only estimates of the RTFs, $\hat{\tilde{\boldsymbol{A}}}(k)$ [and $\widehat{\mathcal{H}}(k)$], rather than their exact values, are assumed to be known. Using this expression, the PSD of the output signal is given by

$$\begin{aligned}
\boldsymbol{\Phi}_{\mathrm{yy}}(k,\ell) &= E[Y(k,\ell)Y^*(k,\ell)] \\
&= E\left\{\left[\frac{1}{\|\hat{\tilde{\boldsymbol{A}}}(k)\|^2}\hat{\tilde{\boldsymbol{A}}}^{\mathrm{H}}(k)\boldsymbol{Z}(k,\ell)\right.\right. \\
&\quad \left.- \boldsymbol{G}^{\mathrm{H}}(k,\ell)\widehat{\mathcal{H}}^{\mathrm{H}}(k)\boldsymbol{Z}(k,\ell),\right] \\
&\quad \times\left[\frac{1}{\|\hat{\tilde{\boldsymbol{A}}}(k)\|^2}\hat{\tilde{\boldsymbol{A}}}^{\mathrm{H}}(k)\boldsymbol{Z}(k,\ell)\right. \\
&\quad \left.\left.- \boldsymbol{G}^{\mathrm{H}}(k,\ell)\widehat{\mathcal{H}}^{\mathrm{H}}(k)\boldsymbol{Z}(k,\ell),\right]^{\mathrm{H}}\right\}\,.
\end{aligned} \tag{47.114}$$

Opening brackets and using the PSD definition $\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell) = E\{\boldsymbol{Z}(k,\ell)\boldsymbol{Z}^{\mathrm{H}}(k,\ell)\}$ yields,

$$\begin{aligned}
\boldsymbol{\Phi}_{\mathrm{yy}}(k,\ell) &= \frac{1}{\|\hat{\tilde{\boldsymbol{A}}}(k)\|^4}\hat{\tilde{\boldsymbol{A}}}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)\hat{\tilde{\boldsymbol{A}}}(k) \\
&\quad - \frac{1}{\|\hat{\tilde{\boldsymbol{A}}}(k)\|^2}\boldsymbol{G}^{\mathrm{H}}(k,\ell)\widehat{\mathcal{H}}^{\mathrm{H}}(k) \\
&\quad \times\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)\hat{\tilde{\boldsymbol{A}}}(k) \\
&\quad - \frac{1}{\|\hat{\tilde{\boldsymbol{A}}}(k)\|^2}\hat{\tilde{\boldsymbol{A}}}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)\widehat{\mathcal{H}}(k)\boldsymbol{G}(k) \\
&\quad + \boldsymbol{G}^{\mathrm{H}}(k,\ell)\widehat{\mathcal{H}}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)\widehat{\mathcal{H}}(k)\boldsymbol{G}(k)\,.
\end{aligned} \tag{47.115}$$

The output PSD depends on the input signal $\boldsymbol{Z}(k,\ell)$ and the optimal multichannel Wiener filter, given by (47.26), calculated during frames for which hypothesis $H_{0\mathrm{s}}$ is true. Using the independence of the desired signal and the noise signal, the NR and the distortion imposed by the algorithm can be calculated separately by deriving

expressions for the output PSD in the following two situations:

$$\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)=\begin{cases}\boldsymbol{\Phi}_{\boldsymbol{N}_\mathrm{s}\boldsymbol{N}_\mathrm{s}}(k,\ell) & H_{0\mathrm{s}}\,,\\ \phi_{\mathrm{ss}}(k,\ell)\boldsymbol{A}(k)\boldsymbol{A}^{\mathrm{H}}(k) & H_1\,,\end{cases}$$

yielding

$$\boldsymbol{\Phi}_{\mathrm{yy}}(k,\ell)=\begin{cases}\boldsymbol{\Phi}^{\mathrm{n}}_{\mathrm{yy}}(k,\ell) & H_{0\mathrm{s}} \quad \Rightarrow \text{noise reduction},\\ \boldsymbol{\Phi}^{\mathrm{s}}_{\mathrm{yy}}(k,\ell) & H_1 \quad \Rightarrow \text{distortion}\,.\end{cases}$$

Note that for simplicity we calculate the NR only during $H_{0\mathrm{s}}$, i. e., while only stationary noise signal is present. For a performance evaluation in the nonstationary case, please refer to [47.96].

Using (47.26) and (47.115), we obtain the output signal PSD:

$$\begin{aligned}\boldsymbol{\Phi}_{\mathrm{yy}}(k,\ell)=\frac{1}{\|\hat{\tilde{\boldsymbol{A}}}(k)\|^4}\times\Big\{&\hat{\tilde{\boldsymbol{A}}}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)\hat{\tilde{\boldsymbol{A}}}(k)\\ &-\hat{\tilde{\boldsymbol{A}}}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{N}_\mathrm{s}\boldsymbol{N}_\mathrm{s}}(k,\ell)\widehat{\mathcal{H}}(k)\\ &\quad\Big[\widehat{\mathcal{H}}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{N}_\mathrm{s}\boldsymbol{N}_\mathrm{s}}(k,\ell)\widehat{\mathcal{H}}(k)\Big]^{-1}\\ &\quad\widehat{\mathcal{H}}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)\hat{\tilde{\boldsymbol{A}}}(k)\\ &-\hat{\tilde{\boldsymbol{A}}}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)\widehat{\mathcal{H}}(k)\\ &\quad\Big[\widehat{\mathcal{H}}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{N}_\mathrm{s}\boldsymbol{N}_\mathrm{s}}(k,\ell)\widehat{\mathcal{H}}(k)\Big]^{-1}\\ &\quad\widehat{\mathcal{H}}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{N}_\mathrm{s}\boldsymbol{N}_\mathrm{s}}(k,\ell)\hat{\tilde{\boldsymbol{A}}}(k)\\ &+\hat{\tilde{\boldsymbol{A}}}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{N}_\mathrm{s}\boldsymbol{N}_\mathrm{s}}(k,\ell)\widehat{\mathcal{H}}(k)\\ &\quad\Big[\widehat{\mathcal{H}}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{N}_\mathrm{s}\boldsymbol{N}_\mathrm{s}}(k,\ell)\widehat{\mathcal{H}}(k)\Big]^{-1}\\ &\quad\widehat{\mathcal{H}}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)\widehat{\mathcal{H}}(k)\\ &\times\Big[\widehat{\mathcal{H}}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{N}_\mathrm{s}\boldsymbol{N}_\mathrm{s}}(k,\ell)\widehat{\mathcal{H}}(k)\Big]^{-1}\\ &\quad\widehat{\mathcal{H}}^{\mathrm{H}}(k)\boldsymbol{\Phi}_{\boldsymbol{N}_\mathrm{s}\boldsymbol{N}_\mathrm{s}}(k,\ell)\hat{\tilde{\boldsymbol{A}}}(k)\Big\}\,.\end{aligned}\tag{47.116}$$

This complicated expression depends on various parameters: the input signal PSD, $\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)$, the stationary noise PSD used for calculating the optimal filters, $\boldsymbol{\Phi}_{\boldsymbol{N}_\mathrm{s}\boldsymbol{N}_\mathrm{s}}(k,\ell)$, and the RTFs estimate $\hat{\tilde{\boldsymbol{A}}}(k)$ [which is also used for the BM $\widehat{\mathcal{H}}(k)$]. This expression will be used in Subsects. 47.7.2 and 47.7.3 for deriving general expressions for the distortion imposed by the algorithm and the obtainable NR, respectively. These general expressions will be evaluated for several interesting cases.

47.7.2 Signal Distortion

The distortion imposed by the algorithm can be calculated by the general expression given in (47.116) for a signal $\boldsymbol{Z}(k,\ell)=\boldsymbol{A}(k)s(k,\ell)$. Assume a perfect estimate of the RTFs $\tilde{\boldsymbol{A}}(k)$, is available, i. e., $\hat{\tilde{\boldsymbol{A}}}(k)=\tilde{\boldsymbol{A}}(k)=\frac{\boldsymbol{A}(k)}{A_1(k)}$. Thus, using the signal PSD expression $\boldsymbol{\Phi}_{\boldsymbol{ZZ}}(k,\ell)=\phi_{\mathrm{ss}}(k,\ell)\boldsymbol{A}(k)\boldsymbol{A}^{\mathrm{H}}(k)$ and the identity $\mathcal{H}^{\mathrm{H}}(k)\boldsymbol{A}(k)=0$, expression (47.116) reduces to

$$\begin{aligned}&\boldsymbol{\Phi}^{\mathrm{s}}_{\mathrm{yy}}(k,\ell)\\ &=\phi_{\mathrm{ss}}(k,\ell)\frac{|\mathcal{F}(k)|^2}{\|\tilde{\boldsymbol{A}}(k)\|^4}\tilde{\boldsymbol{A}}^{\mathrm{H}}(k)\boldsymbol{A}(k)\boldsymbol{A}^{\mathrm{H}}(k)\tilde{\boldsymbol{A}}(k)\\ &\quad\phi_{\mathrm{ss}}(k,\ell)|A_1(k)|^2\,.\end{aligned}\tag{47.117}$$

The filter $A_1(k)$ is the ATF relating the source signal and the first (arbitrarily chosen as the reference) sensor. This distortion cannot be eliminated by the algorithm. Note that this distortion is due to the use of the RTFs, rather than the ATFs themselves. Using direct estimate of the ATFs will avoid this distortion affect. Actually, it imposes on the output signal the same amount of distortion imposed on the arbitrary reference sensor. Hence, we define the total distortion caused by the algorithm by normalizing the output,

$$\mathrm{DIS}(k,\ell)=\frac{\boldsymbol{\Phi}^{\mathrm{s}}_{\mathrm{yy}}(k,\ell)}{|A_1(k)|^2\phi_{\mathrm{ss}}(k,\ell)}\,.\tag{47.118}$$

Hence, a value of $\mathrm{DIS}(k,\ell)=1$ indicates a distortionless output. This value is obtained whenever an exact knowledge of the RTFs is available.

The distortion level demonstrates only weak dependence on the noise field, both for the delay-only and complex ATF cases. For details please refer to [47.77].

47.7.3 Stationary Noise Reduction

We calculate now the amount of obtainable stationary NR. When the noise is nonstationary, the use of a postfilter becomes more important. A performance analysis of the multichannel postfilter (for the two-channel case) in nonstationary noise environment can be found in [47.96].

We will use again the general expression for the output signal given by (47.116), this time with a noise signal as the input signal, i. e., $\boldsymbol{Z}(k,\ell)=\boldsymbol{N}_\mathrm{s}(k,\ell)$ [the same noise signal used for calculating the optimal Wiener filter (47.26)]. The expression (47.116) now reduces to

$$\begin{aligned}
&\Phi_{yy}^{n}(k,\ell)\\
&=\frac{\hat{\tilde{A}}^{H}(k)\Phi_{N_sN_s}(k,\ell)\hat{\tilde{A}}(k)}{\|\hat{\tilde{A}}(k)\|^4}-\frac{\hat{\tilde{A}}^{H}(k)}{\|\hat{\tilde{A}}(k)\|^4}\\
&\quad\times\Phi_{N_sN_s}(k,\ell)\widehat{\mathcal{H}}(k)\left[\widehat{\mathcal{H}}^{H}(k)\Phi_{N_sN_s}(k,\ell)\right]^{-1}\\
&\quad\times\widehat{\mathcal{H}}^{H}(k)\Phi_{N_sN_s}(k,\ell)\hat{\tilde{A}}(k)\,. \qquad (47.119)
\end{aligned}$$

The first term of the equation can be identified as $\Phi_{FBF}^{n}(k,\ell)$, the FBF component of the output PSD when a noise-only signal is applied to the array,

$$\begin{aligned}
\Phi_{FBF}^{n}(k,\ell)&=E\left\{Y_{FBF}^{n}(k,\ell)\left[Y_{FBF}^{n}(k,\ell)\right]^{*}\right\}\\
&=\frac{\hat{\tilde{A}}^{H}(k)\Phi_{N_sN_s}(k,\ell)\hat{\tilde{A}}(k)}{\|\hat{\tilde{A}}(k)\|^4}\,. \qquad (47.120)
\end{aligned}$$

Another interesting figure of merit is the extra NR obtained by the noise cancelling branch (see also [47.40]),

$$\mathrm{NR}_{ANC}(k,\ell)=\frac{\Phi_{FBF}^{n}(k,\ell)}{\Phi_{yy}^{n}(k,\ell)}\,. \qquad (47.121)$$

Expressions (47.119), (47.120), and (47.121) can be used for calculating the NR obtained by the algorithm and to determine the major contributor for this NR. Assuming small errors regime, the error in estimating $\tilde{A}(k)$ has only minor influence on the NR. Therefore, we will assume, throughout the NR analysis, perfect knowledge of the RTFs, i. e., $\hat{\tilde{A}}(k)=\tilde{A}(k)$.

The resulting expressions for the noise cancellation depends on the noise PSD at the sensors. We calculate now the expected NR of the algorithm for three important noise fields: coherent (point source), diffused (spherically isotropic), and incoherent (noise signals generated at the sensors; e.g., amplifier noise, are assumed to be uncorrelated).

Coherent Noise Field

Assume a single stationary point source noise signal with PSD $\Phi_{n_sn_s}(k,\ell)$ and assume that $b_m(t)$ are slowly time-varying ATFs relating the noise source and the m-th sensor. Define,

$$\boldsymbol{N}_s(k,\ell)=\boldsymbol{B}(k)N_s(k,\ell)\,,$$

where

$$\boldsymbol{B}^{T}(k)=(B_1(k)\;\;B_2(k)\;\;\cdots\;\;B_M(k))\,.$$

The PSD matrix of the noise component at the sensors' signals is given by,

$$\Phi_{N_sN_s}(k,\ell)=\Phi_{n_sn_s}(k,\ell)\boldsymbol{B}(k)\boldsymbol{B}^{H}(k)+\varepsilon\mathbf{I}\,,$$

where $\mathbf{I}$ is an $M\times M$ identity matrix, and $\varepsilon\to 0$. The last term is added for stability reasons (see Appendix 47.A). For $\boldsymbol{B}(k)\neq\boldsymbol{A}(k)$, the achievable NR is infinite, i. e.,

$$\Phi_{yy}^{n}(k,\ell)=0\quad\text{for}\quad\boldsymbol{B}(k)\neq\boldsymbol{A}(k)\,.$$

Thus, perfect noise cancellation is achieved. The derivation of this result is given in Appendix 47.A. Note that this is not a surprising result, since for $M\geq 2$ the Wiener filter can entirely eliminate the noise component. This result is valid for all ATFs $\boldsymbol{B}(k)$ provided that $\boldsymbol{B}(k)\neq\boldsymbol{A}(k)$, i. e., the noise and the signal do not originate from the same point. If $\boldsymbol{B}(k)=\boldsymbol{A}(k)$ the noise and desired signal are indistinguishable and no NR is expected. If $\boldsymbol{B}(k)\neq\boldsymbol{A}(k)$ the proposed algorithm can eliminate any point source noise signal as good as the D-GSC can eliminate a directional noise signal in the delay-only propagation case (see [47.40]).

It is also interesting to explore the contribution of the FBF block to the NR,

$$\begin{aligned}
&\Phi_{FBF}^{n}(k,\ell)\\
&=\frac{\tilde{A}^{H}(k)\Phi_{N_sN_s}(k,\ell)\tilde{A}(k)}{\|\tilde{A}(k)\|^4}\\
&=\frac{\tilde{A}^{H}(k)\Phi_{n_sn_s}(k,\ell)\boldsymbol{B}(k)\boldsymbol{B}^{H}(k)\tilde{A}(k)}{\|\tilde{A}(k)\|^4}\\
&=\frac{\Phi_{n_sn_s}(k,\ell)}{\|\tilde{A}(k)\|^4}\tilde{A}^{H}(k)\boldsymbol{B}(k)\left[\tilde{A}^{H}(k)\boldsymbol{B}(k)\right]^{H}\,.
\end{aligned}$$

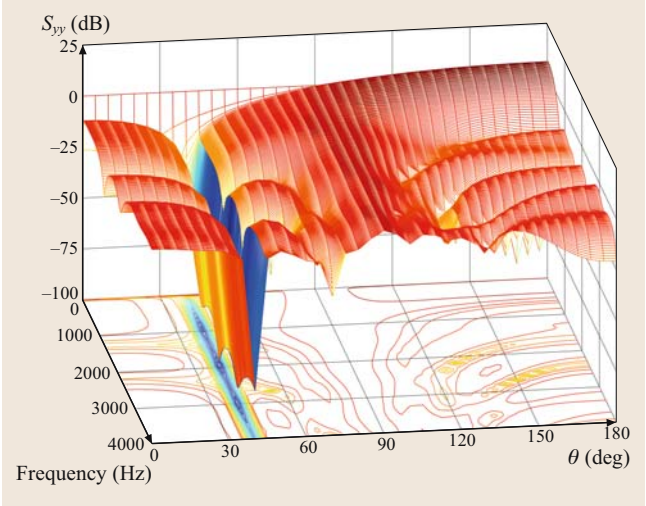


Fig. 47.7 Array output for directional noise field for linear array with $M=5$ sensors for delay-only ATFs

Although the FBF combines the desired signal coherently, it does not necessarily improves the SNR. The infinite NR is due to the ANC branch.

The expected NR in the simple case of delay-only ATFs is shown in Fig. 47.7. We optimize a linear array of $M = 5$ microphones to cancel noise impinging on the array from the direction $\theta = 40°$. We present the output signal of the array as a function of the frequency and the array steering angle θ. It is clearly shown that the main lobe is maintained (i. e., low distortion), while a null is constructed at all frequencies at the noise angle. The main lobe is wider in the lower-frequency band. This result is in good agreement with the theory, since at $\omega = 0\,\mathrm{rad/s}$ there is no phase difference between the signals at the sensors. Similar results were obtained by *Bitzer* et al. [47.40,41]. The general ATFs case is further explored in [47.77, 97].

Diffused Noise Field

In highly reverberant acoustical environment, such as a car enclosure, the noise field tends to be diffused (see for instance [47.42, 98]). A diffused noise source is assumed to be equidistributed on a sphere in the far field of the array. The cross-coherence function between signals received by two sensors (i,j) with distance d_{ij} is given by

$$\Gamma_{N_i N_j}(k) = \frac{\Phi_{N_i N_j}(k)}{\sqrt{\Phi_{N_i N_i}(k)\Phi_{N_j N_j}(k)}} = \frac{\sin(kd_{ij}/c)}{kd_{ij}/c} ,$$

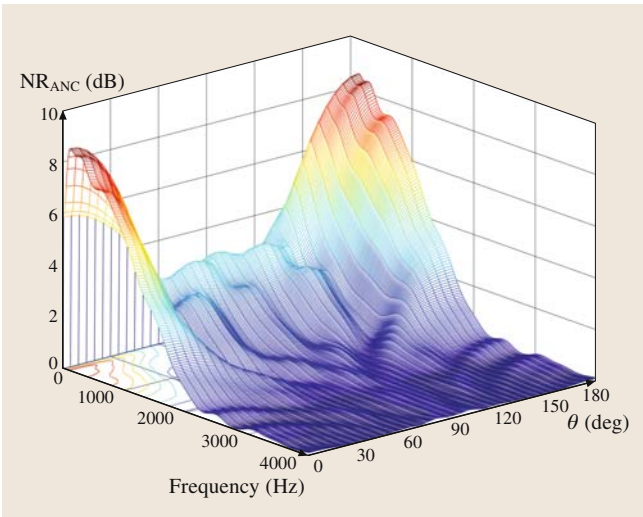


Fig. 47.8 Extra noise reduction of noise cancelling branch for diffused noise field using a linear array with $M = 5$ sensors and delay-only ATFs

where c is the speed of sound [47.98]. Thus, the coherence matrix is given by,

$$\Gamma(k) = \begin{pmatrix} 1 & \Gamma_{N_1 N_2}(k) & \cdots & \Gamma_{N_1 N_M}(k) \\ \Gamma_{N_2 N_1}(k) & 1 & \cdots & \Gamma_{N_2 N_M}(k) \\ \vdots & & \ddots & \vdots \\ \Gamma_{N_M N_1}(k) & & & 1 \end{pmatrix} .$$

The noise PSD at the sensors input is

$$\boldsymbol{\Phi}_{\boldsymbol{N}_\mathrm{s}\boldsymbol{N}_\mathrm{s}}(k,\ell) = \boldsymbol{\Phi}_{\boldsymbol{n}_\mathrm{s}\boldsymbol{n}_\mathrm{s}}(k,\ell)\Gamma(k) .$$

Using (47.120), the noise PSD at the FBF output is given by

$$\boldsymbol{\Phi}^{\mathrm{n}}_{\mathrm{FBF}}(k,\ell) = \frac{\phi_{\mathrm{nn}}(k,\ell)\tilde{\boldsymbol{A}}^{\mathrm{H}}(k)\Gamma(k)\tilde{\boldsymbol{A}}(k)}{\|\tilde{\boldsymbol{A}}(k)\|^4} .$$

The extra NR obtained by the ANC is given by

$$\begin{aligned} &\mathrm{NR}_{\mathrm{ANC}}(k,\ell) \\ &= \left\{1 - \left[\tilde{\boldsymbol{A}}^{\mathrm{H}}(k)\Gamma(k)\mathcal{H}(k)\left[\mathcal{H}^{\mathrm{H}}(k)\Gamma(k)\mathcal{H}(k)\right]^{-1}\right.\right. \\ &\quad \left.\left.\times \mathcal{H}^{\mathrm{H}}(k)\Gamma(k)\tilde{\boldsymbol{A}}(k)[\tilde{\boldsymbol{A}}^{\mathrm{H}}(k)\Gamma(k)\tilde{\boldsymbol{A}}(k)]^{-1}\right]\right\}^{-1} . \end{aligned} \tag{47.122}$$

This expression depends on the RTFs $\tilde{\boldsymbol{A}}(k)$, assumed to be error free, and on the coherence function $\Gamma(k)$.

The same $M = 5$ microphone array used for the directional noise case is now used for the diffused noise field. In Fig. 47.8 we show the extra NR obtained by the ANC for various steering angles and for the entire frequency band. It is clear that almost no NR is obtained in the high-frequency band and only relatively low NR in the low-frequency band. The obtained results are in accordance with the results in [47.39, 42].

Incoherent Noise Field

For incoherent noise field we assume that the noise at the sensors has no spatial correlation.

$$\boldsymbol{\Phi}_{\boldsymbol{N}_\mathrm{s}\boldsymbol{N}_\mathrm{s}}(k,\ell) = \boldsymbol{\Phi}_{\boldsymbol{n}_\mathrm{s}\boldsymbol{n}_\mathrm{s}}(k,\ell)\mathbf{I} ,$$

where $\mathbf{I}$ is an $M \times M$ identity matrix. Using (47.119) with perfect knowledge of the RTFs, i. e., $\hat{\tilde{\boldsymbol{A}}}(k) = \tilde{\boldsymbol{A}}(k)$, and with the prespecified $\boldsymbol{\Phi}_{\boldsymbol{N}_\mathrm{s}\boldsymbol{N}_\mathrm{s}}(k,\ell)$ we obtain,

$$\begin{aligned} &\boldsymbol{\Phi}^{\mathrm{n}}_{\mathrm{yy}}(k,\ell) \\ &= \frac{\boldsymbol{\Phi}_{\boldsymbol{n}_\mathrm{s}\boldsymbol{n}_\mathrm{s}}(k,\ell)}{\|\tilde{\boldsymbol{A}}(k)\|^4}\tilde{\boldsymbol{A}}^{\mathrm{H}}(k) \\ &\quad \times \left\{\mathbf{I} - \mathcal{H}(k)\left[\mathcal{H}^{\mathrm{H}}(k)\mathcal{H}(k)\right]^{-1}\mathcal{H}^{\mathrm{H}}(k)\right\}\tilde{\boldsymbol{A}}(k) . \end{aligned}$$

Amazon EX1010 Page 971

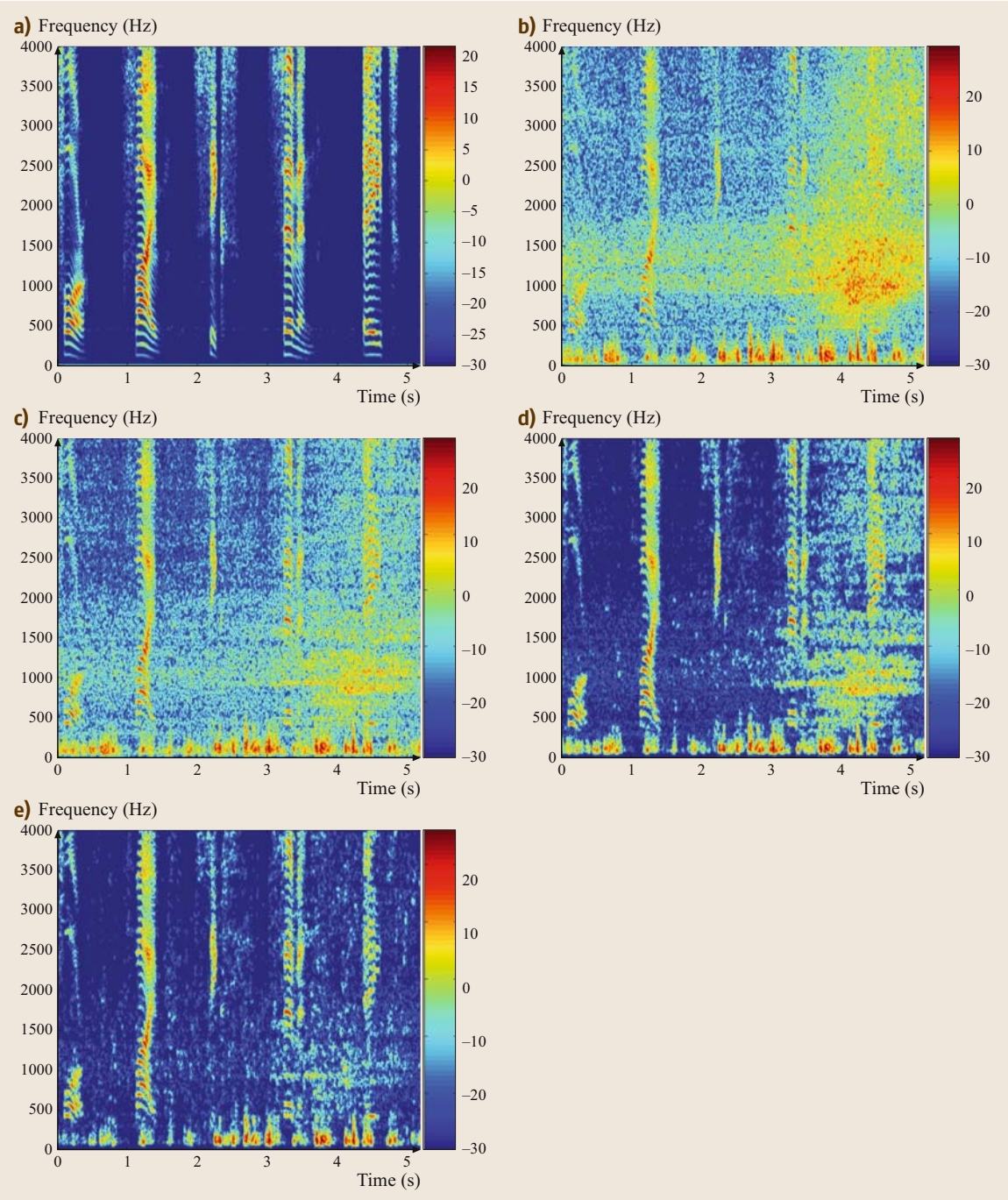


Fig. 47.9 (a) Original clean speech signal at microphone #1: "Five six seven eight nine". (b) Noisy signal at microphone #1. (c) TF-GSC output. (d) Single-channel postfiltering output. (e) Multichannel postfiltering.

It can be easily verified that,

$$\tilde{\boldsymbol{A}}^{\mathrm{H}}(k)\mathcal{H}(k) = \mathbf{0}_{1\times(M-1)} \,.$$

Furthermore, as $\mathcal{H}^{\mathrm{H}}(k)\mathcal{H}(k)$ is a positive matrix, its inverse always exists. Thus, the contribution of the noise cancelling branch is zero, and the NR is only attributed to the FBF. The noise power at the output is thus,

$$\begin{aligned}\boldsymbol{\Phi}^{\mathrm{n}}_{\mathrm{yy}}(k,\ell) = \boldsymbol{\Phi}^{\mathrm{n}}_{\mathrm{FBF}}(k,\ell) &= \boldsymbol{\Phi}_{n_{\mathrm{s}}n_{\mathrm{s}}}(k,\ell)\frac{\tilde{\boldsymbol{A}}^{\mathrm{H}}(k)\tilde{\boldsymbol{A}}(k)}{\|\tilde{\boldsymbol{A}}(k)\|^4} \\ &= \frac{\boldsymbol{\Phi}_{n_{\mathrm{s}}n_{\mathrm{s}}}(k,\ell)}{\|\tilde{\boldsymbol{A}}(k)\|^2} \,.\end{aligned}$$

Again, no NR is guaranteed by this structure, and the result depends on the RTFs involved.

In the case of delay-only ATFs, the FBF branch becomes a simple delay-and-sum beamformer, thus the expected NR is M, the number of sensors.

47.8 Experimental Results

In this section, we compare the performance of a system, consisting of the TF-GSC and multichannel postfilter, to a system consisting of a TF-GSC and a single-channel postfilter.

A linear array, consisting of four microphones with 5 cm spacing, is mounted in a car on the visor. Clean speech signals are recorded at a sampling rate of 8 kHz in the absence of background noise (standing car, silent environment). A car noise signal is recorded while the car speed is about 60 km/h, and the window next to the driver is slightly open (about 5 cm; the other windows are closed). The input microphone signals are generated by mixing the speech and noise signals at various SNR levels in the range $[-5, 10]$ dB.

Offline TF-GSC beamforming [47.38] is applied to the noisy multichannel signals, and its output is enhanced using the OM-LSA estimator [47.93]. The result is referred to as single-channel postfiltering output. Alternatively, the proposed TF-GSC and multichannel postfiltering is applied to the noisy signals. A subjective comparison between multichannel and single-channel postfiltering was conducted using speech sonograms and validated by informal listening tests. Typical examples of speech sonograms are presented in Fig. 47.9. For audio samples please refer to [47.99]. The noise PSD at the beamformer output varies substantially due to the residual interfering components of speech, wind blows, and passing cars. The TF-GSC output is characterized by a high level of noise. Single-channel postfiltering suppresses pseudostationary noise components, but is inefficient at attenuating the transient noise components. By contrast, the system which consists a TF-GSC and multichannel postfilter achieves superior noise attenuation, while preserving the desired source components. This is verified by subjective informal listening tests.

47.9 Summary

In this chapter, we concentrated on the GSC beamformer, and presented a comprehensive study of its components. We showed, that the GSC structure is closely related to other array optimization criteria, such as the Wiener filter. We described multimicrophone postfilters, based on either the MMSE or the log-spectral estimation criteria, which are designed for improving the amount of obtainable NR, with minimal degradation of speech quality. The robustness of the GSC structure to imperfect estimation of its components was analyzed. Various methods were proposed for increasing the robustness of the GSC, and especially, for avoiding leakage of the desired signal into the noise reference signals. Finally, the performance of the TF-GSC was theoretically analyzed and experimentally evaluated under nonstationary noise conditions.

Amazon EX1010 Page 973

47.A Appendix: Derivation of the Expected Noise Reduction for a Coherent Noise Field

For clarity of the exposition we will omit the time and frequency dependence in the derivation. Recall (47.119) and define

$$\mathcal{X}(k,\ell) \triangleq \mathcal{X} = \boldsymbol{\Phi}_{\boldsymbol{N}_s\boldsymbol{N}_s}\mathcal{H}\left(\mathcal{H}^{\mathrm{H}}\boldsymbol{\Phi}_{\boldsymbol{N}_s\boldsymbol{N}_s}\mathcal{H}\right)^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{\Phi}_{\boldsymbol{N}_s\boldsymbol{N}_s}\,. \tag{47.A1}$$

Denote,

$$\mathcal{X} \triangleq \mathcal{K}\times\mathcal{L}\times\mathcal{M}\,,$$

where, $\mathcal{K} = \boldsymbol{\Phi}_{\boldsymbol{N}_s\boldsymbol{N}_s}\mathcal{H}$, $\mathcal{L} = (\mathcal{H}^{\mathrm{H}}\boldsymbol{\Phi}_{\boldsymbol{N}_s\boldsymbol{N}_s}\mathcal{H})^{-1}$, and $\mathcal{M} = \mathcal{H}^{\mathrm{H}}\boldsymbol{\Phi}_{\boldsymbol{N}_s\boldsymbol{N}_s}$. Thus, $\mathcal{X}$ is a multiplication of three terms.

Starting from $\mathcal{L}$ and using the detailed noise structure,

$$\begin{aligned}\mathcal{L} &= \left[\mathcal{H}^{\mathrm{H}}\left(\phi_{\mathrm{nn}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}+\varepsilon\mathbf{I}\right)\mathcal{H}\right]^{-1}\\ &= \left[\phi_{\mathrm{nn}}(\mathcal{H}^{\mathrm{H}}\boldsymbol{B})(\mathcal{H}^{\mathrm{H}}\boldsymbol{B})^{\mathrm{H}}+\varepsilon\mathcal{H}^{\mathrm{H}}\mathcal{H}\right]^{-1}.\end{aligned}$$

If $\boldsymbol{B}=\boldsymbol{A}$, i. e., the noise source is located exactly at the desired signal position, then $\mathcal{H}^{\mathrm{H}}\boldsymbol{B}=0$, and the calculation of the inverse is straightforward, yielding $\mathcal{L} = (\varepsilon\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}$, $\mathcal{K} = \varepsilon\mathcal{H}$, and $\mathcal{M} = \varepsilon\mathcal{H}^{\mathrm{H}}$. Collecting all terms we obtain, $\mathcal{X} = \frac{\varepsilon^2}{\varepsilon}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}} \overset{\varepsilon\to 0}{\longrightarrow} 0$, i. e., the signal at the BM output is zero, as expected. The total noise part of the output is given by,

$$\begin{aligned}\boldsymbol{\Phi}^{\mathrm{n}}_{\mathrm{yy}} &= \boldsymbol{\Phi}^{\mathrm{n}}_{\mathrm{FBF}}\\ &= \frac{\hat{\boldsymbol{A}}^{\mathrm{H}}\boldsymbol{\Phi}_{\boldsymbol{N}_s\boldsymbol{N}_s}\hat{\boldsymbol{A}}}{\|\hat{\boldsymbol{A}}\|^4} = \frac{\hat{\boldsymbol{A}}^{\mathrm{H}}\left(\phi_{\mathrm{nn}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}+\varepsilon\mathbf{I}\right)\hat{\boldsymbol{A}}}{\|\hat{\boldsymbol{A}}\|^4}\\ &\overset{\varepsilon\to 0}{\longrightarrow} \phi_{nn}|A_1|^2\,,\end{aligned}$$

where the last transition is due to $\boldsymbol{B}=\boldsymbol{A}$. This is exactly the no-distortion result obtained in Sect. 47.7.2, for the desired signal direction.

Table 47.4 Twelve terms used for calculating $\mathcal{X} = \mathcal{K}\mathcal{L}\mathcal{M}$

$\mathcal{K}_1\mathcal{L}_1\mathcal{M}_1$	$= \frac{1}{\varepsilon}\boldsymbol{\Phi}^2_{\mathrm{nn}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}$;	(I)
$\mathcal{K}_1\mathcal{L}_2\mathcal{M}_1$	$= -\frac{1}{\varepsilon}\boldsymbol{\Phi}^3_{\mathrm{nn}} \times \frac{\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}}{\phi_{\mathrm{nn}}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}}$	(II)
	$= -\frac{1}{\varepsilon}\boldsymbol{\Phi}^2_{\mathrm{nn}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}$;	
$\mathcal{K}_1\mathcal{L}_3\mathcal{M}_1$	$= \frac{\boldsymbol{\Phi}^3_{\mathrm{nn}}\boldsymbol{B}(\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B})^2\boldsymbol{B}^{\mathrm{H}}}{(\phi_{\mathrm{nn}}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B})^2}$	(III)
	$= \phi_{\mathrm{nn}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}$;	
$\mathcal{K}_2\mathcal{L}_1\mathcal{M}_1$	$= \phi_{\mathrm{nn}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}$;	(IV)
$\mathcal{K}_2\mathcal{L}_2\mathcal{M}_1$	$= -\boldsymbol{\Phi}^2_{\mathrm{nn}}\frac{\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}}{\phi_{\mathrm{nn}}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}}$	(V)
	$= -\phi_{\mathrm{nn}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}$;	
$\mathcal{K}_2\mathcal{L}_3\mathcal{M}_1$	$= \varepsilon\frac{\boldsymbol{\Phi}^2_{\mathrm{nn}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}}{(\phi_{\mathrm{nn}}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B})^2}$	(VI)
	$= \varepsilon\frac{\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}}{\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}}$;	
$\mathcal{K}_1\mathcal{L}_1\mathcal{M}_2$	$= \phi_{\mathrm{nn}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}$;	(VII)
$\mathcal{K}_1\mathcal{L}_2\mathcal{M}_2$	$= -\boldsymbol{\Phi}^2_{\mathrm{nn}}\frac{\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}}{\phi_{\mathrm{nn}}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}}$	(VIII)
	$= -\phi_{\mathrm{nn}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}$;	
$\mathcal{K}_1\mathcal{L}_3\mathcal{M}_2$	$= \varepsilon\frac{\boldsymbol{\Phi}^2_{\mathrm{nn}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}}{(\phi_{\mathrm{nn}}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B})^2}$	(IX)
	$= \varepsilon\frac{\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}}{\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}}$;	
$\mathcal{K}_2\mathcal{L}_1\mathcal{M}_2$	$= \varepsilon\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}$;	(X)
$\mathcal{K}_2\mathcal{L}_2\mathcal{M}_2$	$= -\varepsilon\frac{\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}}{\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}}$;	(XI)
$\mathcal{K}_2\mathcal{L}_3\mathcal{M}_2$	$= \varepsilon^2\frac{\phi_{\mathrm{nn}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}}{(\phi_{\mathrm{nn}}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B})^2}$.	(XII)

For the general case, $\boldsymbol{B} \neq \boldsymbol{A}$, we use the *matrix inversion lemma*, yielding:

$$\mathcal{L} = \frac{1}{\varepsilon}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1} - \frac{\frac{1}{\varepsilon^2}\phi_{\mathrm{nn}}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}}{1+\frac{1}{\varepsilon}\phi_{\mathrm{nn}}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}} .$$

Now, using the approximation $\frac{1}{1+\mu} \approx 1-\mu$, for $\mu \to 0$ (μ properly defined), yields,

$$\mathcal{L} = \frac{1}{\varepsilon}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1} - \frac{\frac{1}{\varepsilon}\phi_{\mathrm{nn}}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}}{\phi_{\mathrm{nn}}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}} + \frac{\phi_{\mathrm{nn}}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}}{\left(\phi_{\mathrm{nn}}\boldsymbol{B}^{\mathrm{H}}\mathcal{H}(\mathcal{H}^{\mathrm{H}}\mathcal{H})^{-1}\mathcal{H}^{\mathrm{H}}\boldsymbol{B}\right)^2} . \quad (47.A2)$$

Now, calculating $\mathcal{X}$,

$$\begin{aligned}\mathcal{X} &= \mathcal{K}\mathcal{L}\mathcal{M} = \boldsymbol{\Phi}_{NN}\mathcal{H}\mathcal{L}\mathcal{H}^{\mathrm{H}}\boldsymbol{\Phi}_{NN} \\ &= \left(\phi_{\mathrm{nn}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}+\varepsilon\mathbf{I}\right)\mathcal{H}\mathcal{L}\mathcal{H}^{\mathrm{H}}\left(\phi_{\mathrm{nn}}\boldsymbol{B}\boldsymbol{B}^{\mathrm{H}}+\varepsilon\mathbf{I}\right) \\ &\triangleq (\mathcal{K}_1+\mathcal{K}_2)\mathcal{H}(\mathcal{L}_1+\mathcal{L}_2+\mathcal{L}_3)\mathcal{H}^{\mathrm{H}}(\mathcal{M}_1+\mathcal{M}_2) , \end{aligned} \quad (47.A3)$$

with the obvious definitions of $\mathcal{K}_1$, $\mathcal{K}_2$, $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$, $\mathcal{M}_1$, and $\mathcal{M}_2$. Opening the brackets we have twelve terms given in Table 47.4.

Note, that terms I and II, terms IV and V, and terms VII and VIII eliminate each other, and that terms VI, IX, X, XI, and XII vanish as ε approaches to zero. Only term III is left, i. e., $\mathcal{X} = \phi_{\mathrm{nn}}BB^{\mathrm{H}}$. Substituting $\mathcal{X}$ into (47.119) we have

$$\boldsymbol{\Phi}_{\mathrm{yy}}^{\mathrm{n}} = \frac{\tilde{\boldsymbol{A}}^{\mathrm{H}}\boldsymbol{\Phi}_{N_{\mathrm{s}}N_{\mathrm{s}}}\tilde{\boldsymbol{A}}}{\|\tilde{\boldsymbol{A}}\|^4} - \frac{\tilde{\boldsymbol{A}}^{\mathrm{H}}\left(\phi_{\mathrm{nn}}BB^{\mathrm{H}}\right)\tilde{\boldsymbol{A}}}{\|\tilde{\boldsymbol{A}}\|^4} = 0 . \quad (47.A4)$$

47.B Appendix: Equivalence Between Maximum SNR and LCMV Beamformers

The LCMV beamformer is given by

$$W^{\mathrm{LCMV}} = \frac{\left(\phi_{\mathrm{ss}}\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}+\boldsymbol{\Phi}_{NN}\right)^{-1}\boldsymbol{A}}{\boldsymbol{A}^{\mathrm{H}}\left(\phi_{\mathrm{ss}}\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}+\boldsymbol{\Phi}_{NN}\right)^{-1}\boldsymbol{A}} . \quad (47.B1)$$

Using the matrix inversion lemma we have in the denominator

$$\begin{aligned}&\boldsymbol{A}^{\mathrm{H}}\left(\boldsymbol{\Phi}_{NN}+\phi_{\mathrm{ss}}\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}\right)^{-1}\boldsymbol{A} \\ &= \boldsymbol{A}^{\mathrm{H}}\left(\boldsymbol{\Phi}_{NN}^{-1}-\frac{\phi_{\mathrm{ss}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}}{1+\phi_{\mathrm{ss}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}}\right)\boldsymbol{A} \\ &= \boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}-\boldsymbol{A}^{\mathrm{H}}\frac{\phi_{\mathrm{ss}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}}{1+\phi_{\mathrm{ss}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}}\boldsymbol{A} \\ &= \boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}-\phi_{\mathrm{ss}}\frac{\left(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}\right)^2}{1+\phi_{\mathrm{ss}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}} \\ &= \frac{\left(1+\phi_{\mathrm{ss}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}\right)\left(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}\right)}{1+\phi_{\mathrm{ss}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}} \\ &\quad -\frac{\phi_{\mathrm{ss}}\left(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}\right)^2}{1+\phi_{\mathrm{ss}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}} \\ &= \frac{\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}}{1+\phi_{\mathrm{ss}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}} \end{aligned} \quad (47.B2)$$

Similarly, we have in the numerator

$$\begin{aligned}&\left(\phi_{\mathrm{ss}}\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}+\boldsymbol{\Phi}_{NN}\right)^{-1}\boldsymbol{A} \\ &= \boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}-\frac{\phi_{\mathrm{ss}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}}{1+\phi_{\mathrm{ss}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}} \\ &= \frac{\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}\left(1+\phi_{\mathrm{ss}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}\right)}{1+\phi_{\mathrm{ss}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}} \\ &\quad -\frac{\phi_{\mathrm{ss}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}}{1+\phi_{\mathrm{ss}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}} \\ &= \frac{\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}}{1+\phi_{\mathrm{ss}}\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}} . \end{aligned} \quad (47.B3)$$

Dividing (47.B3) by (47.B2) we have

$$W^{\mathrm{LCMV}} = \frac{\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}}{\boldsymbol{A}^{\mathrm{H}}\boldsymbol{\Phi}_{NN}^{-1}\boldsymbol{A}} \equiv W^{\mathrm{MSNR}} . \quad (47.B4)$$

References

47.1 S. Haykin: *Adaptive Filter Theory*, 4th edn. (Prentice Hall, Upper Saddle River 2002)

47.2 B. Widrow, S.D. Stearns: *Adaptive Signal Processing* (Prentice-Hall, Upper Saddle River 1985)

47.3 D. Monzingo, T. Miller: *Introduction to Adaptive Arrays* (Wiley, New York 1980)

47.4 H.L.V. Trees: *Detection, Estimation, and Modulation Theory*, Vol. IV (Wiley, New York 2002)

47.5 B.V. Veen, K. Buckley: Beamforming: A versatile approach to spatial filtering, IEEE Acoust. Speech Signal Process. Mag. **5**(2), 4–24 (1988)

47.6 H. Krim, M. Viberg: Two decades of array signal processing research: The parametric approach, IEEE Signal Proc. Mag. **13**, 67–94 (1996)

47.7 H. Cox, R. Zeskind, M. Owen: Robust Adaptive Beamforming, IEEE Trans. Acoust. Speech Signal Process. **35**(10), 1365–1376 (1987)

47.8 S.L. Gay, J. Benesty: *Acoustic Signal Processing for Telecommunication* (Kluwer Academic, Dordrecht 2001)

47.9 M.S. Brandstein, D.B. Ward: *Microphone Arrays: Signal Processing Techniques and Applications* (Springer, Berlin, Heidelberg 2001)

47.10 J. Benesty, Y. Huang (Eds.): *Adaptive Signal Processing: Applications to Real-World Problems* (Springer, Berlin, Heidelberg 2003)

47.11 E. Hänsler, G. Schmidt: *Acoustic Echo and Noise Control: A Pracical Approach* (Wiley, New York 2004)

47.12 J. Benesty, S. Makino, J. Chen: *Speech Enhancement* (Springer, Berlin, Heidelberg 2005)

47.13 W. Herbordt: *Sound Capture for Human/Machine Interfaces – Practical Aspects of Microphone Array Signal Processing* (Springer, Berlin, Heidelberg 2005)

47.14 S. Gannot, J. Benesty, J. Bitzer, I. Cohen, S. Doclo, R. Martin, S. Nordholm: Advances in multimicrophone speech processing, EURASIP J. Appl. Signal Process. **2006**(ID46357) (2006)

47.15 E. Jan, J. Flanagan: Microphone arrays for speech processing, International Symposium on Signals, Systems, and Electronics (ISSSE) (URSI, San Francisco 1995) pp. 373–376

47.16 E. Jan, J. Flanagan: Sound capture from spatial volumes: matched-filter processing of microphone arrays having randomly-distributed sensors, Proc. ICASSP (Atlanta, 1996) pp. 917–920

47.17 D. Rabinkin, R. Renomeron, J. Flanagan, D. Macomber: Optimal truncation time for matched filter array processing, Proc. ICASSP (Seattle, 1998) pp. 3269–3272

47.18 M. Sondhi, G. Elko: Adaptive optimization of microphone arrays under a nonlinear constraint, Proc. ICASSP, Vol. 11 (Tokyo, 1986) pp. 981–984

47.19 Y. Kaneda, J. Ohga: Adaptive microphone-array system for noise reduction, IEEE Trans. Acoust. Speech Signal Process. **34**(6), 1391–1400 (1986)

47.20 D.V. Compernolle: Switching adaptive filters for enhancing noisy and reverberant speech from microphone array recordings, Proc. ICASSP (Albuquerque, 1990) pp. 833–836

47.21 W. Kellermann: Acoustic echo cancellation for beamforming microphone arrays. In: *Microphone Arrays: Signal Processing Techniques and Applications*, ed. by M.S. Brandstein, D.B. Ward (Springer, Berlin, Heidelberg 2001) pp. 281–306

47.22 I. Claesson, S. Nordholm: A spatial filtering approach to robust adaptive beamforming, IEEE Trans. Antennas Propag. **40**(1), 1093–1096 (1992)

47.23 S. Nordholm, I. Claesson, N. Grbić: Optimal and adaptive microphone arrays for speech input in automobiles. In: *Microphone Arrays: Signal Processing Techniques and Applications*, ed. by M.S. Brandstein, D.B. Ward (Springer, Berlin, Heidelberg 2001) pp. 307–330

47.24 R. Martin: Small microphone arrays with postfilters for noise and acoustic echo reduction. In: *Microphone Arrays: Signal Processing Techniques and Applications*, ed. by M.S. Brandstein, D.B. Ward (Springer, Berlin, Heidelberg 2001) pp. 255–280

47.25 B. Widrow, J.R. Glover Jr., J.M. McCool, J. Kaunitz, C.S. Williams, R.H. Hearn, J.R. Zeidler, E. Dong Jr, R.C. Goodlin: Adaptive noise cancelling: Principals and applications, Proc. IEEE **63**(12), 1692–1716 (1975)

47.26 S. Doclo, M. Moonen: GSVD-based optimal filtering for single and multimicrophone speech enhancement, IEEE Trans. Speech Audio Process. **50**(9), 2230–2244 (2002)

47.27 S. Doclo, M. Moonen: Design of far-field and near-field broadband beamformers using eigenfilters, Signal Process. **83**(12), 2641–2673 (2003)

47.28 S. Doclo, M. Moonen: Multi-microphone noise reduction using recursive GSVD-based optimal filtering with ANC postprocessing stage, IEEE Trans. Speech Audio Process. **13**(1), 53–69 (2005)

47.29 A. Spriet, M. Moonen, J. Wouters: A multichannel subband GSVD approach for speech enhancement, Eur. Trans. Telecommun. **13**(2), 149–158 (2002), Special Issue on Acoustic Echo and Noise Control

47.30 G. Rombouts, M. Moonen: QRD-based unconstrained optimal filtering for acoustic noise reduction, Signal Process. **83**(9), 1889–1904 (2003)

47.31 G. Rombouts, M. Moonen: Fast QRD-lattice-based unconstrained optimal filtering for acoustic noise reduction, IEEE Trans. Speech Audio Process. **13**(6), 1130–1143 (2005)

Amazon EX1010 Page 976

47.32 J. Chen, J. Benesty, Y. Huang, S. Doclo: New insights into the noise reduction Wiener filter, IEEE Trans. Speech Audio Process. **14**(4), 1218–1234 (2006)

47.33 A. Spriet, M. Moonen, J. Wouters: Spatially pre-processed speech distortion weighted multi-channel Wiener filtering for noise reduction, Signal Process. **84**(12), 2367–2387 (2004)

47.34 S. Doclo, A. Spriet, J. Wouters, M. Moonen: Speech distortion weighted multichannel wiener filtering techniques for noise reduction. In: *Speech Enhancement*, Signals and Communication Technology, ed. by J. Benesty, S. Makino, J. Chen (Springer, Berlin, Heidelberg 2005) pp. 199–228

47.35 O.L. Frost: An algorithm for linearly constrained adaptive array processing, Proc. IEEE **60**(8), 926–935 (1972)

47.36 L.J. Griffiths, C.W. Jim: An alternative approach to linearly constrained adaptive beamforming, IEEE Trans. Antennas Propag. **30**(1), 27–34 (1982)

47.37 S. Affes, Y. Grenier: A source subspace tracking array of microphones for double talk situations, Proc. ICASSP (Munich, 1997) pp. 269–272

47.38 S. Gannot, D. Burshtein, E. Weinstein: Signal enhancement using beamforming and nonstationarity with applications to speech, IEEE Trans. Signal Proces. **49**(8), 1614–1626 (2001)

47.39 S. Nordholm, Y.H. Leung: Performance limits of the broadband generalized sidelobe cancelling structure in an isotropic noise field, J. Acoust. Soc. Am. **107**(2), 1057–1060 (2000)

47.40 J. Bitzer, K.U. Simmer, K.D. Kammeyer: Theoretical noise reduction limits of the generalized sidelobe canceller (GSC) for speech enhancement, Proc. ICASSP (Phoenix, 1999) pp. 2965–2968

47.41 J. Bitzer, K. Simmer, K. Kammeyer: Multichannel noise reduction – algorithms and theoretical limits, Proc. EUSIPCO, Vol. I (Rhodes, 1998) pp. 105–108

47.42 J. Bitzer, K.-D. Kammeyer, K. Simmer: An alternative implementation of the superdirective beamformer, Workshop on Application of Signal Processing to Audio and Acoustics (IEEE, New Paltz, NY 1999)

47.43 C. Marro, Y. Mahieux, K. Simmer: Analysis of noise reduction and dereverberation techniques based on microphone arrays with postfiltering, IEEE Trans. Speech Audio Process. **6**(3), 240–259 (1998)

47.44 S. Nordholm, I. Claesson, M. Dahl: Adaptive microphone array employing calibration signals: an analytical evaluation, IEEE Trans. Speech Audio Process. **7**(3), 241–252 (1999)

47.45 K.-C. Huarng, C.-C. Yeh: Performance analysis of derivative constraint adaptive arrays with pointing errors, IEEE Trans. Acoust. Speech Signal Process. **38**(2), 209–219 (1990)

47.46 S. Nordholm, I. Claesson, P. Eriksson: The broadband wiener solution for griffiths-jim beamformers, IEEE Trans. Signal Process. **40**(2), 474–478 (1992)

47.47 H. Cox: Resolving power and sensitivity to mismatch of optimum array processors, J. Acoust. Soc. Am. **54**(3), 771–785 (1973)

47.48 D. Brandwood: A complex gradient operator and its application in adaptive array theory, Proc. IEEE **130**(1 Parts F and H), 11–16 (1983)

47.49 G. Strang: *Linear Algebra and its Application*, 2nd edn. (Academic, New York 1980)

47.50 R. Crochiere: A weighted overlap-add method for short-time Fourier analysis/synthesis, IEEE Trans. Acoust. Speech Signal Process. **28**(1), 99–102 (1980)

47.51 J. Shynk: Frequency-domain and multirate and adaptive filtering, IEEE Signal Process. Mag. **9**(1), 14–37 (1992)

47.52 J. Allen, D. Berkley: Image method for efficiently simulating small-room acoustics, J. Acoust. Soc. Am. **65**(4), 943–950 (1979)

47.53 S. Doclo, M. Moonen: Combined frequency-domain dereverberation and noise reduction technique for multi-microphone speech enhancement, Int. Workshop Acoust. Echo Noise Control (IWAENC) (Darmstadt 2001) pp. 31–34

47.54 B. Yang: Projection approximation subspace tracking, IEEE Trans. Signal Process. **43**(1), 95–107 (1995)

47.55 O. Hoshuyama, A. Sugiyama: Robust adaptive beamforming. In: *Microphone Arrays*, ed. by M. Brandstein, D. Ward (Springer, Berlin, Heidelberg 2001) pp. 87–109

47.56 E. Weinstein, M. Feder, A. Oppenheim: Multichannel signal separation by decorrelation, IEEE Trans. Speech Audio Process. **1**(4), 405–413 (1993)

47.57 S. Gannot: *Array Processing of Nonstationary Signals with Application to Speech* (Tel-Aviv University, Tel-Aviv 2000), available on line, http://www.biu.ac.il/ gannot

47.58 C. Knapp, G. Carter: The generalized correlation method for estimation of time delay, IEEE Trans. Acoust. Speech Signal Process. **24**(4), 320–327 (1976)

47.59 T. Dvorkind, S. Gannot: Time difference of arrival estimation of speech source in a noisy and reverberant environment, Signal Process. **85**(1), 177–204 (2005)

47.60 J. Chen, J. Benesty, Y. Huang: Time delay estimation in room acoustic environments: an overview, EURASIP J. Appl. Signal Process. **26**(503), 19 (2006)

47.61 I. Cohen: Relative transfer function identification using speech signals, IEEE Trans. Speech Audio Process. **12**(5), 451–459 (2004)

47.62 I. Cohen: Identification of speech source coupling between sensors in reverberant noisy environments, IEEE Signal Process. Lett. **11**(7), 613–616 (2004)

47.63 O. Shalvi, E. Weinstein: System identification using nonstationary signals, IEEE Trans. Signal Process. **44**(8), 2055–2063 (1996)

47.64 D.G. Manolakis, V.K. Ingle, S.M. Kogan: *Statistical and Adaptive Signal Processing: Spectral Estima-*

Amazon EX1010 Page 977

tion, Signal Modeling, Adaptive Filtering and Array Processing (McGraw-Hill, Singapore 2000)
47.65 R. Martin: Noise power spectral density estimation based on optimal smoothing and minimum statistics, IEEE Trans. Speech Audio Process. **9**(5), 504–512 (2001)
47.66 I. Cohen, B. Berdugo: Noise estimation by minima controlled recursive averaging for robust speech enhancement, IEEE Signal Process. Lett. **9**(1), 12–15 (2002)
47.67 I. Cohen: Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging, IEEE Trans. Speech Audio Process. **11**(5), 466–475 (2003)
47.68 S. Nordholm, H.Q. Dam, N. Grbić, S.Y. Low: Adaptive microphone array employing spatial quadratic soft constraints and spectral shaping. In: *Speech Enhancement*, Signals and Communication Technology, ed. by J. Benesty, S. Makino, J. Chen (Springer, Berlin, Heidelberg 2005) pp. 229–246
47.69 O. Hoshuyama, A. Sugiyama, A. Hirano: A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters, IEEE Trans. Signal Process. **47**(10), 2677–2684 (1999)
47.70 C. Fancourt, L. Parra: The generalized sidelobe decorrelator, Proc. IEEE Workshop on Applications of Signal Process. to Audio and Acoustics (New Paltz, NY 2001) pp. 167–170
47.71 W. Herbordt, W. Kellermann: Computationally efficient frequency-domain robust generalized sidelobe canceller, Int. Workshop Acoust. Echo Noise Control (IWAENC) (Darmstadt 2001) pp. 51–54
47.72 W. Herbordt, W. Kellermann: Adaptive beamforming for audio signal acquisition. In: *Adaptive Signal Processing: Applications to Real-World Problems*, Signals and Communication Technology, ed. by J. Benesty, Y. Huang (Springer, Berlin, Heidelberg 2003) pp. 155–194
47.73 S. Doclo, M. Moonen: On the output SNR of the speech-distortion weighted multichannel Wiener filter, IEEE Signal Process. Lett. **11**(12), 809–811 (2005)
47.74 A. Spriet, M. Moonen, J. Wouters: Stochastic gradient-based implementation of spatially preprocessed speech distortion weighted multichannel wiener filtering for noise reduction in hearing aids, IEEE Trans. Signal Process. **53**(3), 911–925 (2005)
47.75 A. Spriet, M. Moonen, J. Wouters: Robustness analysis of multichannel wiener filtering and generalized sidelobe cancellation for multimicrophone noise reduction in hearing aid applications, IEEE Trans. Speech Audio Process. **13**(4), 487–503 (2005)
47.76 S. Low, S. Nordholm, R. Togneri: Convolutive blind signal separation with post-processing, IEEE Trans. Speech Audio Process. **12**(5), 539–548 (2004)
47.77 S. Gannot, D. Burshtein, E. Weinstein: Analysis of the power spectral deviation of the general transfer function GSC, IEEE Trans. Signal Process. **52**(4), 1115–1121 (2004)
47.78 K.U. Simmer, J. Bitzer, C. Marro: Post-filtering techniques. In: *Microphone Arrays: Signal Processing Techniques and Applications*, ed. by M.S. Brandstein, D.B. Ward (Springer, Berlin, Heidelberg 2001) pp. 39–60
47.79 R. Zelinski: A Microphone array with adaptive post-filtering for noise reduction in reverberant rooms, IEEE ICASSP (New York, 1988) pp. 2578–2581
47.80 R. Zelinski: Noise reduction based on microphone array with LMS adaptive post-filtering, Electron. Lett. **26**(24), 2036–2581 (1990)
47.81 S. Fischer, K.U. Simmer: An adaptive microphone array for hands-free communication, Int. Workshop Acoust. Echo Noise Control (IWAENC) (Røros, 1995) pp. 44–47
47.82 S. Fischer, K.U. Simmer: Beamforming microphone arrays for speech acquisition in noisy environments, Speech Commun. **20**(3–4), 215–227 (1996)
47.83 S. Fischer, K.-D. Kammeyer: Broadband beamforming with adaptive postfiltering for speech acquisition in noisy environment, Proc. ICASSP, Vol. 1 (Munich, 1997) pp. 359–362
47.84 J. Meyer, K.U. Simmer: Multi-channel speech enhancement in a car environment using Wiener filtering and spectral subtraction, Proc. ICASSP (Munich, 1997) pp. 21–24
47.85 K.U. Simmer, S. Fischer, A. Wasiljeff: Suppression of coherent and incoherent noise using a microphone array, Annales des Télécommun. **49**(7–8), 439–446 (1994)
47.86 J. Bitzer, K. Simmer, K.-D. Kammeyer: Multimicrophone noise reduction by post-filter and superdirective beamformer, Int. Workshop Acoust. Echo Noise Control (IWAENC) (Pocono Manor, 1999) pp. 100–103
47.87 J. Bitzer, K.U. Simmer, K.-D. Kammeyer: Multimicrophone noise reduction techniques as front-end devices for speech recognition, Speech Commun. **34**, 3–12 (2001)
47.88 K.U. Simmer, A. Wasiljeff: Adptive microphone arrays for noise supression in the frequency domain, Proc. Second Cost 229 Workshop on Adaptive Algorithms in Communications (Bordeaux, 1992) pp. 185–194
47.89 I. McCowan, H. Bourlard: Microphone array post-filter for diffuse noise field, Proc. ICASSP, Vol. 1. (Orlando, 2002) pp. 905–908
47.90 I. McCowan, H. Bourlard: Microphone array post-filter based on noise field coherence, IEEE Trans. Speech Audio Process. **11**(6), 709–716 (2003)
47.91 I. Cohen, S. Gannot, B. Berdugo: An integrated real-time beamforming and postfiltering system for non-stationary noise environments, EURASIP J. Appl. Signal Process. **2003**(11), 1064–1073 (2003), special issue on Signal Processing for Acoustic Communication Systems

Amazon EX1010 Page 978

47.92 I. Cohen, B. Berdugo: Microphone array post-filtering for non-stationary noise suppression, Proc. ICASSP (Orlando, 2002) pp. 901–904
47.93 I. Cohen, B. Berdugo: Speech enhancement for non-stationary noise environments, Signal Process. **81**(11), 2403–2418 (2001)
47.94 Y. Ephraim, D. Malah: Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator, IEEE Trans. Acoust. Speech Signal Process. **ASSP-32**(6), 1109–1121 (1984)
47.95 Y. Ephraim, D. Malah: Speech enhancement using a minimum mean square error log-spectral amplitude estimator, IEEE Trans. Acoust. Speech Signal Process. **33**(2), 443–445 (1985)
47.96 I. Cohen: Analysis of two-channel generalized sidelobe canceller (GSC) with post-filtering, IEEE Trans. Speech Audio Process. **11**(6), 684–699 (2003)
47.97 S. Gannot, D. Burshtein, E. Weinstein: Theoretical analysis of the general transfer function GSC, Int. Workshop Acoust. Echo Noise Control (IWAENC) (Darmstadt 2001) pp. 27–30
47.98 N. Dal-Degan, C. Prati: Acoustic noise analysis and speech enhancement techniques for mobile radio application, Signal Process. **15**(4), 43–56 (1988)
47.99 S. Gannot, I. Cohen: Audio Sample Files http://www.biu.ac.il/ gannot (2005)

48. Feedback Control in Hearing Aids

A. Spriet, S. Doclo, M. Moonen, J. Wouters

Acoustic feedback limits the maximum amplification that can be used in a hearing aid without making it unstable. This chapter gives an overview of existing techniques for feedback suppression and, in particular, adaptive feedback cancellation. Because of the presence of a closed signal loop, standard adaptive filtering techniques for open-loop systems fail to provide a reliable feedback path estimate if the desired signal is spectrally colored. Several approaches for improving the estimation accuracy of the adaptive feedback canceller will be reviewed and evaluated for acoustic feedback paths measured in a commercial behind-the-ear hearing aid.

This chapter is organized as follows. Section 48.1 gives a mathematical formulation of the acoustic feedback problem in hearing aids and briefly describes the two possible approaches to reduce its negative effects, i. e., feedforward suppression and feedback cancellation. In addition, performance measures for feedback cancellation are defined.

Section 48.2 discusses the standard continuous adaptation feedback (CAF) cancellation algorithm that is widely studied for application in hearing aids. We demonstrate that the standard CAF suffers from a model error or bias when the desired signal is spectrally colored (e.g., a speech signal). In the literature, several solutions have been proposed to reduce the bias of the CAF. A common approach is to incorporate signal decorrelating operations (such as delays) in the signal processing path of the hearing aid or to reduce the adaptation speed of the adaptive feedback canceller. Other techniques, discussed in Sect. 48.3, exploit prior knowledge of the acoustic feedback path to improve the adaptation of the feedback canceller. In Sect. 48.4, a final class of techniques is presented that view the feedback path as a part of a closed-loop system and apply closed-loop system identification theory [48.1]. Among the different closed-loop identification methods, especially the direct method is an appealing approach for feedback cancellation. In contrast to the other methods, this technique does not require the use of an external probe signal. The direct method reduces the bias of the feedback canceller by incorporating a (stationary or time-varying) model of the desired signal $x[k]$ in the identification.

Finally, Sect. 48.5 compares the steady-state performance as well as the tracking performance of different algorithms for acoustic feedback paths measured in a commercial behind-the-ear hearing aid.

48.1 **Problem Statement** 980
48.1.1 Acoustic Feedback in Hearing Aids. 980
48.1.2 Feedforward Suppression Versus Feedback Cancellation 981
48.1.3 Performance of a Feedback Canceller 982

48.2 **Standard Adaptive Feedback Canceller** 982
48.2.1 Adaptation of the CAF 982
48.2.2 Bias of the CAF 984
48.2.3 Reducing the Bias of the CAF 985

48.3 **Feedback Cancellation Based on Prior Knowledge of the Acoustic Feedback Path** 986
48.3.1 Constrained Adaptation (C-CAF)..... 986
48.3.2 Bandlimited Adaptation (BL-CAF) .. 988

48.4 **Feedback Cancellation Based on Closed-Loop System Identification** 990
48.4.1 Closed-Loop System Setup..... 990
48.4.2 Direct Method..... 991
48.4.3 Desired Signal Model 992
48.4.4 Indirect and Joint Input–Output Method.... 994

48.5 **Comparison** 995
48.5.1 Steady-State Performance..... 995
48.5.2 Tracking Performance 995
48.5.3 Measurement of the Actual Maximum Stable Gain..... 995

48.6 **Conclusions** 997

References 997

Amazon EX1010 Page 980

Acoustic feedback in hearing aids refers to the acoustical coupling between the loudspeaker and the microphone of the hearing aid. Because of this coupling, the hearing aid produces a severe distortion of the desired signal and an annoying howling sound when the gain is increased. With the growing use of open fittings, i. e., without an earmold, and the decreasing distance between the loudspeaker and the microphone of the hearing aid, the demand for effective feedback suppression or cancellation techniques increases.

This chapter gives an overview of existing techniques for feedback suppression and, in particular, adaptive feedback cancellation. Because of the presence of a closed signal loop, standard adaptive filtering techniques for open-loop systems (as used in echo cancellation) fail to provide a reliable feedback path estimate if the desired signal is spectrally colored. Several approaches for improving the estimation accuracy of the adaptive feedback canceller will be discussed.

48.1 Problem Statement

Section 48.1.1 defines the notation and gives a mathematical formulation of the acoustic feedback problem in hearing aids. Section 48.1.2 describes the two approaches that can be used to reduce acoustic feedback, i. e., feedforward suppression and feedback cancellation. In Sect. 48.1.3, we define the performance measures that will be used to evaluate the feedback cancellation techniques.

48.1.1 Acoustic Feedback in Hearing Aids

Notation

The superscripts T and H denote matrix/vector transpose and complex conjugate transpose, respectively. The expectation operator is denoted by $\mathcal{E}\{\cdot\}$. The matrix $\mathcal{F}$ is the $M \times M$ discrete Fourier transform (DFT) matrix. The matrix $\mathbf{I}_L$ is the $L \times L$ identity matrix; the matrix $\mathbf{0}_L$ is an $L \times L$ dimensional matrix of zeros.

The discrete-time index is denoted by k. The symbol q^{-1} denotes the discrete-time delay operator, i. e.,

$$q^{-1}u[k] = u[k-1] \,. \tag{48.1}$$

A discrete-time filter with coefficient vector $\boldsymbol{f}[k]$

$$\boldsymbol{f}[k] = \left[f_0[k] \;\; f_1[k] \;\; \cdots \;\; f_{L_F-1}[k]\right]^{\mathrm{T}} \tag{48.2}$$

and filter length L_F is represented as a polynomial transfer function $F(q,k)$ in q, i. e.,

$$F(q,k) = \boldsymbol{f}^{\mathrm{T}}[k]\boldsymbol{q} \,, \tag{48.3}$$

with $\boldsymbol{q} = \left[1 \;\; q^{-1} \;\; \ldots \;\; q^{-L_F+1}\right]^{\mathrm{T}}$. This representation, which is adopted from [48.2], allows the following notation for the filtering of $u[k]$ with $F(q,k)$:

$$F(q,k)u[k] = \boldsymbol{f}^{\mathrm{T}}[k]\boldsymbol{u}[k] \,, \tag{48.4}$$

with

$$\boldsymbol{u}[k] = \left[u[k] \;\; u[k-1] \;\; \cdots \;\; u[k-L_F+1]\right]^{\mathrm{T}} \,. \tag{48.5}$$

The spectrum of $F(q,k)$ is denoted by $F(\mathrm{e}^{\mathrm{i}\omega},k)$ with $\omega \in [0,\pi]$ being the normalized angular frequency.

Acoustic Feedback

Figure 48.1 illustrates the problem of acoustic feedback for a hearing aid with a single microphone. The so-called *forward path* $G(q,k)$ represents the regular signal processing path of the hearing aid, typically including a frequency-specific gain, compression, and/or noise reduction. We assume that $G(q,k)$ has a delay d_G of at least one sample, i. e., $g_0 = 0$.

The *feedback path* between the loudspeaker and the microphone is denoted by $F(q,k)$. It includes the frequency characteristic of the loudspeaker and the microphone, the acoustic feedback path, and the characteristics of the digital-to-analog (DAC) and analog-to-digital (ADC) converters. The feedback path $F(q,k)$ typically contains a delay d_F that arises from the processing delay of the ADC and DAC convert-

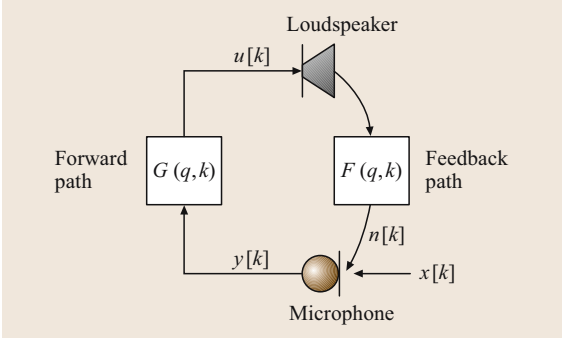


Fig. 48.1 Acoustic feedback

Amazon EX1010 Page 981

ers and the propagation time of the sound from the loudspeaker to the microphone. For simplicity, we will include this delay d_F in the delay d_G of the forward path $G(q,k)$. The loudspeaker and microphone signal are $u[k]$ and $y[k]$, respectively. The desired sound signal (i. e., without acoustic feedback) is denoted by $x[k]$ and the feedback signal is denoted by $n[k] = F(q,k)u[k]$.

Ideally, the loudspeaker signal $u[k]$ equals the desired signal $x[k]$ processed by the forward path $G(q,k)$, i. e.,

$$u[k] = G(q,k)x[k]\,. \tag{48.6}$$

However, because of acoustic feedback, the amplified sound $u[k]$ sent through the loudspeaker is fed back into the microphone, resulting in a closed-loop system $C(q,k)$

$$C(q,k) = \frac{G(q,k)}{1 - G(q,k)F(q,k)} \tag{48.7}$$

from $x[k]$ to the loudspeaker signal $u[k]$.

Instability occurs if the loop gain $|G(\mathrm{e}^{\mathrm{i}\omega},k)F(\mathrm{e}^{\mathrm{i}\omega},k)|$ exceeds one at an angular frequency $\omega \in [0,\pi]$ with positive feedback, i. e., an angular frequency where the loop phase $\angle(G(\mathrm{e}^{\mathrm{i}\omega},k)F(\mathrm{e}^{\mathrm{i}\omega},k))$ equals a multiple of 2π. If the gain $|G(\mathrm{e}^{\mathrm{i}\omega},k)|$ is increased beyond this point, the system starts to oscillate. However also at low gain margins, acoustic feedback already degrades the sound quality by introducing ringing or howling. To avoid significant audible distortion, a gain margin of at least 6 dB is advisable: the effect of the acoustic feedback on the closed-loop frequency response $C(\mathrm{e}^{\mathrm{i}\omega},k)$ is then limited to [48.3]

$$\frac{2}{3}G(\mathrm{e}^{\mathrm{i}\omega},k) \le C(\mathrm{e}^{\mathrm{i}\omega},k) \le 2G(\mathrm{e}^{\mathrm{i}\omega},k). \tag{48.8}$$

As an illustration, Fig. 48.2 depicts the feedback path frequency response measured in a commercial (GN ReSound) behind-the-ear hearing aid. During the measurement, the hearing aid was attached to the right ear of a Cortex MK II artificial head. An earmold with a vent size of 2 mm was used. In this example, the minimum feedback path attenuation, and hence, the maximum amplification without instability lies around 20 dB. The acoustic feedback path when a mobile phone is attached to the head is also shown. This demonstrates that placing a telephone handset close to the ear can temporarily reduce the feedback path attenuation by another 10–20 dB [48.4–6].

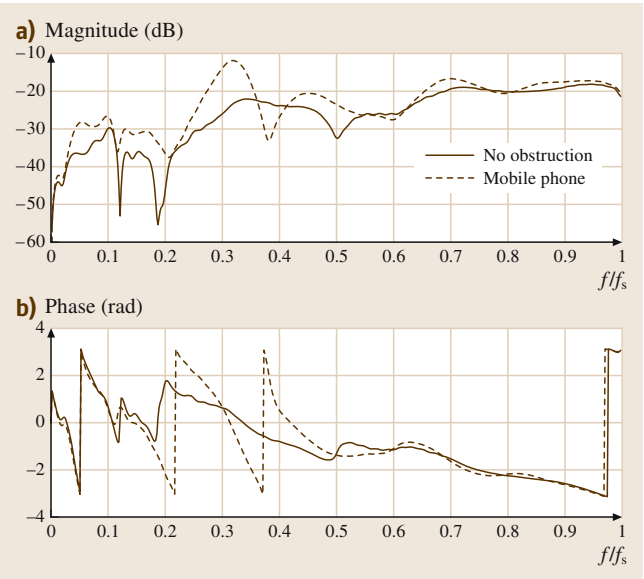


Fig. 48.2 Feedback path frequency response of a commercial, GN ReSound behind-the-ear hearing aid connected to a vented earmold (with a vent size of 2 mm) for two conditions, i. e., without obstruction and with a mobile phone attached to the ear

48.1.2 Feedforward Suppression Versus Feedback Cancellation

To reduce the negative effects introduced by acoustic feedback, several techniques have been proposed in the literature. They can be broadly classified into feedforward suppression and feedback cancellation techniques.

Feedforward Suppression

In *feedforward suppression* techniques, the regular signal processing path $G(q,k)$ of the hearing aid is modified in such a way that it is stable in conjunction with the feedback path $F(q,k)$. The most common technique is the use of a notch filter. In a notch filter, the gain is reduced in a narrow frequency band around the critical frequencies, whenever feedback occurs [48.7–10]. The main disadvantage is that this approach is reactive: in order to identify the oscillation frequencies, howling first has to occur. Other examples include equalizing the phase of the open-loop response [48.11] and using time-varying elements (such as frequency shifting, delay and phase modulation) in the forward path $G(q,k)$ [48.7, 12, 13].

The achievable increase in maximum stable gain with feedforward suppression techniques was generally found to be limited [48.7, 12]. In addition, feedforward suppression techniques all compromise the basic fre-

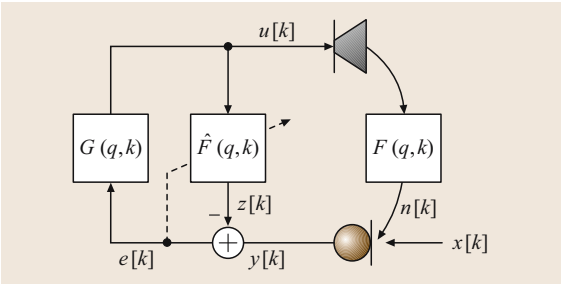


Fig. 48.3 Adaptive feedback cancellation

quency response of the hearing aid, and hence, may seriously affect the sound quality.

Feedback Cancellation

A more-promising solution for acoustic feedback is the use of a *feedback cancellation* algorithm, which is illustrated in Fig. 48.3. The feedback canceller $\hat{F}(q,k)$ produces an estimate $z[k]$ of the feedback signal $n[k]$ and subtracts this estimate $z[k]$ from the microphone signal $y[k]$. As a result, the closed-loop transfer function $C(q,k)$ from the desired signal $x[k]$ to the loudspeaker signal $u[k]$ is modified to

$$C(q,k) = \frac{G(q,k)}{1 - G(q,k)[F(q,k) - \hat{F}(q,k)]} \,. \qquad (48.9)$$

Ideally, $\hat{F}(q,k) = F(q,k)$ such that the closed-loop system $C(q,k)$ equals the desired hearing aid response $G(q,k)$.

The acoustic path $F(q,k)$ between the loudspeaker and the microphone can vary significantly depending on the acoustical environment (see Fig. 48.2) [48.4–6, 14]. Hence, adaptive feedback cancellers $\hat{F}(q,k)$ are called for. In the remainder of this chapter, existing adaptive feedback cancellation techniques will be discussed.

48.1.3 Performance of a Feedback Canceller

The most common measures to assess the performance of feedback cancellation algorithms are the misalignment between the true and estimated feedback path $\hat{F}(q,k)$ and the maximum stable gain [48.14–18].

The *misalignment* $\zeta_k(F,\hat{F})$ between the true feedback path $F(q,k)$ and the feedback path estimate $\hat{F}(q,k)$ is computed in the frequency domain as

$$\zeta_k(F,\hat{F}) = \sqrt{\frac{\int_0^\pi |F(\mathrm{e}^{\mathrm{i}\omega},k) - \hat{F}(\mathrm{e}^{\mathrm{i}\omega},k)|^2 \,\mathrm{d}\omega}{\int_0^\pi |F(\mathrm{e}^{\mathrm{i}\omega},k)|^2 \,\mathrm{d}\omega}} \,. \qquad (48.10)$$

The misalignment reflects the accuracy of the feedback path estimate $\hat{F}(q,k)$.

To get an idea of the actual benefit of a feedback cancellation algorithm for the hearing aid user, the *maximum stable gain* (MSG) is often used [48.14,16]. In this chapter, the MSG is defined as the largest gain for which the hearing aid does not oscillate and the desired signal is not substantially degraded by algorithm artifacts (e.g., signal distortion or howling) [48.16]. The MSG is determined by gradually and slowly increasing the gain while the feedback canceller is running.

48.2 Standard Adaptive Feedback Canceller

Adaptive feedback cancellers can be divided into two classes: algorithms with a continuous adaptation and algorithms with a *noncontinuous adaptation* [48.9,16,19]. The latter only adapt the coefficients of the feedback canceller when instability is detected or when the input signal level is low [48.19–21]. Due to this reactive, rather than proactive, adaptation, such systems may be objectionable. A *continuous adaptation* feedback canceller, on the other hand, continuously adapts the coefficients of the feedback canceller.

This section describes a standard continuous adaptation feedback (CAF) cancellation technique that is widely studied for hearing-aid applications [48.9,16,22]. In Sect. 48.2.2, we show that, because of the presence of a closed signal loop, the standard CAF suffers from a large model error or bias if the desired signal is spectrally colored. Classical approaches for reducing the bias of the CAF are discussed in Sect. 48.2.3.

48.2.1 Adaptation of the CAF

The CAF continuously adapts the coefficients

$$\hat{\boldsymbol{f}}[k] = \left[f_0[k] \;\; f_1[k] \;\; \cdots \;\; f_{L_{\hat{F}}-1}[k] \right]^{\mathrm{T}} \qquad (48.11)$$

of the feedback canceller based on standard adaptive filtering (Wiener filtering) procedures (Fig. 48.3). The CAF minimizes the energy of the feedback compensated

Amazon EX1010 Page 983

signal $e[k]$, i. e.,

$$J(\hat{\boldsymbol{f}}[k]) = \mathcal{E}\{|y[k] - \hat{\boldsymbol{f}}^{\mathrm{T}}[k]\boldsymbol{u}[k]|^2\}\,, \tag{48.12}$$

with

$$\boldsymbol{u}[k] = \left[u[k] \;\; \cdots \;\; u[k - L_{\hat{F}} + 1]\right]^{\mathrm{T}}, \tag{48.13}$$

resulting in the well-known Wiener filter

$$\hat{\boldsymbol{f}}[k] = \mathcal{E}\{\boldsymbol{u}[k]\boldsymbol{u}^{\mathrm{T}}[k]\}^{-1}\mathcal{E}\{\boldsymbol{u}[k]y[k]\}\,. \tag{48.14}$$

Assuming the sufficient-order case, where $L_{\hat{F}} = L_F$, and using

$$y[k] = \boldsymbol{f}^{\mathrm{T}}[k]\boldsymbol{u}[k] + x[k], \tag{48.15}$$

(48.14) can be written as:

$$\begin{aligned}\hat{\boldsymbol{f}}[k] &= \mathcal{E}\{\boldsymbol{u}[k]\boldsymbol{u}^{\mathrm{T}}[k]\}^{-1}\mathcal{E}\{\boldsymbol{u}[k]\boldsymbol{u}^{\mathrm{T}}[k]\boldsymbol{f}[k]\} \\ &\quad + \mathcal{E}\{\boldsymbol{u}[k]\boldsymbol{u}^{\mathrm{T}}[k]\}^{-1}\mathcal{E}\{\boldsymbol{u}[k]x[k]\}. \\ &= \boldsymbol{f}[k] + \mathcal{E}\{\boldsymbol{u}[k]\boldsymbol{u}^{\mathrm{T}}[k]\}^{-1}\mathcal{E}\{\boldsymbol{u}[k]x[k]\}\,. \end{aligned} \tag{48.16}$$

From (48.16), it can be seen that the desired signal $x[k]$ acts as a disturbance to the feedback canceller.

Standard adaptive filter procedures (e.g., least mean squares (LMS), recursive least squares (RLS)) can be used for adapting the filter coefficients $\hat{\boldsymbol{f}}[k]$.

Time-Domain Normalized LMS

The most popular adaptive filtering technique is the NLMS algorithm, i. e.,

$$\begin{aligned}\hat{\boldsymbol{f}}[k] &= \hat{\boldsymbol{f}}[k-1] + \mu[k]\boldsymbol{u}[k] \\ &\quad \times (y[k] - \hat{\boldsymbol{f}}^{\mathrm{T}}[k-1]\boldsymbol{u}[k])\,. \end{aligned} \tag{48.17}$$

Unlike in acoustic echo cancellation, the feedback canceller adapts during a continuous double-talk situation [see (48.16)]. To reduce the excess mean square error (MSE) in the presence of desired signals $x[k]$ with large power fluctuations or signal onsets, the step size $\mu[k]$ of the NLMS algorithm is normalized with the sum of the short-term average input power $\overline{|u[k]|^2}$ and error power $\overline{|e[k]|^2}$ [48.16, 23]:

$$\mu[k] = \frac{\bar{\mu}}{L_{\hat{F}}\left(\overline{|u[k]|^2} + \overline{|e[k]|^2}\right) + c}\,, \tag{48.18}$$

where $\bar{\mu}$ is a dimensionless step size constant, c is a small positive constant to prevent singularities, and

$$\overline{|u[k]|^2} = (1-\gamma)|u[k]|^2 + \gamma\overline{|u[k-1]|^2}\,, \tag{48.19}$$

$$\overline{|e[k]|^2} = (1-\gamma)|e[k]|^2 + \gamma\overline{|e[k-1]|^2}\,, \tag{48.20}$$

with $\gamma \in (0, 1]$. For a highly time-varying signal $x[k]$ such as a speech signal, a burst in $e[k]$ most often originates from an increase in the short-term power of $x[k]$. Hence, normalization with the error power reduces the step size $\mu[k]$ when the desired signal $x[k]$ is strong and thus mitigates the negative effect of a strong desired signal segment on the excess MSE.

Partitioned-Block Frequency-Domain NLMS

Better performance can be achieved when updating the feedback canceller in the frequency domain [48.24–27]. Partitioned-block frequency-domain (PBFD) adaptive filters combine the beneficial properties of frequency-domain algorithms, i. e., faster convergence (thanks to a frequency-dependent step size control) and a reduced complexity, with a small processing delay as required for hearing aids.

In this chapter, we concentrate on a PBFD implementation based on overlap-save [48.27]. In this implementation, the $L_{\hat{F}}$-taps feedback canceller $\hat{\boldsymbol{f}}[k]$ is partitioned into $L_{\hat{F}}/P$ segments $\hat{\boldsymbol{f}}_p[k]$ of length P each. Each segment $\hat{\boldsymbol{f}}_p[k]$, $p = 0, \ldots, L_{\hat{F}}/P - 1$ is then transformed to the frequency-domain vector $\hat{\boldsymbol{F}}_p[k]$ by means of the M-point DFT matrix $\mathcal{F}$:

$$\hat{\boldsymbol{f}}_p[k] = \left[\hat{f}_{pP}[k] \;\; \ldots \;\; \hat{f}_{(p+1)P-1}[k]\right]^{\mathrm{T}}, \tag{48.21}$$

$$\hat{\boldsymbol{F}}_p[k] = \mathcal{F}\begin{bmatrix}\hat{\boldsymbol{f}}_p[k] \\ \boldsymbol{0}\end{bmatrix}\begin{matrix}\updownarrow P \\ \updownarrow M-P\end{matrix}. \tag{48.22}$$

(It is assumed that $L_{\hat{F}}/P \in \mathbb{N}_0$. Otherwise $\hat{\boldsymbol{f}}_p[k]$ has to be padded with zeros.) Let the L-dimensional block signal $\boldsymbol{u}_{\bar{k}}$ be defined as

$$\boldsymbol{u}_{\bar{k}} = \left[u[\bar{k}L+1] \;\; \ldots \;\; u[(\bar{k}+1)L]\right]^{\mathrm{T}}, \tag{48.23}$$

with $\bar{k}$ the block index. For each block of L input samples $\boldsymbol{u}_{\bar{k}}$, the PBFD NLMS filter produces L output samples $\boldsymbol{z}_{\bar{k}} = [z[\bar{k}L+1] \; \ldots \; z[(\bar{k}+1)L]]^{\mathrm{T}}$:

$$\boldsymbol{U}_p[\bar{k}] = \mathrm{diag}\left\{\mathcal{F}\begin{bmatrix}u[(\bar{k}+1)L - pP - M + 1] \\ \vdots \\ u[(\bar{k}+1)L - pP]\end{bmatrix}\right\}, \tag{48.24}$$

$$\boldsymbol{z}_{\bar{k}} = \begin{bmatrix}\boldsymbol{0} & \mathbf{I}_L\end{bmatrix}\mathcal{F}^{-1}\sum_{p=0}^{\frac{L_{\hat{F}}}{P}-1}\boldsymbol{U}_p[\bar{k}]\hat{\boldsymbol{F}}_p[\bar{k}]\,. \tag{48.25}$$

The parameter L is called the block length and hence the corresponding input/output delay of the PBFD implementation equals $2L - 1$. To ensure proper operation, it is required that the DFT length $M \geq P + L - 1$.

The PBFD update equation is

$$\hat{\boldsymbol{F}}_p[\bar{k}+1] = \hat{\boldsymbol{F}}_p[\bar{k}] + \boldsymbol{\Delta}[\bar{k}]\mathcal{F}\boldsymbol{g}\mathcal{F}^{-1}\boldsymbol{U}_p^{\mathrm{H}}[\bar{k}]\boldsymbol{E}[\bar{k}]\,, \tag{48.26}$$

where

$$\boldsymbol{E}[\bar{k}] = \mathcal{F}\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{I}_L\end{bmatrix}(\boldsymbol{y}_{\bar{k}} - \boldsymbol{z}_{\bar{k}})\,, \tag{48.27}$$

$$\boldsymbol{y}_{\bar{k}} = \big[y[\bar{k}L+1] \;\; \ldots \;\; y[(\bar{k}+1)L]\big]^{\mathrm{T}}\,, \tag{48.28}$$

and

$$\boldsymbol{g} = \begin{bmatrix}\boldsymbol{I}_P & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}_{M-P}\end{bmatrix}. \tag{48.29}$$

The matrix $\boldsymbol{\Delta}[\bar{k}]$ in (48.26) is a diagonal matrix that contains the step sizes $\mu_m[\bar{k}]$, $m = 0, \ldots, M-1$ of the different frequency bins m, i. e.,

$$\boldsymbol{\Delta}[\bar{k}] = \mathrm{diag}\big\{\mu_0[\bar{k}], \;\; \ldots \;\; \mu_{M-1}[\bar{k}]\big\}. \tag{48.30}$$

Each step size $\mu_m[\bar{k}]$ is normalized according to the sum of the input and the error power in each frequency bin m:

$$\mu_m[\bar{k}] = \frac{\bar{\mu}_m}{\sum_{p=0}^{\frac{L_{\hat{F}}}{P}-1}|U_{p,m}[\bar{k}]|^2 + \frac{L_{\hat{F}}}{P}|E_{p,m}[\bar{k}]|^2 + c}\,, \tag{48.31}$$

where $U_{p,m}[\bar{k}]$ and $E_{p,m}[\bar{k}]$ are the m-th element of $\boldsymbol{U}_p[\bar{k}]$ and $\boldsymbol{E}_p[\bar{k}]$, respectively, and c is a small constant.

The update equations of the PBFD $\hat{\boldsymbol{F}}_p[\bar{k}]$ are summarized in Table 48.1. In the simulations, we set $M = 64$, $P = 32$, and $\bar{\mu}_m = 0.01$. The block length L is chosen such that the processing delay $2L-1$ does not exceed the desired total processing delay d_G (e.g., $L = 32$ for $d_G \geq 4$ ms; $L = 8$ for $d_G = 1$ ms; $L = 16$ for $d_G = 2$–3 ms).

48.2.2 Bias of the CAF

The presence of the closed signal loop $G(q, k)$ introduces specific signal correlation when the desired signal $x[k]$ is spectrally colored (e.g., a speech or music signal). As a result, the CAF fails to provide a reliable feedback path estimate [48.28].

Let us again assume the sufficient-order case where $L_{\hat{F}} = L_F$. Then, the feedback path estimate $\hat{\boldsymbol{f}}[k]$ can be decomposed as [see (48.16)]:

$$\hat{\boldsymbol{f}}[k] = \boldsymbol{f}[k] + \underbrace{\mathcal{E}\{\boldsymbol{u}[k]\boldsymbol{u}^{\mathrm{T}}[k]\}^{-1}\mathcal{E}\{\boldsymbol{u}[k]x[k]\}}_{\text{bias}}\,. \tag{48.32}$$

If $\mathcal{E}\{\boldsymbol{u}[k]x[k]\} = \boldsymbol{0}$, the feedback path estimate $\hat{\boldsymbol{f}}[k]$ is unbiased.

However, because of the presence of the forward path $G(q, k)$, the input signal $u[k]$ to the adaptive filter

Table 48.1 PBFD implementation of CAF

For each block of L input samples $[u[\bar{k}L+1], \ldots, u[(\bar{k}+1)L]$:
$\forall p = 0, \ldots, \frac{L_{\hat{F}}}{P} - 1$:
$\boldsymbol{U}_p[\bar{k}] = \mathrm{diag}\left\{\mathcal{F}\begin{bmatrix}u[(\bar{k}+1)L - pP - M + 1]\\ \vdots \\ u[(\bar{k}+1)L - pP]\end{bmatrix}\right\}$.
Block of output samples $\boldsymbol{z}_{\bar{k}} = [z[\bar{k}L+1] \;\ldots\; z[(\bar{k}+1)L]]^{\mathrm{T}}$:
$\boldsymbol{z}_{\bar{k}} = [\boldsymbol{0} \;\; \boldsymbol{I}_L]\mathcal{F}^{-1}\sum_{p=0}^{\frac{L_{\hat{F}}}{P}-1}\boldsymbol{U}_p[\bar{k}]\hat{\boldsymbol{F}}_p[\bar{k}]$.
Update formula:
$\boldsymbol{y}_{\bar{k}} = [y[\bar{k}L+1] \;\ldots\; y[(\bar{k}+1)L]]^{\mathrm{T}}$,
$\boldsymbol{E}[\bar{k}] = \mathcal{F}\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{I}_L\end{bmatrix}\left(\boldsymbol{y}_{\bar{k}} - [\boldsymbol{0} \;\; \boldsymbol{I}_L]\mathcal{F}^{-1}\sum_{p=0}^{\frac{L_{\hat{F}}}{P}-1}\boldsymbol{U}_p[\bar{k}]\hat{\boldsymbol{F}}_p[\bar{k}]\right)$,
$\hat{\boldsymbol{F}}_p[\bar{k}+1] = \hat{\boldsymbol{F}}_p[\bar{k}] + \boldsymbol{\Delta}[\bar{k}]\mathcal{F}\boldsymbol{g}\mathcal{F}^{-1}\boldsymbol{U}_p^{\mathrm{H}}[\bar{k}]\boldsymbol{E}[\bar{k}]$.
Step size:
$\boldsymbol{\Delta}[\bar{k}] = \mathrm{diag}\{\mu_0[\bar{k}], \ldots, \mu_{M-1}[\bar{k}]\}$,
$\mu_m[\bar{k}] = \frac{\bar{\mu}_m}{\sum_{p=0}^{\frac{L_{\hat{F}}}{P}-1}|U_{p,m}[\bar{k}]|^2 + \frac{L_{\hat{F}}}{P}|E_{p,m}[\bar{k}]|^2 + c}$.

$\hat{\boldsymbol{f}}[k]$ relates to $x[k]$ as

$$u[k] = C(q,k)x[k]\,, \tag{48.33}$$

where $C(q,k)$ is defined in (48.9).

Usually, $G(q,k)$ contains a processing delay d_G of at least one sample. (Note that the delay d_G includes the processing delay of the DAC and the ADC of the hearing aid (Sect. 48.1).) Assuming that $G(q,k)$, $F(q,k)$, and $\hat{F}(q,k)$ are causal, the closed-loop system $C(q,k)$ can be specified as

$$\begin{aligned} C(q,k) = {} & c_{d_G}[k]q^{-d_G} + c_{d_G+1}[k]q^{-d_G-1} \\ & + \cdots + c_{L_C-1}[k]q^{-L_C+1} \end{aligned} \tag{48.34}$$

and hence

$$\begin{aligned} u[k] = {} & \big(c_{d_G}[k] + \ldots \\ & + c_{L_C-1}[k]q^{-L_C+d_G+1}\big)x[k-d_G]\,. \end{aligned} \tag{48.35}$$

As a result, $\mathcal{E}\{\boldsymbol{u}[k]x[k]\} = \mathbf{0}$ if and only if

$$\mathcal{E}\{x[k-\delta]x[k]\} = 0 \tag{48.36}$$

for $d_G \le \delta \le L_C + L_{\hat{F}} - 2$.

Most practical sound signals $x[k]$ are *spectrally colored*, meaning that the signal values $x[k]$ are correlated in time (e.g., speech, music, etc.). Many of these audio signals may be well approximated as low-order time-varying autoregressive (AR) random processes

$$x[k] = H(q,k)w[k] = \frac{1}{A(q,k)}w[k]\,, \tag{48.37}$$

where $A(q,k)$ is an finite impulse response (FIR) with $a_0 = 1$ and where $w[k]$ is a white noise signal. Hence, the signal model $H(q)$ is often infinite impulse response (IIR), so that the length L_H of $H(q)$ typically exceeds d_G and hence, $\mathcal{E}\{\boldsymbol{u}[k]x[k]\} \neq 0$. The CAF will then cancel the desired signal $x[k]$ instead of the feedback signal $n[k] = F(q,k)u[k]$ [see (48.16)], leading to signal distortion.

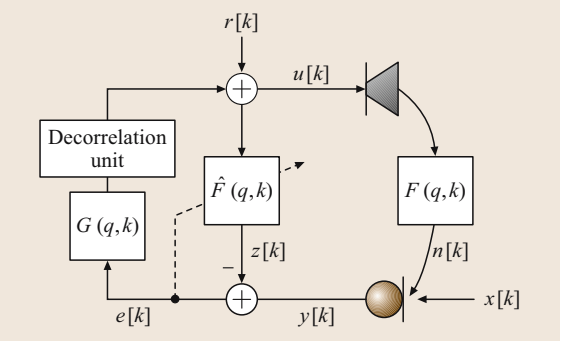


Fig. 48.4 CAF with decorrelating signal operations (i. e., inserting a decorrelation unit or adding a probe signal $r[k]$)

From (48.16), we observe that the bias in the feedback path estimate $\hat{\boldsymbol{f}}[k]$ decreases with increasing power ratio of the feedback signal $n[k]$ (i. e., the signal to identify) to the desired signal $x[k]$ (which acts as a disturbance). Since

$$n[k] = F(q,k)C(q,k)x[k]\,, \tag{48.38}$$

the larger the loop gain $|G(q,k)F(q,k)|$, the smaller the bias will be. This indicates that the bias will be smallest for large gains $G(q,k)$ and for frequencies that are closest to instability, which is a desirable property.

48.2.3 Reducing the Bias of the CAF

Classical approaches to reduce the bias of the CAF include inserting signal decorrelation operations in the forward path $G(q,k)$ and reducing the adaptation speed of the feedback canceller.

Inserting Signal Decorrelation Operations

A first solution to reduce the bias is to include decorrelating signal operations in cascade with the forward path $G(q,k)$, as depicted in Fig. 48.4. Ideally, the decorrelation unit removes any correlation between the desired signal $x[k]$ and the input $u[k]$ to the adaptive filter $\hat{F}(q,k)$, so that the bias term in (48.32) equals zero. Often, the decorrelation unit however degrades the sound quality, making full decorrelation impossible.

For many signals $x[k]$, the autocorrelation sequence $\mathcal{E}\{x[k]x[k-\delta]\}$ decreases with increasing lag δ. A common technique to reduce the correlation is to increase the delay d_G by inserting a delay into the forward path [48.28]. Many audio signals however have a relatively strong autocorrelation. A large delay d_G may then be required to efficiently decorrelate $x[k]$ and $u[k]$. However, the delay d_G should not exceed 10 ms in order not to degrade intelligibility [48.29] and sound quality, e.g., by comb-filter effects resulting from destructive and constructive interference between the direct sound to the eardrum and the amplified and delayed sound [48.30, 31].

As an illustration, Fig. 48.5 depicts the misalignment of the PBFD CAF as a function of the delay d_G of $G(q,k) = Gq^{-d_G}$ with $G = 10\,\mathrm{dB}$ for the acoustic feedback path in Fig. 48.2 with a mobile phone attached to the head. Three desired signals $x[k]$ are considered: a white-noise signal, a stationary speech weighted noise signal, and a real speech signal. The real speech signal

Amazon EX1010 Page 986

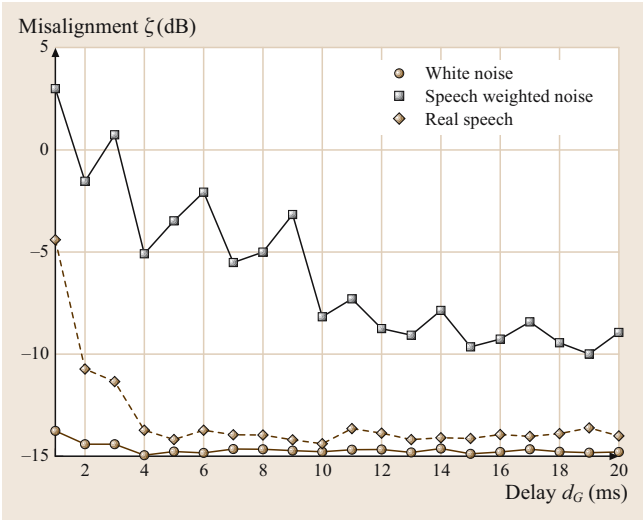


Fig. 48.5 Misalignment of the PBFD CAF as a function of the delay d_G for white noise, speech weighted noise, and real speech as a desired signal. $G(q,k) = Gq^{-d_G}$ with $G = 10\,\text{dB}$

consists of English sentences spoken by a male speaker from the hearing in noise test (HINT) database [48.32]. For speech weighted noise and real speech, the CAF is clearly biased. For both signals, the bias (or misalignment) decreases with increasing delay d_G. A delay of 4 ms provides sufficient decorrelation for the speech weighted noise signal. However, for the real speech signal, a much larger delay $d_G > 10\,\text{ms}$ is required.

Other techniques to reduce the correlation between $x[k]$ and $u[k]$ are inserting a *probe signal* $r[k]$ (noise signal) at the input $u[k]$ of the loudspeaker (see Fig. 48.4) or adding *nonlinearities* to the forward path $G(q,k)$, such as frequency shifting, phase or delay modulation [48.12, 33–35]. Such methods may, however, degrade the sound quality. In the former case, e.g., the probe signal should be inaudible so that it does not degrade the signal-to-noise ratio. Its level will thus have to be lower than the level of the ordinary output $u[k]$, resulting in a significant residual bias. A small amount of modulation or a small frequency shift (e.g., a frequency shift of up to 5 Hz [48.13]) may still be acceptable for speech signals, however, its effect on music is objectionable. These techniques will not be further discussed.

Reducing the Adaptation Speed

In [48.36–38], distortion of the desired signal $x[k]$ is reduced by keeping the adaptation speed of the feedback canceller small. When feedback or a sudden change in the feedback path is detected, the adaptation speed is temporarily increased. A slow adaptation speed reduces the ability to track changes in the spectrum of the desired signal $x[k]$, and hence, limits the negative effect of a temporary signal with a strong autocorrelation on the feedback path estimate [48.30]. Convergence of this technique may however be too slow to track variations in the feedback path successfully so that the maximum amplification may still be limited.

In addition to inserting signal decorrelating operations or reducing the adaptation speed, other approaches have been proposed in the literature for improving the estimation accuracy of the adaptive feedback canceller. A first class of techniques (described in Sect. 48.3) exploits prior knowledge of the acoustic feedback path to improve the adaptation of the feedback canceller. A second class of techniques (discussed in Sect. 48.4) views the feedback path as a part of a closed-loop system and applies closed-loop system identification theory.

48.3 Feedback Cancellation Based on Prior Knowledge of the Acoustic Feedback Path

In [48.15, 22, 39], prior knowledge of the acoustic feedback path is exploited to reduce the bias of the standard CAF. The adaptation of the feedback canceller is controlled based on the prior knowledge through constrained (Sect. 48.3.1) or bandlimited adaptation (Sect. 48.3.2).

48.3.1 Constrained Adaptation (C-CAF)

In [48.22, 39], the bias is reduced by constrained adaptation: the filter coefficients $\hat{\boldsymbol{f}}[k]$ of the adaptive feedback canceller $\hat{F}(q,k)$ are not allowed to deviate too much from reference filter coefficients $\hat{\boldsymbol{f}}_{\text{ref}}$. These reference filter coefficients $\hat{\boldsymbol{f}}_{\text{ref}}$ are measured during start-up or fitting, by means of a probe signal $r[k]$.

A cheap method to impose this constraint is to add a term to the cost function (48.12) of the feedback canceller that penalizes excessive deviation of the filter $\hat{\boldsymbol{f}}[k]$ from the reference filter $\hat{\boldsymbol{f}}_{\text{ref}}$, i. e.,

$$J(\hat{\boldsymbol{f}}[k]) = \mathcal{E}\left\{|y[k] - \hat{\boldsymbol{f}}^{\mathrm{T}}[k]\boldsymbol{u}[k]|^2\right\} + \eta(\hat{\boldsymbol{f}}[k] - \hat{\boldsymbol{f}}_{\text{ref}})^{\mathrm{T}}(\hat{\boldsymbol{f}}[k] - \hat{\boldsymbol{f}}_{\text{ref}})\,. \quad (48.39)$$

This results in the solution

$$\hat{\boldsymbol{f}}[k] = (\mathcal{E}\{\boldsymbol{u}[k]\boldsymbol{u}^{\mathrm{T}}[k]\} + \eta\mathbf{I})^{-1} \left(\mathcal{E}\{\boldsymbol{u}[k]y[k]\} + \eta\hat{\boldsymbol{f}}_{\mathrm{ref}}\right) . \tag{48.40}$$

The scalar η trades off between signal distortion reduction and the ability to model deviations from $\hat{\boldsymbol{f}}_{\mathrm{ref}}$. The larger η, the more $\hat{\boldsymbol{f}}[k]$ evolves towards $\hat{\boldsymbol{f}}_{\mathrm{ref}}$.

Time-Domain NLMS

From (48.39), the following time-domain NLMS update equation can be derived

$$\hat{\boldsymbol{f}}[k+1] = \hat{\boldsymbol{f}}[k] + \frac{\bar{\mu}}{L_{\hat{F}}\left(\overline{|u[k]|^2} + \overline{|e[k]|^2}\right) + \eta + c} \times \left[\boldsymbol{u}[k](y[k] - \hat{\boldsymbol{f}}^{\mathrm{T}}[k]\boldsymbol{u}[k]) - \eta(\hat{\boldsymbol{f}}[k] - \hat{\boldsymbol{f}}_{\mathrm{ref}})\right] . \tag{48.41}$$

Partitioned-Block Frequency-Domain NLMS

Converting (48.41) into the partitioned-block frequency domain results in [48.27]:

$$\hat{\boldsymbol{F}}_p[\bar{k}+1] = \hat{\boldsymbol{F}}_p[\bar{k}] + \boldsymbol{\Delta}[\bar{k}]\left(\mathcal{F}\boldsymbol{g}\mathcal{F}^{-1}\boldsymbol{U}_p^{\mathrm{H}}[\bar{k}]\boldsymbol{E}[\bar{k}] - \boldsymbol{\eta}(\hat{\boldsymbol{F}}_p[\bar{k}] - \hat{\boldsymbol{F}}_{p,\,\mathrm{ref}})\right) , \tag{48.42}$$

where

$$\hat{\boldsymbol{f}}_{p,\,\mathrm{ref}} = \left[\hat{f}_{\mathrm{ref},\,pP} \;\; \cdots \;\; \hat{f}_{\mathrm{ref},\,(p+1)P-1}\right]^{\mathrm{T}} , \tag{48.43}$$

$$\hat{\boldsymbol{F}}_{p,\,\mathrm{ref}} = \mathcal{F}\begin{bmatrix} \hat{\boldsymbol{f}}_{p,\,\mathrm{ref}} \\ \mathbf{0} \end{bmatrix} \begin{matrix} \updownarrow P \\ \updownarrow M-P \end{matrix} . \tag{48.44}$$

$\boldsymbol{\eta}$ is a diagonal matrix that contains the trade-off parameters η_m, $m = 0, \ldots, M-1$ for the different frequency bins m. The diagonal step-size matrix $\boldsymbol{\Delta}[\bar{k}]$ is

$$\boldsymbol{\Delta}[\bar{k}] = \mathrm{diag}\left\{\mu_0[\bar{k}], \;\; \ldots \;, \;\; \mu_{M-1}[\bar{k}]\right\} \tag{48.45}$$

with

$$\mu_m[\bar{k}] = \frac{\bar{\mu}_m}{\sum_{p=0}^{\frac{L_{\hat{F}}}{P}-1} |U_{p,m}[\bar{k}]|^2 + \frac{L_{\hat{F}}}{P}|E_{p,m}[\bar{k}]|^2 + \eta_m + c} . \tag{48.46}$$

In the sequel, the CAF with constrained adaptation will be referred to as C-CAF.

Performance

Figure 48.6 illustrates the effect of $\eta_m = \eta$ on the performance of the PBFD C-CAF for different delays d_G

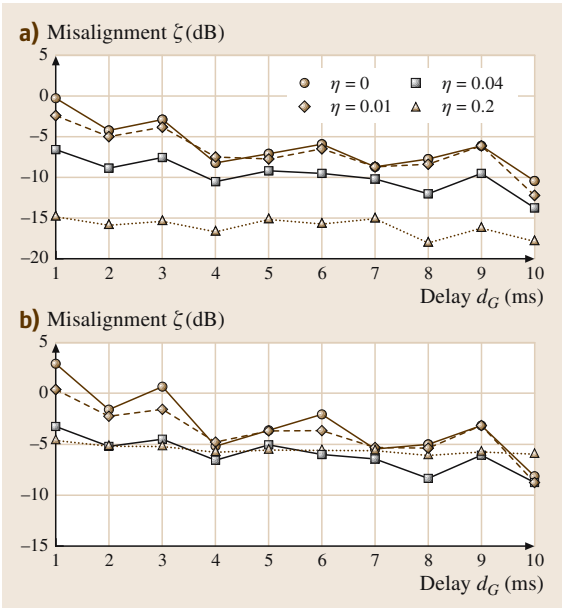


Fig. 48.6a,b Misalignment of the C-CAF as a function of the delay d_G of $G(q,k) = Gq^{-d_G}$ for different settings of η and real speech as a desired signal. **(a)** Without obstruction ($G = 17$ dB). **(b)** With a mobile phone attached to the head ($G = 10$ dB)

and real speech as a desired signal. The reference filter $\hat{\boldsymbol{f}}_{\mathrm{ref}}$ is a 20-taps filter. It was computed for the scenario without obstruction by disconnecting the forward path $G(q,k)$ and inserting a white-noise signal $r[k]$ into the loudspeaker. The frequency response of the reference filter $\hat{\boldsymbol{f}}_{\mathrm{ref}}$ together with the frequency response of the true feedback path is depicted in Fig. 48.7. The feedback path when a mobile phone is attached to the ear is also shown. Figure 48.6(a) shows the misalignment for the feedback path without obstruction. Figure 48.6b depicts the misalignment for the feedback path with a mobile phone attached to the head, illustrating the performance of the C-CAF in the presence of a deviation in the feedback path model. The forward path gain was set close to instability, i. e., $G = 17$ dB for the scenario without obstruction and $G = 10$ dB for the scenario with a mobile phone.

For the scenario without obstruction, the reference filter is a good model of the actual feedback path. As a result, the misalignment of the C-CAF decreases with increasing η. When the mobile phone is attached to the head, the actual feedback path deviates from the reference path (Fig. 48.6). If η is chosen too large, the feedback path deviation cannot be successfully mod-

Amazon EX1010 Page 988

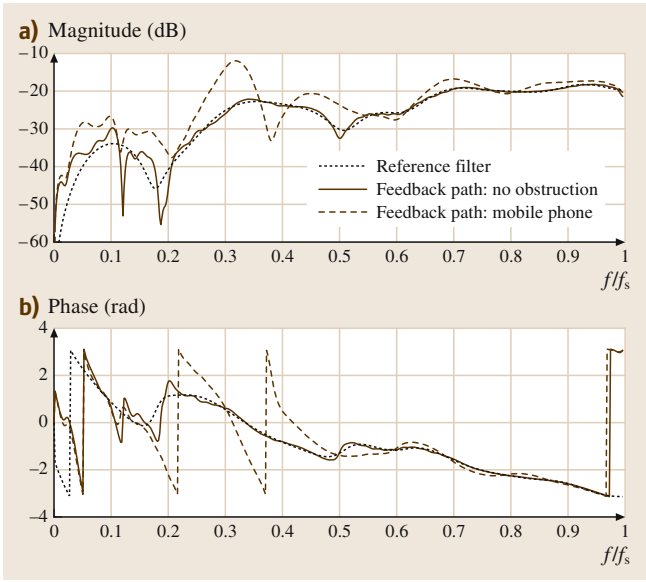


Fig. 48.7a,b Frequency response of the reference filter $\hat{\boldsymbol{f}}_{\text{ref}}$ used in the C-CAF. For comparison, the true feedback path for the scenario without obstruction and with a mobile phone are also depicted

eled. Indeed, Fig. 48.6b shows that for $\eta = 0.2$, a similar or worse misalignment is obtained than for $\eta = 0.04$. In the sequel, $\eta = 0.04$.

48.3.2 Bandlimited Adaptation (BL-CAF)

In [48.15, 38, 40–42], feedback cancellation is restricted to the frequency band that encompasses the unstable frequencies through bandlimited adaptation. Indeed, to avoid instability, it suffices that the feedback canceller cancels the feedback signal at the critical frequencies, where instability is about to occur. Typically, the acoustic feedback path of a hearing aid provides less attenuation at high frequencies, where most hearing aid users have the largest hearing loss (Fig. 48.2) [48.3]. As a result, the risk for instability is often highest in the high-frequency range, while most of the desired signal energy is typically concentrated at low frequencies, e.g., for speech signals. By concentrating the adaptation of the feedback canceller on the higher frequencies only, the feedback canceller may be more efficient and will introduce less distortion.

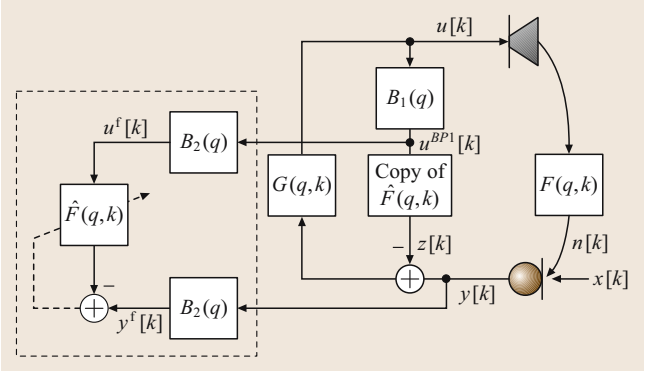


Fig. 48.8 Bandlimited adaptive feedback cancellation (BL-CAF)

Figure 48.8 depicts the block diagram of the bandlimited feedback canceller (abbreviated as BL-CAF). The filters $B_1(q)$ and $B_2(q)$ are high- or bandpass filters chosen such that all critical frequencies are preserved while the desired signal components are removed as much as possible. The filter $B_1(q)$ in the feedback cancellation path limits the feedback cancellation signal to the frequency band of interest and hence prevents the desired signal components outside the critical band from being distorted by the feedback canceller $\hat{F}(q,k)$. To focus the modeling effort of the adaptive feedback canceller $\hat{\boldsymbol{f}}[k]$ on the regions that contain the critical frequencies, the adaptive filter minimizes the frequency-weighted error energy

$$J(\hat{\boldsymbol{f}}[k]) = \mathcal{E}\left\{|B_2(q)(y[k] - \hat{F}(q,k)u^{\text{BP}_1}[k])|^2\right\} = \mathcal{E}\left\{|y^{\text{f}}[k] - \hat{F}(q,k)u^{\text{f}}[k]|^2\right\}, \quad (48.47)$$

where

$$u^{\text{BP}_1}[k] = B_1(q)u[k], \quad (48.48)$$

$$y^{\text{f}}[k] = B_2(q)y[k], \quad (48.49)$$

$$u^{\text{f}}[k] = B_2(q)u^{\text{BP}_1}[k]\,. \quad (48.50)$$

The filter $B_2(q)$ reduces the energy of the residual error signal (which acts as a disturbance to the adaptive feedback canceller). As a result, the misadjustment and bias of the bandlimited adaptive feedback canceller is reduced compared to the standard CAF.

Note that the filter $B_1(q)$ introduces a group delay in the feedback cancellation path. This group delay should not exceed the delay in the acoustic feedback path, otherwise the feedback signal cannot be cancelled [48.15]. To guarantee a small group delay and to save computational complexity, low-order elliptic IIR filters $B_1(q)$ and $B_2(q)$ are used in [48.15].

In [48.15], minimization of (48.47) is realized through a filtered-X algorithm. In the filtered-X algorithm, the input signal $u^{\text{BP}_1}[k]$ and the feedback-compensated signal $e[k] = y[k] - \hat{F}(q,k)u^{\text{BP}_1}[k]$ are

Amazon EX1010 Page 989

filtered by $B_2(q)$ before being used to update the adaptive filter $\hat{F}(q,k)$. Equivalently, $J(\boldsymbol{f}[\bar{k}])$ can be minimized by performing standard adaptive filtering techniques on the prefiltered microphone and loudspeaker data $y^{\mathrm{f}}[k]$ and $u^{\mathrm{f}}[k]$ (as illustrated in Fig. 48.8) [48.27]. In contrast to the filtered-X algorithm, any change to the adaptive filter $\hat{\boldsymbol{f}}[k]$ then has an immediate effect on the frequency-weighted adaptation error even if $B_2(q)$ has a group delay. As a result, no stability problems may occur [48.15, 43].

Time-Domain NLMS

For NLMS adaptation, the update equation becomes

$$\hat{\boldsymbol{f}}[k] = \hat{\boldsymbol{f}}[k-1] + \mu[k]\boldsymbol{u}[k]\underbrace{\left(y^{\mathrm{f}}[k] - \hat{\boldsymbol{f}}^{\mathrm{T}}[k-1]\boldsymbol{u}^{\mathrm{f}}[k]\right)}_{e^{\mathrm{f}}[k]}, \tag{48.51}$$

with

$$\boldsymbol{u}^{\mathrm{f}}[k] = \left[u^{\mathrm{f}}[k] \;\; \ldots \;\; u^{\mathrm{f}}[k-L_{\hat{F}}]\right] \tag{48.52}$$

$$\mu[k] = \frac{\bar{\mu}}{L_{\hat{F}}\left(\overline{|u^{\mathrm{f}}[k]|^2} + \overline{|e^{\mathrm{f}}[k]|^2}\right) + c}, \tag{48.53}$$

Partitioned-Block Frequency-Domain NLMS

The PBFD implementation equals [48.27]

$$\hat{\boldsymbol{F}}_p[\bar{k}+1] = \hat{\boldsymbol{F}}_p[\bar{k}] + \boldsymbol{\Delta}[\bar{k}]\mathcal{F}\boldsymbol{g}\mathcal{F}^{-1}\boldsymbol{U}_p^{\mathrm{f,H}}[\bar{k}]\boldsymbol{E}^{\mathrm{f}}[\bar{k}], \tag{48.54}$$

where

$$\boldsymbol{U}_p^{\mathrm{f}}[\bar{k}] = \mathrm{diag}\left\{\mathcal{F}\begin{bmatrix} u^{\mathrm{f}}[(\bar{k}+1)L - pP - M + 1] \\ \vdots \\ u^{\mathrm{f}}[(\bar{k}+1)L - pP] \end{bmatrix}\right\}, \tag{48.55}$$

$$\boldsymbol{E}^{\mathrm{f}}[\bar{k}] = \mathcal{F}\begin{bmatrix}\boldsymbol{0} \\ \mathbf{I}_L\end{bmatrix}\left(\boldsymbol{y}_{\bar{k}}^{\mathrm{f}} - \begin{bmatrix}\boldsymbol{0} & \mathbf{I}_L\end{bmatrix}\mathcal{F}^{-1} \times \sum_{p=0}^{\frac{L_{\hat{F}}}{P}-1}\boldsymbol{U}_p^{\mathrm{f}}[\bar{k}]\hat{\boldsymbol{F}}_p[\bar{k}]\right), \tag{48.56}$$

$$\boldsymbol{y}_{\bar{k}}^{\mathrm{f}} = \left[y[\bar{k}L+1] \;\; \ldots \;\; y[(\bar{k}+1)L]\right]^{\mathrm{T}}. \tag{48.57}$$

The step sizes $\mu_m[\bar{k}]$ are computed as

$$\frac{\bar{\mu}_m}{\sum_{p=0}^{\frac{L_{\hat{F}}}{P}-1}\left|U_{p,m}^{\mathrm{f}}[\bar{k}]\right|^2 + \frac{L_{\hat{F}}}{P}\left|E_{p,m}^{\mathrm{f}}[\bar{k}]\right|^2 + c}. \tag{48.58}$$

Performance

The bandwidths and the stopband attenuation of the filters $B_1(q)$ and $B_2(q)$ determine the efficiency of the bandlimited feedback canceller $\hat{F}(q,k)$: the larger the bandwidth (or the smaller the stopband attenuation), the larger the frequency range where acoustic feedback is cancelled, but the larger the bias and misadjustment of the feedback path estimate. To specify the bandwidth of $B_1(q)$ and $B_2(q)$, identification of the critical frequency region for different hearing aids and different hearing aid users is required, e.g., during fitting. However, the critical frequency region expands with increasing gain in the forward path $G(q,k)$ as well as, e.g., with the presence of a telephone handset or hand palm close to the ear (Fig. 48.2). Hence, to guarantee stability, the bandwidth and stopband attenuation of $B_1(q)$ and

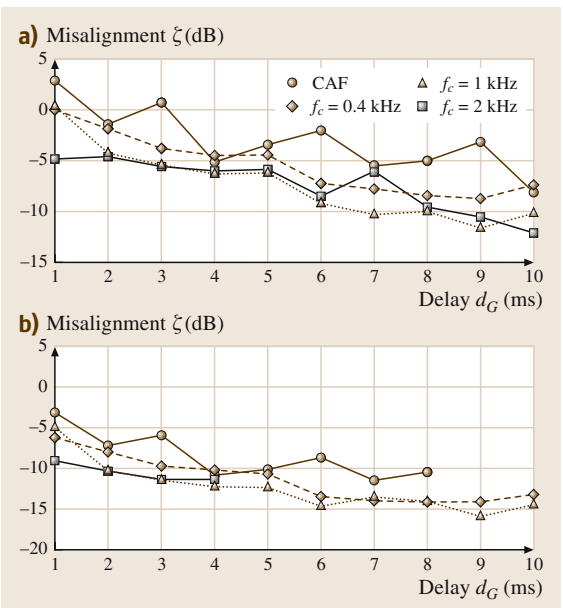


Fig. 48.9a,b Misalignment of BL-CAF as a function of the delay d_G for different cut-off frequencies f_c (i. e., f_c = 0.4 kHz, f_c = 1 kHz, and f_c = 2 kHz) and real speech as a desired signal. **(a)** G = 10 dB. **(b)** G = 17 dB (for d_G = 5–6 ms and d_G = 8–10 ms, instability occurs for the BL-CAF with f_c = 2 kHz; for d_G = 9–10 ms, instability occurs for the CAF)

$B_2(q)$ should not be chosen too small. In this Chapter, a second-order IIR filter $B_1(q) = B_2(q)$ with a cut-off frequency of 1 kHz and a stopband attenuation of 20 dB is used.

As an illustration, Fig. 48.9 depicts the misalignment of the BL-CAF as a function of the forward path delay d_G for three different cut-off frequencies f_c, i. e., $f_c =$ 0.4 kHz, $f_c = 1$ kHz and $f_c = 2$ kHz. For comparison, the misalignment of the CAF is also shown. The acoustic feedback path with a mobile phone attached to the head is used. The desired signal $x[k]$ is a real speech signal. Two forward path gains G are depicted, i.e., $G = 10$ dB and $G = 17$ dB. For all delays d_G, the BL-CAF achieves a smaller misalignment than the CAF, especially for low gains G. For $G = 10$ dB, a cut-off frequency of 1 kHz or 2 kHz results in a smaller misalignment than a cut-off frequency of 0.5 kHz. For $G = 17$ dB, the cut-off frequency of 2 kHz performs worse than $f_c = 1$ kHz: for $d_G \geq 5$ ms, instability even occurs. At high gains, the region with critical frequencies increases, explaining the worse performance. This illustrates that f_c should not be chosen too large.

48.4 Feedback Cancellation Based on Closed-Loop System Identification

All of the aforementioned feedback cancellation techniques initially ignore the presence of the closed signal loop $G(q, k)$ and apply standard adaptive filtering techniques for open-loop systems. In a second stage, signal processing operations are included to reduce the negative effect of $G(q, k)$ on the performance.

An alternative approach is to view the feedback path immediately as a part of a closed-loop system and to apply closed-loop system identification theory [48.1, 17, 30, 44, 45]. Closed-loop identification methods can be classified as direct, indirect, or joint input–output methods [48.1]. In contrast to the indirect and the joint input–output approach, the direct method requires neither assumptions about the forward path $G(q, k)$ (e.g., linearity) nor the presence of an external probe signal $r[k]$, making it an appealing approach for feedback cancellation.

Section 48.4.1 defines the closed-loop system setup. Section 48.4.2 describes the direct method of closed-loop system identification in the context of feedback cancellation. In this method, the bias of the standard CAF is reduced by incorporating a (stationary or time-varying) model of the desired signal $x[k]$ in the identification [48.18, 30, 45, 46]. The estimation of the desired signal model is discussed in Sect. 48.4.3. Section 48.4.4 discusses the other two closed-loop system identification procedures, i. e., the indirect method and the joint input–output method [48.17, 47].

48.4.1 Closed-Loop System Setup

Figure 48.10 depicts the closed-loop system setup. The *open-loop system to be identified* is given by

$$y[k] = F(q, k)u[k] + x[k] \,. \tag{48.59}$$

For the time being, we assume that the desired signal $x[k]$ can be modeled as

$$x[k] = H(q, k)w[k] \,, \tag{48.60}$$

with $w[k]$ a zero-mean white-noise sequence and $H(q, k)$ monic and inversely stable.

The output signal $y[k]$ is fed back to the input $u[k]$ according to

$$u[k] = G(q, k)(y[k] - \hat{F}(q, k)u[k]) + r[k] \,. \tag{48.61}$$

The signal $r[k]$ is an optional probe signal that is independent of $x[k]$, and can be viewed as an extra excitation signal of the feedback path $F(q, k)$. The probe signal is generally a noise signal.

By combining (48.59) and (48.61), we obtain the closed-loop relations [48.1, 44]

$$y[k] = (1 + F(q, k)C(q, k))x[k] + F(q, k)S(q, k)r[k] \,, \tag{48.62}$$

$$u[k] = C(q, k)x[k] + S(q, k)r[k] \,, \tag{48.63}$$

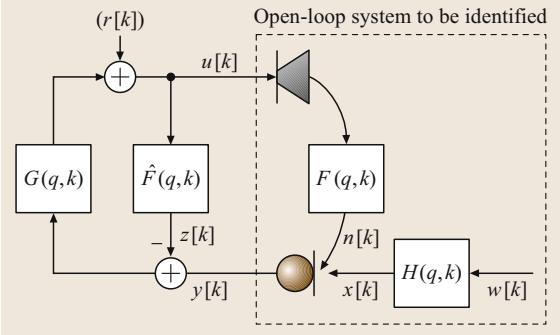


Fig. 48.10 Closed-loop system set-up

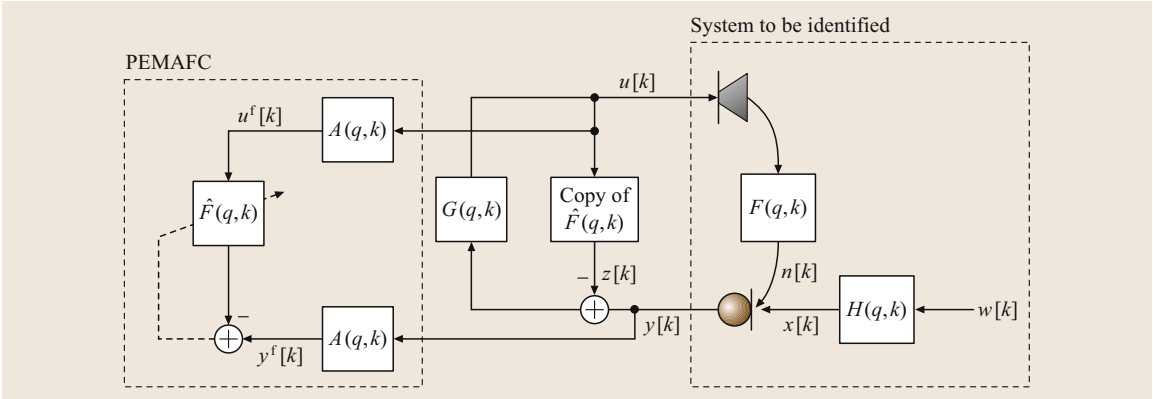


Fig. 48.11 Feedback cancellation with the prediction error method (PEMAFC)

where $S(q,k)$ is the so-called sensitivity function

$$S(q,k) = \frac{1}{1 - G(q,k)(F(q,k) - \hat{F}(q,k))} \tag{48.64}$$

and $C(q,k)$ is the closed-loop transfer function from $x[k]$ to $u[k]$ [see(48.9)]

$$C(q,k) = S(q,k)G(q,k)\,. \tag{48.65}$$

For future use, we denote the closed-loop transfer function from $r[k]$ to $y[k]$ as $F^c(q,k)$:

$$F^c(q,k) = F(q,k)S(q,k)\,. \tag{48.66}$$

Depending on the assumptions that are made about the forward path $G(q,k)$, closed-loop identification methods can be classified as *indirect, joint input–output*, or *direct* methods [48.1, 44].

48.4.2 Direct Method

In the direct method of closed-loop identification, the open-loop system $\{F(q,k),\ H(q,k)\}$

$$y[k] = F(q,k)u[k] + H(q,k)w[k]\,, \tag{48.67}$$

is identified from the loudspeaker signal $u[k]$ and the microphone signal $y[k]$ by an open-loop identification method, thereby ignoring the presence of the closed signal loop $G(q,k)$ [48.18, 30, 45, 48]. Only some specific open-loop identification methods can be applied to closed-loop systems, such as the prediction error method (PEM) [48.1,44]. The PEM produces an estimate $\hat{F}(q,k)$ and $\hat{H}(q,k)$ of the feedback path $F(q,k)$ and the desired signal model $H(q,k)$, respectively, by minimizing the energy of the so-called prediction error

$$e^p[k] = \hat{H}(q,k)^{-1}(y[k] - \hat{F}(q,k)u[k])\,,$$

i. e.,

$$\begin{aligned} J(\hat{\boldsymbol{f}}[k]) &= \mathcal{E}\{|\hat{H}(q,k)^{-1}(y[k] - \hat{F}(q,k)u[k])|^2\}\,, \\ &= \mathcal{E}\{|y^f[k] - \hat{\boldsymbol{f}}^T[k]\boldsymbol{u}^f[k]|^2\}\,, \end{aligned} \tag{48.68}$$

where

$$\boldsymbol{u}^f[k] = \left[u^f[k]\ \ \ldots\ \ u^f[k - L_{\hat{F}} + 1]\right]^T\,, \tag{48.69}$$

$$y^f[k] = \hat{H}(q,k)^{-1}y[k]\,, \tag{48.70}$$

$$u^f[k] = \hat{H}(q,k)^{-1}u[k]\,. \tag{48.71}$$

Minimization of (48.68) results in

$$\hat{\boldsymbol{f}}[k] = \mathcal{E}\{\boldsymbol{u}^f[k]\boldsymbol{u}^{f,\,T}[k]\}^{-1}\mathcal{E}\{\boldsymbol{u}^f[k]y^f[k]\}\,, \tag{48.72}$$

where $y^f[k]$ can be decomposed as:

$$y^f[k] = \hat{H}(q,k)^{-1}x[k] + F(q,k)u^f[k]\,. \tag{48.73}$$

If $\hat{H}(q,k) = H(q,k)$,

$$y^f[k] = w[k] + F(q,k)u^f[k]\,. \tag{48.74}$$

The desired signal $x[k]$ is converted into a white-noise signal $w[k]$. As a result, $\hat{\boldsymbol{f}}[k]$ results in an unbiased feedback path estimate in the sufficient-order case (i. e., $L_{\hat{F}} = L_F$) [48.18, 45].

From (48.68) it follows that $J(\hat{\boldsymbol{f}}[k])$ can be minimized by performing standard adaptive filtering techniques on the pre-whitened data $y^f[k]$ and $u^f[k]$ [48.43]. This is illustrated in Fig. 48.11, where $A(q,k) = \hat{H}(q,k)^{-1}$. The update equations of the time-domain and PBFD NLMS algorithm correspond to (48.51–48.58) with $y^f[k]$ and $u^f[k]$ now defined in (48.70–48.71) (Table 48.2). In the sequel, we refer to the feedback canceller based on the PEM as PEMAFC.

Amazon EX1010 Page 992

Table 48.2 PBFD implementation of PEMAFC-a

For each block of L input samples $[u[\bar{k}L+1], \ldots, u[(\bar{k}+1)L]$

Block of output samples $z_{\bar{k}} = [z[\bar{k}L+1] \ \ldots \ z[(\bar{k}+1)L]]^{\mathrm{T}}$

$$z_{\bar{k}} = [\boldsymbol{0} \ \boldsymbol{I}_L]\mathcal{F}^{-1}\sum_{p=0}^{\frac{L_{\hat{F}}}{P}-1}\boldsymbol{U}_p[\bar{k}]\hat{\boldsymbol{F}}_p[\bar{k}]\,,$$

with

$$\boldsymbol{U}_p[\bar{k}] = \mathrm{diag}\left\{\mathcal{F}\begin{bmatrix} u[(\bar{k}+1)L-pP-M+1] \\ \vdots \\ u[(\bar{k}+1)L-pP]\end{bmatrix}\right\},\ p=0,\ \ldots,\ \frac{L_{\hat{F}}}{P}-1\,.$$

Prefilter loudspeaker and microphone signal with $A(q,\bar{k}_a-1)$, $\bar{k}_a = \left\lceil\frac{\bar{k}L+i}{N}\right\rceil$ (N is the framelength):

$$u^{\mathrm{f}}[\bar{k}L+i] = A(q,\bar{k}_a-1)u[\bar{k}L+i-N]\,,$$
$$y^{\mathrm{f}}[\bar{k}L+i] = A(q,\bar{k}_a-1)y[\bar{k}L+i-N],\ i=1,\ \ldots,\ L,$$
$$\boldsymbol{U}_p^{\mathrm{f}}[\bar{k}] = \mathrm{diag}\left\{\mathcal{F}\begin{bmatrix} u^{\mathrm{f}}[(\bar{k}+1)L-pP-M+1] \\ \vdots \\ u^{\mathrm{f}}[(\bar{k}+1)L-pP]\end{bmatrix}\right\}.$$

Update AR model $A(q,\bar{k}_a)$:

If $\bar{k}L+i = \bar{k}_a N$ with $\bar{k}_a$ an integer:

$$A(q,\bar{k}_a) = \text{Levinson–Durbin}\left([e(\bar{k}L+i-N+1] \ \ldots \ e[\bar{k}L+i]]^{\mathrm{T}}\right)$$

with $e[\bar{k}L+i] = y[\bar{k}L+i] - z[\bar{k}L+i]$.

Update formula:

$$\boldsymbol{y}_{\bar{k}}^{\mathrm{f}} = [y^{\mathrm{f}}[\bar{k}L+1] \ \ldots \ y^{\mathrm{f}}[(\bar{k}+1)L]]^{\mathrm{T}}\,,$$
$$\boldsymbol{E}^{\mathrm{f}}[\bar{k}] = \mathcal{F}\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{I}_L\end{bmatrix}\left(\boldsymbol{y}_{\bar{k}}^{\mathrm{f}} - [\boldsymbol{0} \ \boldsymbol{I}_L]\mathcal{F}^{-1}\sum_{p=0}^{\frac{L_{\hat{F}}}{P}-1}\boldsymbol{U}_p^{\mathrm{f}}[\bar{k}]\hat{\boldsymbol{F}}_p[\bar{k}]\right),$$
$$\hat{\boldsymbol{F}}_p[\bar{k}+1] = \hat{\boldsymbol{F}}_p[\bar{k}] + \boldsymbol{\Delta}[\bar{k}]\mathcal{F}\boldsymbol{g}\mathcal{F}^{-1}\boldsymbol{U}_p^{\mathrm{f},\mathrm{H}}[\bar{k}]\boldsymbol{E}^{\mathrm{f}}[\bar{k}]\,.$$

Step size:

$$\boldsymbol{\Delta}[\bar{k}] = \mathrm{diag}\{\mu_0[\bar{k}],\ \ldots,\ \mu_{M-1}[\bar{k}]\}\,,$$
$$\mu_m[\bar{k}] = \frac{\tilde{\mu}_m}{\sum_{p=0}^{\frac{L_{\hat{F}}}{P}-1}\left|U_{p,m}^{\mathrm{f}}[\bar{k}]\right|^2 + \frac{L_{\hat{F}}}{P}\left|E_{p,m}^{\mathrm{f}}[\bar{k}]\right|^2 + c}\,.$$

48.4.3 Desired Signal Model

The PEMAFC requires an estimate of the desired signal model $H(q,k)$. Both, fixed and adaptive estimates of the desired signal model $H^{-1}(q,k)$ have been considered in the literature. In the sequel, PEMAFC with a fixed model is referred to as PEMAFC-f and PEMAFC with an adaptive model is referred to as PEMAFC-a.

Fixed Desired Signal Model (PEMAFC-f)

In [48.30, 45], the desired signal model $H(q,k)$ is approximated by a *fixed* low-pass all-pole filter $\hat{H}(q,k) = A^{-1}(q)$, which represents the long-term average speech spectrum. A simple model for the average speech spectrum is a first-order all-pole model:

$$\hat{H}(q,k) = \frac{1}{1-\alpha q^{-1}}\,, \qquad (48.75)$$

with $\alpha < 1$ (e.g., $\alpha = 0.9$) [48.22].

Adaptive Desired Signal Model (PEMAFC-a)

In practice, the desired signal model $H^{-1}(q,k)$ is unknown and highly time varying. In addition, the quality of the feedback canceller $\hat{F}(q,k)$ strongly depends on the accuracy of the signal model estimate $\hat{H}(q,k)$ [48.45], so that it is desirable to identify the feedback path $F(q,k)$ as well as the desired signal model $H(q,k)$ with the PEM method. However, $F(q,k)$ and $H(q,k)$ are not always identifiable in the closed-loop system at hand [48.45].

Since most audio signals $x[k]$ can be approximated by a low-order AR model, it is assumed that

$$H^{-1}(q,k) = A(q,k) = 1 + q^{-1}\bar{A}(q,k) \qquad (48.76)$$

with $\bar{A}(q,k)$ an FIR filter. In [48.18], it has been shown that the AR model $A(q,k)$ and the feedback path $F(q,k)$ can be both identified in closed-loop without adding nonlinearities or a probe signal, if the delay $d_G \geq L_{H^{-1}} = L_A$.

Since the filter length L_A is short, the required processing delay d_G is usually much shorter (e.g., 1–2 ms) than the delay d_G that is needed in the standard CAF to decorrelate $x[k]$ and $u[k]$. For example, a 10–20 ms speech segment at a sampling frequency $f_s = 16$ kHz can be modeled by a 10–20-order AR model with $w[k]$ a white-noise excitation (in case of unvoiced sounds) or a pulse-train excitation (in case of voiced sounds).

The AR model $A(q,k)$ is computed through linear prediction of the feedback compensated signal

$$e[k] = y[k] - \hat{\boldsymbol{f}}^{\mathrm{T}}[k]\boldsymbol{u}[k]\,, \tag{48.77}$$

on subsequent frames of $N = 160$ samples (i. e., 10 ms), e.g., using the Levinson–Durbin algorithm [48.49]. Note that $e[k] = x[k]$, if $\hat{\boldsymbol{f}}[k] = \boldsymbol{f}[k]$.

So far, we have assumed that the excitation $w[k]$ of $H(q,k)$ is a white-noise sequence. This assumption does not apply for voiced speech, where the excitation $w[k]$ approximates a pulse train that is periodical with the pitch period P (expressed as number of samples). A low-order (e.g., 10th to 20th order) AR model will not suffice to notch out the discrete frequencies caused by the pulse train excitation. As a result, the prefiltered loudspeaker signal $u^{\mathrm{f}}[k]$ will still be correlated with the excitation signal $w[k]$ at the pitch frequency $\frac{f_s}{P}$ and its harmonics $k\frac{f_s}{P}$ with $k \in \mathbb{N}$ and $\frac{k}{P} < \frac{1}{2}$. The residual correlation caused by the pulse train excitation can be removed by cascading the low-order short-term AR model $A_{\mathrm{ST}}(q,k)$ with a long-term predictor $A_{\mathrm{LT}}(q,k) = (1 - bq^{-\hat{P}[k]})$, where $\hat{P}[k]$ is an estimate of the pitch period P [48.48]:

$$A(q,k) = A_{\mathrm{ST}}(q,k)(1 - bq^{-\hat{P}[k]})\,. \tag{48.78}$$

For speech, the pitch frequency f_s/P lies between 50 and 400 Hz and hence for $f_s = 16$ kHz, P lies between 40 and 320 samples. To guarantee identifiability, the total filter length $\hat{P}[k] + L_{A_{\mathrm{ST}}}$ of $A(q,k)$ should not exceed the maximum allowable processing delay d_G.

For feedback cancellers $\hat{F}(q,k)$ with a short filter length $L_{\hat{F}} \leq P$ and for small delays d_G, the bias in the feedback path estimate due to the periodic pulse train excitation $w[k]$ is negligible [48.27]. In hearing aids, the dominant part of the feedback path $F(q,k)$ is quite short, so that a relatively short filter length $L_{\hat{F}}$ is typically used, i. e., $L_{\hat{F}} \leq 100$ [48.14]. In addition, the delay d_G should be kept small (i. e., ≤ 10 ms). As a result, the benefit of a long-term predictor is limited for hearing-aid applications.

The equations of the PBFD NLMS implementation of the PEMAFC-a are summarized in Table 48.2.

Performance

As an illustration, Fig. 48.12 depicts the misalignment of PEMAFC-f (with the fixed AR model (48.75)) and PEMAFC-a (with $L_A = 21$) as a function of the delay d_G for the acoustic feedback path with a mobile phone attached to the head and a forward path gain $G = 10$ dB. (To guarantee identifiability, $L_A = 16$ for $d_G = 1$ ms.) For comparison, the misalignment of the CAF is also shown.

In Fig. 48.12a, the desired signal $x[k]$ is speech weighted noise, generated by passing white noise through a 20th-order all-pole model. In this example, the desired signal $x[k]$ can be perfectly whitened by a 20th-order AR model $A(q,k)$, since the white-noise assumption is satisfied and $L_{H^{-1}} = 21$. Indeed, for $d_G \leq 2$ ms, the PEMAFC succeeds to fully decorrelate $x[k]$ and $u[k]$: the same misalignment is obtained as for the CAF with white noise as a desired signal $x[k]$ (Fig. 48.5). (For $d_G = 16$ ms, the AR model is undermodeled since $L_A = 16$, explaining the worse performance.) The fixed all-pole model in the PEMAFC-f whitens the desired signal to some extent such that better performance than the

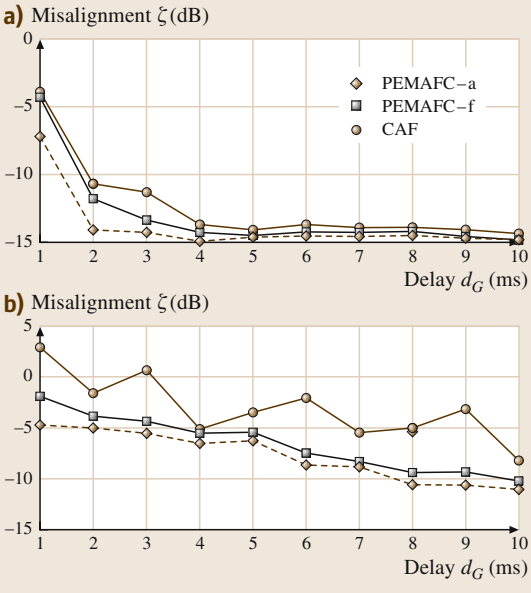


Fig. 48.12a,b Misalignment of the PEMAFC with a fixed (PEMAFC-f) and adaptive AR model (PEMAFC-a) as a function of the delay d_G for the acoustic feedback path with a mobile phone attached to the head. **(a)** $x[k]$ = speech weighted noise. **(b)** $x[k]$ = real speech

CAF is achieved. For $d_G \geq 4$ ms, $x[k]$ and $u[k]$ are uncorrelated so that all methods achieve similar performance.

In Fig. 48.12b, $x[k]$ is a real speech signal (male sentences from the HINT database [48.32]). The model assumptions for $x[k]$ (i. e., the assumption of a AR-model for $H^{-1}(q,k)$ and white noise for $w[k]$) are not completely satisfied, explaining the residual bias. However, for all delays $d_G \leq 10$ ms, the PEMAFC-f and PEMAFC-a achieve better performance than the CAF. PEMAFC-a outperforms PEMAFC-f thanks to the adaptive AR model.

48.4.4 Indirect and Joint Input–Output Method

In contrast to the direct method, the indirect and joint input–output method both require the use of a probe signal $r[k]$.

Indirect Method

In addition to the use of a probe signal $r[k]$, the indirect method [48.1, 44] requires knowledge of the forward path $G(q,k)$. This makes it not very suitable for hearing aids because $G(q,k)$ often contains user-dependent and adaptive signal processing features, e.g., dynamic-range compression.

In the indirect method, the closed-loop transfer function $F^{\mathrm{c}}(q,k)$ (see (48.66)) from $r[k]$ to $y[k]$ is estimated by means of an adaptive filter $\hat{F}^{\mathrm{c}}(q,k)$. If the forward path $G(q,k)$ is known, the feedback path estimate $\hat{F}(q,k)$ can then be extracted from this closed-loop transfer function estimate $\hat{F}^{\mathrm{c}}(q,k)$. In [48.50, 51], the estimate of the closed-loop transfer function $F^{\mathrm{c}}(q,k)$ is used as an approximation of $F(q,k)$, i. e., the sensitivity function $S(q,k)$ is assumed to be 1. This generally results in a biased feedback path estimate $\hat{F}(q,k)$, especially close to or at instability. In [48.52], it is shown that the first L_F coefficients of $F^{\mathrm{c}}(q,k)$ are equal to the coefficients of $F(q,k)$ if the forward path delay $d_G \geq L_{\hat{F}}$. For a white-noise signal $r[k]$ and $L_{\hat{F}} = L_F$, the estimate $\hat{F}^{\mathrm{c}}(q,k)$ is then unbiased.

The identification of the closed-loop system $F^{\mathrm{c}}(q,k)$ is seriously degraded by the presence of the desired signal $x[k]$ (see (48.62)). As a result, a large noise level or a slow adaptation speed is required to obtain an accurate estimate $\hat{F}^{\mathrm{c}}(q,k)$ (i. e., with a small variance).

Joint Input–Output Method

In [48.47], the joint input–output method is applied to obtain an unbiased feedback path estimate $\hat{F}(q,k)$. In this approach, the forward path $G(q,k)$ is assumed to have a certain structure (e.g., a linear filter) and the microphone signal $y[k]$ and the loudspeaker signal $u[k]$ are jointly viewed as the output of a system driven by the probe signal $r[k]$ (see (48.62)-(48.63)) [48.1, 44]:

$$\begin{bmatrix} y[k] \\ u[k] \end{bmatrix} = \begin{bmatrix} F^{\mathrm{c}}(q,k) \\ S(q,k) \end{bmatrix} r[k] + \begin{bmatrix} 1 + F(q,k)C(q,k) \\ C(q,k) \end{bmatrix} x[k] \,. \tag{48.79}$$

The closed-loop transfer function $F^{\mathrm{c}}(q,k)$ and the sensitivity function $S(q,k)$ are determined by means of standard adaptive filtering techniques performed on the excitation signal $r[k]$. The feedback path estimate $\hat{F}(q,k)$ is then derived from these two model estimates $\hat{F}^{\mathrm{c}}(q,k)$ and $\hat{S}(q,k)$ as:

$$\hat{F}(q,k) = \hat{F}^{\mathrm{c}}(q,k)\hat{S}^{-1}(q,k) \,. \tag{48.80}$$

In [48.47], $\hat{F}(q,k)$ is realized as an adaptive FIR filter that minimizes the error energy

$$\mathcal{E}\{|\hat{F}^{\mathrm{c}}(q,k)r[k] - \hat{F}(q,k)\hat{S}(q,k)r[k]|^2\} \,. \tag{48.81}$$

In [48.17], the feedback path $F(q,k)$ is identified with a two-stage method, requiring only two adaptive filters. In a first stage, the sensitivity function $S(q,k)$ from $r[k]$ to $u[k]$ is determined using a standard adaptive filter. The output $z_1[k] = \hat{S}(q,k)r[k]$ of the first adaptive filter is then fed to the input of a second adaptive filter $\hat{F}(q,k)$ that minimizes

$$\mathcal{E}\left\{\left|y[k] - \hat{F}(q)z_1[k]\right|^2\right\} \,. \tag{48.82}$$

Both implementations require the use of an external identification signal $r[k]$. The identification of $F^{\mathrm{c}}(q,k)$, $S(q,k)$ and $F(q,k)$ is impeded by the presence of the desired signal $x[k]$. To preserve sound quality, the level of the probe signal $r[k]$ should be low with respect to the desired output $G(q)x[k]$. To avoid a large variance in the feedback path estimate, a slow adaptation speed should be used for the adaptive filters [48.47], at the expense of a poor tracking performance. In addition, large filter lengths are required to identify the closed-loop transfer functions $\hat{F}^{\mathrm{c}}(q,k)$ and $\hat{S}(q,k)$ accurately, as they can easily contain narrowband peaks when the closed-loop system $F^{\mathrm{c}}(q,k)$ is close to instability (e.g., for high forward path gains).

Amazon EX1010 Page 995

48.5 Comparison

This section compares the performance of different adaptive feedback cancellation techniques. We concentrate on techniques that do not require the use of an external probe signal $r[k]$ during adaptation, i. e., CAF, C-CAF, BL-CAF, and PEMAFC (with fixed and adaptive AR model).

48.5.1 Steady-State Performance

Figure 48.13 compares the steady-state performance, i. e., the misalignment upon convergence, of the C-CAF ($\eta = 0.04$), BL-CAF, PEMAFC-f, and PEMAFC-a for the acoustic feedback path with a mobile phone attached to the head and real speech as a desired signal $x[k]$. The forward path gain is 10 dB. Figure 48.13a depicts the misalignment of CAF, C-CAF, and BL-CAF as a function of the delay d_G; Fig. 48.13b compares the misalignment of CAF, PEMAFC-f, and PEMAFC-a. For all delays $d_G \in [1, 10]$ ms, C-CAF, BL-CAF, and PEMAFC achieve a smaller misalignment than the CAF: depending on the delay d_G and the algorithm, a reduction in misalignment of 1 to 8.5 dB is achieved. PEMAFC-a generally achieves the best performance. PEMAFC-f performs 1–2 dB worse than the PEMAFC-a. For delays $d_G > 3$ ms, BL-CAF and PEMAFC-a have about the same steady-state misalignment.

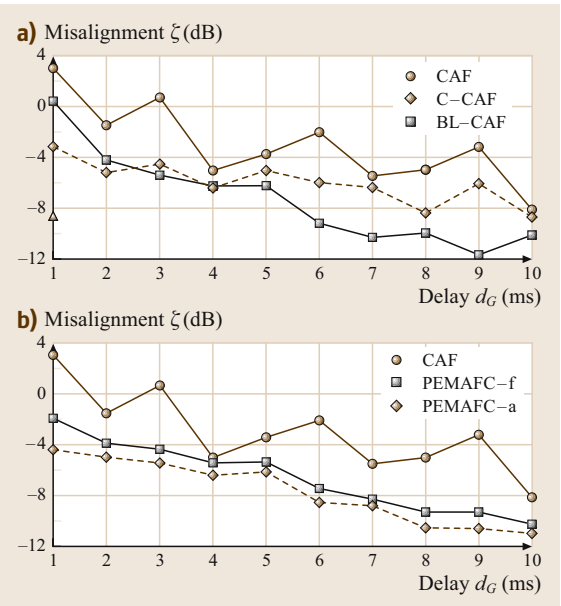


Fig. 48.13a,b Comparison of steady-state performance as a function of the delay d_G of $G(q, k)$ ($G = 10$ dB) for the acoustic feedback path with a mobile phone attached to the head. **(a)** Misalignment of CAF, C-CAF, and BL-CAF. **(b)** Misalignment of CAF, PEMAFC-f, and PEMAFC-a. Real speech as a desired signal $x[k]$

48.5.2 Tracking Performance

To illustrate the convergence and tracking performance of the algorithms, an abrupt change of the feedback path was simulated after 10 s by switching the test condition from the scenario without obstruction to the scenario with a mobile phone attached to the head. After 20 s the mobile phone was removed again. The forward path gain and delay were $G = 17$ dB and $d_G = 4$ ms. Figure 48.14a depicts the misalignment of CAF, C-CAF, and BL-CAF as a function of time; Fig. 48.14b depicts the misalignment of CAF, PEMAFC-f, and PEMAFC-a.

When the sudden changes in feedback path occur, the hearing-aid system temporarily becomes unstable. For this scenario, all algorithms succeed in restabilizing the hearing aid after the feedback path changes. The faster the convergence of the algorithm, the shorter the duration of instability. Before 10 s and after 20 s, C-CAF and BL-CAF converge faster than CAF and have a smaller misalignment. Between 10 s and 20 s, the CAF performs quite well. During this time period, the loop gain and hence the energy ratio of the feedback signal to the (disturbing) desired signal is high, explaining the good performance. The PEMAFC methods pre-whiten the desired signal (which acts as a disturbance) to some extent, generally resulting in a faster convergence and hence, better tracking compared to CAF, C-CAF, and BL-CAF. Among all algorithms, PEMAFC-a has the fastest convergence and best tracking performance thanks to the adaptive pre-whitening of the desired signal.

48.5.3 Measurement of the Actual Maximum Stable Gain

To illustrate the effective benefit of the different algorithms for a hearing aid user, the MSG of the algorithms was determined by increasing the forward path gain until audible distortion (such as howling) occurred. (This gain was smaller than the gain at which instability occurred.) The desired signal $x[k]$ was a real speech sig-

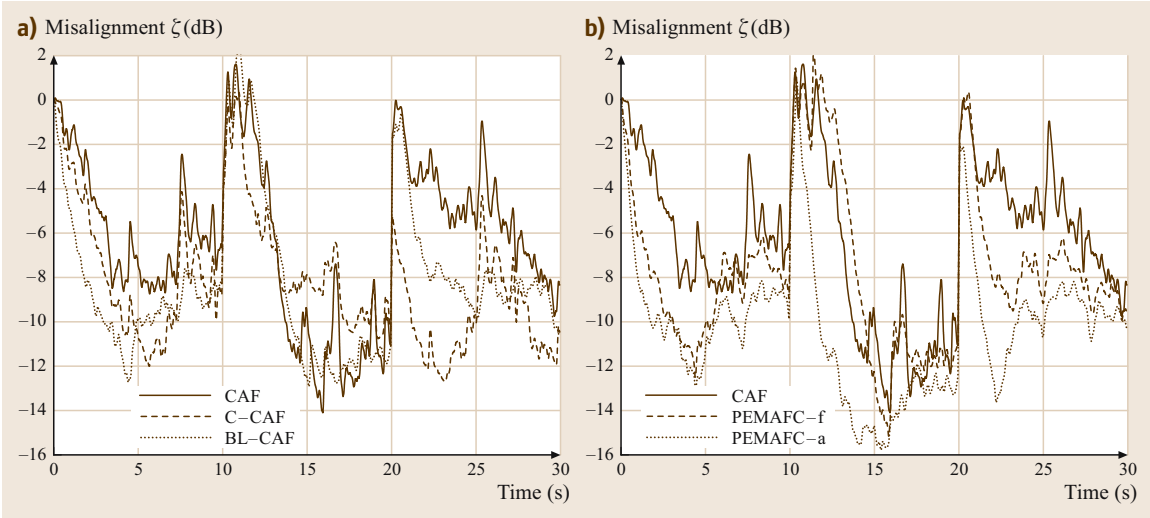


Fig. 48.14a,b Convergence and tracking performance of the various algorithms. At 10 s and 20 s a sudden change in the feedback path occurs. **(a)** Misalignment as a function of time of CAF, C-CAF, and BL-CAF. **(b)** Misalignment as a function of time of CAF, PEMAFC-f, and PEMAFC-a

nal, consisting of sentences spoken by a male speaker. The initial gain in the forward path was set at 6 dB below the stability margin. After 1 s of convergence, the gain was gradually and slowly increased in steps of 0.1 dB each 0.5 s, while the adaptive algorithm was running.

Table 48.3 compares the MSG without feedback cancellation (abbreviated in Table 48.3 as No FC) with the MSG that was achieved with the different algorithms. Three forward path delays d_G are considered, i. e., $d_G = 2$ ms, $d_G = 4$ ms, and $d_G = 6$ ms. For $d_G = 2$ ms, the benefit in MSG by the CAF is small: without obstruction, the MSG even decreases. Increasing the delay d_G significantly improves the performance of the CAF (see Fig. 48.5). Indeed, for $d_G = 4$ ms and 6 ms, the CAF on average increases the MSG of the hearing aid by 13.5 dB (without obstruction) and 19.5 dB (with a mobile phone).

For the scenario without obstruction, the C-CAF increases the MSG of the hearing aid by 15 dB. Especially for $d_G = 2$ ms, the C-CAF outperforms the CAF. For the scenario with a mobile phone, the C-CAF does not achieve better performance than the CAF. For this scenario, the reference filter in the C-CAF significantly deviates from the true feedback path (see Fig. 48.7), explaining the worse performance. The BL-CAF improves the MSG of the hearing aid by 14 dB (without obstruction) and 11 dB (with a mobile phone). For $G > 30$ dB, the frequencies below the cut-off frequency $f_c = 1$ kHz become critical. This explains why the MSG of the

Table 48.3 Maximum stable gain (MSG) in dB achieved with the various algorithms for a real speech signal $x[k]$. For comparison, the MSG achieved without feedback canceller is also shown ('No FC')

	No FC	CAF	C-CAF	BL-CAF	PEMAFC-f	PEMAFC-a
No obstruction						
$d_G = 2$ ms	17 dB	14 dB	32 dB	31 dB	30 dB	41 dB
$d_G = 4$ ms	17 dB	33 dB	33 dB	27 dB	35 dB	37 dB
$d_G = 6$ ms	16 dB	27 dB	31 dB	35 dB	37 dB	39 dB
Mobile phone						
$d_G = 2$ ms	9 dB	16 dB	17 dB	27 dB	33 dB	39 dB
$d_G = 4$ ms	9 dB	31 dB	22 dB	26 dB	38 dB	38 dB
$d_G = 6$ ms	10 dB	27 dB	23 dB	27 dB	37 dB	39 dB

Amazon EX1010 Page 997

BL-CAF with $f_c = 1\,\text{kHz}$ is limited to about 30 dB. PEMAFC-f and PEMAFC-a increase the MSG of the hearing aid by 17 dB and 22 dB, respectively, for the scenario without obstruction and by 27 dB and 29 dB, respectively, for the scenario with a mobile phone. For all three delays, they outperform the CAF. PEMAFC-a especially offers a benefit compared to PEMAFC-f for low delays d_G.

48.6 Conclusions

This chapter gave an overview of existing feedforward suppression and, in particular, feedback cancellation techniques. The most promising approach to reduce acoustic feedback is the use of a continuous adaptation feedback (CAF) canceller. However, standard continuous adaptation feedback cancellers (CAF) fail to provide a reliable feedback path estimate when the desired signal is spectrally colored.

Existing techniques to reduce the bias of the standard CAF were discussed. Common approaches include introducing signal-decorrelating operations in the forward path or reducing the adaptation speed of the feedback canceller. Other techniques control the adaptation of the feedback canceller based on prior knowledge of the acoustic feedback path through constrained (C-CAF) or bandlimited adaptation (BL-CAF). A final class of techniques views the feedback path as part of a closed-loop system and applies closed-loop system identification using the direct, the indirect or joint input–output method. The direct method seems especially appealing for feedback cancellation, since it relies on neither assumptions about the forward path nor on the use of an external excitation signal. In this method, the bias is reduced by incorporating a stationary (PEMAFC-f) or a time-varying (PEMAFC-a) model of the desired signal.

The performance of CAF, C-CAF, BL-CAF, and PEMAFC was compared for real speech as a desired signal based on acoustic feedback paths measured in a commercial hearing aid. Simulations showed that C-CAF, BL-CAF, and PEMAFC all reduce the bias of the CAF, especially for small processing delays. Among all these methods, PEMAFC with adaptive desired signal model (PEMAFC-a) achieved the best tracking performance and largest maximum stable gain.

References

48.1 U. Forssell: Closed-loop Identification – Methods, Theory and Applications. Ph.D. Thesis (Linköping Universitet, Linköping 1999)

48.2 L. Ljung, T. Söderström: *Theory and Practice of Recursive Identification* (MIT, Cambridge 1983)

48.3 J. Hellgren, T. Lunner, S. Arlinger: System identification of feedback in hearing aids, J. Acoust. Soc. Am. **105**(6), 3481–3496 (1999)

48.4 J. Hellgren, T. Lunner, S. Arlinger: Variations in the feedback of hearing aids, J. Acoust. Soc. Am. **106**(5), 2821–2833 (1999)

48.5 B. Rafaely, M. Roccasalva-Firenze, E. Payne: Feedback path variability modeling for robust hearing aids, J. Acoust. Soc. Am. **107**(5), 2665–2673 (2000)

48.6 M.R. Stinson, G.A. Daigle: Effect of handset proximity on hearing aid feedback, J. Acoust. Soc. Am. **115**(3), 1147–1156 (2004)

48.7 D.K. Bustamante, T.L. Worrall, M.J. Williamson: Measurement and adaptive suppression of acoustic feedback in hearing aids, Proc. ICASSP, Vol. 3 (Glasgow 1989) pp. 2017–2020

48.8 V. Hohmann, V. Hamacher, I. Holube, B. Kollmeier, T. Wittkop: Method for the operation of a hearing aid device or hearing device system as well as hearing aid device or hearing aid device system, US Patent 0176595 (2002), A1

48.9 J.A. Maxwell, P.M. Zurek: Reducing acoustic feedback in hearing aids, IEEE Trans. Speech Audio Process. **3**(4), 304–313 (1995)

48.10 R. Porayath, J. Daniel: Acoustic feedback elimination using adaptive notch filter algorithm, US Patent 5999631 (1999)

48.11 R. Wang, R. Harjani: Acoustic feedback cancellation in hearing aids, Proc. ICASSP, Vol. 1 (1993) pp. 137–140

48.12 J.L. Nielsen, U.P. Svensson: Performance of some linear time-varying systems in control of acoustic feedback, J. Acoust. Soc. Am. **1**(106), 240–254 (1999)

48.13 M.R. Schroeder: Improvement of acoustic-feedback stability by frequency shifting, J. Acoust. Soc. Am. **36**(9), 1718–1724 (1976)

48.14 J.M. Kates: Room reverberation effects in hearing aid feedback cancellation, J. Acoust. Soc. Am. **109**(1), 367–378 (2001)

48.15 H.-F. Chi, S.X. Goa, S.D. Soli, A. Alwan: Band-limited feedback cancellation with a modified

Amazon EX1010 Page 998

filtered-X LMS algorithm for hearing aids, Speech Commun. **39**(1–2), 147–161 (2003)
48.16 J.E. Greenberg, P.M. Zurek, M. Brantley: Evaluation of feedback-reduction algorithms for hearing aids, J. Acoust. Soc. Am. **108**(5), 2366–2376 (2000)
48.17 N.A. Shusina, B. Rafaely: Unbiased adaptive feedback cancellation in hearing aids by closed-loop identification, IEEE Trans. Audio Speech Lang. Process. **14**(2), 658–665 (2006)
48.18 A. Spriet, I. Proudler, M. Moonen, J. Wouters: Adaptive feedback cancellation in hearing aids with linear prediction of the desired signal, IEEE Trans. Signal Process. **53**(10), 3749–3763 (2005)
48.19 J.M. Kates: Feedback cancellation in hearing aids: Results from a computer simulation, Signal Process. **39**(3), 553–562 (1991)
48.20 Y. Park, D. Kim, I. Kim: An efficient feedback cancellation for multiband compression hearing aids, Proc. 20th Annu. Int. Conf. Eng. Med. Biol. Soc., Vol. 5 (1998) pp. 2706–2709
48.21 S. Thipphayathetthana, C. Chinrungrueng: Variable step-size of the least-mean-square algorithm for reducing acoustic feedback in hearing aids, IEEE Asi-Pacific Conf. on Circuits and Systems (2000) pp. 407–410
48.22 J.M. Kates: Adaptive feedback cancellation in hearing aids. In: *Adaptive Signal Processing: Applications to Real-World Problems*, ed. by J. Benesty, Y. Huang (Springer, Berlin, Heidelberg 2003) pp. 23–55
48.23 J.E. Greenberg: Modified LMS algorithms for speech processing with an adaptive noise canceller, IEEE Trans. Speech Audio Process. **6**(4), 338–350 (1998)
48.24 T. Fillon, J. Prado: Acoustic feedback cancellation for hearing-aids, using multi-delay filter, 5th Nordic Signal Process. Symp. (NORSIG) (on board Hurtigruten, 2002)
48.25 A. Kaelin, A. Lindgren, S. Wyrsch: A digital frequency-domain implementation of a very high gain hearing aid with compensation for recruitment of loudness and acoustic echo cancellation, Signal Process. **64**(1), 71–85 (1998)
48.26 J.-S. Soo, K. Pang: Multidelay block frequency domain adaptive filter, IEEE Trans. Acoust. Speech Signal Process. **38**(4), 788–798 (1990)
48.27 A. Spriet, G. Rombouts, M. Moonen, J. Wouters: Adaptive feedback cancellation in hearing aids, J. Franklin Inst. *343*, 545–572 (2006)
48.28 M.G. Siqueira, A. Alwan: Steady-state analysis of continuous adaptation in acoustic feedback reduction systems for hearing-aids, IEEE Trans. Speech Audio Process. **8**(4), 443–453 (2000)
48.29 M.A. Stone, B.C.J. Moore: Tolerable hearing aid delays. I. estimation of limits imposed by the auditory path alone using simulated hearing losses, Ear Hear. **20**(3), 182–191 (1999)
48.30 J. Hellgren: Analysis of feedback cancellation in hearing aids with filtered-X LMS and the direct method of closed loop identification, IEEE Trans. Speech Audio Process. **10**(2), 119–131 (2002)
48.31 S. Laugesen, K.V. Hansen, J. Hellgren: Acceptable delays in hearing aids and implications for feedback cancellation, J. Acoust. Soc. Am. **105**(2), 1211–1212 (1999)
48.32 M. Nilsson, S.D. Soli, A. Sullivan: Development of the Hearing in Noise Test for the measurement of speech reception thresholds in quiet and in noise, J. Acoust. Soc. Am. **95**(2), 1085–1099 (1994)
48.33 A.M. Engebretson, M.F. George: Properties of an adaptive feedback equalization algorithm, J. Rehabil. Res. Devel. **30**(1), 8–16 (1993)
48.34 C.P. Janse, P.A.A. Timmermans: Signal amplifier system with improved echo cancellation, US Patent 5748751 (1998)
48.35 H.A.L. Joson, F. Asano, Y. Suzuki, S. Toshio: Adaptive feedback cancellation with frequency compression for hearing aids, J. Acoust. Soc. Am. **94**(6), 3248–3254 (1993)
48.36 T. Kaulberg: A hearing aid with an adaptive filter for suppression of acoustic feedback, European Patent 1191814 (2002)
48.37 J. Nielsen, M. Ekelid: Feedback cancellation using bandwidth detection, WO Patent 01/06746*A2* (2001)
48.38 J. Nielsen, M. Ekelid: Feedback cancellation with low frequency input, WO Patent 01/06812*A1* (2001)
48.39 J.M. Kates: Constrained adaptation for feedback cancellation in hearing aids, J. Acoust. Soc. Am. **106**(2), 1010–1019 (1999)
48.40 S. Gao, S. Soli, H.-F. Chi: Band-limited adaptive feedback canceller for hearing aids, European Patent 1118247 (1999)
48.41 B. Rafaely, N.A. Shusina, J.L. Hayes: Robust compensation with adaptive feedback cancellation in hearing aids, Speech Commun. **39**(1–2), 163–170 (2003)
48.42 S.D. Soli, P. Widin, K.M. Buckley: Auditory prosthesis, noise suppression apparatus and feedback suppression apparatus having focused adaptive filtering, US Patent 5402496 (1993)
48.43 A. Spriet, I. Proudler, M. Moonen, J. Wouters: An instrumental variable method for adaptive feedback cancellation in hearing aids, Proc. ICASSP, Vol. 3 (Philadelphia 2005) pp. 129–132
48.44 U. Forssell, L. Ljung: Closed-loop identification revisited, Automatica **35**, 1215–1241 (1999)
48.45 J. Hellgren, U. Forssell: Bias of feedback cancellation algorithms in hearing aids based on direct closed loop identification, IEEE Trans. Speech Audio Process. **9**(7), 906–913 (2001)
48.46 R. Leber, W. Schaub: Circuit and method for the adaptive suppression of an acoustic feedback, US Patent 6611600 (2003)

Amazon EX1010 Page 999

48.47 N.A. Shusina, B. Rafaely: Feedback cancellation in hearing aids based on indirect closed-loop identification, Proc. IEEE Benelux Signal Process. Symp. (SPS) (Leuven 2002) pp. 177–180

48.48 G. Rombouts, T. van Waterschoot, K. Struyve, M. Moonen: Acoustic feedback suppression for long acoustic paths using a nonstationary source model, IEEE Trans. Signal Process. **54**(9), 3426–3434 (2004), accepted for publication, Available as Technical Report ESAT-SISTA/TR 2004-151, K.U.

48.49 J.R. Deller, J.G. Proakis, J.H.L. Hansen: *Discrete-Time Processing of Speech Signals* (Macmillan, Englewood Cliffs 1993)

48.50 N. Bisgaard, O. Dyrlund: Acoustic feedback part 2: A digital system for suppression of feedback, Hear. Instrum. **42**, 44–45 (1991)

48.51 H.R. Skovgaard: Hearing aid compensating for acoustic feedback, US Patent 5680467 (1997)

48.52 J.H. Stott, N.D. Wells: Method and apparatus for reduction of unwanted feedback, US Patent 6269165 (2001)

Amazon EX1010 Page 1000

49. Active Noise Control

S. M. Kuo, D. R. Morgan

This chapter introduces principles, algorithms, and applications of active noise control (ANC) systems. We emphasize the practical aspects of ANC systems in terms of adaptive algorithms for real-world applications. The basic algorithm for ANC is first developed and analyzed based on broadband feedforward control. This algorithm is then modified for narrowband feedforward and adaptive feedback ANC systems. Finally, these single-channel ANC algorithms are expanded to multichannel systems for three-dimensional applications.

An ANC system can be categorized as either feedforward or feedback control. Feedforward ANC systems are classified into

1. broadband control with a reference sensor, which will be discussed in Sect. 49.1, and
2. narrowband control with a reference sensor that is not influenced by the control field, which will be introduced in Sect. 49.2.

The concept of adaptive feedback ANC will be developed in Sect. 49.3 from the standpoint of reference signal synthesis. These single-channel ANC systems will be expanded to multichannel systems in Sect. 49.4.

49.1 **Broadband Feedforward Active Noise Control** 1002
49.1.1 Filtered-X LMS Algorithm 1003
49.1.2 Analysis of the FXLMS Algorithm 1004
49.1.3 Leaky FXLMS Algorithm 1004
49.1.4 Feedback Effects and Solutions 1005

49.2 **Narrowband Feedforward Active Noise Control** 1006
49.2.1 Introduction 1006
49.2.2 Waveform Synthesis Method 1007
49.2.3 Adaptive Notch Filters 1008
49.2.4 Multiple-Frequency ANC 1009
49.2.5 Active Noise Equalization 1010

49.3 **Feedback Active Noise Control** 1010

49.4 **Multichannel ANC** 1011
49.4.1 Principles 1011
49.4.2 Multichannel FXLMS Algorithms 1012
49.4.3 Frequency-Domain Convergence Analysis 1013
49.4.4 Multichannel IIR Algorithm 1014
49.4.5 Multichannel Adaptive Feedback ANC Systems 1015

49.5 **Summary** 1015

References 1015

Traditional acoustic noise control uses passive techniques such as enclosures, barriers, and silencers to attenuate undesired noise. These passive methods achieve high attenuation over a broad frequency range; however, they are relatively large, costly, and ineffective at low frequencies. Active noise control (ANC) [49.1–7] uses an electroacoustic or electromechanical system to cancel the primary (unwanted) noise, based on the principle of superposition. That is, an antinoise of equal amplitude but opposite phase is generated through a secondary source and combined with the primary noise, thus resulting in the cancelation of both noises. ANC is developing rapidly because it efficiently attenuates low-frequency noises, often with potential benefits in size, weight, volume, and cost.

An acoustic ANC system utilizing a microphone and a loudspeaker to generate a canceling sound was first proposed in [49.8]. Since the characteristics of the undesired noise are changing, an ANC system must be adaptive to track these variations. An adaptive filter [49.9, 10] updates its coefficients to minimize an error signal based on a predetermined adaptive algorithm. The most commonly used adaptive filter for practical ANC applications is the finite impulse response (FIR) filter updated by the least-mean-square (LMS) algorithm. An early duct ANC system based on adaptive filter theory was developed in [49.11].

Most ANC systems are digital, where signals from sensors are sampled and processed in real

Amazon EX1010 Page 1001

time using digital signal processing (DSP) systems. Advanced development of DSP chips [49.12] enabled low-cost implementation of powerful adaptive algorithms and thus encouraged real-world application of ANC systems. The continuous progress of ANC involves the development of advanced DSP algorithms, transducers, and hardware. More-sophisticated adaptive algorithms allow faster convergence and greater noise attenuation, and more-powerful DSP hardware allows these advanced ANC systems to be implemented in real time to improve performance.

Many practical considerations arise when an ANC system is deployed in real applications. An approach to analyze ANC performance involves a hierarchy of techniques, starting with an ideal simplified problem and progressively adding practical constraints and other complexities [49.13]. Performance analysis resolves the following issues:

1. fundamental performance limitations
2. practical constraints that limit performance
3. how to balance performance against complexity
4. how to determine a practical design architecture

For industrial applications, the ANC system must have the following properties [49.14]:

1. maximum efficiency over the frequency band to cancel a wide range of noise,
2. the system can be built and preset in the factory and then installed on site,
3. self-adaptability to deal with any variations in the physical parameters, and
4. robustness and reliability.

49.1 Broadband Feedforward Active Noise Control

This section considers broadband feedforward ANC systems that have a single reference sensor, single secondary source, and single error sensor. The general single-channel acoustic ANC system in a duct is illustrated in Fig. 49.1 [49.2], where the reference signal $x(n)$ is picked up by a microphone. This reference signal is processed by the ANC system to generate the secondary signal $y(n)$ to drive a loudspeaker. The error microphone senses the residual noise $e(n)$ and uses it to optimize the performance of the ANC system.

The ANC system shown in Fig. 49.1 can be modeled in an adaptive system identification framework as illustrated in Fig. 49.2 [49.2], in which an adaptive filter $W(z)$ is used to estimate an unknown plant $P(z)$. The primary path $P(z)$ consists of the acoustic response from the reference sensor to the error sensor. The adaptive filter $W(z)$ minimizes the residual error $e(n)$ by generating the filter output $y(n)$ to approximate the primary noise $d(n)$. When $d(n)$ and $y(n)$ are acoustically combined, the residual error is reduced, which results in cancelation of both sounds based on the principle of superposition. The most important difference between Fig. 49.2 and the traditional system identification scheme is the use of an acoustic summing junction instead of the subtraction of $y(n)$ from $d(n)$ in the electrical domain. The performance of ANC is dependent on the coherence of $d(n)$ and $x(n)$. In order to minimize the residual error, it is necessary to maximize the coherence at frequencies for which there is significant noise energy.

The use of an adaptive filter for ANC applications is complicated by the acoustic summing junction, which represents acoustic superposition in the acous-

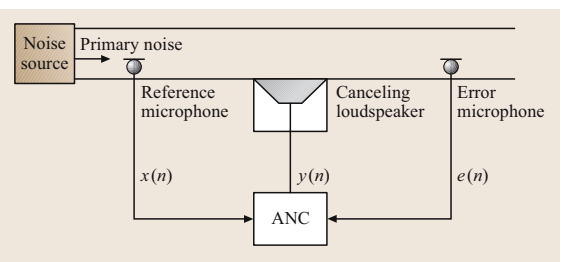


Fig. 49.1 Configuration of single-channel acoustic ANC system in a duct

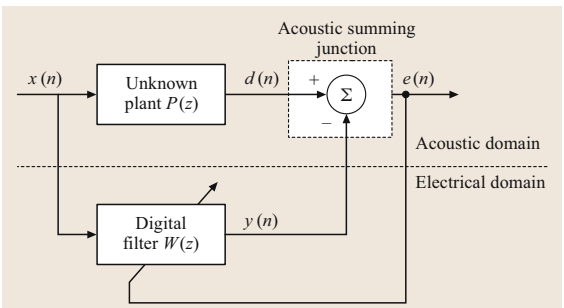


Fig. 49.2 Modeling of ANC using adaptive system identification scheme

Amazon EX1010 Page 1002

tic domain from the canceling loudspeaker to the error microphone. Therefore, it is necessary to compensate for the secondary-path transfer function $S(z)$ from $y(n)$ to $e(n)$, which includes the digital-to-analog converter, reconstruction filter, power amplifier, loudspeaker, the acoustic path from loudspeaker to error microphone, error microphone, preamplifier, anti-aliasing filter, and analog-to-digital converter.

49.1.1 Filtered-X LMS Algorithm

The introduction of the secondary path $S(z)$ into an ANC system using the LMS algorithm will cause instability because the error signal is not correctly *aligned* in time with the reference signal. There are several techniques that can be used to compensate for the effect of $S(z)$, and an effective solution is to place an identical filter in the reference signal path to the weight update of the LMS algorithm, which realizes the filtered-X LMS (FXLMS) algorithm [49.15]. The FXLMS algorithm was independently derived by *Widrow* [49.16] in the context of adaptive control and *Burgess* [49.11] for ANC applications.

The placement of the secondary-path transfer function $S(z)$ following the digital filter $W(z)$ updated by the LMS algorithm is shown in Fig. 49.3 [49.2]. The residual noise is expressed as

$$e(n) = d(n) - s(n) * [\boldsymbol{w}^{\mathrm{T}}(n)\, \boldsymbol{x}(n)] \,, \tag{49.1}$$

where $s(n)$ is the impulse response of secondary path $S(z)$, the asterisk denotes linear convolution, T denotes vector transpose, $\boldsymbol{w}(n) = [w_0(n)\ w_1(n)\ \ldots\ w_{L-1}(n)]^{\mathrm{T}}$ and $\boldsymbol{x}(n) = [x(n)\, x(n-1)\ \ldots\ x(n-L+1)]^{\mathrm{T}}$ are the coefficient and signal vectors, respectively, and L is the filter length.

We assume that the adaptive filter minimizes the instantaneous squared error $e^2(n)$ using the steepest-descent algorithm, which updates the coefficient vector in the negative gradient direction with step size μ. The FXLMS algorithm can then be derived as [49.2]

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu \boldsymbol{x}'(n) e(n) \,, \tag{49.2}$$

where the step size (or convergence factor) μ determines the convergence speed, $\boldsymbol{x}'(n) = [x'(n)\, x'(n-1)\ \ldots\ x'(n-L+1)]^{\mathrm{T}}$ is the filtered signal vector, and $x'(n) = s(n) * x(n)$. In practical ANC applications, $S(z)$ is unknown and must be estimated by an additional filter $\hat{S}(z)$. Therefore, the filtered reference signal is generated by passing the reference signal $x(n)$ through the secondary-path estimation filter described by

$$\boldsymbol{x}'(n) = \hat{s}(n) * \boldsymbol{x}(n) \,, \tag{49.3}$$

where $\hat{s}(n)$ is the impulse response of the estimated secondary-path filter $\hat{S}(z)$.

Analysis shows that, within the limit of slow adaptation, the algorithm will converge with nearly 90° of phase error between $\hat{S}(z)$ and $S(z)$ [49.15]. Therefore, offline modeling can be used in most practical applications to estimate $S(z)$ during an initial training stage. An experimental setup for offline secondary-path modeling is illustrated in Fig. 49.4 [49.2], where an internally-generated random noise $x(n)$ is used as the input to

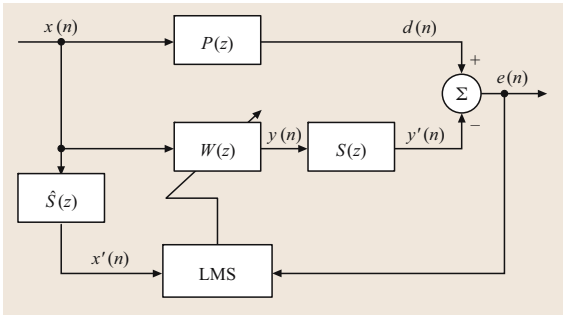


Fig. 49.3 Block diagram of single-channel ANC system using the FXLMS algorithm

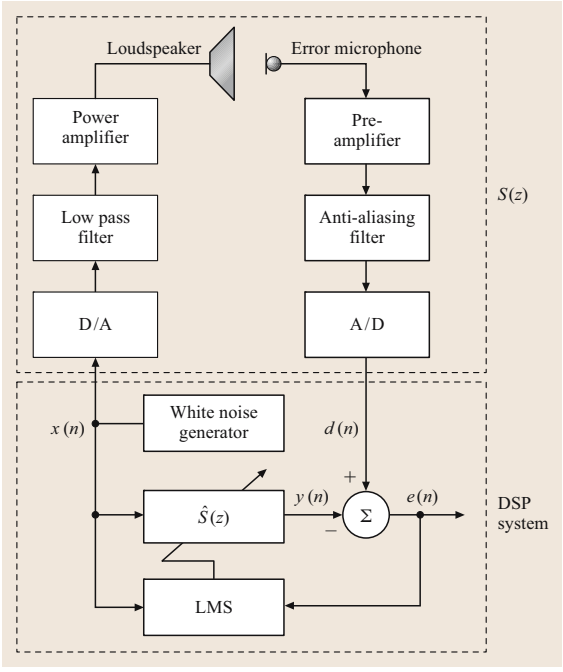


Fig. 49.4 Experimental setup for offline modeling of the secondary path

both the loudspeaker and the adaptive filter $\hat{S}(z)$ updated by the LMS algorithm, which is used to model the secondary path. After convergence of the algorithm, the adaptation is stopped and the coefficients $\hat{s}_l$, $l = 0, 1, \ldots, L-1$ are used to form the secondary-path estimate $\hat{S}(z)$. Algorithms developed for online secondary-path modeling are summarized in [49.2], which may be useful for some applications with significant changes in secondary path during ANC online operation.

49.1.2 Analysis of the FXLMS Algorithm

From Fig. 49.3, the residual error is ideally zero (i. e., $E(z) = 0$) after convergence of the adaptive filter. Therefore, the adaptive filter $W(z)$ will converge to the optimal transfer function

$$W^{\mathrm{o}}(z) = \frac{P(z)}{S(z)} \,. \tag{49.4}$$

This requires the adaptive filter $W(z)$ to simultaneously model $P(z)$ and inversely model $S(z)$. With a correct model, the system can respond instantaneously to changes in the primary noise caused by changes in the noise sources.

Assuming that the adaptive filter $W(z)$ is changing slowly and $\hat{S}(z) = S(z)$, the order of $W(z)$ and $S(z)$ in Fig. 49.3 can be commuted [49.15]. The output of the adaptive filter now carries through directly to the error signal, thus the traditional LMS algorithm analysis method can be used. This method gives accurate results if adaptation is slow by using a small step size μ. The analysis shows that the maximum step size that can be used in the FXLMS algorithm is approximately [49.17]

$$\mu_{\max} = \frac{1}{P_{x'}(L+\Delta)} \,, \tag{49.5}$$

where $P_{x'}$ is the power of the filtered reference signal $x'(n)$, and Δ is the overall delay in the secondary path in terms of the number of samples. Therefore, the secondary-path delay decreases the convergence speed of the ANC system by reducing the maximum step size that can be used in the FXLMS algorithm.

When $\hat{S}(z) \neq S(z)$, the secondary-path estimate has to match the phase within 90° for the algorithm to be stable. The effects of secondary-path modeling errors on the performance of ANC with the FXLMS algorithm are reported in [49.18, 19]. Analysis suggests that phase errors of 40° hardly affect the convergence speed of the algorithm. However, convergence will slow appreciably as the phase difference approaches 90°. Any magnitude estimation error will proportionally change the autocorrelation matrix, and hence scale the ideal stability bound accordingly. However, there is no simple relationship between phase modeling error and stability in the range ±90°.

In a broadband ANC system, measurement noises $u(n)$ and $v(n)$ are present in the reference and error signals, respectively. The optimal unconstrained transfer function $W^{\mathrm{o}}(z)$ becomes [49.2]

$$W^{\mathrm{o}}(z) = \frac{P(z)S_{xx}(z)}{[S_{xx}(z) + S_{uu}(z)]S(z)} \,, \tag{49.6}$$

where $S_{xx}(z)$ and $S_{uu}(z)$ are the power spectra of $x(n)$ and $u(n)$, respectively. This equation shows that $W^{\mathrm{o}}(z)$ is independent of the measurement noise $v(n)$ associated with the error sensor. However, the measurement noise $u(n)$ associated with the reference sensor affects the optimum weight vector, and hence reduces the cancelation performance. The best frequency response of the controller is a compromise between cancelation of the primary noise $x(n)$ and amplification of the measurement noise through the controller [49.14].

If the secondary-path transfer function $S(z)$ is modeled as a pure delay Δ, then $\hat{S}(z)$ in Fig. 49.3 can be replaced by a delay Δ. This special case of the FXLMS algorithm is known as the delayed LMS algorithm [49.20]. The upper bound for the step size depends on the delay Δ and is in close agreement with the approximation given in (49.5). Therefore, efforts should be made to keep the delay small, such as decreasing the distance between the error sensor and the secondary source and reducing the delay in electrical components as shown in Fig. 49.4.

49.1.3 Leaky FXLMS Algorithm

In an ANC application, high noise levels associated with low-frequency resonances may cause nonlinear distortion by overloading the secondary source. A solution to this problem is the introduction of output power constraints. Similar results can be obtained by constraining the adaptive filter weights by modifying the cost function as [49.21]

$$\hat{\xi}(n) = e^2(n) + \gamma \boldsymbol{w}^{\mathrm{T}}(n)\boldsymbol{w}(n) \,, \tag{49.7}$$

where γ is a weighting factor on the control effort. Following the derivation of the FXLMS algorithm, the update algorithm can be derived as [49.2]

$$\boldsymbol{w}(n+1) = \nu \boldsymbol{w}(n) + \mu \boldsymbol{x}'(n)e(n) \,, \tag{49.8}$$

Amazon EX1010 Page 1004

where $\nu = 1 - \mu\gamma$ is the leakage factor. This leaky FXLMS algorithm also reduces numeric error in the finite-precision implementation [49.22]. The introduction of a leakage factor has a considerable stabilizing effect on the adaptive algorithm, especially when very large source strengths are used [49.23].

The leakage has the effect of modifying the autocorrelation matrix of the input process [49.2]. All eigenvalues are positive even if some of the original input eigenvalues are zero. This guarantees a unique solution and a bounded time constant for all modes. The price of leakage is increased complexity of the weight update equation and the introduction of a bias into the converged solution.

49.1.4 Feedback Effects and Solutions

The acoustic ANC system shown in Fig. 49.1 uses a reference microphone to pick up the reference noise. Unfortunately, the antinoise output to the loudspeaker also radiates upstream to the reference microphone, resulting in a corrupted reference signal $x(n)$. The coupling of the acoustic wave from the canceling loudspeaker to the reference microphone is called acoustic feedback.

A more-general block diagram of an ANC system that includes feedback from the secondary source to the reference sensor is shown in Fig. 49.5 [49.2], where $u(n)$ is the primary noise, $x(n)$ is the signal picked up by the reference sensor, and $F(z)$ is the feedback path transfer function from the output of the adaptive filter $W(z)$ to the reference sensor. With the feedback path, the steady-state transfer function of the adaptive filter becomes [49.2]

$$W^{o}(z) = \frac{P(z)}{S(z) + P(z)F(z)} \,. \tag{49.9}$$

The simplest approach to solving the feedback problem is to use a feedback cancelation (or neutralization) filter, which is used for acoustic echo cancelation [49.24]. The electrical model of the feedback path is driven by the secondary signal, and its output estimates the acoustic feedback. A duct acoustic ANC system using the FXLMS algorithm with feedback neutralization is illustrated in Fig. 49.6 [49.2]. The feedback component of the reference signal is canceled electronically using the output of feedback neutralization filter $\hat{F}(z)$, which models the feedback path $F(z)$. Thus, the input signal $x(n)$ is computed as

$$x(n) = u(n) - \sum_{m=0}^{M-1} \hat{f}_m y(n-m-1) \,, \tag{49.10}$$

where $\hat{f}_m$ are the coefficients of the feedback neutralization filter $\hat{F}(z)$ with length M.

Since the primary noise is highly correlated with the antinoise, the adaptation of the feedback neutralization filter must be inhibited when the ANC system is in operation. This is similar to adaptive echo cancelation during periods of double-talk. Thus, the feedback neutralization filter can be obtained by using an offline modeling method for estimating the transfer function of the feedback path. Moreover, the models $\hat{S}(z)$ and $\hat{F}(z)$ can be estimated simultaneously using offline modeling [49.2].

Equation (49.9) shows that the optimal solution of the adaptive filter is generally an infinite impulse response (IIR) function. The poles of an IIR filter provide a well-matched solution with a lower-order filter. However, the disadvantages of adaptive IIR filters are

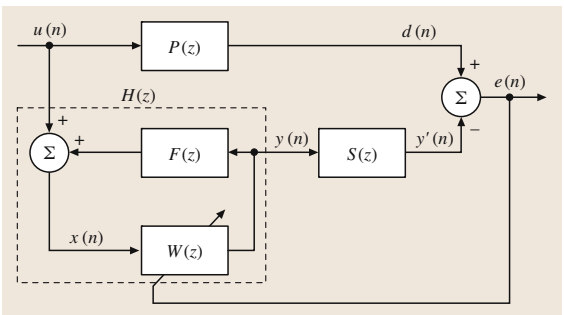


Fig. 49.5 Block diagram of ANC with acoustic feedback

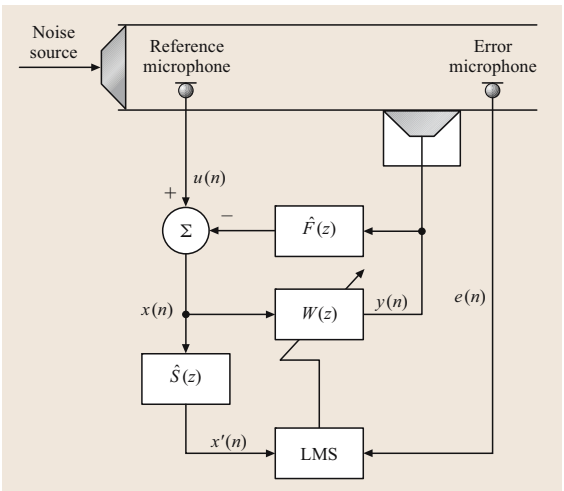


Fig. 49.6 ANC system with acoustic feedback cancelation

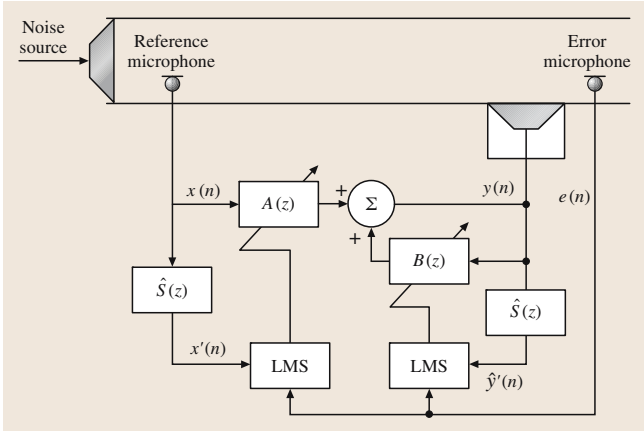


Fig. 49.7 ANC system using an adaptive IIR filter

1. IIR filters may be unstable if some pole(s) of the filter move outside of the unit circle during the adaptive process,
2. the adaptation may converge to a local minimum because the cost function of adaptive IIR filters is generally nonquadratic, and
3. IIR adaptive algorithms can have a relatively slow convergence rate in comparison with that of FIR filters.

A comprehensive discussion of adaptive IIR filters can be found in [49.25].

A block diagram of an ANC system using an adaptive IIR filter [49.26] is illustrated in Fig. 49.7 [49.2]. The output signal of the IIR filter $y(n)$ is computed as

$$y(n) = \boldsymbol{a}^{\mathrm{T}}(n)\boldsymbol{x}(n) + \boldsymbol{b}^{\mathrm{T}}(n)\boldsymbol{y}(n-1)\,, \tag{49.11}$$

where $\boldsymbol{a}(n) \equiv [a_0(n)\, a_1(n)\, \ldots\, a_{L-1}(n)]^{\mathrm{T}}$ is the weight vector of $A(z)$, $\boldsymbol{x}(n)$ is the reference signal vector, $\boldsymbol{b}(n) \equiv [b_1(n)\, b_2(n)\, \ldots\, b_M(n)]^{\mathrm{T}}$ is the weight vector of $B(z)$, and $\boldsymbol{y}(n-1)$ is the output signal vector delayed by one sample. Based on the recursive LMS algorithm [49.27], the filtered-U recursive LMS algorithm for ANC is derived as [49.28]

$$\boldsymbol{a}(n+1) = \boldsymbol{a}(n) + \mu \boldsymbol{x}'(n) e(n)\,, \tag{49.12}$$

$$\boldsymbol{b}(n+1) = \boldsymbol{b}(n) + \mu \hat{\boldsymbol{y}}'(n-1) e(n)\,, \tag{49.13}$$

where $\hat{\boldsymbol{y}}'(n-1) = \hat{s}(n) * \boldsymbol{y}(n-1)$ is the filtered version of the canceling signal vector at time $n-1$. Real-time experiments have been conducted to test the system performance for various reference microphone locations, error microphone locations, and different time-varying sources, such as a centrifugal fan and diesel engine [49.29].

49.2 Narrowband Feedforward Active Noise Control

Many noises generated by engines, compressors, motors, fans, and propellers are periodic. Direct observation of the rotation of such sources is generally possible by using an appropriate sensor, which provides information for generating an electrical reference signal that contains the fundamental frequency and all the harmonics of the primary noise.

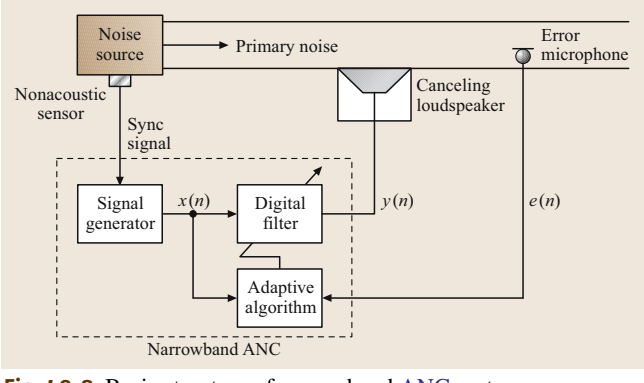


Fig. 49.8 Basic structure of narrowband ANC system

49.2.1 Introduction

A basic block diagram of narrowband ANC for reducing periodic acoustic noise in a duct is illustrated in Fig. 49.8. This system uses a synthesized reference signal $x(n)$ internally generated by the ANC system. This technique has the following advantages:

1. undesired acoustic feedback from the canceling loudspeaker back to the reference microphone is avoided,
2. nonlinearities and aging problems associated with the reference microphone are avoided,
3. the periodicity of the noise removes the causality constraint,

Amazon EX1010 Page 1006

4. the use of an internally generated reference signal results in the ability to control each harmonic independently, and
5. it is only necessary to model the transfer function over frequencies of the harmonic tones; thus, an FIR filter with substantially lower order may be used.

As shown in Fig. 49.8 [49.2], the reference signal generator is triggered by a synchronization signal from a nonacoustic sensor, such as a tachometer from an automotive engine. Two types of reference signals are commonly used in narrowband ANC systems:

1. an impulse train with a period equal to the inverse of the fundamental frequency of the periodic noise [49.30]
2. sine waves that have the same frequencies as the corresponding tones to be canceled

The first technique is called the waveform synthesis method [49.31], and the second technique is called the adaptive notch filter, which was originally developed for the cancelation of tonal interference [49.32].

49.2.2 Waveform Synthesis Method

The waveform synthesizer stores canceling noise waveform samples $\{w_l(n), l = 0, 1, \ldots, L-1\}$ in memory, where L is the number of samples over one cycle of the waveform. These samples represent the required waveform, and are sequentially sent to the secondary loudspeaker producing the actual canceling noise waveform. The data pointer to the memory buffer address can be incremented in a circular fashion between 0 and $L-1$ for each sampling period T, controlled by interrupts generated from the synchronization signal. It is important to note that most advanced digital signal processors support circular addressing in hardware [49.12].

The residual noise picked up by the error microphone is also synchronously sampled with the reference signal. In a practical system, there is a delay between the time the signal $y(n)$ is fed to the loudspeaker and the time it is received at the error microphone. This delay can be accommodated by subtracting a time offset from the circular pointer. The waveform synthesis method is equivalent to an adaptive FIR filter of order $L = N$ excited by a Kronecker impulse train of period $N = T_0/T$ samples [49.30], where $T_0 = 2\pi/\omega_0$ is the period of the noise with fundamental frequency ω_0.

For an adaptive filter with length $L = N$, the transfer function $H(z)$ between the primary input $D(z)$ and the error output $E(z)$ is derived as [49.30]

$$H(z) = \frac{E(z)}{D(z)} = \frac{1 - z^{-L}}{1 - (1-\mu)z^{-L}} \,. \tag{49.14}$$

The zeros have constant amplitude ($|z| = 1$) and are equally spaced ($2\pi/L$) on the unit circle of the z-plane to create nulls at frequencies $k\omega_0$. Therefore, the tonal components of the periodic noise at the fundamental and harmonic frequencies are attenuated by this multiple notch filter. The poles have the same frequencies as the zeros, but are equally spaced on a circle at distance $(1-\mu)$ from the origin. The effect of the poles is to reduce the bandwidth of the notch.

Equation (49.14) suggests the use of $0 < \mu < 1$ to guarantee stability. The 3 dB bandwidth of each notch for $\mu \ll 1$ is approximated as $B \approx \mu/\pi T$ (Hz) [49.30]. Thus, the bandwidth of the notch filter is proportional to the step size μ. However, the time constant of the response envelope decay is approximately $\tau \approx T/\mu$ (s). Therefore, there is a tradeoff between the notch bandwidth and the duration of the transient response, which is determined by the step size and the sampling rate of the narrowband ANC system.

As discussed in Sect. 49.1, the effects of the secondary path $S(z)$ must be compensated for by using the FXLMS algorithm. The presence of $S(z)$ modifies the transfer function of the controller to [49.33]

$$H(z) = \frac{1 - z^{-L}}{1 - [1 - \mu S(z)]z^{-L}} \,. \tag{49.15}$$

For the case of a single sinusoid, the secondary path can be modeled by a pure delay. Therefore, the compensator can be approximated as $\hat{S}(z) = z^{-\Delta}$, where Δ is the number of samples of delay from $y(n)$ to $e(n)$. The behavior of the synchronous periodic ANC with delayed coefficient update changes significantly as the delay exceeds each integer multiple of L. The steady-state transfer function $H(z)$ from $D(z)$ to $E(z)$ for the delayed LMS algorithm is [49.34]

$$H(z) = \frac{1 - z^{-L}}{1 - (1 - \mu z^{-\lfloor \Delta/L \rfloor L})z^{-L}} \,. \tag{49.16}$$

This transfer function shows that the delayed coefficient adaptation changes the pole structure. In addition to reducing the bandwidth of the notch, the inclusion of delay shifts the angle of the poles away from $k\omega_0$, and accordingly increases the out-of-band overshoot of the frequency response [49.35]. As the step size μ and/or the delay in the secondary path are increased, these out-of-band peaks become larger until the system finally becomes unstable.

49.2.3 Adaptive Notch Filters

An adaptive notch filter can be realized by using an adaptive filter with a sinusoidal reference signal. The advantages of the adaptive notch filter are that it offers easy control of bandwidth, an infinite null, and the capability to track the exact frequency of the interference adaptively. The reference input is a cosine wave $x(n) = x_0(n) = A\cos(\omega_0 n)$, where A and ω_0 are the amplitude and frequency, respectively, of the reference signal. A 90° phase shifter is used to produce the quadrature reference signal $x_1(n) = A\sin(\omega_0 n)$. The steady-state transfer function $H(z)$ from the primary input $d(n)$ to the noise canceler output $e(n)$ is [49.32]

$$H(z) = \frac{E(z)}{D(z)} = \frac{z^2 - 2z\cos\omega_0 + 1}{z^2 - (2-\mu A^2)z\cos\omega_0 + 1 - \mu A^2} \,. \quad (49.17)$$

The zeros of $H(z)$ are located on the unit circle in the z-plane at $z = e^{\pm i\omega_0}$. The adaptive noise canceler therefore acts as a tunable notch filter, with a notch located at the reference frequency ω_0. For a general adaptive filter of length L, (49.17) becomes [49.36]

$$H(z) = \frac{z^2 - 2z\cos\omega_0 + 1}{z^2 - \left(2 - \frac{\mu L A^2}{2}\right) z\cos\omega_0 + 1 - \frac{\mu L A^2}{2}} \,. \quad (49.18)$$

The sharpness of the notch is determined by the closeness of the poles to the zeros. The 3 dB bandwidth of the notch filter is estimated as [49.36]

$$B \approx \frac{\mu L A^2}{4\pi T} \quad \text{(Hz)} \,. \quad (49.19)$$

When the interfering sinusoid frequency changes rapidly or jitters around, the notch width must cover a wider frequency range. This can be accomplished by using a larger μ, which has the effect of providing faster tracking.

For $L = 2$, the autocorrelation matrix $\boldsymbol{R}$ is a diagonal matrix where the diagonal terms are equal to $A^2/2$ and the off-diagonal terms are zero [49.2]. The eigenvalues λ_1 and λ_2 of the $\boldsymbol{R}$ matrix are identical. Therefore, the system has very fast convergence since the eigenvalue spread equals 1. The $1/e$ time constant of the adaptation is approximated as

$$\tau_{\text{mse}} \le \frac{T}{\mu\lambda} = \frac{2T}{\mu A^2} \quad \text{(s)} \,, \quad (49.20)$$

which is determined by the amplitude square of the reference sine wave and the step size μ.

The application of the adaptive notch filter to periodic ANC has been developed by *Ziegler* [49.37]. A recursive quadratic oscillator provides two orthogonal components $x_0(n)$ and $x_1(n)$, which are used as reference inputs for the adaptive filter. These two signals are separately weighted and then summed to produce the canceling signal $y(n)$. The delayed LMS algorithm updates the filter weights to minimize the residual error $e(n)$:

$$w_l(n+1) = w_l(n) + \mu e(n) x_l(n-\Delta) \,, \quad l = 0, 1 \,, \quad (49.21)$$

where Δ is used to compensate for the secondary path. This frequency-dependent delay can be determined offline from the estimated secondary path.

The delay unit can be replaced by a secondary-path estimate $\hat{S}(z)$ as in the FXLMS algorithm illustrated in Fig. 49.9 [49.2]. The adaptive weights are updated as

$$w_l(n+1) = w_l(n) + \mu x_l'(n) e(n), \quad l = 0, 1 \,, \quad (49.22)$$

where $x_0'(n)$ and $x_1'(n)$ are the filtered versions of $x_0(n)$ and $x_1(n)$, respectively. In the limit of slow adaptation, the transfer function of the narrowband ANC system then becomes [49.2]

$$H(z) = \frac{z^2 - 2z\cos\omega_0 + 1}{z^2 - [2\cos\omega_0 - \beta\cos(\omega_0 - \phi_\Delta)]z + 1 - \beta\cos\phi_\Delta} \,, \quad (49.23)$$

where $\beta = \mu A^2 A_s$, A_s is the amplitude of $S(z)$ at frequency ω_0, and $\phi_\Delta = \phi_s - \phi_{\hat{s}}$ is the phase difference between $S(z)$ and $\hat{S}(z)$ at ω_0. For small β, $H(z)$ has complex conjugate poles at radius $r_p = \sqrt{1 - \beta\cos\phi_\Delta}$. Since all the terms composing β are positive, the radius of the pole can be greater than 1 only if $\cos\phi_\Delta$ is negative. Accordingly, the stability condition is

$$-90° < \phi_\Delta < 90° \quad (49.24)$$

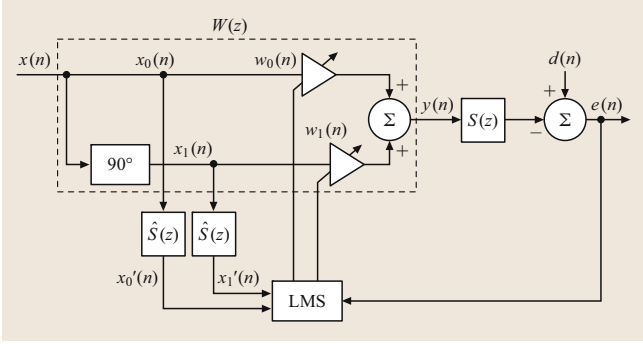


Fig. 49.9 Single-frequency ANC system using the FXLMS algorithm

Amazon EX1010 Page 1008

and the convergence time constant is slowed down by a factor of $1/\cos\phi_\Delta$ [49.15].

Another method for analyzing the stability and transient response of the adaptive notch filter using the FXLMS algorithm is to formulate the problem in the complex weight domain and apply standard control theory [49.38]. Application of this analysis technique showed that large out-of-band gain can lead to instability. The origin and characteristics of asymmetric out-of-band overshoot were investigated in [49.39] using a perturbation technique to quantify the pole displacements due the secondary path. One solution to the out-of-band gain problem is to equalize the secondary-path transfer function $S(z)$ in phase and amplitude over the entire band. An alternative solution is to control the out-of-band response by either employing a band-pass filter in the secondary path before demodulation or a low-pass filter after demodulation [49.38]. However, there is an inherent tradeoff here because in addition to attenuating out-of-band gain, a band-pass filter will also introduce delay. Another solution consists of two interconnected adaptive filters using the same internally generated reference signal [49.40].

49.2.4 Multiple-Frequency ANC

In practical ANC applications, periodic noise usually contains multiple tones at several harmonic-related frequencies. This type of noise can be attenuated by a filter with multiple notches. An adaptive filter with multiple notches can be implemented by direct, parallel, direct/parallel, or cascade form.

A method for eliminating multiple sinusoidals or other periodic interference was proposed by *Glover* [49.36]. The reference signal is a sum of M sinusoids:

$$x(n) = \sum_{m=1}^{M} A_m \cos(\omega_m n) \,, \tag{49.25}$$

where A_m and ω_m are, respectively, the amplitude and the frequency of the m-th sinusoid. When the frequencies of the reference sinusoids are close together, a long filter ($L \gg 2M$) is required to give good resolution between adjacent frequencies. This is an undesired solution since a higher-order adaptive FIR filter results in slower convergence, higher excess mean-square error, and larger numeric errors. An application of Glover's method for actively attenuating engine-generated noise was proposed in [49.41]. The relationship between the convergence rate and required filter length with the frequency separation between two adjacent harmonics is studied in [49.42] based on the analysis of eigenvalue spread.

If the primary noise contains M sinusoids, M two-weight adaptive filters can be connected in parallel to attenuate these narrowband components [49.37]. The canceling signal is a sum of M adaptive filter outputs $y_m(n)$:

$$y(n) = \sum_{m=1}^{M} y_m(n) \,. \tag{49.26}$$

Since only one error sensor is used, there is only one error signal $e(n)$ to update all M adaptive filters based on the FXLMS algorithm.

A configuration using multiple reference signal generators and corresponding adaptive filters has been developed [49.43] to improve the performance of ANC systems. The idea is to separate a collection of many harmonically related sinusoids into mutually exclusive sets that individually have frequencies spaced out as far as possible. In general, if there are M harmonics to be canceled and K ($K < M$) signal generators are used, each reference signal $x_k(n)$, $k = 1, 2, \ldots, K$ contains staggered sinusoidal frequencies of every other K-th harmonic. These reference signals are processed by their corresponding adaptive filters. By partitioning signal component frequencies in this fashion, the rate of convergence of each adaptive filter can be significantly improved. This is because the frequency difference between any two adjacent sinusoidal components in $x_k(n)$ is effectively increased, as compared to the direct implementation technique [49.42].

Ideally, multiple-sinusoid references are more effectively employed in a cascade of M second-order single-frequency notch filters. The overall response of such an arrangement is given by [49.35]

$$H(z) = \prod_{m=1}^{M} H_m(z) = \prod_{m=1}^{M} \frac{1}{1 + S(z)W_m(z)} \,, \tag{49.27}$$

where $W_m(z)$ is the m-th section adaptive filter. Each $W_m(z)$ produces a pole at frequency ω_m, and so each $H_m(z)$ produces a notch at ω_m. If an estimate of the secondary-path transfer function is available, it is possible to configure a *pseudocascade* arrangement [49.35] that ideally performs as a true cascade but requires only one secondary-path estimate $\hat{S}(z)$.

A rectangular wave $x(t)$ potentially contains the fundamental and all harmonic components of the periodic noise. The shape of the spectrum of $x(t)$ is dependent on the duty-cycle ratio τ/T_0, where τ

and T_0 are the pulse width and the fundamental period of the rectangular wave. When τ approaches zero, the rectangular wave becomes an impulse train. Usually satisfactory performance can be achieved for [49.2]

$$\frac{\tau}{T_0} \leq \frac{1}{2M} , \tag{49.28}$$

where M is the largest significant harmonic index of the primary noise. Reference-signal components above the largest significant harmonic must be removed by a low-pass filter in order to avoid aliasing problems. Applications of this principle were developed for attenuating engine noise in an earth-moving machine [49.44] and tonal blade passage noise in a multiple-blade fan [49.45].

49.2.5 Active Noise Equalization

The design of an ANC system usually pursues maximal attenuation of the primary noise. However, in some applications, it is desirable to retain a small residual error with a specified spectral shape. An ANC system that is capable of controlling residual noise spectrum shape is called an active noise equalizer [49.46]. The principle of narrowband active noise equalization can also be applied to broadband noise [49.47].

In a narrowband active noise equalizer, the output of the two-weight filter, $y(n)$, is split into two branches. The gains β and $(1-\beta)$ are inserted in these two branches to adjust the residual noise. If the secondary-path estimate is perfect (i. e., $\hat{S}(z) = S(z)$), the pseudo-error $e'(n) = d(n) - y(n)$ is the residual noise of the conventional ANC system. The adaptive filter minimizes the pseudo-error using the FXLMS algorithm. After the filter has converged, the pseudo-error $e'(n) \approx 0$, however, the real residual noise is $e(n) \approx \beta d(n)$. Thus, $e(n)$ contains a residual narrowband noise whose amplitude is totally controlled by adjusting the gain value β.

The functions of the active noise equalizer can be classified into the following four operation modes:

1. Cancelation mode: $\beta = 0$, the active noise equalizer attenuates the sinusoid completely at frequency ω_0;
2. Attenuation mode: $0 < \beta < 1$, the amount of attenuation is determined by the factor β;
3. Neutral mode: $\beta = 1$, the noise was passed without attenuation;
4. Enhancement mode: $\beta > 1$, the active noise equalizer functions as an amplifier.

49.3 Feedback Active Noise Control

In a single-channel feedback ANC system, the error sensor output is processed by an ANC system to generate the secondary signal [49.1]. A single-channel adaptive feedback ANC system was proposed in [49.48] and extended to the multichannel case in [49.49]. We view this technique as an adaptive feedforward system that synthesizes (or regenerates) its own reference signal based only on the adaptive filter output and error signal. Adaptive feedback ANC has been applied in the design of active headsets [49.50, 51].

In Fig. 49.3, the primary noise is expressed in the z-domain as $D(z) = E(z) + S(z)Y(z)$, where $E(z)$ is the signal obtained from the error sensor and $Y(z)$ is the secondary signal generated by the adaptive filter. If $\hat{S}(z) \approx S(z)$, we can estimate the primary noise $d(n)$ and use this as a synthesized reference signal $x(n)$. That is,

$$X(z) \equiv \hat{D}(z) = E(z) + \hat{S}(z)Y(z) . \tag{49.29}$$

This is the principle for reference signal synthesis (regeneration) technique, whereby the secondary signal $y(n)$ is filtered by the secondary-path estimate $\hat{S}(z)$ and then combined with $e(n)$ to regenerate the primary noise.

The complete single-channel adaptive feedback ANC system using the FXLMS algorithm is illustrated in Fig. 49.10 [49.2], where $\hat{S}(z)$ is also required to com-

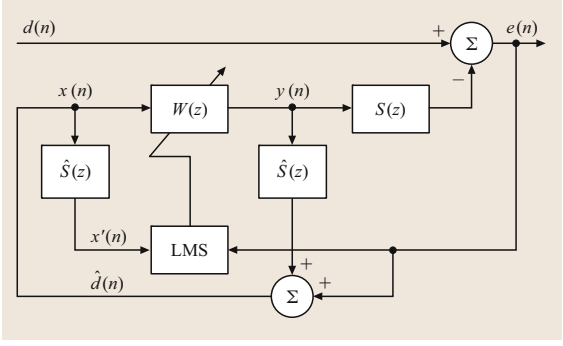


Fig. 49.10 Adaptive feedback ANC system using the FXLMS algorithm

pensate for the secondary path. The reference signal $x(n)$ is synthesized as

$$x(n) \equiv \hat{d}(n) = e(n) + \sum_{m=0}^{M-1} \hat{s}_m y(n-m) \,, \qquad (49.30)$$

where $\hat{s}_m$, $m = 0, 1, \ldots, M-1$ are the coefficients of the FIR filter $\hat{S}(z)$ used to estimate the secondary path.

From Fig. 49.10, we have $x(n) = d(n)$ if $\hat{S}(z) = S(z)$. If the step size μ of the LMS algorithm is small (slow convergence), the adaptive filter $W(z)$ can be commuted with $S(z)$ [49.9]. If we further assume that the secondary path $S(z)$ can be modeled by a pure delay, that is, $S(z) = z^{-\Delta}$, Fig. 49.10 is identical to the standard adaptive prediction scheme. The system response from $d(n)$ to $e(n)$ is called the prediction error filter $H(z)$. The adaptive filter $W(z)$ acts as an adaptive predictor of the primary noise $d(n)$ to minimize the residual noise $e(n)$, so the performance of the adaptive feedback ANC system depends on the predictability of the primary noise $d(n)$.

The feedforward ANC systems discussed in Sect. 49.1 use a reference sensor to measure the primary noise and an error sensor to monitor the performance. The adaptive feedback ANC system only uses an error sensor and thus cancels only the predictable noise components of the primary noise. A combination of the feedforward and feedback control structures is called a hybrid ANC system with both reference and error sensors, as illustrated in Fig. 49.11 [49.52]. The reference sensor is kept close to the noise source and provides a coherent reference signal for the feedforward ANC system. The error sensor is placed downstream

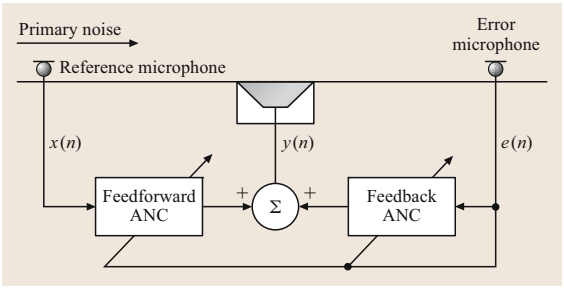


Fig. 49.11 Combination of feedforward and adaptive feedback ANC with the FXLMS algorithm

and senses the residual noise, which is used to synthesize the reference signal for the adaptive feedback ANC filter, as well as to adapt the coefficients of both the feedforward and feedback ANC filters. The feedforward ANC attenuates primary noise that is correlated with the reference signal, while the feedback ANC cancels the predictable components of the primary noise that are not observed by the reference sensor.

In the hybrid ANC system, the secondary signal $y(n)$ is generated using the outputs of both the feedforward and feedback ANC filters. A similar hybrid ANC system using the adaptive IIR feedforward ANC and the adaptive feedback ANC can be found in [49.2]. The advantage of hybrid ANC over other ANC systems is that a lower-order filter can be used to achieve the same performance. The hybrid systems also clearly demonstrate an advantage over either the simple feedforward ANC or adaptive feedback ANC system alone when there is significant plant noise.

49.4 Multichannel ANC

The noise field in an enclosure or a large-dimension duct is more complicated than in a narrow duct, so it is generally necessary to use a multichannel ANC system with several secondary sources, error sensors, and perhaps even several reference sensors.

49.4.1 Principles

Theoretical predictions, computer simulations, and laboratory experiments on harmonic ANC in a shallow enclosure were presented in [49.53–55]. The total acoustic potential energy E_p is approximated by the sum of the squares of the outputs of many error sensors distributed throughout the enclosure. This suggests that the multichannel ANC system should minimize a cost function

$$\xi = \frac{V}{4\rho c^2 M} \sum_{m=1}^{M} |p_m|^2 \,, \qquad (49.31)$$

where V is the volume, ρ is the density of the acoustic medium, c is the propagation velocity, and p_m is the sound pressure at the m-th error sensor with a total of M error microphones. Equation (49.31) shows that ξ is a quadratic function, and thus allows the efficient gradient descent method to be employed in the multichannel ANC algorithm.

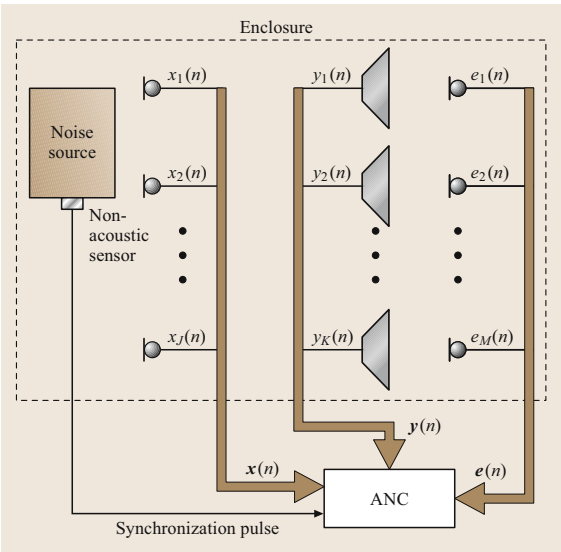


Fig. 49.12 Multichannel acoustic ANC system

For canceling low-frequency tones, only a few error sensors are needed to achieve a substantial reduction in total noise energy [49.56]. The locations of these error sensors are very important in order to obtain the best estimate of the total acoustic potential energy. The secondary sources should be located in positions where they can couple well to the acoustic modes of the enclosure. To improve the performance of the system, the distance from the primary to the secondary sources must be less than a quarter-wavelength at the highest frequency [49.57]. Subtle physical attributes of the multichannel ANC system that affect convergence are the placement of the error sensors and the location of the secondary sources.

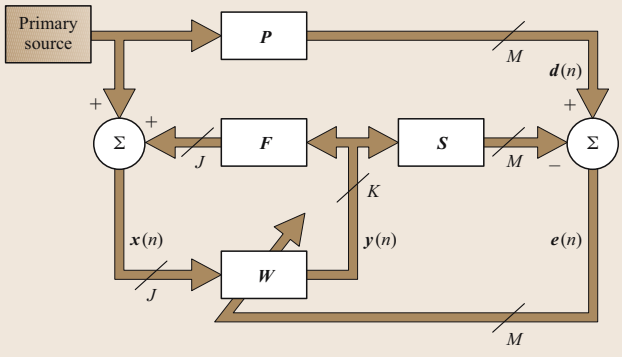


Fig. 49.13 Multichannel FXLMS algorithm

49.4.2 Multichannel FXLMS Algorithms

A multichannel acoustic ANC system is illustrated in Fig. 49.12 [49.2]. For narrowband feedforward multichannel ANC systems, a nonacoustic sensor can be used as the reference sensor to generate a reference signal. For broadband feedforward ANC, the multichannel system employs J reference sensors to form the reference signal vector. For adaptive feedback ANC, the reference signals are internally synthesized based on the secondary and error signals. The multichannel ANC system generates K canceling signals to drive the corresponding secondary sources, and M error sensors are distributed over the desired locations to measure the residual noise components.

A block diagram of a multichannel ANC system that includes feedback paths from the secondary sources to the reference sensors is shown in Fig. 49.13 [49.2]. The wide arrows represent an array of signals (acoustic or electrical) that are symbolically expressed as vectors. The matrix $\boldsymbol{P}$ represents $M \times J$ primary-path transfer functions from the primary source to each error sensor output $e_m(n)$. The matrix $\boldsymbol{S}$ represents $M \times K$ secondary-path transfer functions, $S_{mk}(z)$, from K secondary sources to M error sensors. Also, the matrix $\boldsymbol{F}$ represents $J \times K$ feedback paths from K secondary sources to J reference sensors. There are $K \times J$ possible feedforward channels, each demanding a separate adaptive filter, and these $K \times J$ adaptive filters are represented by the matrix $\boldsymbol{W}$. A complete set of multichannel FXLMS algorithms for broadband feedforward, narrowband feedforward, and adaptive feedback ANC systems are introduced in [49.2].

The single-reference/multiple-output ANC system aims to control a multidimensional noise field produced by rotating machinery such as engines [49.21]. The k-th secondary signal $y_k(n)$ is obtained by filtering the reference signal $x(n)$ through the corresponding adaptive FIR filter:

$$y_k(n) = \boldsymbol{w}_k^{\mathrm{T}}(n)\boldsymbol{x}(n)\,, \quad k = 1, 2, \ldots, K\,, \qquad (49.32)$$

where $\boldsymbol{w}_k(n) \equiv [w_{k,0}(n)\ w_{k,1}(n)\ \ldots\ w_{k,L-1}(n)]^{\mathrm{T}}$ is the weight vector of the k-th adaptive filter and $\boldsymbol{x}(n) = [x(n)\ x(n-1)\ \ldots\ x(n-L+1)]^{\mathrm{T}}$ is the common reference signal vector for all adaptive filters.

The single-reference/multiple-output FXLMS algorithm can be partitioned into K equations [49.2]

$$\boldsymbol{w}_k(n+1) = \boldsymbol{w}_k(n) + \mu \sum_{m=1}^{M} \boldsymbol{x}'_{km}(n) e_m(n)\,,$$
$$k = 1, 2, \ldots, K\,, \qquad (49.33)$$

Amazon EX1010 Page 1012

where

$$\boldsymbol{x}'_{km}(n) \equiv \hat{s}_{mk}(n) * \boldsymbol{x}(n) \,, \tag{49.34}$$

for $k = 1, 2, \ldots, K$ and $m = 1, 2, \ldots, M$, are the filtered reference signal vectors, which are formed by filtering $\boldsymbol{x}(n)$ by the secondary-path estimates $\hat{S}_{mk}(z)$ from the k-th secondary source to the m-th error sensor. In (49.34), $\hat{s}_{mk}(n)$ represents the impulse response of an FIR filter $\hat{S}_{mk}(z)$ that is used to estimate $S_{mk}(z)$. The multichannel leaky FXLMS algorithm is similar to (49.33), except that $\nu \boldsymbol{w}_k(n)$ is used to replace $\boldsymbol{w}_k(n)$.

In the general multiple-reference/multiple-output ANC system using the FXLMS algorithm, the ANC filter $\boldsymbol{W}$ has J reference input signals $x_j(n)$. Each controller in the matrix $\boldsymbol{W}$ is represented by $W_{kj}(z)$, where j is the reference input index and k is the secondary source index. The secondary signal output to the k-th secondary source is

$$y_k(n) = \sum_{j=1}^{J} \boldsymbol{w}_{kj}^{\mathrm{T}}(n) \boldsymbol{x}_j(n) \,,$$
$$k = 1, 2, \ldots, K \,, \tag{49.35}$$

where $\boldsymbol{x}_j(n) \equiv [x_j(n)\, x_j(n-1) \ldots x_j(n-L+1)]^{\mathrm{T}}$, $j = 1, 2, \ldots, J$ are the reference signal vectors. There are $M \times K$ different secondary paths $S_{mk}(z)$ between the secondary sources and error sensors, which are modeled by $\hat{S}_{mk}(z)$ to generate an array of filtered reference signals $\boldsymbol{x}'_{jkm}(n)$ for the multichannel FXLMS algorithm

$$\boldsymbol{w}_{kj}(n+1) = \boldsymbol{w}_{kj}(n) + \mu \sum_{m=1}^{M} \boldsymbol{x}'_{jkm}(n) e_m(n) \tag{49.36}$$

for $k = 1, 2, \ldots, K$ and $j = 1, 2, \ldots, J$, where the vectors

$$\boldsymbol{x}'_{jkm}(n) = \hat{s}_{mk}(n) * \boldsymbol{x}_j(n) \tag{49.37}$$

are the filtered reference signal vectors.

A practical application of multiple-reference/multiple-output active control for propeller blade-passage noise inside a 50 seat aircraft has been reported [49.17]. That system uses three reference signals (internally generated sinusoids) for the fundamental frequency and its first two harmonics, 16 secondary sources, and 32 error sensors, resulting in a $3 \times 16 \times 32$ multiple-reference/multiple-output ANC system. Since the reference signal is an internally generated sinusoid, only two coefficients are required for each of the 1536 $(3 \times 16 \times 32)$ individual secondary-path models to generate the filtered reference signals.

In some cases, it is advantageous to locally minimize the sum of the squared signals from only a few error sensors. This concept is formalized by employing independent weighting factors q_{km} to control the influence of the M error signals on each of the K adaptive filters [49.58]. The single-reference/multiple-output FXLMS algorithm given in (49.33) is then modified by using $e'_{km}(n)$ to replace $e_{km}(n)$, where

$$e'_{km}(n) \equiv q_{km} e_m(n) \tag{49.38}$$

is the error signal $e_m(n)$ weighted by q_{km} for the adaptation of $\boldsymbol{w}_k(n)$. Thus, the influence of the error signals is limited by setting some of the q_{km} equal to zero.

49.4.3 Frequency-Domain Convergence Analysis

The convergence behavior of a multichannel ANC system can be analyzed in the frequency domain in terms of the convergence of the individual secondary signals, the cost function, and the control effort [49.56]. This frequency-domain analysis provides insight into the physical process of controlling one or more harmonics in narrowband ANC systems.

To cancel a narrowband noise component of known frequency, the reference signal is a pure sinusoid, which can be considered a constant of value $X = 1$ in the frequency domain. The multichannel frequency-domain FXLMS algorithm is expressed as [49.56]

$$\boldsymbol{W}(n+1) = (\boldsymbol{I} - \mu \boldsymbol{S}^{\mathrm{H}} \boldsymbol{S}) \boldsymbol{W}(n) + \mu \boldsymbol{S}^{\mathrm{H}} \boldsymbol{D} \,, \tag{49.39}$$

where the superscript H (Hermitian) denotes conjugate transpose. Therefore, the eigenvalues of $\boldsymbol{S}^{\mathrm{H}} \boldsymbol{S}$ determine the convergence characteristics of the multichannel frequency-domain FXLMS algorithm. It is noteworthy that disparate eigenvalues arise here because of the spatial nature of the secondary-path matrix $\boldsymbol{S}$. Thus, the convergence of the general broadband multichannel FXLMS algorithm is limited by both the spatial and temporal characteristics of the system.

As with the single-channel leaky FXLMS algorithm, the cost function can be modified to include the control effort as

$$\xi(n) = \boldsymbol{E}^{\mathrm{H}} \boldsymbol{E} + \gamma \boldsymbol{W}^{\mathrm{H}} \boldsymbol{W} \,, \tag{49.40}$$

where γ is a real constant that determines the balance between reducing the total mean-square error $\boldsymbol{E}^{\mathrm{H}} \boldsymbol{E}$ and moderating the control effort $\boldsymbol{W}^{\mathrm{H}} \boldsymbol{W}$. The convergence

of the multichannel frequency-domain leaky FXLMS algorithm is guaranteed if [49.56]

$$\mu < \frac{2}{\sigma_k^2 + \gamma}\,, \quad k = 1, 2, \ldots, K\,, \tag{49.41}$$

where σ_k^2 is the k-th eigenvalue of $\boldsymbol{S}^{\mathrm{H}}\boldsymbol{S}$. The effect of γ on the convergence of the ANC system is to speed up the very slow modes of convergence, which are associated with small values of σ_k^2, by adding a constant factor γ.

Consider the unconstrained solution for $\gamma = 0$. In this case the effort can be written [49.56]

$$\boldsymbol{W}^{\mathrm{H}}(n)\boldsymbol{W}(n) \approx \sum_{k=1}^{K} \frac{|P_k|^2}{\sigma_k^2}\left[1 - \left(1 - \mu\sigma_k^2\right)^n\right]^2\,. \tag{49.42}$$

For $\mu\sigma_k^2 \ll 1$, the m-th mode requires a control effort of $|P_k|^2/\sigma_k^2$ and it converges exponentially with a time constant $1/\mu\sigma_k^2$ and contributes to the sum of the squared error $\xi(n)$ by an amount $|P_k|^2$. If σ_k^2 is small and $|P_k|^2$ is not correspondingly small, a considerable control effort may be required to reduce this component of $\xi(n)$, even though it may not be significant compared to the minimum mean-square error.

For the constrained solution $\gamma \neq 0$, the steady-state value of the control effort for a convergent system is [49.56]

$$\boldsymbol{W}^{\mathrm{o}^{\mathrm{H}}}\boldsymbol{W}^{\mathrm{o}} = \sum_{k=1}^{K} \frac{\sigma_k^2 |P_k|^2}{\left(\sigma_k^2 + \gamma\right)^2}\,. \tag{49.43}$$

The first K components of the residual error are largely removed by the action of the ANC system assuming that $\sigma_k^2 \gg \gamma$, $k = 1, 2, \ldots, K$. If $\sigma_k^2 \gg \gamma$, the factor γ plays no role in the convergence of that component of $\xi(n)$. However, if $\sigma_k^2 \ll \gamma$, the control effort is limited for the k-th mode to a value $\sigma_k^2|P_k|^2/\gamma^2$, so smaller σ_k^2 requires smaller effort.

A suitably chosen value of γ will not only improve convergence but can also prevent unreasonable values of control effort on an insignificant noise component. Moderating the control effort also considerably reduces the risk of instability due to errors in the secondary-path models [49.18]. Furthermore, the constrained algorithm is equivalent to the leaky FXLMS algorithm, which has the additional benefit of reducing the effects of numerical errors and preventing algorithm stalling.

49.4.4 Multichannel IIR Algorithm

The broadband feedforward multiple-reference/multiple-output ANC system uses adaptive IIR filters to provide longer impulse responses and feedback compensation [49.23, 59]. As illustrated in Fig. 49.14 [49.2], the controller contains two filter sections. The first section is a block of FIR filters from each reference sensor to each secondary source, while the second section implements matrix IIR filters. The feedforward section $\boldsymbol{A}(z)$ consists of elements $A_{kj}(z)$ from input $x_j(n)$ to the k-th summing node to produce $y_k(n)$, the k-th component of the output vector $\boldsymbol{y}(n)$. The transfer function matrix $\boldsymbol{B}(z)$ represents the recursive section of the controller with element $B_{ki}(z)$ from $y_i(n-1)$ to $y_k(n)$. Therefore, the secondary signal $y_k(n)$ that drives the k-th secondary source is expressed as

$$y_k(n) = \sum_{j=1}^{J} \boldsymbol{a}_{kj}^{\mathrm{T}}(n)\boldsymbol{x}_j(n) + \sum_{i=1}^{K} \boldsymbol{b}_{ki}^{\mathrm{T}}(n)\boldsymbol{y}_i(n-1)$$
$$\text{for } k = 1, 2, \ldots, K\,, \tag{49.44}$$

where $\boldsymbol{a}_{kj}(n) \equiv [a_{kj,0}(n)\, a_{kj,1}(n)\, \ldots\, a_{kj,L-1}(n)]^{\mathrm{T}}$ and $\boldsymbol{b}_{ki}(n) \equiv [b_{ki,0}(n)\, b_{ki,1}(n)\, \ldots\, b_{ki,I-1}(n)]^{\mathrm{T}}$ are the feedforward and feedback filter coefficients, respectively; $\boldsymbol{x}_j(n)$ and $\boldsymbol{y}_k(n)$ are the reference input signal vectors and output signal vectors, respectively. A multiple-reference/multiple-output filtered-U recursive LMS

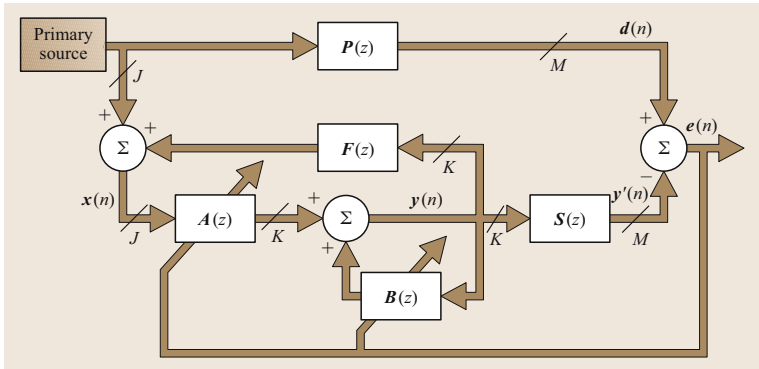


Fig. 49.14 Multichannel ANC algorithm using adaptive IIR filters

algorithm [49.59] for the IIR filter structure minimizes the sum of the M mean-square error signals:

$$\boldsymbol{a}_{kj}(n+1) = \boldsymbol{a}_{kj}(n) + \mu \sum_{m=1}^{M} \boldsymbol{x}'_{jkm}(n) e_m(n) \,, \quad (49.45)$$

$$\boldsymbol{b}_{ki}(n+1) = \boldsymbol{b}_{ki}(n) + \mu \sum_{m=1}^{M} \boldsymbol{y}'_{ikm}(n) e_m(n) \,, \quad (49.46)$$

where $\boldsymbol{x}'_{jkm}(n) \equiv \hat{s}_{mk}(n) * \boldsymbol{x}_j(n)$ and $\boldsymbol{y}'_{ikm}(n) \equiv \hat{s}_{mk}(n) * \boldsymbol{y}_i(n)$ are, respectively, the filtered reference and output signal vectors.

Multichannel adaptive IIR filtering has been successfully applied to the active control of random noise in a small reverberant room [49.23]. The matrix IIR structure results in a more-stable configuration in the presence of feedback, especially if a small leakage factor is included. The experimental results also show that far better performance is achieved by using IIR filters rather than FIR filters when the primary noise source has a lightly damped dynamic behavior.

49.4.5 Multichannel Adaptive Feedback ANC Systems

The single-channel feedback ANC system can be extended to a $K \times 1$ system that has K secondary sources and one error sensor. The reference signal $x(n)$ of the feedback ANC system is synthesized as an estimate of the primary noise

$$x(n) \equiv \hat{d}(n) = e(n) + \sum_{k=1}^{K} \hat{s}_k(n) * y_k(n) \,, \quad (49.47)$$

where $\hat{s}_k(n)$ is the impulse responses of the filter $\hat{S}_k(z)$ that models the secondary path $S_k(z)$ from the k-th secondary source to the error sensor, and the $y_k(n)$ are the secondary signals obtained from the adaptive filters $W_k(z)$.

The FXLMS algorithm is used to minimize the error signal $e(n)$ by adjusting the weight vector for each adaptive filter $W_k(z)$ according to

$$\boldsymbol{w}_k(n+1) = \boldsymbol{w}_k(n) + \mu \boldsymbol{x}'_k(n) e(n) \,, \quad k = 1, 2, \ldots, K \,, \quad (49.48)$$

where

$$\boldsymbol{x}'_k(n) \equiv \hat{s}_k(n) * \boldsymbol{x}(n) \quad (49.49)$$

is the reference signal vector filtered by the secondary-path estimate $\hat{S}_k(z)$.

49.5 Summary

This chapter has reviewed the practical aspects of ANC systems in terms of adaptive algorithms. The most widely used ANC system using the adaptive FIR filter and the FXLMS algorithm was introduced and analyzed based on single-channel broadband feedforward control. This basic system was extended to narrowband feedforward and adaptive feedback ANC systems. These single-channel ANC algorithms were then expanded to multichannel cases for controlling the noise field in an enclosure or a large-dimension duct.

References

49.1 P.A. Nelson, S.J. Elliott: *Active Control of Sound* (Academic, San Diego 1992)

49.2 S.M. Kuo, D.R. Morgan: *Active Noise Control Systems – Algorithms and DSP Implementations* (Wiley, New York 1996)

49.3 C.R. Fuller, S.J. Elliott, P.A. Nelson: *Active Control of Vibration* (Academic, San Diego 1996)

49.4 C.H. Hansen, S.D. Snyder: *Active Control of Noise and Vibration* (E&FN Spon, London 1997)

49.5 S.J. Elliott: *Signal Processing for Active Control* (Academic, San Diego 2001)

49.6 O. Tokhi, S. Veres: *Active Sound and Vibration Control: Theory and Applications* (Institute of Electrical Engineers, London 2002)

49.7 S.M. Kuo, D.R. Morgan: Active noise control: A tutorial review, Proc. IEEE **87**, 943–973 (1999)

49.8 P. Lueg: Process of silencing sound oscillations, U.S. Patent 2043416, June 9 (1936).

49.9 B. Widrow, S.D. Stearns: *Adaptive Signal Processing* (Prentice-Hall, Englewood Cliffs 1985)

49.10 S. Haykin: *Adaptive Filter Theory*, 4th edn. (Prentice Hall, Upper Saddle River 2002)

Amazon EX1010 Page 1015

49.11 J.C. Burgess: Active adaptive sound control in a duct: A computer simulation, J. Acoust. Soc. Am. **70**, 715–726 (1981)

49.12 S.M. Kuo, W.S. Gan: *Digital Signal Processors* (Prentice Hall, Upper Saddle River 2005)

49.13 D.R. Morgan: A hierarchy of performance analysis techniques for adaptive active control of sound and vibration, J. Acoust. Soc. Am. **89**, 2362–2369 (1991)

49.14 A. Roure: Self-adaptive broadband active sound control system, J. Sound Vib. **101**, 429–441 (1985)

49.15 D.R. Morgan: An analysis of multiple correlation cancellation loops with a filter in the auxiliary path, IEEE Trans. Acoust. Speech Signal Process. **ASSP-28**, 454–467 (1980)

49.16 B. Widrow, D. Shur, S. Shaffer: On adaptive inverse control, Proc. 15th Asilomar Conf. (1981) pp. 185–189

49.17 S.J. Elliott, P.A. Nelson: Active noise control, IEEE Signal Process. Mag. **10**, 12–35 (1993)

49.18 C.C. Boucher, S.J. Elliott, P.A. Nelson: Effect of errors in the plant model on the performance of algorithms for adaptive feedforward control, IEE Proc. F Radar Signal Process. **138**, 313–319 (1991)

49.19 S.D. Snyder, C.H. Hansen: The effect of transfer function estimation errors on the filtered-X LMS algorithm, IEEE Trans. Signal Process. **42**, 950–953 (1994)

49.20 G. Long, F. Ling, J.G. Proakis: The LMS algorithm with delayed coefficient adaptation, IEEE Trans. Acoust. Speech Signal Process. **37**, 1397–1405 (1989)

49.21 S.J. Elliott, I.M. Stothers, P.A. Nelson: A multiple error LMS algorithm and its application to the active control of sound and vibration, IEEE Trans. Acoust. Speech Signal Process. **ASSP-35**, 1423–1434 (1987)

49.22 R.D. Gitlin, H.C. Meadors, S.B. Weinstein: The tap-leakage algorithm: An algorithm for the stable operation of a digital implemented, fractional adaptive spaced equalizer, Bell Syst. Tech. J. **61**, 1817–1839 (1982)

49.23 S. Laugesen, S.J. Elliott: Multichannel active control of random noise in a small reverberant room, IEEE Trans. Signal Process. **1**, 241–249 (1993)

49.24 M.M. Sondhi, D.A. Berkley: Silencing echoes on the telephone network, Proc. IEEE **68**, 948–963 (1980)

49.25 J.R. Treichler: Adaptive algorithms for infinite impulse response filters. In: *Adaptive Filters*, ed. by C.F.N. Cowan, P.M. Grant (Prentice-Hall, Englewood Cliffs 1985), Chap. 4

49.26 L.J. Eriksson, M.C. Allie, R.A. Greiner: The selection and application of an IIR adaptive filter for use in active sound attenuation, IEEE Trans. Acoust. Speech Signal Process. **ASSP-35**, 433–437 (1987)

49.27 P.L. Feintuch: An adaptive recursive LMS filter, Proc. IEEE **64**, 1622–1624 (1976)

49.28 L.J. Eriksson: Development of the filtered-U algorithm for active noise control, J. Acoust. Soc. Am. **89**, 257–265 (1991)

49.29 L.J. Eriksson, M.C. Allie, C.D. Bremigan, J.A. Gilbert: Active noise control on systems with time-varying sources and parameters, Sound Vib. **23**, 16–21 (1989)

49.30 S.J. Elliott, P. Darlington: Adaptive cancellation of periodic, synchronously sampled interference, IEEE Trans. Acoust. Speech Signal Process. **ASSP-33**, 715–717 (1985)

49.31 B. Chaplin: The cancellation of repetitive noise and vibration, Proc. Inter-noise **1980**, 699–702 (1980)

49.32 B. Widrow, J.R. Glover, J.M. McCool, J. Kaunitz, C.S. Williams, R.H. Hern, J.R. Zeidler, E. Dong, R.C. Goodlin: Adaptive noise canceling: principles and applications, Proc. IEEE **63**, 1692–1716 (1975)

49.33 P. Darlington, S.J. Elliott: Stability and adaptively controlled systems – a graphical approach, Proc. ICASSP **1987**, 399–402 (1987)

49.34 P. Darlington, S.J. Elliott: Synchronous adaptive filters with delayed coefficient adaptation, Proc. ICASSP **1988**, 2586–2589 (1988)

49.35 D.R. Morgan, J. Thi: A multitone pseudocascade filtered-X LMS adaptive notch filter, IEEE Trans. Signal Process. **41**, 946–956 (1993)

49.36 J.R. Glover Jr.: Adaptive noise canceling applied to sinusoidal interferences, IEEE Trans. Acoust. Speech Signal Process. **ASSP-25**, 484–491 (1977)

49.37 E. Ziegler Jr.: Selective active cancellation system for repetitive phenomena, U.S. Patent 4878188, Oct. 31 (1989)

49.38 D.R. Morgan, C. Sanford: A control theory approach to the stability and transient analysis of the filtered-X LMS adaptive notch filter, IEEE Trans. Signal Process. **40**, 2341–2346 (1992)

49.39 X. Kong, S.M. Kuo: Analysis of asymmetric out-of-band overshoot in narrowband active noise control systems, IEEE Trans. Speech Audio Process. **7**, 587–591 (1999)

49.40 S.M. Kuo, M. Ji: Passband disturbance reduction in periodic active noise control systems, IEEE Trans. Speech Audio Process. **4**, 96–103 (1996)

49.41 D. P. Pfaff, N. S. Kapsokavathis, N. A. Parks: Method for actively attenuating engine generated noise, U.S. Patent 5146505, Sept. 8 (1992).

49.42 S.M. Kuo, A.B. Puvvala: Effects of frequency separation in periodic active noise control systems, IEEE Trans. Speech Audio Process. **14**, 1857–1866 (2006)

49.43 Y. Yuan, N. S. Kapsokavathis, K. Chen, S. M. Kuo: Active noise control system, U.S. Patent 5359662, Oct. 25 (1994).

49.44 S. M. Kuo, M. J. Ji, M. K. Christensen, R. A. Herold: Indirectly sensed signal processing in active periodic acoustic noise cancellation, U.S. Patent 5502770, Mar. 26 (1996).

49.45 P. D. Hill: Active acoustic attenuation system for reducing tonal noise in rotating equipment, U.S. Patent 5010576, Apr. 23 (1991).

Amazon EX1010 Page 1016

49.46 S.M. Kuo, M.J. Ji: Development and analysis of an adaptive noise equalizer, IEEE Trans. Speech Audio Process. **3**, 217–222 (1995)

49.47 S.M. Kuo, Y. Yang: Broadband Adaptive Noise Equalizer, IEEE Signal Process. Lett. **3**, 234–235 (1996)

49.48 L.J. Eriksson: Recursive algorithms for active noise control, Proc. Int. Symp. Active Control of Sound Vib. (1991) pp. 137–146

49.49 S.J. Elliott, T.J. Sutton: Performance of feedforward and feedback systems for active control, IEEE Trans. Speech Audio Process. **4**, 214–223 (1996)

49.50 Y. Song, Y. Gong, S.M. Kuo: A robust hybrid feedback active noise cancellation headset, IEEE Trans. Speech Audio Process. **13**, 607–617 (2005)

49.51 S.M. Kuo, S. Mitra, W.S. Gan: Active noise control system for headphone applications, IEEE Trans. Contr. Syst. Technol. **14**, 331–335 (2006)

49.52 D.C. Swanson: Active noise attenuation using a self-tuning regulator as the adaptive control algorithm, Proc. Inter-noise **1989**, 467–470 (1989)

49.53 P.A. Nelson, A.R.D. Curtis, S.J. Elliott, A.J. Bullmore: The active minimization of harmonic enclosed sound fields, Part I: Theory, J. Sound Vib. **117**, 1–13 (1987)

49.54 A.J. Bullmore, P.A. Nelson, A.R.D. Curtis, S.J. Elliott: The active minimization of harmonic enclosed sound fields, Part II: A computer simulation, J. Sound Vib. **117**, 15–33 (1987)

49.55 S.J. Elliott, A.R.D. Curtis, A.J. Bullmore, P.A. Nelson: The active minimization of harmonic enclosed sound fields, Part III: Experimental verification, J. Sound Vib. **117**, 35–58 (1987)

49.56 S.J. Elliott, C.C. Boucher, P.A. Nelson: The behavior of a multiple channel active control system, IEEE Trans. Signal Process. **40**, 1041–1052 (1992)

49.57 P.A. Nelson, A.R.D. Curtis, S.J. Elliott, A.J. Bullmore: The minimum power output of free field point sources and the active control of sound, J. Sound Vib. **116**, 397–414 (1987)

49.58 S.J. Elliott, C.C. Boucher: Interaction between multiple feedforward active control systems, IEEE Trans. Speech Audio Process. **2**, 521–530 (1994)

49.59 D.E. Melton, R.A. Greiner: Adaptive feedforward multiple-input, multiple-output active noise control, Proc. ICASSP **1992**, 229–232 (1992)

Amazon EX1010 Page 1017

Part I Multichannel Speech Processing

Ed. by J. Benesty, I. Cohen, Y. Huang

50 Microphone Arrays
G. W. Elko, Summit, USA
J. Meyer, Summit, USA

51 Time Delay Estimation and Source Localization
Y. Huang, Murray Hill, USA
J. Benesty, Montreal, Canada
J. Chen, Murray Hill, USA

52 Convolutive Blind Source Separation Methods
M. S. Pedersen, Smørum, Denmark
J. Larsen, Kongens Lyngby, Denmark
U. Kjems, Smørum, Denmark
L. C. Parra, New York, USA

53 Sound Field Reproduction
R. Rabenstein, Erlangen, Germany
S. Spors, Berlin, Germany

50. Microphone Arrays

G. W. Elko, J. Meyer

This chapter introduces various types of microphone array beamforming systems and discusses some of the fundamental theory of their operation, design, implementation, and limitations. It is shown that microphone arrays have the ability to offer directional gains that can significantly improve the quality of signal pickup in reverberant and noisy environments.

Hands-free audio communication is now a major feature in mobile communication systems as well as audio and video conferencing systems. One problem that becomes evident to users of these systems is the decrease in communication quality due to the pickup of room reverberation and background noise. In the past, this problem was dealt with by using microphones placed close to the desired talker or source. Although this simple solution has proven to be quite effective, it also has its drawbacks. First, it is not always possible or desirable to place the microphone very close to the talker's mouth. Second, by placing the microphone close to the talker's mouth, one has to deal with rapid level variation as the talker moves his or her mouth relative to the microphone. Third is the negative impact of speech plosives (airflow transients generated by plosive sounds) and forth, microphone structure-borne handling noise has a detrimental effect. Finally, for directional microphone elements, there is also a nearfield proximity effect (where the frequency response of the microphone is modulated by the relative position of the microphone to the mouth). With these issues in mind, it is of interest to investigate other potential solutions. One solution is to use beamforming microphone arrays, which can offer significant directional gain so as to result in similar audio performance to that of closely placed microphones.

50.1 **Microphone Array Beamforming** 1021
50.1.1 Delay-and-Sum Beamforming 1023
50.1.2 Filter-and-Sum Beamforming 1028
50.1.3 Arrays with Directional Elements ... 1028
50.2 **Constant-Beamwidth Microphone Array System** 1029
50.3 **Constrained Optimization of the Directional Gain** 1030
50.4 **Differential Microphone Arrays** 1031
50.5 **Eigenbeamforming Arrays** 1034
50.5.1 Spherical Array 1034
50.5.2 Eigenbeamformer 1035
50.5.3 Modal Beamformer 1037
50.6 **Adaptive Array Systems** 1037
50.6.1 Constrained Broadband Arrays 1038
50.7 **Conclusions** 1040
References 1040

50.1 Microphone Array Beamforming

The propagation of acoustic waves in space is a function of both space and time coordinates. In general, the mathematical representation of a propagating acoustic wave is a four-dimensional function: three spatial dimensions and one time variable. Acoustic wave propagation can be modeled by a linearized scalar acoustic wave equation that describes the relationships between the physical quantities of acoustic pressure and density variation of the medium [50.1],

$$\nabla^2 p = \frac{1}{c^2}\frac{\partial^2 p}{\partial t^2}, \tag{50.1}$$

where p is the instantaneous acoustic pressure fluctuation of the sound and c is the propagation speed of sound in the medium. There are a few major underlying assumptions in the derivation of (50.1), but this simplified linear model is adequate for laying the foundation for array beamforming of acoustic signals. The dimen-

Amazon EX1010 Page 1019

sionality of the wave equation can easily be seen from (50.1), where p is a function of three space variables and one time variable. Therefore the scalar acoustic pressure field in space can be represented as $p(\boldsymbol{r}, t)$, where $\boldsymbol{r}$ is the measurement position in the acoustic field and t is the time dependence. The scalar acoustic pressure field must satisfy (50.1) at all points in space. By using the spatial Fourier transform, the acoustic field can also equivalently be represented in the wavevector–frequency domain. This representation has been widely used in the fields of geophysics, structural, underwater and aeroacoustics, and can be quite useful in the analysis and design of microphone array beamformers.

Applying the four-dimensional Fourier transform to the time–space scalar acoustic field yields,

$$P(\boldsymbol{k}, \omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(\boldsymbol{r}, t) \mathrm{e}^{-\mathrm{i}(\omega t - \boldsymbol{k}^{\mathrm{T}} \boldsymbol{r})} \, \mathrm{d}\boldsymbol{r} \, \mathrm{d}t \, , \qquad (50.2)$$

where the acoustic pressure P is capitalized to indicate a transform to the frequency domain, $\boldsymbol{k}$ is the wavevector and the superscript 'T' represents the transpose operator. It may seem that this transformation has unnecessarily complicated the description of the scalar acoustic pressure field, but as will be seen later, it has transformed the field description into a form that makes the analysis of acoustic field space–time functions analogous to the field of multidimensional signal processing. Since the Fourier transform is a linear transformation, one can write an inverse relationship,

$$p(\boldsymbol{r}, t) = \frac{1}{(2\pi)^4} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P(\boldsymbol{k}, \omega) \mathrm{e}^{\mathrm{i}(\omega t - \boldsymbol{k}^{\mathrm{T}} \boldsymbol{r})} \, \mathrm{d}\boldsymbol{k} \, \mathrm{d}\omega \, . \qquad (50.3)$$

Equation (50.3) directly shows that any acoustic field $p(\boldsymbol{r}, t)$ can be represented as an infinite number of propagating plane waves with appropriate complex weighting. This interpretation can easily be seen from the equation for a single propagating plane wave with frequency ω_0 and wavevector $\boldsymbol{k}_0$,

$$p(\boldsymbol{r}, t) = A_0 \mathrm{e}^{-\mathrm{i}(\omega_0 t - \boldsymbol{k}_0^{\mathrm{T}} \boldsymbol{r})} \, , \qquad (50.4)$$

where A_0 is the plane-wave amplitude. Note that for generality, one can allow A_0 to be complex, but for this discussion it is assumed that A_0 is real. The wavenumber–frequency spectrum of this single propagating plane wave is simply,

$$P(\boldsymbol{k}, \omega) = A_0 \delta(\boldsymbol{k} - \boldsymbol{k}_0) \delta(\omega - \omega_0) \, . \qquad (50.5)$$

Equation (50.5) shows the Fourier mapping of infinite continuous functions in both time and space to be a point in the wavevector–frequency space.

Now consider a general weighting function of the space–time pressure distribution with a four-dimensional linear shift-invariant filter having an impulse response $h(\boldsymbol{r}, t)$. Then, the output signal is the convolution of the space–time acoustic pressure signal and the spatial weighting function such that,

$$y(\boldsymbol{r}, t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(\boldsymbol{r} - \hat{\boldsymbol{r}}, t - \tau) p(\hat{\boldsymbol{r}}, \tau) \, \mathrm{d}\hat{\boldsymbol{r}} \, \mathrm{d}\tau \, . \qquad (50.6)$$

Equation (50.6) is a filtering operation that can equivalently be represented in the wavevector–frequency domain as,

$$Y(\boldsymbol{k}, \omega) = H(\boldsymbol{k}, \omega) P(\boldsymbol{k}, \omega) \, . \qquad (50.7)$$

Equation (50.6) and (50.7) show the direct analogy between the space–time wavevector–frequency representation of spatial filtering and the well-known results from multidimensional linear systems theory. Thus, by using specific spatial weighting functions, one can filter the acoustic field to investigate the propagating directions, amplitude, and phases of plane waves traveling in space. Filtering the spatiotemporal acoustic scalar pressure field is known in the field of array signal processing as *beamforming*. Thus, if one wants to *look* at plane waves propagating from the direction of the unit vector $\boldsymbol{k}_0 / \|\boldsymbol{k}_0\|$, one would design a filter such that,

$$H(\boldsymbol{k}, \omega) = \delta(\boldsymbol{k} - \boldsymbol{k}_0) G(\omega) \, , \qquad (50.8)$$

where $G(\omega)$ is the desired frequency response to waves propagating in the direction of the wavevector $\boldsymbol{k}_0$.

Although it may seem limiting to use plane waves as the underlying basis functions for the representation of a sound-field, it should be noted that other orthogonal representations such as spherical and cylindrical basis functions can themselves be represented as a series of plane waves.

The representation of the wavevector–frequency domain is a natural framework for analysis of the spatial and frequency filtering of beamforming arrays, as will be seen in the following.

In spatial filtering by beamforming, one deals with spatial apertures. An aperture is a region over which energy is received. Where an aperture can either be continuous, as in parabolic dishes, or discretely realized as in microphone arrays with multiple microphones, mixtures of continuous and discrete apertures are possible.

Amazon EX1010 Page 1020

In fact, even discrete microphone arrays are a mixture since no microphone is a perfect point receiver. Although sampled aperture systems must generally perform more signal processing, they offer several advantages over continuous aperture systems. A main advantage in using sampled apertures is that they allow for the possibility of using digital signal processing on the individual array signals. Arrays can be electronically steered, can form multiple simultaneous beams, and can be made adaptive purely by electronic means.

50.1.1 Delay-and-Sum Beamforming

Delay-and-sum beamforming, also known as *classical beamforming*, is one of the simplest and oldest techniques for realizing directional array systems. Although not fundamentally limited in bandwidth, early delay-and-sum arrays were used in *narrowband* operation to focus arrays onto a particular point or direction. Since a time delay for narrowband applications can be accomplished with a unique phase shift for each element, narrowband beamforming is typically implemented with phase shifts and is therefore commonly referred to as *phased-array* beamforming. One typical application for microphone arrays is to pick up speech or acoustic signals that are wideband (the desired signals cover many octaves in bandwidth). Therefore most microphone arrays for speech acquisition are implemented as delay-sum beamformers since they are inherently wideband.

Both continuous and discrete beamformers can be seen as implementations of wavevector–frequency filtering as discussed in the previous section. It should be noted that these two beamformers are not mutually exclusive; one can, for instance, have a discrete set of continuous transducers so that the beamformer is a mix of continuous and discrete sampling. Of special interest here is the spatial and frequency response of a finite array of microphones that sample the acoustic field. A classic delay-sum beamformer uses the sum of weighted and delayed samples from an array of N discretely sampled points of the pressure field. In general, a delay-sum beamformer output y is formed as,

$$y(t) = \sum_{n=1}^{N} w_n s(\boldsymbol{r}_n, t - \tau_n) \,, \tag{50.9}$$

where the weights w_n are real and the time delay τ_n is applied to the measured signals $s(\boldsymbol{r}_n, t)$, at microphone n. If the incident field is a planewave with wavevector $\boldsymbol{k}_0$, amplitude A_0 and frequency ω_0, then,

$$s(\boldsymbol{r}_n, t) = A_0 \, \mathrm{e}^{\mathrm{i}(\omega_0 t - \boldsymbol{k}_0^{\mathrm{T}} \boldsymbol{r}_n)} \,, \tag{50.10}$$

and

$$y(t) = A_0 \, \mathrm{e}^{\mathrm{i}\omega_0 t} \sum_{n=1}^{N} w_n \, \mathrm{e}^{-\mathrm{i}(\omega_0 \tau_n + \boldsymbol{k}_0^{\mathrm{T}} \boldsymbol{r}_n)} \,. \tag{50.11}$$

The real weights w_n scale the signals measured at the positions $\boldsymbol{r}_n$. Thus, from (50.9), the origin of the term *delay-and-sum* can be seen as simply a description of how the classical delay-sum beamformer is realized.

For a causal delay-sum beamformer, the exponent in (50.11) must be less than or equal to zero for all n. Typically, one element position is selected as the spatial origin. This reference element position therefore defines the vectors $\boldsymbol{r}_n$. Since the array can be steered to any direction in space and one can select any position to define $\boldsymbol{r}_n$, an additional delay may be required in order to maintain causality. (Recall that $\boldsymbol{k}_0^{\mathrm{T}} \boldsymbol{r}_n$ can be either positive or negative depending on the direction of the incident wave relative to the array.) If an end element is selected as the spatial origin, then the causal delay is equal to the maximum time that an acoustic wave takes to transit the array. Smaller causal delays are also possible depending on which microphone position in the array is chosen as the spatial origin reference and the maximum desired steering angle.

If we set the causal time delay equal to T_0, then the output for an incident plane wave with angular frequency ω_0 and wavevector $\boldsymbol{k}_0$ can be written as,

$$y(t) = A_0 \, \mathrm{e}^{\mathrm{i}\omega_0 t} \sum_{n=1}^{N} w_n \, \mathrm{e}^{-\mathrm{i}\left[\omega_0 (T_0 + \tau_n) + \boldsymbol{k}_0^{\mathrm{T}} \boldsymbol{r}_n\right]} . \tag{50.12}$$

The maximum plane-wave response corresponds to the direction where the delays τ_n compensate for the propagation delay $\boldsymbol{k}_0^{\mathrm{T}} \boldsymbol{r}_n$ of a plane wave propagating with wavevector $\boldsymbol{k}_0$. This is done by setting the values of τ_n such that these delays offset the time lead (or lag) of a desired direction plane wave propagating over the array. Thus, setting the delays to steer the array to the direction of $\boldsymbol{k}_0$, means that the last two terms in the exponential in (50.12) cancel out. The delayed and summed output $y(t)$ is now such that all microphone signals propagating from the desired direction are added in phase. This direction corresponds to a maximum amplitude output for any selection of τ_n. Plane waves propagating from directions other than $\boldsymbol{k}_0$ will result in the addition of position-dependent phase variations, and as a result, the output amplitude will be smaller due to destructive interference.

To summarize, classical delay-and-sum beamforming uses delays between each array element that compensate for differences in the propagation delay of

the desired signal direction across the array. Signals originating from a desired direction (or location if the source is in the near field) are summed in phase, while other signals undergo destructive interference. By adjusting the weights of the delay-sum beamformer, the shape of the beam and sidelobes as well as the position of the nulls of an array can be controlled.

Uniformly Spaced Linear Arrays

While array geometries are in principle arbitrary, certain configurations are especially useful and amenable to mathematical analysis. For example, consider a simple array geometry of uniformly spaced collinear elements as shown in Fig. 50.1. Assume that an odd number of elements, numbered from $-N$ to N, are positioned on the x-axis such that element n has coordinate nd, and d is the interelement spacing. Thus, *endfire* directions lie on the x-axis, while *broadside* directions lie in the y–z plane. Much as linear time-domain systems are analyzed by their response to complex exponential time functions, an array response can be considered in terms of its response to complex exponential plane waves (as shown above).

For a collinear delay-and-sum beamformer steered towards a direction making an angle θ_0 measured clockwise from the y-axis, the relative time delays τ_n are,

$$\tau_n = n\frac{d}{c}\sin\theta_0, \tag{50.13}$$

where θ_0 is the *steering* direction, and represents the direction from which plane waves are received with maximum response.

Now, consider the unit-amplitude plane wave of frequency ω_0, arriving from a direction θ (not necessarily equal to θ_0). This wave is represented by the expression,

$$p(\boldsymbol{r}, t) = \mathrm{e}^{\mathrm{i}(\omega_0 t - \boldsymbol{k}_0^{\mathrm{T}}\boldsymbol{r})}\,. \tag{50.14}$$

Note that it has been assumed that the microphones are located along a line so the vector $\boldsymbol{r}$ points along the axis of the array. Since the array is axisymmetric about the axis of the array, the components of the wavevector can be assumed, without loss of generality, to lie in the x–y plane. With this assumption, one only needs to investigate the array response in the plane. The wavevector can be written into its three orthogonal components as,

$$\boldsymbol{k}_0 = -\frac{\omega_0}{c}\begin{pmatrix}\sin\theta\\ \cos\theta\\ 0\end{pmatrix}\,. \tag{50.15}$$

Substituting the wave field for a planewave and time delay expressions into the beamformer output expression yields,

$$y(t) = \mathrm{e}^{\mathrm{i}\omega_0 t}\sum_{n=-N}^{N} w_n\,\mathrm{e}^{\mathrm{i}\frac{\omega_0}{c}nd(\sin\theta-\sin\theta_0)}\,. \tag{50.16}$$

This expression can be simplified by removing the time-varying $\mathrm{e}^{\mathrm{i}\omega_0 t}$ term and substituting $u = \sin\theta$ and $u_0 = \sin\theta_0$ rather than by the angle itself. This convention has several advantages beyond mere compactness. First, a change of steering direction amounts to a single linear translation in u-space. Thus, the shape of the beam measured in the u-domain does not change with array steering angle, while beams measured in θ space change as a function of the steering direction. Second, one can logically divide the description of the array function into two regions: a *visible* and *invisible* region [50.2]. The visible region corresponds to values of u that have real angles in space and corresponds mathematically to $|u| \le 1$. Values of $|u| > 1$ correspond to angles that are not physical and is therefore referred to as the invisible region.

Making the substitutions for $\sin\theta$ and $\sin\theta_0$ yields an expression for the *array response*,

$$H(u, \omega) = \sum_{n=-N}^{N} w_n\,\mathrm{e}^{\mathrm{i}\frac{\omega}{c}nd(u-u_0)}. \tag{50.17}$$

Note that the description of the array response given in (50.17) has been written as a function of the two independent variables u and ω instead of the variables $\boldsymbol{k}$, t, and ω. This change of variables allows one to better visualize the response of the array in terms of angle and frequency.

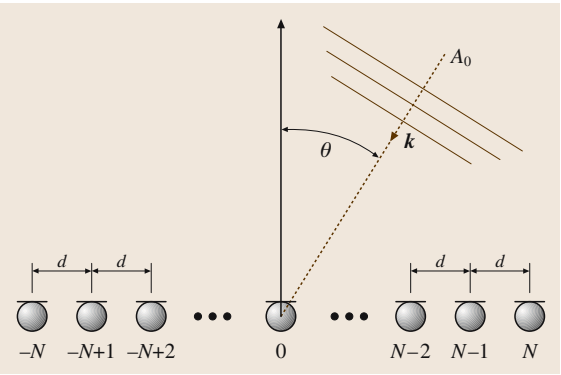


Fig. 50.1 Schematic of uniformly spaced linear array containing $2N+1$ microphones spaced by d

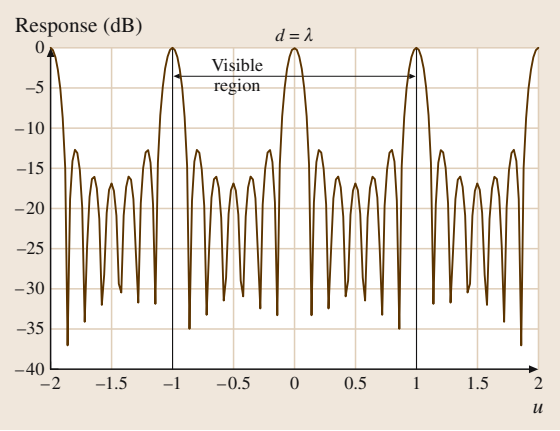


Fig. 50.2 Response of a seven-element uniformly spaced and weighted array as a function of the dimensionless variable u for element spacing equal to the wavelength of the incident acoustic signal

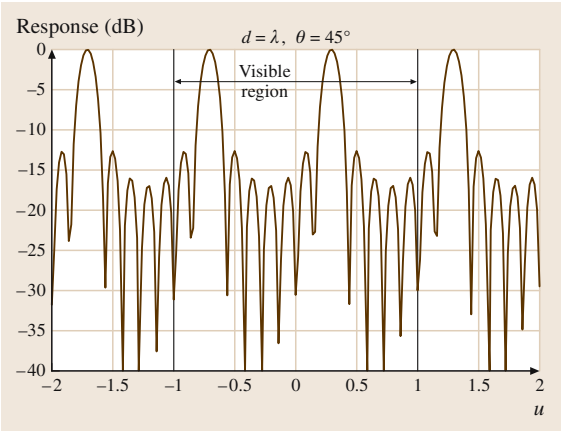


Fig. 50.3 u-response of a seven-element uniformly spaced and weighted array steered 45° as a function of the dimensionless variable u for element spacing equal to the wavelength of sound

The array response as expressed in (50.17) is a function in a two-dimensional u–ω space. For any real constant K the following identity holds,

$$H\left[\left(\frac{u}{K}+u_0\right), K\omega\right] = H(u+u_0, \omega)\,. \tag{50.18}$$

From (50.18), it can be seen that the spatial extent (beamwidth) of the array becomes commensurately smaller as the frequency increases directly proportionally to the frequency ω. Thus if the incident wave frequency is doubled, the beamwidth of the array is cut in half and vice versa. Another important aspect of the array function is that it is periodic in u for any given ω,

$$H(u, \omega) = H\left(u + \frac{2\pi mc}{\omega d}, \omega\right)\,, \tag{50.19}$$

where m is an integer. Thus, the main lobe has an infinite set of identical copies, which are called *grating lobes*. Since the *visible region* [50.2] is defined for values of $-1 \le u \le 1$, grating lobes may or may not be seen in the array response. As can be seen in (50.19), the appearance of grating lobes is a function of both microphone spacing and incident frequency.

Grating lobes are visible when another lobe appears at $|u| \le 1$. When fully visible, a grating lobe is equal in amplitude to the main lobe of the array. This phenomena is called spatial aliasing and is directly analogous to time aliasing in sampled time systems. As either the element spacing or the frequency increases, grating lobes move in an angular direction towards the main lobe and can be thought of as either a dilatation or contraction of the array function as defined in (50.19). As the spacing or frequency increases, new grating lobes become visible, appearing first along the endfire direction. If the array is not steered, grating lobes occur when the first periodic replication $(2\pi c)/\omega$ falls into the visible region $|u| \le 1$. In terms of the spacing d, grating lobes occur when

$$d \ge \frac{2\pi c}{\omega} = \lambda\,, \tag{50.20}$$

where λ is the incident sound wavelength. As shown above, the spatiofrequency response $H(u, \omega)$ is periodic, and thus *spatial aliasing* for spatiotemporal signals will occur before the grating lobe appears at $|u| = 1$. Thus, to preclude spatial aliasing while having any steering angle $|u_0| \le 1$, the grating lobe must be be out of visual space, $|u| + |u_0| \ge 2$. With this constraint,

$$d < \frac{\pi c}{\omega} = \frac{\pi c}{2\pi f} = \frac{\lambda}{2}\,. \tag{50.21}$$

Equations (50.20) and (50.21) define the *spatial sampling* requirement, which is analogous to the Nyquist sampling rate for sampled time-domain signals. If the array is steered to endfire which is the maximum steering angle in visible space, grating lobes can be avoided if adjacent lobes are separated by a distance greater than 2 in u-space (the same restriction as for no spatial aliasing for an unsteered array). It should also be noted that the lowest frequency at the onset of a grating lobe can be increased if the array is composed of directional elements that have low sensitivity in the region of $|u| \approx 1$.

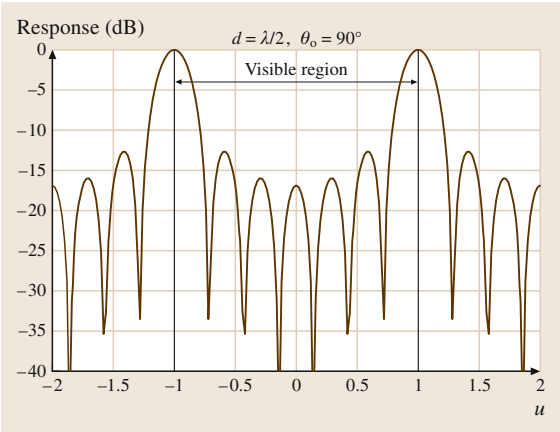


Fig. 50.4 u-response of a seven-element uniformly spaced and weighed array steered 90° as a function of the dimensionless variable u for element spacing equal to half the wavelength of sound

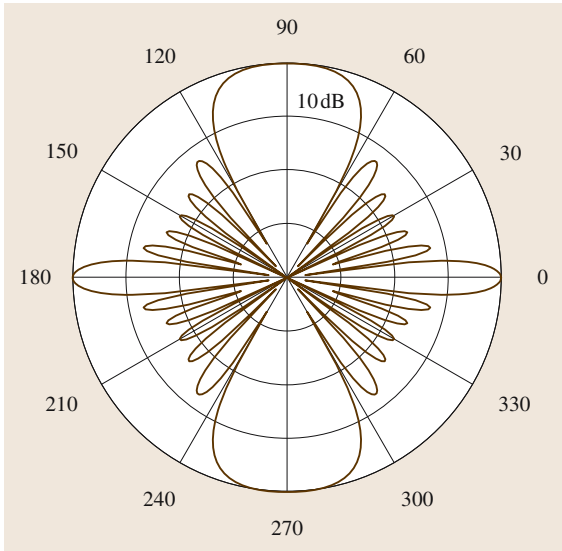


Fig. 50.5 Polar directivity pattern for a seven-element uniformly spaced and weighted unsteered array when $d = \lambda$. Note the large grating lobe at $\theta = 90°$. Also note that the directional response is axisymmetric around the $\theta = 0°$ axis

Uniformly Sampled and Weighted Linear Arrays

To better understand spatial aliasing and how $H(u, \omega)$ changes with increased spacing or frequency, as well as steering angle, it is instructive to examine a uniformly weighted linear array. Figure 50.2 shows the response of a uniformly weighted array as a function of the variable u for the case $d = \lambda$. The visible region is denoted on the figure as the region between the two vertical red lines. One can clearly see the grating lobes beginning to move into the visible region in the figure at $u = \pm 1$. This behavior was predicted by (50.20) for an unsteered array. Figure 50.3 shows how the response is rotated in u-space when the array is steered 45° from the unsteered direction (broadside). One can now clearly see that the grating lobe is fully within the visible region. For comparison purposes, Fig. 50.4 shows the directional response for a 90° steered array for the case of $d = \lambda/2$. Equation (50.21) predicts that for this case, a grating lobe should appear in the opposite direction (along the axis of the array but in the opposite of the steered direction). Figure 50.4 confirms that this is indeed the case as one can see the grating lobe at 270° when the array is steered to 90° (endfire).

Figures 50.5 and 50.6 show directivity patterns when $d = \lambda$ and $d = \lambda/2$ for a seven-element uniformly spaced and weighted unsteered array. It is not coincidental that there are six zeros (nulls) in the response since as can be seen in (50.17), a seven-element array response can be written as a seventh-order polynomial. As the frequency or spacing changes, the spatial response either contracts

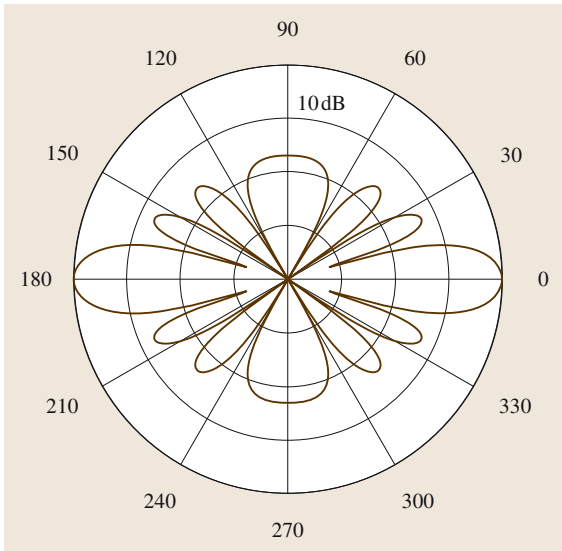


Fig. 50.6 Polar directivity pattern for a seven-element uniformly spaced and weighted unsteered array for $d = \lambda/2$. Note that compared to Fig. 50.5, there is no grating lobe in the $\theta = 90°$ direction

or dilates, so at lower frequencies or smaller element spacing, some or all of these nulls might not be contained within the visible region $|u| \leq 1$. One way to visualize

this effect is to plot the angular response of the array as a function of frequency.

Figure 50.7 shows a surface plot of the magnitude of the response of a seven-element array with 8 cm spacing versus angle and frequency. The colors indicate a logarithmic range limited to 40 dB. The narrowing of the main lobe as frequency increases is clearly evident in the figure. It can also be seen that the array is essentially omnidirectional below 500 Hz and that the first spatial aliasing lobe appears at around 4 kHz while a second grating lobe appears at around 8 kHz. These two frequencies correspond to sound with wavelengths of 8 and 16 cm, respectively. Thus grating lobes appear for the unsteered array at frequencies where the element spacing is an integer multiple of the incident sound wavelength, as predicted by (50.20). Six distinct zeros can be seen in the response between the grating lobes in Fig. 50.7 and it can also be seen that they gradually disappear from the visible region below 4 kHz. No nulls occur in the visible region below 500 Hz.

The angular position of the first nulls on either side of the main lobe can be computed for an unsteered uniformly weighted array by noting that (50.17) can be rewritten

$$H(u,\omega) = \frac{\sin(\hat{N}\omega u d/2c)}{\sin(\omega u d/2c)}\,, \tag{50.22}$$

where $\hat{N} = 2N+1$ is the total number of elements, The first null occurs when the argument of the numerator of (50.22) is $\pm\pi$. The beamwidth, $\mathrm{BW}_{\mathrm{null}}$ is twice the angle between the main lobe and the first null

$$\begin{aligned}\mathrm{BW}_{\mathrm{null}} &= 2\sin^{-1}\left(\frac{2\pi c}{\hat{N}\omega d}\right)\\ &= 2\sin^{-1}(\lambda/L)\,,\end{aligned} \tag{50.23}$$

where λ is the wavelength of the incident sound and $L = \hat{N}d$ is close to the array length $[(\hat{N}-1)d]$. (Note that other definitions for beamwidth are also used, such as the angles where the response falls to half the main-lobe level.) If the array length L is much larger than the incident wavelength, then one can use the small-angle approximation and obtain,

$$\mathrm{BW}_{\mathrm{null}} \approx \frac{2\lambda}{L}\,. \tag{50.24}$$

Equations (50.23) and (50.24) show that the beamwidth is proportional to the wavelength and inversely proportional to the length of the array.

Although the above analysis is only for uniformly weighted and spaced linear arrays, the general qualitative relationships between the array length, frequency, and beamwidth are similar for weighted and nonuniformly sampled arrays.

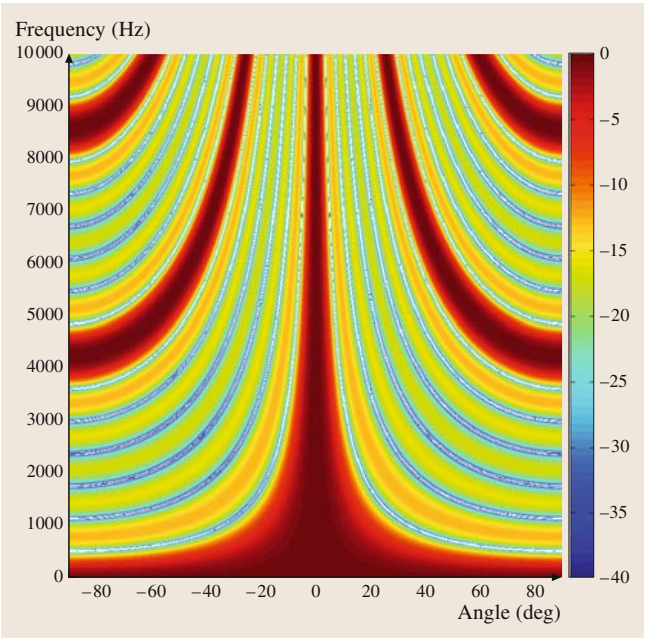


Fig. 50.7 Response of a seven-element uniformly spaced and weighted array for an element spacing of 8 cm

Analogy with FIR Filter Design

The discussion thus far expresses the beamformer behavior in terms of the array weights w_n. The opposite process, finding the weights for a given desired behavior, is also extremely important. This process is now shown to be related to finite impulse response (FIR) filter design.

Consider the Fourier transform of the array weights,

$$X(\mathrm{e}^{\mathrm{i}\omega_{\mathrm{t}}}) = \sum_{n=-N}^{N} w_n\,\mathrm{e}^{-\mathrm{i}\omega_{\mathrm{t}}n}\,, \tag{50.25}$$

where $\omega_{\mathrm{t}} = (u-u_0)\omega_0 d/c$. Comparing this with (50.17) yields the identity

$$H\left(u_0 + \frac{\omega_{\mathrm{t}}c}{\omega_0 d}, \omega_0\right) = X(\mathrm{e}^{\mathrm{i}\omega_{\mathrm{t}}})\,, \tag{50.26}$$

for any ω_0. From (50.26), it can be seen that finding the weights w_n for a given $H(u,\omega_0)$ is equivalent to an FIR filter design problem for a given desired frequency response $X(\mathrm{e}^{\mathrm{i}\omega_{\mathrm{t}}})$. Thus, like FIR filter design, array beamformer design is based on finding the appropriate coefficients of a finite-order polynomial that match a design goal in some desired way.

Classical delay-and-sum beamformer element weights are constants and therefore not a function of frequency. Since the spatial response of a delay-and-sum beamformer is a function of frequency, where the beamwidth of the array is commensurately smaller as the frequency increases, it is of interest to find a more-flexible design method that allows control of the beam pattern as a function of frequency. An obvious modification is to generalize the delay-and-sum beamformer by replacing the set of scalar weightings by a set of general filters. For obvious reasons, this generalized beamformer architecture is widely known as a filter-and-sum beamformer.

50.1.2 Filter-and-Sum Beamforming

As discussed in the previous section, it is desirable to enable the control of a beamformer design over a wide frequency range. The analysis of the delay-and-sum beamformer can be extended to a discrete-time filter-and-sum beamformer. Thus, without a loss in generality, the elemental filters are assumed to be FIR filters of length M. The beamformer output sequence, for integer κ, is

$$y[\kappa] = \sum_{n=-N}^{N} \sum_{m=0}^{M-1} p\left[\boldsymbol{r}_n, (\kappa - m)\mathrm{T}\right]\ h_n[m]\,, \quad (50.27)$$

where κ is the time sample, $h_n[m]$ is the impulse response of the filter associated with the n–th element (i. e., the n-th elemental filter), the number of microphones is $2N+1$, T is the sampling period, and $p(\boldsymbol{r}, t)$ is the incident wave field. As before, we consider the complex exponential plane wave,

$$p(\boldsymbol{r}, \kappa\mathrm{T}) = \mathrm{e}^{\mathrm{i}(\omega_0 \kappa \mathrm{T} - \boldsymbol{k}_0^{\mathrm{T}} \boldsymbol{r})}\,, \quad (50.28)$$

for some frequency ω_0 and wavevector $\boldsymbol{k}_0$. A uniformly spaced beamformer with spacing d has an output

$$y[\kappa] = \mathrm{e}^{\mathrm{i}\omega_0 \kappa \mathrm{T}} \sum_{n=-N}^{N} \sum_{m=0}^{M-1} h_n[m] \mathrm{e}^{\mathrm{i}(-\omega_0 m \mathrm{T} + \frac{\omega_0}{c} u_0 n d)}\,, \quad (50.29)$$

where $u_0 = \sin\theta_0$ as before.

Once again, the time-dependent portion is omitted to give the array response,

$$H(u, \omega) = \sum_{n=-N}^{N} \sum_{m=0}^{M-1} h_n[m] \mathrm{e}^{-\mathrm{i}(\omega m \mathrm{T} - \frac{\omega}{c} u n d)}\,. \quad (50.30)$$

From (50.30) it can now be seen that a filter-sum beamformer design problem is equivalent to a two-dimensional FIR filter design problem.

Two-Dimensional Filter Design

Just as delay-and-sum beamformer design is related to one-dimensional (1-D) FIR filter design, the filter-and-sum beamformer design is related to two-dimensional (2-D) FIR filter design that can be seen by considering an $N \times M$ function of 2-D Dirac delta functions, weighted according to the elemental filter coefficients,

$$f(x, y) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \delta(x - n, y - m) h_n[m]\,. \quad (50.31)$$

Now, the discrete 2-D Fourier transform of this function is:

$$X(\omega_1, \omega_2) = \sum_{n=-N}^{N} \sum_{m=0}^{M-1} h_n[m] \mathrm{e}^{-\mathrm{i}(\omega_1 n + \omega_2 m)}\,. \quad (50.32)$$

Note that this function is periodic with period 2π in both the ω_1 and ω_2 dimensions. Combining this with (50.30) yields

$$H\left(\frac{\omega_1 c \mathrm{T}}{\omega_2 d}, \frac{\omega_2}{\mathrm{T}}\right) = X(\omega_1, \omega_2)\,. \quad (50.33)$$

Equation (50.33) implies that a filter-and-sum beamformer can be conceptualized as a 2-D filter whose Fourier transform is closely related to the desired array response. Note that $X(\omega_1, \omega_2)$ is exactly the familiar wavenumber–frequency response [50.3], after converting the wavenumber and frequency variables $k_x = \|\boldsymbol{k}\| \sin\theta$, ω to the quantities $\omega_1 = k_x d$, $\omega_2 = \omega \mathrm{T}$, which have units of radians.

50.1.3 Arrays with Directional Elements

A microphone element frequency response can be effectively corrected with a single inverse filter placed at the output of the beamformer. Microphone elements with varying directivity patterns cannot generally be compensated for, but one can mathematically model their behavior using the *pattern multiplication* theorem [50.2]. This principle states that the total far-field response can be obtained from the far-field response of an equivalent array constructed using omnidirectional

Amazon EX1010 Page 1026

elements, multiplied by the far-field response of the individual microphones. Thus, the total array response is the product of the two array responses,

$$H(u,\omega) = H_{\mathrm{m}}(u,\omega)H_{\mathrm{a}}(u,\omega)\,,$$

where $H_{\mathrm{m}}(u,\omega)$ is the far-field response of a single microphone element, and $H_{\mathrm{a}}(u,\omega)$ is the far-field response of the equivalent array composed of omnidirectional elements. (Note that all of the directional elements must be identical and oriented in the same manner).

50.2 Constant-Beamwidth Microphone Array System

Due to hardware limitations, early digital beamforming microphone arrays were implemented as delay-sum beamformers [50.4]. As discussed previously in (50.19), a delay-and-sum beamformer has a decreasing beamwidth as the frequency increases. In fact, (50.23) showed that these functions are inversely related. Narrowing of the beamwidth as frequency increases can result in an undesired low-pass-colored output from an array if it is placed into a typical room with reverberation. This is due to the larger beamwidth at lower frequencies that allows more input power from a reverberant acoustic field. To minimize the coloration of a desired wide-band signal, it is desirable to consider beamformers whose directivity is constant over frequency.

By examining the results given in (50.23) and (50.24), an obvious solution towards attaining a constant-beamwidth beamformer design would be to design a filter-sum beamformer where the ratio of the acoustic wavelength to effective array length was held constant or nearly constant.

One proposed solution to this problem was to split the array into subarrays that covered different

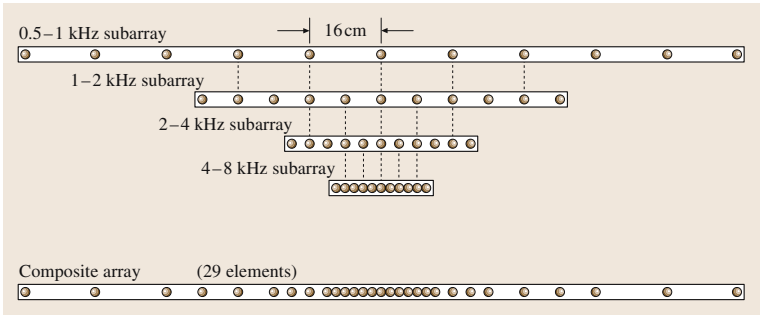


Fig. 50.8 Diagram of nested four-subarray microphone array

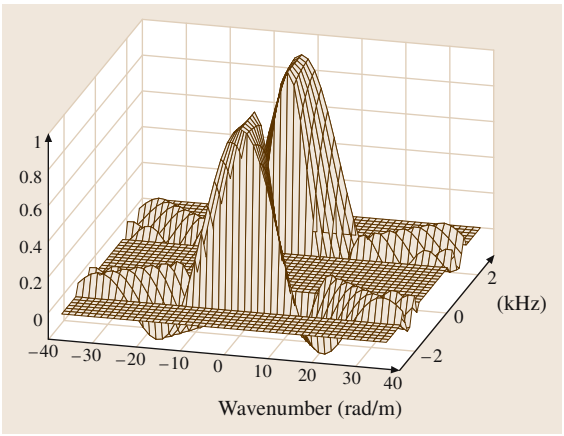


Fig. 50.9 Frequency–wavenumber design for a constant-beamwidth beamformer over one subarray

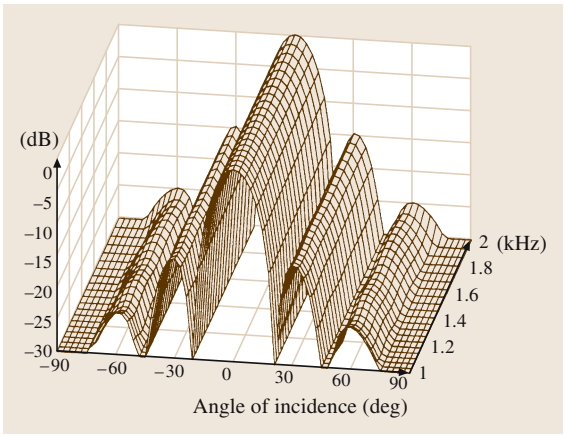


Fig. 50.10 Array response over one octave for a constant directivity array using a fan filter as shown in Fig. 50.9

frequency bands [50.4]. Some early microphone arrays were hybrid digital systems composed of three harmonically-nested subarrays covering a bandwidth from 300 Hz to 4 kHz [50.4, 5], but were later modified to include four harmonically nested subarrays (200–8 kHz). A four-band harmonically nested subarray is shown diagrammatically in Fig. 50.8. One interesting observation that can be made here is that, by harmonically nesting the subarrays, many elements are commonly shared in the array. Thus, the total number of array elements can be significantly reduced.

Further refinements were made on the multi-octave arrays to allow experimentation with a filter-sum configuration [50.6, 7]. These were developed to allow investigation into array algorithms for constant-beamwidth and dynamic control of the array beamwidth [50.7, 8]. Figure 50.9 shows the frequency–wavenumber response of the *fan filter* used to keep the beamwidth constant over the bandwidth of one octave subarray. A basic detail to notice here is that, as the frequency rises, the wavenumber bandwidth commensurately increases so that the beamwidth remains essentially constant. Figure 50.10 shows the array response over the 1–2 kHz octave when using the fan-filter design as shown in Fig. 50.9. It can be seen that the directivity is effectively constant over the whole octave. Similar fan filters are used for all octaves of the microphone array resulting in a constant beamwidth over most of the frequency range [50.6, 7].

The fan-filter design is equivalent to constant directivity beamformers based on the use of different low-pass filters on each element where the low-pass cutoff frequency is proportional the position of the element relative to the center of the array [50.9]. Intuitively the low-pass design acts to shorten the length of the array inversely proportional to frequency, thereby exactly counteracting the inverse relationship between array beamwidth and frequency.

50.3 Constrained Optimization of the Directional Gain

Directional gain is a common measure used in the design and analysis of microphone arrays. Optimization of the directional gain may not yield practical or realizable array designs, and therefore it is necessary to constrain the optimization to attain a useful design.

The directivity factor metric is typically used in quantifying the change in the acoustic signal-to-noise ratio (SNR) of a beamforming array in an isotropic noise field. The directivity factor (Q) expresses the ratio of sound energy received from the steered direction to the average energy received from all directions. Mathematically, the definition is,

$$Q(r,\theta_0,\phi_0)=\frac{4\pi|y(r,\theta_0,\phi_0)|^2}{\int_0^{2\pi}\int_0^{\pi}|y(r,\theta,\phi)|^2\sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi}\,, \tag{50.34}$$

where $y(r,\theta,\phi)$ is the response to a wave arriving from spherical coordinate directions (r,θ,ϕ), and (θ_0,ϕ_0) the steering directions. It can be shown that this function can be maximized for any array geometry by solving for the maximum of the ratio of two Hermitian forms [50.10]. Although it is not always the goal to maximize the directivity factor, this quantity is typically used to quantify beamforming acoustic SNR gains for far-field sources in isotropic noise fields. It is standard engineering practice to express the directivity factor in decibels, and this form is called the directivity index. The directivity index (DI) is defined as

$$\mathrm{DI}(r,\theta_0,\phi_0)=10\log\left[Q(r,\theta_0,\phi_0)\right]. \tag{50.35}$$

Optimization of the directional gain at frequencies where the array size is smaller than the acoustic wavelength results in array designs that are referred to as superdirectional arrays. Superdirectional arrays obtain directivity by forming the differences between subarray beampatterns. Superdirectional arrays have element weighting functions that oscillate in and out of phase between elements and can have directional indices (DI) potentially twice as high as classical delay-sum beamformers [$20\log(N)$ versus $10\log(N)$] [50.11]. Pattern-differencing beamformers result in a reduced acoustic signal output relative to microphone self-noise and electronics. Thus the array output SNR that can be significantly compromised. This problem has been known in the past and many authors have written on the subject [50.12]. The earliest publication containing a solution that constrains the gain to deal with the problem was a paper by *Gilbert* and *Morgan* [50.13]. However, it was not until a publication by *Cox* et al. [50.14] that a practical design methodology emerged. The following development follows that of Cox et al.

Consider a linear array of N sensors with desired signal and noise spatial cross-correlation matrices de-

noted by $\boldsymbol{P}$ and $\boldsymbol{Q}$, respectively. If one assumes that the signal and noise signals are stationary and independent processes, then the spatial cross-correlation matrix for the array can be written as

$$\boldsymbol{R}(\omega) = \sigma_{\mathrm{s}}^2 \boldsymbol{P}(\omega) + \sigma_{\mathrm{n}}^2 \boldsymbol{Q}(\omega)\,, \tag{50.36}$$

where the frequency domain is denoted by the explicit use of the temporal frequency ω. Thus the input signal-to-noise ratio is simply $\sigma_{\mathrm{s}}^2/\sigma_{\mathrm{n}}^2$. The output power spectrum of the beamformer that combines each microphone signal by a weight $w(\omega)$ can be written

$$Y = \boldsymbol{w}^{\mathrm{H}} \boldsymbol{R} \boldsymbol{w}, \tag{50.37}$$

where $\boldsymbol{w}$ is the complex weight vector containing the beamformer weights for each microphone element at frequency ω_0, and ^{H} denotes the complex conjugate transpose. Also note that the explicit frequency dependence has been omitted for compactness of the expression. The array gain (equivalent to the directivity index for an isotropic noise field) can now be defined as the relative gain in SNR at the beamformer output, and can be written as,

$$G = \frac{\boldsymbol{w}^{\mathrm{H}} \boldsymbol{P} \boldsymbol{w}}{\boldsymbol{w}^{\mathrm{H}} \boldsymbol{Q} \boldsymbol{w}}\,, \tag{50.38}$$

Note that in general, $\boldsymbol{Q}$ can also contain components of sensor self-noise.

In order to quantify the gain in SNR through a beamformer, *Cox* [50.14] proposed a measure called *white-noise gain* (WNG), which simplifies the above equations by assuming that the noise matrix can be assumed as an identity matrix. An expression of the WNG is then simply

$$G_{\mathrm{w}} = \frac{\boldsymbol{w}^{\mathrm{H}} \boldsymbol{P} \boldsymbol{w}}{\boldsymbol{w}^{\mathrm{H}} \boldsymbol{w}} \tag{50.39}$$

From (50.39), it can be seen that lower values of WNG imply a higher relative sensitivity to white-noise signals in each microphone relative to the sensitivity of acoustic signals. Therefore, the term WNG is best understood when interpreted as the gain of the array *over* white noise.

A constrained design can be obtained by the use of Lagrange multipliers such that

$$\begin{aligned} &\arg\min_{\boldsymbol{w}} \left\{ \frac{1}{G} + \epsilon \frac{1}{G_{\mathrm{w}}} \right\} \\ &= \arg\min_{\boldsymbol{w}} \left\{ \frac{\boldsymbol{w}^{\mathrm{H}} [\boldsymbol{Q} + \epsilon \boldsymbol{I}] \boldsymbol{w}}{\boldsymbol{w}^{\mathrm{H}} \boldsymbol{P} \boldsymbol{w}} \right\}\,. \end{aligned} \tag{50.40}$$

Equation (50.40) is a generalized eigenvalue problem and the constrained optimum weight can also be found [50.10] as the eigenvector corresponding to the largest eigenvalue of the middle term in the numerator in (50.40). For a single desired source propagating from a known direction θ relative to the array, one can obtain the optimum weight as,

$$\boldsymbol{w}_{\mathrm{opt}} = [\boldsymbol{Q} + \epsilon \boldsymbol{I}]^{-1} \boldsymbol{s}^*(\theta), \tag{50.41}$$

where $\boldsymbol{s}^*$ is the complex-conjugate signal across the array generated by the desired signal at angle θ. Equation (50.41) can be seen to be the constrained optimum beamformer that operates by the process of *whiten then match*. The *whitening* is done by the inverse of the term in the brackets where areas of higher spatial noise are commensurately attenuated by the beamformer. The second term within the brackets is a regularization of the inverse of the noise correlation matrix and acts as a constraint on the inverse. Matching to waves propagating from the desired direction is accomplished by the steering vector $\boldsymbol{s}^*$ that steers the array in the direction of the desired sound source.

By adjusting the Lagrange multiplier between the values of 0 and ∞, one can obtain a monotonic change in directional gain from an unconstrained array for $\epsilon = 0$ to that of a uniformly weighted array ($\epsilon \approx \infty$).

50.4 Differential Microphone Arrays

Differential microphone arrays are part of a regime of arrays that are of general interest due to their small size relative to the acoustic wavelength. For a given number of microphone elements in an array, differential arrays theoretically have the potential to attain maximum directional gain [50.11]. However, as will be shown later in this section, there are some significant caveats in realizing high directional gain with small differential arrays.

Earlier discussion on delay and filter-sum beamformers showed that these arrays operate as FIR spatial low-pass filters. As a result, directional gain is obtained when the array length is large or on the order of the incident sound wavelength. On the contrary, differential arrays operate as spatial high-pass filters, and as such, can be developed and analyzed differently than the delay and filter-sum arrays. (To be precise, a differential array does fall into the general filter-sum beamformer

category. However, since the analysis of differential array operation can be developed more compactly, it will be assumed that these arrays are different.)

Differential array output can be understood as a finite-difference approximation to the sum of spatial derivatives of the scalar acoustic pressure field. The term *first-order* differential array applies to any array whose response is proportional to the combination of two components: a zeroth-order (acoustic pressure) signal and another proportional to the first-order spatial derivative of a scalar acoustic pressure field. Similarly, the term n-th-*order* differential array is used for arrays that have a response proportional to a linear combination of signal derived from spatial derivatives up to, and including order n.

Before discussing various implementations of n-th-order finite-difference arrays, expressions are developed for the n-th-order spatial acoustic pressure derivative in a direction $\boldsymbol{r}$. Since realizable differential arrays are approximations to acoustic pressure differentials, equations for general order differentials provide significant insight into the operation of these arrays.

The acoustic pressure field for a propagating acoustic plane wave can be written

$$p(\boldsymbol{r}, t) = A_0 \, \mathrm{e}^{\mathrm{i}(\omega_0 t - \boldsymbol{k}_0^{\mathrm{T}} \boldsymbol{r})} = A_0 \, \mathrm{e}^{\mathrm{i}(\omega_0 t - k_0 r \cos\phi)} \,, \tag{50.42}$$

where A_0 is the plane-wave amplitude. The angle ϕ is the angle between the position vector $\boldsymbol{r}$ and the wavevector $\boldsymbol{k}_0$ (note that the angle ϕ here is not the same as the standard spherical coordinate angle used to describe the angle in the x–y plane). Dropping the time dependence and taking the n-th-order spatial derivative along the direction of the position vector $\boldsymbol{r}$ yields:

$$\frac{\mathrm{d}^n}{\mathrm{d}r^n} p(k, r) = A_0 (-\mathrm{i}k \cos\phi)^n \, \mathrm{e}^{-\mathrm{i}kr\cos\phi} \,. \tag{50.43}$$

The plane-wave solution is valid for the response to sources that are *far* from the microphone array. The term *far* implies that the distance between the source and receiver is many times the square of the relevant dimension divided by the acoustic wavelength where the relevant dimension is the greater of the source or receiver dimension. Using (50.43) one can interpret that the n-th-order differential has a bidirectional pattern component with the shape of $(\cos\phi)^n$. It can also be seen that the frequency response of a differential microphone is high-pass with a slope of $6n$ dB per octave. If the far-field assumption is relaxed, the response of the differential system to a point source located at the coordinate origin is

$$p(k, r) = A_0 \frac{\mathrm{e}^{-\mathrm{i}(kr\cos\phi)}}{r} \,, \quad r > 0. \tag{50.44}$$

The n-th-order spatial derivative in the radial direction r is

$$\frac{\mathrm{d}^n}{\mathrm{d}r^n} p(k, r, \phi) = A_0 \frac{n!}{r^{n+1}} \, \mathrm{e}^{-\mathrm{i}kr\cos\phi} (-1)^n \sum_{m=0}^{n} \frac{(\mathrm{i}kr\cos\phi)^m}{m!} \,. \tag{50.45}$$

As can be seen in (50.45), a fundamental property for differential arrays is that the general n-th-order array response is a weighted sum of bidirectional terms of the form $\cos(\phi)^n$.

Implicit in the development of differential arrays is the inherent assumption that the true differential can be approximated by discrete microphones that are used to estimate the pressure differentials by finite differences. For reasonable estimation of the differential by finite differences of the acoustic pressure, the microphones must be placed such that the element spacing is much less than the acoustic wavelength over the desired frequency range. With the small-spacing requirements, one can see both the main advantage and disadvantage of differential arrays: they can attain high directional gain with small size, but also have commensurately small WNG values, indicating a high sensitivity to self-noise as well as microphone amplitude and phase mismatch.

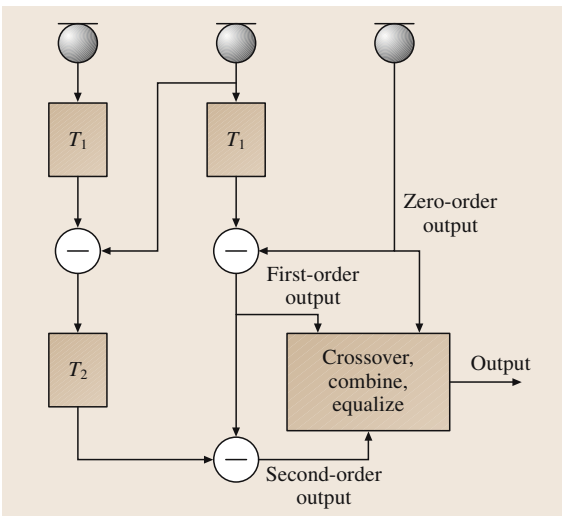


Fig. 50.11 Construction of a generalized second-order differential array as first-order differential combinations

An n-th-order differential array can be written as a sum of the n-th-order spatial differences and lower-order terms. An n-th-order array can also be written as the product of n first-order response terms as

$$y_n(\omega,\phi) = A_0 \prod_{i=1}^{n} \left(1 - \mathrm{e}^{-\mathrm{i}\omega(T_i + d_i/c\cos\phi)}\right) , \quad (50.46)$$

where the d_i relate to the microphone spacings, and the T_i relate to the chosen time delays. There is a design advantage in expressing the array response in terms of the products of first-order terms: it is now simple to represent higher-order systems as cascaded systems of lower order. Figure 50.11 shows how differential arrays can be constructed for up to second order. Extension of the design technique to higher orders is straightforward.

Values of T_i can easily be determined from the microphone spacing and the desired null angle. The ordering of T_i is not important as long as $\omega T_i \ll \pi$. If again it is assumed that $\boldsymbol{k}d_i \ll \pi$ and $\omega T_i \ll \pi$, then (50.46) can be approximated as

$$y_n(\omega,\phi) \approx A_0 \omega^n \prod_{i=1}^{n} (T_i + d_i/c\cos\phi) . \quad (50.47)$$

Equation (50.47) can be further simplified by making some simple substitutions for the arguments in the product term. Setting $\alpha_i = T_i/(T_i + d_i/c)$, then

$$y_n(\omega,\phi) \approx A_0 \omega^n \prod_{i=1}^{n} [\alpha_i + (1-\alpha_i)\cos\phi] . \quad (50.48)$$

If the product in (50.48) is expanded, a power series in $\cos\phi$ can be written for the response of the n-th-order array to an incident plane wave

$$\begin{aligned} &y_n(\omega,\phi) \\ &= A_0 G \omega^n \left(a_0 + a_1\cos\phi + a_2\cos^2\phi + \ldots \right. \\ &\quad \left. + a_n \cos^n\phi\right) , \end{aligned} \quad (50.49)$$

where the constant G is an overall gain factor. The only frequency-dependent term in (50.49) is ω^n. Thus the frequency response of an n-th-order differential array can be easily compensated by a low-pass filter whose frequency response is proportional to ω^{-n}. By choosing a structure that places only a delay behind each element in a differential array, the coefficients in the power series in (50.49) are independent of frequency, resulting in an array whose beampattern is independent of frequency.

To simplify the following exposition on the directional properties of differential arrays, it is assumed that the amplitude factor can be neglected. Also, since the directional pattern described by the power series in $\cos\phi$ can have any general scaling, the directional response can be written as solely a function of ϕ:

$$\begin{aligned} y_n(\phi) &= a_0 + a_1\cos\phi + a_2\cos^2\phi + \ldots \\ &\quad + a_n\cos^n\phi . \end{aligned} \quad (50.50)$$

Without loss of generality, the coefficients a_i can be defined to attain a normalized response at $\phi = 0°$, which implies

$$\sum_{i=0}^{n} a_i = 1. \quad (50.51)$$

In general, n-th-order differential microphones have, at most, n nulls (zeros). This follows directly from (50.50) and the fundamental theorem of algebra.

As an instructive example, let us examine the specific case for a second-order array,

$$y_2(\phi) = a_0 + a_1\cos\phi + a_2\cos^2\phi . \quad (50.52)$$

Equation (50.52) can also be factored into two first-order terms and written as

$$y_2(\phi) = [\alpha_1 + (1-\alpha_1)\cos\phi][\alpha_2 + (1-\alpha_2)\cos\phi] , \quad (50.53)$$

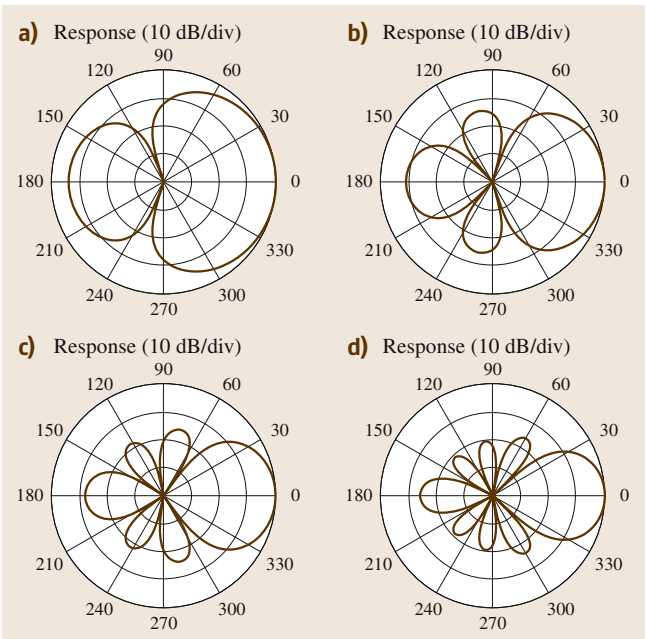


Fig. 50.12a–d Directional patterns that have maximum directional gain for differential microphone arrays for up to forth-order arrays

where

$$a_0 = \alpha_1\alpha_2,$$
$$a_1 = \alpha_1(1-\alpha_2)+\alpha_2(1-\alpha_1),$$
$$a_2 = (1-\alpha_1)(1-\alpha_2)\,, \tag{50.54}$$

or

$$\alpha_1 = a_0 + a_1/2 \pm \sqrt{(a_0+a_1/2)^2 - a_0}\,,$$
$$\alpha_2 = a_0 + a_1/2 \mp \sqrt{(a_0+a_1/2)^2 - a_0}\,. \tag{50.55}$$

As shown, the general form of the second-order system is the sum of second-, first-, and zero-order terms. If certain constraints are placed on the values of a_0 and a_1, it can be seen that there are two nulls (zeros) in the interval $0 \le \phi < \pi$. The array response pattern is axisymmetric about $\phi = 0$. These zeros can be explicitly found at the angles ϕ_1 and ϕ_2:

$$\phi_1 = \cos^{-1}\left(\frac{-\alpha_1}{1-\alpha_1}\right)\,, \tag{50.56}$$

$$\phi_2 = \cos^{-1}\left(\frac{-\alpha_2}{1-\alpha_2}\right)\,, \tag{50.57}$$

where now α_1 and α_2 can take on either positive or negative values where the magnitude of the inverse cosine is less than or equal to one. If the resulting beam pattern is constrained to have a maximum at $\phi = 0°$, then the values of α_1 and α_2 are also constrained. One interesting thing to note is that negative values of α_1 or α_2 correspond to a null moving into the front half-plane. Negative values of α_1 for the first-order microphone can be shown to have a rear-lobe sensitivity that exceeds the sensitivity at 0°. Since (50.53) is the product of two first-order terms, emphasis of the rear lobe caused by a negative value of α_2 can be counteracted by the zero from the term containing α_1. As a result, a beampattern can be found for the second-order microphone that has maximum sensitivity at $\phi = 0°$ and a null in the front-half plane. This result also implies that the beamwidth of a second-order microphone with a negative value of α_2 is narrower than that of the second-order dipole ($\cos^2(\phi)$ directional dependence). Figure 50.12 shows the beam patterns that have the highest attainable directivity for first through forth-order differential arrays [50.11].

50.5 Eigenbeamforming Arrays

The scalar acoustic pressure sound field obeys the Helmholtz equation [50.1], which states that the distribution of acoustic pressure and particle velocity on a surface uniquely defines the sound field within this surface if no sources or obstacles are enclosed within the surface. One can also use the Helmholtz equation to compute the external sound field if there are no obstacles or sources external to the measured surface. Thus, the interior and exterior sound field is uniquely determined if the sound pressure and particle velocity on a closed surface are known. Therefore it follows that only the sound pressure and particle velocity on a closed surface is required to capture all information of the sound field. This observation leads to a different beamformer implementation where, instead of forming beams in the classical sense by combining filtered individual microphone signals, one can use spatially decomposed *eigenbeams* to realize a general beamformer.

50.5.1 Spherical Array

Although the array surface can be any general surface, one can greatly simplify the analysis and array construction by using microphones placed on a rigid surface. Using a rigid body has the advantage that the radial particle velocity on its surface is zero. This greatly simplifies the general solution since only the sound pressure needs to be measured. Using a spherical geometry also simplifies the mathematics and allows for a design that puts equal weight on all directions due to its spherical symmetry. Theoretically many other shapes are possible, but the most amenable are shapes that are congruous to separable coordinate systems like cylindrical, oblate,

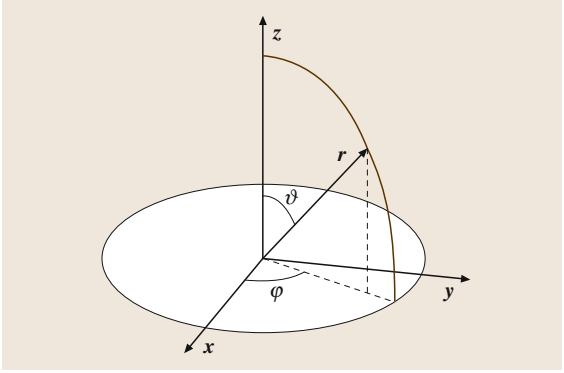


Fig. 50.13 Definition of the spherical coordinate system

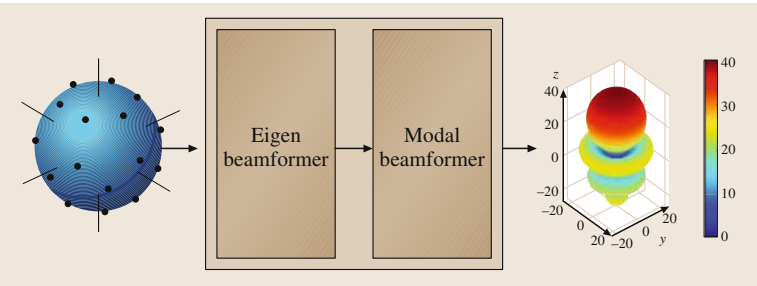


Fig. 50.14 Block diagram of the functional blocks of the spherical array

prolate, etc. Fig. 50.13 shows the notation for the standard spherical coordinate system that is used for the spherical array.

50.5.2 Eigenbeamformer

The overall structure of the spherical array can be decomposed into two main cascaded sections as shown in Fig. 50.14: an eigenbeamformer and a modal beamformer [50.15, 16].

The task of the eigenbeamformer is to transform the microphone signals into an orthonormal beam space. Since these beams are characteristic of the sound field in a similar way as eigenvectors are for a matrix, they are referred to as eigenbeams. Hence, the beamforming preprocessor is referred to as an eigenbeamformer. The second stage is a *modal* beamformer that linearly combines the eigenbeams to form a desired output beam pattern. In the following sections, it is shown that a spherical harmonic decomposition of the sound field leads to a computationally simple realization of the modal beamformer.

To begin an analysis of the spherical eigenbeamformer, we first need to introduce the orthonormal spherical harmonic eigenfunctions $Y_n^m(\vartheta, \varphi)$ which have order n and degree m, and are functions of the spherical angles ϑ, φ. The spherical harmonics Y_n^m are defined as [50.17],

$$Y_n^m(\vartheta, \varphi) = \sqrt{\frac{(2n+1)}{4\pi}\frac{(n-m)!}{(n+m)!}} P_n^m(\cos\vartheta)\mathrm{e}^{\mathrm{i}m\varphi}, \tag{50.58}$$

where $P_n^m(\cos\vartheta)$ are the standard associated Legendre functions. To calculate the response of the spherical eigenbeamformer to a planewave with incident angle ϑ, φ, and wavenumber k, it is judicious to express the plane wave in terms of spherical harmonics. By matching the boundary conditions of the rigid sphere of radius a to the impinging plane wave from the direction (ϑ, φ), one can compute the total acoustic pressure on the surface at location $(a, \vartheta_\mathrm{s}, \varphi_\mathrm{s})$ as

$$p_\mathrm{s}(\vartheta_\mathrm{s}, \varphi_\mathrm{s}, ka, \vartheta, \varphi) = 4\pi \sum_{n=0}^{\infty} i^n b_n(ka) \sum_{m=-n}^{n} Y_n^m(\vartheta, \varphi) Y_n^{m^*}(\vartheta_\mathrm{s}, \varphi_\mathrm{s}) . \tag{50.59}$$

The b_n are the *modal coefficients*, defined as

$$b_n(ka) = \left[j_n(ka) - \frac{j_n'(ka)}{h_n^{(2)'}(ka)} h_n^{(2)}(ka) \right] , \tag{50.60}$$

where $j_n(ka)$ are the spherical Bessel functions, and the prime indicates the derivative with respect to its argument, and the function $h_n^{(2)}(k)$ is the spherical Hankel of the second kind.

From the previous development, it can be seen that the rigid spherical baffle has a unique pressure field for plane waves impinging from different angles. A desired beam in a specific direction can be formed by filtering the surface pressure by a general weighting of the acoustic pressure on the surface. The spherical eigenbeamformer operates by decomposing this general surface weighting function into spherical harmonics. Thus, the output of an eigenbeamforming microphone to an incident plane wave for a spherical harmonic weighting function of $Y_{n'}^{m'}(\vartheta_\mathrm{s}, \varphi_\mathrm{s})$ can be written

$$\begin{aligned} F_{n',m'}(\vartheta, \varphi, ka) &= \int_{\Omega_\mathrm{s}} p_\mathrm{s}(\vartheta_\mathrm{s}, \varphi_\mathrm{s}, ka, \vartheta, \varphi) Y_{n'}^{m'}(\vartheta_\mathrm{s}, \varphi_\mathrm{s}) \mathrm{d}\Omega_\mathrm{s}, \\ &= 4\pi i^{n'} b_{n'}(ka) Y_{n'}^{m'}(\vartheta, \varphi) \end{aligned} \tag{50.61}$$

where $\mathrm{d}\Omega_\mathrm{s}$ represents an integration over the surface of the sphere. The far-field directivity of the microphone has the same directional dependence as the applied spherical harmonic sensitivity function, namely $Y_{n'}^{m'}$. Thus, it can now be seen that forming a desired beam

can be accomplished by using spherical harmonics as the basis functions expansion of the general surface weighting and then appropriately weighting and summing the outputs of these basis functions. The paragraph below on eigenbeams gives a detailed analysis of these eigenbeams. One final thing to note here is that the modal coefficients b_n introduce a frequency dependance. In order to combine different eigenbeams appropriately to form a desired beam, the b_n coefficients must be equalized so that they have the same amplitude and frequency response.

Discrete Orthonormality

In order to realize a spherical array, the continuous aperture has to be sampled. To achieve the same result as obtained in (50.61), the sample positions must fulfill the following *discrete orthonormality* condition:

$$A_{nm} \sum_{s=0}^{S-1} Y_n^m(\vartheta_s, \varphi_s) Y_{n'}^{m'}(\vartheta_s, \varphi_s) = \delta_{nn'} \delta_{mm'} \; . \quad (50.62)$$

where the A_{nm} are scale factors that are required to fullfill the discrete orthonomality condition. Although is not trivial to find a set of sensor locations that fulfill this discrete orthonormality, one sensor arrangement that fulfills this constraint up to fourth-order spherical harmonics are sample points at the center of the 32 faces of a truncated icosahedron [50.16].

The resulting structure of the eigenbeamformer can be derived from (50.62): for a specific eigenbeam Y_n^m the microphone signals are weighted by the sampled values of the corresponding surface sensitivity, $Y_n^m(\vartheta_s, \varphi_s)$. It can therefore be seen that the eigenbeamformer is effectively a set of simple delay-sum beamformers per eigenbeam (with no delay required).

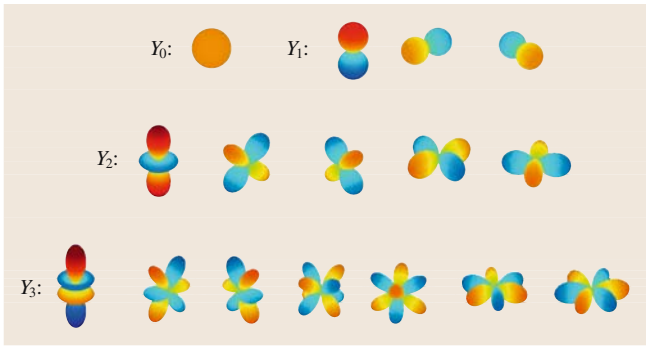


Fig. 50.15 Real and imaginary components of the spherical harmonics from zero to third order

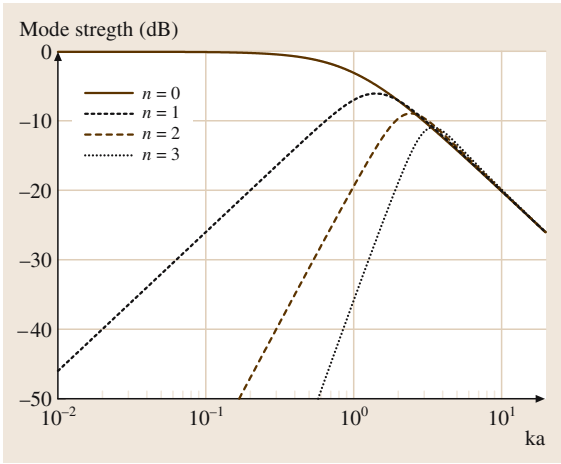


Fig. 50.16 Modal coefficients b_n for the first four orders

Besides the orthonormal constraint, spatial aliasing has to be taken into account when sampling a discrete aperture as in the design of classical beamforming arrays. Since there are $(N+1)^2$ spherical harmonics for a spatial resolution of order N, a minimum of $(N+1)^2$ sample locations are required to distinguish the spherical harmonics.

Eigenbeams

The outputs of the eigenbeamformer are a set of orthonormal beampatterns that are referred to as eigenbeams. These eigenbeams represent a spatially orthonormal decomposition of the sound field and represent a complete description of the original sound field up to the highest-order spherical harmonic used in the orthonormal expansion.

Figure 50.15 shows some example eigenbeams. From (50.58), it can be seen that the elevation dependence follows the Legendre function while the azimuth dependence has a sine-cosine dependance. The order n determines the number of zeros in the ϑ-direction while two times the degree m gives the number of zeros in the φ-direction.

The total number of eigenbeams required to capture N-th-order spherical harmonics of the sound field is $(N+1)^2$. As noted previously, the maximum achievable directional gain is $20\log(N+1)$. Thus, to obtain a maximum directional gain of 12 dB for any arbitrary direction, one would require eigenbeams up to third order. Note here that the term *order* has the same context as that used in differential arrays.

Similar to differential arrays, eigenbeams have the desirable quality that their directional response is fre-

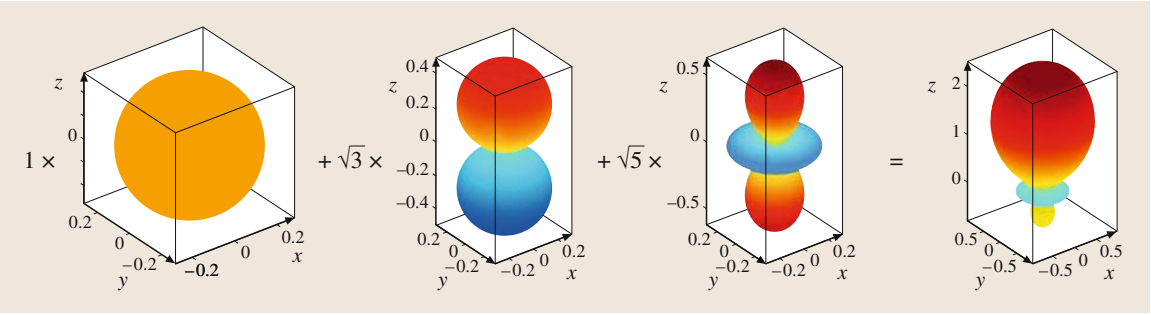


Fig. 50.17 Generating a second order hypercardioid pattern (note that only the directional properties are shown)

quency independent; however, as shown in Fig. 50.16 and (50.60), the magnitude response of the eigenbeams are dependent on the eigenbeam order and have high-pass responses that are $6n$ dB per octave. Thus one needs to equalize the different eigenbeams when combining them to realize a desired beampattern.

50.5.3 Modal Beamformer

The modal beamformer forms the second stage of the overall eigenbeamformer processing structure and comprises a simple weight-and-sum beamformer

$$D(\vartheta, \varphi) = \sum_{n=0}^{N} c_n Y_n(\vartheta, \varphi) \,, \tag{50.63}$$

where the beamformer multiplies each input beam by a factor c_n and adds up all weighted beams. As an example, Fig. 50.17 shows the generation of a second-order hypercardioid pattern steered along the z-axis.

To steer the resulting beam pattern towards a desired look-direction $[\vartheta_0, \varphi_0]$ one can use the Legendre polynomial addition theorem [50.1],

$$P_n(\cos\Theta) = \sum_{m=-n}^{n} \frac{(n-m)!}{(n+m)!} P_n^m(\cos\vartheta) P_n^m(\cos\vartheta_s) \mathrm{e}^{\mathrm{i}m(\varphi-\varphi_s)} \tag{50.64}$$

and (50.63) to write the overall steered array response as

$$D_{\vartheta_0\varphi_0}(\vartheta, \varphi) = \underbrace{\sum_{n=0}^{N} c_n}_{\text{combine}} \underbrace{\sum_{m=-n}^{n} \sqrt{\frac{(n-m)!}{(n+m)!}} P_n^m(\cos\vartheta_0) \mathrm{e}^{-\mathrm{i}m\varphi_0}}_{\text{steer}} \times Y_n^m(\vartheta, \varphi) \,. \tag{50.65}$$

Although modal beamforming arrays seem quite different from classical arrays, they are equivalent in operation but offer a more-intuitive and efficient way to realize beamforming arrays. As noted earlier, eigenbeamforming is not limited to using a spherical harmonic expansion to realize a modal beamformer, and any other spatially orthogonal basis function expansion can be used. Modal beamforming offers the capability of reducing the number of signals that need to be stored for processing since, in principal, the number of eigenbeams required can be much less than the number of microphone elements. Modal beamforming also has the desirable property that forming multiple simultaneous beams is computationally very efficient since the modal beamformer requires M multiply-adds, where M is the number of eigenbeams to form each unique output. Finally, the inherent computational simplicity of modal beamformers leads to efficient dynamic and adaptive microphone array implementations based on multiple beams and not on individual elements.

50.6 Adaptive Array Systems

The microphone arrays described in the previous sections are beamformers whose spatial responses are static since the weights, once designed, are fixed. Designs of fixed beamformers are done with assumptions of the signal and noise temporal and spatial statistics. This approach often leads to good designs when the a priori

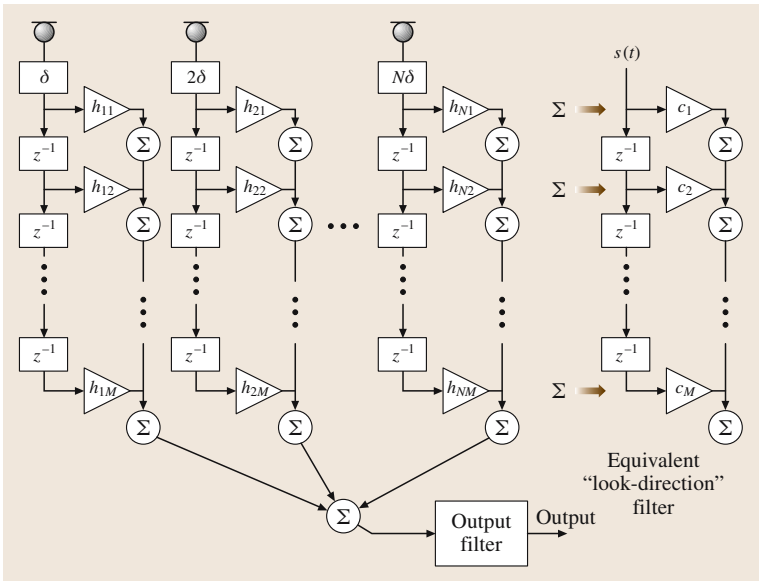


Fig. 50.18 Block diagram of a broadband adaptive microphone array implemented as an FIR filter-sum beamformer

assumptions match the acoustic conditions in which the array is placed. However, in acoustic and noise environments that dynamically change, it can be advantageous to use arrays that self-optimize their performance based on the sound field that the array is placed. Arrays that self-optimize their spatial response are generally referred to as adaptive arrays. There are many forms of adaptive array systems [50.18], but only a constrained broadband array is described here since most other schemes are based on similar approaches.

50.6.1 Constrained Broadband Arrays

Adaptive beamformers typically optimize a quadratic metric that is related to the output SNR. A common optimization is to minimize the output power under the constraint that the desired signal is not affected or only minor modifications are imparted to the signal. Thus one needs to minimize the output power under a constraint that the desired signal is not harmed. In most cases, like that of speech pickup in a room, the statistics of the desired source are not precisely known or are highly nonstationary. For this case, common adaptive array designs are based on the assumption that the position or direction of the desired source is known. The position can be found by beam-steering techniques based on power or known signal statistics, or by more sophisticated methods such as eigendecomposition [50.3] or other time-delay estimation direction-finding schemes [50.19]. It is further assumed that the locations of the desired source and noise sources change slowly. In a typical scenario, there may be a desired source at one angle and undesired interferers at other angles. In view of these requirements, adaptive optimization of the array output can result in better performance than designs based on assumed noise field models.

A block diagram of an adaptive filter-sum array is shown in Fig. 50.18. Each microphone feeds a weighted tapped delay line (FIR filter) whose intermediate outputs are summed. If there are N microphones and M taps on each delay line, there are a total of NM tap weights h_{nm}. All weights are adapted to optimize the output according to some specified criterion which, as mentioned above, is typically based on minimizing the output power under the constraint that the desired source direction is not affected or only slightly degraded.

The considerations governing the choice of N and M for a uniform linear array are described as follows. A linear combination of the microphone signals can produce at most $N-1$ controllable nulls in the directivity pattern of the array at a single frequency. If an M-tap FIR filter is placed behind each microphone, then the same number of nulls can be produced at M frequencies ($M/2$ if the weights are real). Thus, for broadband sources, N is governed by the expected number of noise sources and M by the frequency range over which the desired signal has appreciable energy. The goal of the adaptive algorithm is to search for the optimum location of the nulls under the constraints that signals arriving from the desired direction are not too distorted in frequency response.

Adaptive arrays of this type have been considered in the past. Constrained optimization of an adaptive array like the one shown in Fig. 50.18 can be written in matrix form as

$$\min_{\boldsymbol{h}} \boldsymbol{h}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{h} \quad \text{subject to} \quad \boldsymbol{C}\boldsymbol{h} = \boldsymbol{c}\,, \tag{50.66}$$

where $\boldsymbol{R}$ is the input cross-correlation matrix, the $M \times N$ matrix $\boldsymbol{C}$ describes the M linearly independent constraints, and $\boldsymbol{c}$ is the set of M constraining values. *Frost* [50.20] developed an algorithm that minimizes the output power of the array while maintaining a flat magnitude and linear phase response for signals arriving from the desired source direction. The Frost algorithm uses a *hard constraint* such that the output contains only a delayed version of the signal arriving from the *look direction* (the look direction is the direction that is known a priori and is set by the delay values δ in Fig. 50.18). Constraints are enforced by noting that for signals propagating from the *look direction*, the signals moving through the FIR tapped-delay lines are all in phase. The matrix $\boldsymbol{C}$ sums all of the columns to form an equivalent vector $\boldsymbol{c}$ which is the response of the array to the look direction. The look-direction equivalent filter is the column sum and is shown in Fig. 50.18 as the rightmost FIR filter. For the Frost hard constraint of flat magnitude and linear phase, the vector $\boldsymbol{c}$ must be a Dirac delta function (where all taps but one tap are zero and the single nonzero tap is set to the desired gain of signals propagating from the look direction). Although the Frost algorithm is straightforward, one can extend the degrees of freedom by softening the constraint, which is useful when the number of taps is small.

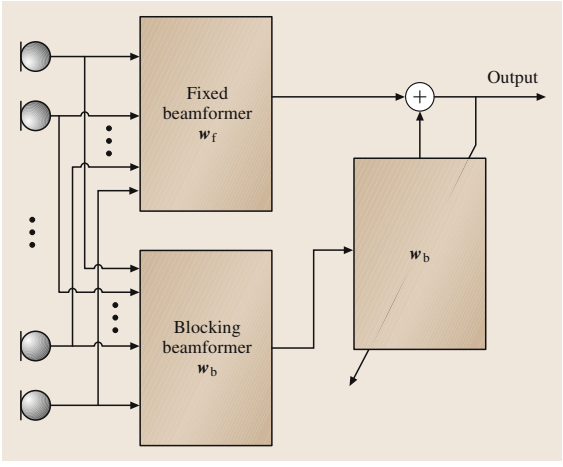


Fig. 50.19 Schematic of a generalized sidelobe canceler (GSC) adaptive array

One variant of the Frost beamformer that can simplify the computational cost by forcing the constraint into the front-end of the array processing, was proposed by *Griffiths* and *Jim* [50.21]. Their solution was to preform a set of $N-1$ difference beams that have nulls in the look direction. These *prebeams* are then used to adjust an adaptive filter-sum beamformer to minimize the total output power. Since the prebeams do not contain any look-direction signal, they implicitly invoke the look-direction constraint when combined adaptively with the main adaptive beamformer. Figure 50.19 shows a general schematic of how the Griffiths–Jim beamformer operates. In Fig. 50.19 the fixed filter-sum beamformer is denoted as $\boldsymbol{W}_{\mathrm{f}}$, the subtractive prebeamformer *blocking matrix* as $\boldsymbol{W}_{\mathrm{b}}$, and the adaptive beamformer as $\boldsymbol{W}_{\mathrm{a}}$. The general structure shown in Fig. 50.19 is known as a *generalized sidelobe canceler* (GSC) [50.18] and this structure was actually the basis of the first adaptive beamformers, which predated the development of the Frost beamformer. Intuitively, the GSC structure is appealing since it allows one to directly visualize the operation of the adaptive beamformer.

A GSC beamformer generally has one fixed main beam steered in the known look direction, and one or more secondary beams that have nulls in the look direction. These secondary beams are used to identify spatial noise sources not in the look-direction and then linearly combined with the main beam to adaptively minimize the output power or some other defined performance measure. Specific works on GSC-based beamformers for microphone arrays have been presented by *Hoshuyama* et al. [50.22], and *Herbordt* et al. [50.23].

It is well known that the Frost beamformer and GSC beamformers are sensitive to multiple paths [50.24], which manifest themselves as *signal cancelation*. Signal cancelation is due to correlated off-axis signals that cancel the desired look-direction signal by destructive interference within the filter-sum beamformer. This effect can be easily visualized by considering the operation of a GSC beamformer. In GSC beamforming, reflected look-direction signals pass the *blocking matrix* and can therefore be used by the adaptive beamformer to cancel the look-direction signal. The exact amount of signal cancelation depends on the geometry of the array as well as the acoustic environment. One practical implementation to mitigate this multipath effect is to use a smaller number of taps so that reflected signals do not enter into the adaptive FIR tapped-delay lines [50.24].

Kaneda and *Ohga* [50.25] replaced the Frost *hard* constraint with a softer constraint. Since they were

mainly interested in speech signals and realized that small perturbations to magnitude and phase were not detectable on a speech signal, their *soft* constraint allowed for some deviation in magnitude and phase for signals propagating from the look direction.

The *soft* constraint can be seen if one writes the transfer function $G(\mathrm{i}\omega)$ from the desired source to the output,

$$D = \int |G(\mathrm{i}\omega) - 1|^2 \, \mathrm{d}\omega \leq D_0, \tag{50.67}$$

where D_0 is some specified positive number.

Neither the Frost or Kaneda approaches exploited the fact that the quality of speech is less sensitive to the phase of the transfer function from the desired source direction to the array output. In order to capitalize on this property, a different soft-constraint algorithm was proposed by *Sondhi* et al. [50.24] that places a constraint on only the magnitude of the transfer function. This modified constraint results in an output magnitude response variation defined as

$$D = \int \left[|G(\mathrm{i}\omega)|^2 - 1 \right]^2 \mathrm{d}\omega. \tag{50.68}$$

Fortunately a computationally simple algorithm can be found that uses a gradient search to minimize the normalized output power with the constraint that

$$D \leq D_0 \, . \tag{50.69}$$

A constraint such as (50.69) is difficult to impose directly. Therefore a search to minimize another cost function C defined as

$$C = P + \epsilon D \tag{50.70}$$

leads to a solution, where P is the normalized output power and ϵ is a Lagrange multiplier. The value D obtained at the optimum setting of the filters is then implicitly a function of the parameter ϵ. By varying ϵ, a desired value of the distortion is realized.

Figure 50.18 contains a block labeled *output filter*. The purpose of this filter is to compensate for the frequency response deviation that occurs in the softly constrained adaptive beamformer. In order for the array to steer nulls at frequencies where wavelengths are longer than the array length, the array must resort to pattern differencing, which results in a commensurate high-pass response in the look direction. If one constrains the array too tightly, the adaptive algorithm focuses on maintaining flat array response and not on minimizing the output power. By allowing larger-magnitude distortion at lower frequencies, the adaptive algorithm focuses on minimizing the output power at these lower frequencies. The inverse filter acts to compensate the commensurate low-frequency roll-off distortion. An approach that accomplishes a similar result was introduced by *Gooch* and *Shynk* [50.26], who proposed an adaptive array containing both pole and zero filters. This can effectively handle the constrained adaptive array and the frequency response change due to superdirectional operation.

50.7 Conclusions

This chapter has introduced various types of beamforming microphone array systems and discussed some of the fundamental theory of their operation, design, and implementation. With the continuing trend of lower cost electronics, it is now feasible to implement microphone arrays in a variety of low-cost consumer audio communication devices. It was shown that appropriately designed microphone arrays have the ability to offer directional gains that can significantly improve the quality of hands-free speech pickup in reverberant and noisy environments.

References

50.1 P.M. Morse, K.U. Ingard: *Theoretical Acoustics* (McGraw Hill, New York 1968)

50.2 B.D. Steinberg: *Principles of Aperture and Array System Design* (Wiley, New York 1976)

50.3 D.H. Johnson, D.E. Dudgeon: *Array Signal Processing: Concepts and Techniques* (Prentice-Hall, Upper Saddle River 1993)

50.4 J.L. Flanagan, J.D. Johnston, R. Zahn, G.W. Elko: Computer-steered microphone arrays for sound transduction in large rooms, J. Acoust. Soc. Am. **78**, 1508–1518 (1985)

50.5 W. Kellermann: A self steering digital microphone array, Proc. IEEE ICASSP **1991**, 3581–3584 (1991)

50.6 R.J. Lustberg: *Acoustic Beamforming Using Acoustic Arrays* (MIT Masters Thesis, MIT Cambridge 1993)

50.7 T.C. Chou: *Broadband Frequency-Independent Beamforming* (MIT Masters Thesis, MIT Cambridge 1994)

50.8 M.M. Goodwin: *Implementation and Applications of Electroacoustic Array Beamformers* (MIT Masters Thesis, MIT Cambridge 1992)

50.9 D.B. Ward, R.A. Kennedy, R.C. Williamson: FIR filter design for frequency invariant beamformers, IEEE Signal Process. **3**, 69–71 (1996)

50.10 D.K. Cheng, F.I. Tseng: Gain optimization for arbitrarily antenna arrays, IEEE Trans. Antennas Prop. **AP-13**, 973–974 (1965)

50.11 G.W. Elko: Differential microphone arrays. In: *Audio Signal Processing for Next Generation Multimedia Communication Systems*, ed. by Y. Huang, J. Benesty (Kluwer Academic, Dordrecht 2004)

50.12 N. Yaru: A note on super-gain antenna arrays, Proc IRE **39**, 1081–1085 (1951)

50.13 E.N. Gilbert, S.P. Morgan: Optimum design of antenna arrays subject to random variations, Bell System Tech. J. **34**, 637–663 (1955)

50.14 H. Cox, R. Zeskind, T. Kooij: Practical supergain, IEEE Trans. ASSP **34**, 393–398 (1986)

50.15 J. Meyer, G.W. Elko: A highly scalable spherical microphone array based on an orthonormal decomposition of the soundfield, IEEE ICASSP **2002**, 1781–1784 (2002)

50.16 J. Meyer, G.W. Elko: Spherical Microphone Arrays for 3D Sound Recording. In: *Audio Signal Processing for Next Generation Multimedia Communication Systems*, ed. by Y. Huang, J. Benesty (Kluwer Academic, Dordrecht 2004)

50.17 E.G. Williams: *Fourier Acoustics* (Academic Press, San Diego 1999)

50.18 R.A. Monzingo, T.W. Miller: *Introduction to Adaptive Arrays* (Wiley, New York 1980)

50.19 L. Chen, J. Benesty, Y. Huang: Time delay estimation in room acoustic environments: an overview, EURASIP Appl. Signal Process. **2006**, ID26503 (1972)

50.20 O.L. Frost: An algorithm for linear constrained adaptive array processing, Proc. IEEE **60**, 926–935 (1972)

50.21 L.J. Griffiths, C.W. Jim: An alternative approach to linearly constrained beamforming, IEEE Trans. Antennas Prop. **AP-30**, 27–34 (1982)

50.22 O. Hoshuyama, A. Sugiyama, A. Hirano: A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters, IEEE Trans. Signal Process. **47**, 2677–2684 (1999)

50.23 W. Herboldt, W. Kellermann: Analysis of blocking matrices for generalized sidelobe cancellers for non-stationary broadband signals, Proc. IEEE ICASSP **2002**, 4187–4191 (2002)

50.24 M.M. Sondhi, G.W. Elko: Adaptive optimization of microphone arrays under a nonlinear constraint, Proc. IEEE ICASSP **2002**, 981–984 (2002)

50.25 Y. Kaneda, J. Ohga: Adaptive microphone array system for noise reduction, IEEE Trans. ASSP **34**, 1391–1400 (1986)

50.26 R.P. Gooch, J.J. Shynk: Wide-band adaptive array processing using pole-zero digital filters, IEEE Trans. Antennas Prop. **AP-34**, 355–367 (1985)

Amazon EX1010 Page 1039

51. Time Delay Estimation and Source Localization

Y. Huang, J. Benesty, J. Chen

A fundamental requirement of microphone arrays is the capability of instantaneously locating and continuously tracking a speech sound source. The problem is challenging in practice due to the fact that speech is a nonstationary random process with a wideband spectrum, and because of the simultaneous presence of noise, room reverberation, and other interfering speech sources. This Chapter presents an overview of the research and development on this technology in the last three decades. Focusing on a two-stage framework for speech source localization, we survey and analyze the state-of-the-art time delay estimation (TDE) and source localization algorithms.

This chapter is organized into two sections. In Sect. 51.2, we will study the TDE problem and review a number of cutting-edge TDE algorithms, ranging from the generalized cross-correlation methods to blind multichannel-identification-based algorithms and the second-order statistics-based multichannel cross-correlation coefficient method to the higher-order statistics-based entropy-minimization approach. In Sect. 51.3, we will investigate the source localization problem from the perspective of estimation theory. The emphasis is on least-squares estimators with closed-form estimates. The spherical intersection, spherical interpolation, and linear-correction spherical interpolation algorithms will be presented.

51.1 **Technology Taxonomy** 1043
51.2 **Time Delay Estimation** 1044
51.2.1 Problem Formulation and Signal Models 1044
51.2.2 The Family of the Generalized Cross-Correlation Methods 1045
51.2.3 Adaptive Eigenvalue Decomposition Algorithm 1047
51.2.4 Adaptive Blind Multichannel Identification Based Methods 1048
51.2.5 Multichannel Spatial Prediction and Interpolation Methods 1049
51.2.6 Multichannel Cross-Correlation Coefficient Algorithm 1050
51.2.7 Minimum-Entropy Method 1052
51.3 **Source Localization** 1054
51.3.1 Problem Formulation 1054
51.3.2 Measurement Model and Cramèr–Rao Lower Bound 1054
51.3.3 Maximum-Likelihood Estimator 1055
51.3.4 Least-Squares Estimators 1056
51.3.5 Least-Squares Error Criteria 1056
51.3.6 Spherical Intersection (SX) Estimator 1057
51.3.7 Spherical Interpolation (SI) Estimator 1057
51.3.8 Linear-Correction Least-Squares Estimator 1058
51.4 **Summary** 1061
References 1062

51.1 Technology Taxonomy

Thanks to the explosive growth in computing power and the continuous drop in the cost and size of acoustic electric transducers, a growing number of speech processing and communication systems can afford to employ multiple microphones. There are two major application areas that are driving the interest in microphone array technologies:

- distant speech acquisition that permits good-quality sound pickup due to the capability to mitigate the effects of room reverberation, ambient noise, and other interfering sound sources; and
- acoustic surveillance and monitoring where sound caused by the events of particular interest is detected, localized, and tracked.

In both applications, there is a primary need to localize a speech source by analyzing the acoustic signals captured by various microphones placed at different positions. A good example of the former is microphone

array beam steering [51.1–4] and a good example of the latter is automatic camera pointing for videoconferencing [51.5–8].

The problem of locating radiative point sources using sensor arrays has long been of great research interest given its theoretical as well as practical importance in a great variety of applications, e.g., radar [51.9, 10], underwater sonar [51.11], and seismology [51.12]. In these applications, source localization is more commonly referred to as direction of arrival (DOA) estimation. A class of celebrated approaches is based on high-resolution spectral analysis, including Capon's minimum variance (MV) spectral estimation method [51.13], and eigenanalysis-based techniques, such as the popular multiple signal classification (MUSIC) algorithm [51.14]. These methods perform statistical fit for DOA with respect to a spatiospectral correlation matrix derived from the signals received at the sensors. By assumption, the source signal needs to be statistically stationary and narrowband, and the source is located in the far field of the sensor array. However, these basic premises are barely met by speech sources. Moreover, the multipath effect is not taken into account in the formulation of DOA estimation. Therefore, the high-resolution DOA estimation algorithms were found to be incompetent for speech source localization, particularly in reverberant acoustic environments.

For speech source localization, microphone array beam scanning and time delay estimation (TDE)-based localization methods are the two most widely used approaches. A beamformer is a spatial filter that operates on the microphone outputs in order to enhance the signal coming from one direction while suppressing noise and interference from other directions. Therefore, steering a microphone array beamformer and scanning across a room for the highest-energy output leads to an estimate of the direction of an active speech source. Using two arrays and by intersecting the two corresponding direction estimates, the source location is determined [51.15, 16]. While the beam scanning method has the remarkable advantage that it can easily be extended to the case of simultaneously localizing multiple speech sources [51.17], it has some intrinsic drawbacks. Since speech is a typical broadband signal, ideally a broadband beamformer whose beam pattern would be the same across the entire speech spectrum needs to be developed, which is a technically challenging problem. In addition, most beamforming algorithms assume that the speech sound source is in the far field in order to make the design problem analytically tractable. When the speech source is close to the microphone array, performance degradation can be expected. On the contrary, if the speech source is in the far field, the localization resolution may not be satisfactory as the source direction is only selected from a set of discrete beam scanning angles. The number of beam scanning angles is therefore a trade-off between resolution and computational complexity.

Alternatively, a TDE-based source localization algorithm involves a two-step procedure [51.18]. In the first step, a set of relative time differences of arrival (TDOAs) for different microphone pairs are calculated. In the second step, the acoustic source location is estimated from the TDOAs and using the a priori knowledge about the locations of the microphones. TDE-based source localization algorithms have many merits: they cope well with both narrowband and broadband source signals; their localization resolution can be flexibly adjusted by varying the sampling rate and microphone array size; the effect of room reverberation is only of concern in the first-step TDE and several recently developed robust TDE algorithms are promising for future practical use; and in general they are computationally efficient. Therefore, after continuous investigation over the last two decades, TDE-based speech source localization schemes have become the technique of choice, especially in recent digital systems. In this chapter, an overview of the state of the art of this class of source localization methods is presented.

51.2 Time Delay Estimation

51.2.1 Problem Formulation and Signal Models

Suppose that there is one speech sound source and N microphones. The TDE problem is concerned with the computation of the relative time difference of arrival (TDOA) between different microphone signals.

Depending on the surrounding acoustic environment, there are two signal models for the TDE problem: the ideal free-field model and the real reverberant model. The former assumes no room reverberation while the latter uses an acoustic channel impulse response, usually a finite impulse response (FIR) filter, to describe the effect of room reverberation.

Ideal Free-Field Model

In an anechoic open space, as shown in Fig. 51.1a, the speech source signal $s(k)$ propagates radiatively and the sound level falls off as a function of distance from the source. Then the signal captured by the n-th microphone at time k can be expressed as follows:

$$x_n(k) = \alpha_n s(k-\tau_n) + b_n(k), \quad n = 1, 2, \cdots, N, \tag{51.1}$$

where α_n $(0 \le \alpha_n \le 1)$ is an attenuation factor due to propagation loss, τ_n is the propagation time, and $b_n(k)$ is additive noise. The time difference of arrival (TDOA) between the i-th and j-th microphones is defined as,

$$\tau_{ij} \triangleq \tau_i - \tau_j, \quad i, j = 1, 2, \cdots, N. \tag{51.2}$$

The noise signal $b_n(k)$ is presumed to be a zero-mean, white, Gaussian random process, and is uncorrelated with the source signal as well as the noise signals at other microphones.

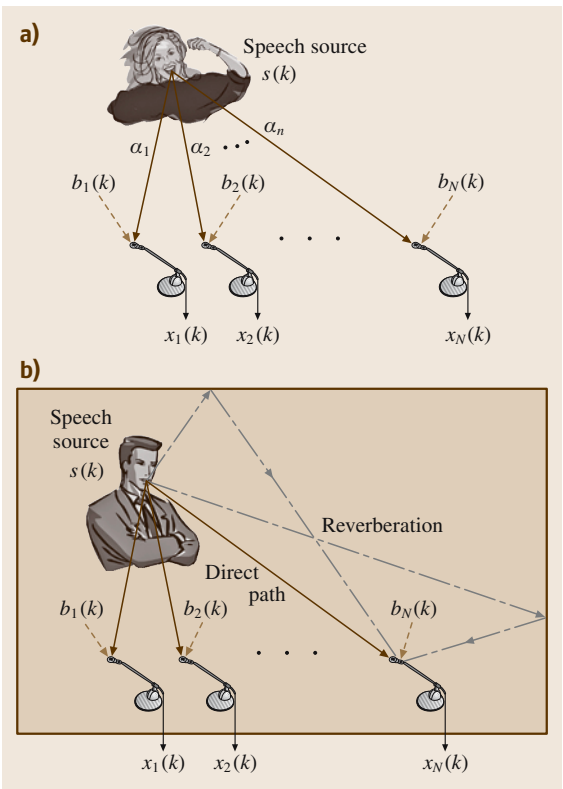


Fig. 51.1a,b Illustration of the two acoustic signal models for time delay estimation: **(a)**ideal free-field model and **(b)**real reverberant model

Real Reverberant Model

While the ideal free-field model has the merit of being simple, it does not take into account room reverberation. Therefore, in a real reverberant environment, the ideal free-field model is inadequate and problematic, and we need a more-comprehensive and more-informative alternative to describe the effect of multipath propagation. The real reverberant model treats the acoustic impulse response with an FIR filter, as illustrated by Fig. 51.1b. In such a single-input multiple-output (SIMO) system, the n-th microphone signal is given by

$$x_n(k) = h_n * s(k) + b_n(k), \quad n = 1, 2, \cdots, N, \tag{51.3}$$

where h_n is the n-th channel impulse response, and the symbol $*$ denotes the linear convolution operator. In vector/matrix form, (51.3) can be rewritten as

$$\boldsymbol{x}_n(k) = \boldsymbol{H}_n \cdot \boldsymbol{s}(k) + \boldsymbol{b}_n(k), \quad n = 1, 2, \cdots, N, \tag{51.4}$$

where

$$\boldsymbol{x}_n(k) = \left[x_n(k) \ \cdots \ x_n(k-L+1) \right]^{\mathrm{T}},$$

$$\boldsymbol{H}_n = \begin{pmatrix} h_{n,0} & \cdots & h_{n,L-1} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & h_{n,0} & \cdots & h_{n,L-1} \end{pmatrix},$$

$$\boldsymbol{s}(k) = \begin{bmatrix} s(k) & s(k-1) & \cdots & s(k-L+1) \\ \cdots & s(k-2L+2) & \end{bmatrix}^{\mathrm{T}},$$

$$\boldsymbol{b}_n(k) = \left[b_n(k) \ \cdots \ b_n(k-L+1) \right]^{\mathrm{T}},$$

where $[\cdot]^{\mathrm{T}}$ denotes the transpose of a vector or a matrix, and L is the length of the longest channel impulse response in this SIMO system.

In contrast to the ideal free-field model, the time delay τ_n in the real reverberant model is an implicit or hidden parameter. Using such a model, TDE can be performed only after the SIMO system is *blindly* identified (since the source signal is unknown), which looks like a more-difficult problem but is fortunately not insurmountable.

51.2.2 The Family of the Generalized Cross-Correlation Methods

The generalized cross-correlation (GCC) algorithm proposed by *Knapp* and *Carter* [51.19] is so far the most widely used approach to TDE. It employs the ideal free-field model (51.1) and considers only two microphones,

i. e., $N = 2$. Then the TDOA estimate between the two microphones is obtained as the time lag that maximizes the cross-correlation between the filtered signals of the microphone outputs:

$$\hat{\tau}_{12}^{\mathrm{GCC}} = \arg\max_{\tau} r_{x_1x_2}^{\mathrm{GCC}}(\tau)\,, \tag{51.5}$$

where

$$\begin{aligned} r_{x_1x_2}^{\mathrm{GCC}}(\tau) &\triangleq \mathcal{F}^{-1}\left\{\Psi_{x_1x_2}(f)\right\} \\ &= \int_{-\infty}^{\infty} \Psi_{x_1x_2}(f)\mathrm{e}^{\mathrm{i}2\pi f\tau}\,\mathrm{d}f \\ &= \int_{-\infty}^{\infty} \Phi(f)S_{x_1x_2}(f)\mathrm{e}^{\mathrm{i}2\pi f\tau}\,\mathrm{d}f \end{aligned} \tag{51.6}$$

is the GCC function, $\mathcal{F}^{-1}\{\cdot\}$ stands for the inverse discrete-time Fourier transform (IDTFT),

$$S_{x_1x_2}(f) \triangleq E\left\{X_1(f)X_2^*(f)\right\} \tag{51.7}$$

is the cross spectrum, $\Phi(f)$ is a frequency-domain weighting function,

$$\Psi_{x_1x_2}(f) = \Phi(f)S_{x_1x_2}(f) \tag{51.8}$$

is the generalized cross spectrum,

$$X_n(f) = \sum_k x_n(k)\mathrm{e}^{-\mathrm{i}2\pi fk}, \quad n = 1, 2\,,$$

where $E\{\cdot\}$ denotes mathematical expectation, and $(\cdot)^*$ denotes complex conjugate.

There are many different choices of the frequency-domain weighting function $\Phi(f)$, leading to a variety of different GCC methods for TDE. In the following, several most known and practically most useful algorithms in this family will be discussed.

Classical Cross-Correlation

The classical cross-correlation (CCC) method takes $\Phi(f) = 1$. We know from the ideal free-field model (51.1) that

$$X_n(f) = \alpha_n\mathrm{e}^{-\mathrm{i}2\pi f\tau_n}S(f) + B_n(f), \quad n = 1, 2\,. \tag{51.9}$$

Substituting (51.9) into (51.8) and noting that the noise signal at one microphone is uncorrelated with the source signal and the noise signal at the other microphone by assumption, we have

$$\Psi_{x_1x_2}^{\mathrm{CCC}}(f) = \alpha_1\alpha_2\mathrm{e}^{-\mathrm{i}2\pi f\tau_{12}}E\left\{|S(f)|^2\right\}\,. \tag{51.10}$$

The fact that $\Psi_{x_1x_2}^{\mathrm{CCC}}(f)$ depends on the source signal is detrimental for TDE since speech is inherently nonstationary.

Smoothed Coherence Transform

In order to overcome the impact of fluctuating levels of the speech source signal on TDE, an effective way is to pre-whiten the microphone outputs before their cross spectrum is computed. This is equivalent to choosing

$$\Phi(f) = \frac{1}{\sqrt{E\left\{|X_1(f)|^2\right\}E\left\{|X_2(f)|^2\right\}}}\,, \tag{51.11}$$

which leads to the so-called smoothed coherence transform (SCOT) method [51.20]. Substituting (51.9) and (51.11) into (51.8) produces the SCOT cross-spectrum:

$$\begin{aligned} \Psi_{x_1x_2}^{\mathrm{SCOT}}(f) &= \frac{\alpha_1\alpha_2\mathrm{e}^{-\mathrm{i}2\pi f\tau_{12}}E\left\{|S(f)|^2\right\}}{\sqrt{E\left\{|X_1(f)|^2\right\}E\left\{|X_2(f)|^2\right\}}} \\ &= \frac{\alpha_1\alpha_2\mathrm{e}^{-\mathrm{i}2\pi f\tau_{12}}E\{|S(f)|^2\}}{\sqrt{\alpha_1^2E\{|S(f)|^2\}+\sigma_{b_1}^2(f)}\cdot} \\ &\quad\times\frac{1}{\sqrt{\alpha_2^2E\{|S(f)|^2\}+\sigma_{b_2}^2(f)}} \\ &= \frac{\mathrm{e}^{-\mathrm{i}2\pi f\tau_{12}}}{\sqrt{1+\frac{1}{\mathrm{SNR}_1(f)}}\cdot\sqrt{1+\frac{1}{\mathrm{SNR}_2(f)}}}\,, \end{aligned} \tag{51.12}$$

where

$$\begin{aligned} \sigma_{b_n}^2(f) &= E\left\{|B_n(f)|^2\right\}, \\ \mathrm{SNR}_n(f) &= \frac{\alpha_nE\{|S(f)|^2\}}{E\left\{|B_n(f)|^2\right\}}, \quad n = 1, 2\,. \end{aligned}$$

If the signal-to-noise ratios (SNRs) are the same at the two microphones, then we get

$$\Psi_{x_1x_2}^{\mathrm{SCOT}}(f) = \left(\frac{\mathrm{SNR}(f)}{1+\mathrm{SNR}(f)}\right)\cdot\mathrm{e}^{-\mathrm{i}2\pi f\tau_{12}}. \tag{51.13}$$

Therefore, the performance of the SCOT algorithm for TDE would vary with the SNR. But when the SNR is large enough,

$$\Psi_{x_1x_2}^{\mathrm{SCOT}}(f) \approx \mathrm{e}^{-\mathrm{i}2\pi f\tau_{12}}\,,$$

which implies independent TDE performance of the power of the source signal. The SCOT method is theoretically superior to the CCC method only when the noise level is low.

Phase Transform

It becomes clear by examining (51.6) that the TDOA information is conveyed in the phase rather than the amplitude of the cross-spectrum. Therefore, we can simply discard the amplitude and only keep the phase. By setting

$$\Phi(f) = \frac{1}{\left|S_{x_1x_2}(f)\right|}\,, \tag{51.14}$$

we get the phase transform (PHAT) method [51.19]. In this case, the generalized cross spectrum is given by

$$\Psi_{x_1x_2}^{\mathrm{PHAT}}(f) = \mathrm{e}^{-\mathrm{i}2\pi f\tau_{12}}, \tag{51.15}$$

which depends only on the TDOA τ_{12}. Substituting (51.15) into (51.6), we obtain an ideal GCC function:

$$r_{x_1x_2}^{\mathrm{PHAT}}(\tau) = \int_{-\infty}^{\infty} \mathrm{e}^{\mathrm{i}2\pi f(\tau-\tau_{12})}\,\mathrm{d}f = \begin{cases} \infty, & \tau = \tau_{12}, \\ 0, & \text{otherwise.} \end{cases} \tag{51.16}$$

As a result, the PHAT method in general performs better than the CCC and SCOT methods for TDE with respect to a speech sound source.

Discussion: The GCC methods are computationally efficient. They induce very short decision delays and hence have good tracking capability: an estimate is produced almost instantaneously. The GCC methods have been well studied and are found to perform fairly well in moderately noisy and nonreverberant environments [51.21, 22]. In order to improve their robustness to additive noise, many amendments have been proposed [51.6, 23–25]. However, these methods still tend to break down when room reverberation is high. This is insightfully explained by the fact that the GCC methods model the surrounding acoustic environment as an ideal free field and thus have a fundamental weakness in their inability to cope with room reverberation. Cepstral prefiltering was suggested to overcome the multipath propagation effect of room reverberation in [51.26]. However, the improvement is limited and shortcomings still remain. Therefore, the main focus of current research in TDE is on combating the effect of room reverberation.

51.2.3 Adaptive Eigenvalue Decomposition Algorithm

The adaptive eigenvalue decomposition (AED) algorithm [51.27] approaches the TDE problem from a different point of view from the traditional GCC methods. While the AED also considers only two microphones, it adopts the real reverberant rather than the ideal free-field model. It first identifies the two channel impulse responses, and then measures the time difference between the two recognized direct paths as the TDOA estimate. Since the source signal is unknown, the channel identification has to be a blind method.

Following the real reverberant model (51.3) and the fact that, in the absence of additive noise,

$$x_1(k) * h_2 = s(k) * h_1 * h_2 = x_2(k) * h_1\,, \tag{51.17}$$

we deduce the following cross relation in vector/matrix form at time k:

$$\boldsymbol{x}^{\mathrm{T}}(k)\boldsymbol{u} = \boldsymbol{x}_1^{\mathrm{T}}(k)\boldsymbol{h}_2 - \boldsymbol{x}_2^{\mathrm{T}}(k)\boldsymbol{h}_1 = 0, \tag{51.18}$$

where

$$\begin{aligned} \boldsymbol{x}(k) &= \left[\boldsymbol{x}_1^{\mathrm{T}}(k)\ \ \boldsymbol{x}_2^{\mathrm{T}}(k)\right]^{\mathrm{T}}, \\ \boldsymbol{u} &= \left[\boldsymbol{h}_2^{\mathrm{T}}\ \ -\boldsymbol{h}_1^{\mathrm{T}}\right]^{\mathrm{T}}, \\ \boldsymbol{h}_n &= \left[h_{n,0}\ \ h_{n,1}\ \ \cdots\ \ h_{n,L-1}\right]^{\mathrm{T}}, \quad n = 1, 2\,. \end{aligned}$$

Multiplying (51.18) by $\boldsymbol{x}(k)$ from the left-hand side and taking the expectation yields

$$\boldsymbol{R}_{xx}\boldsymbol{u} = \boldsymbol{0}\,, \tag{51.19}$$

where $\boldsymbol{R}_{xx} \triangleq E\left\{\boldsymbol{x}(k)\boldsymbol{x}^{\mathrm{T}}(k)\right\}$ is the covariance matrix of the two microphone signals. This indicates that the vector $\boldsymbol{u}$, which consists of the two impulse responses, is in the null space of $\boldsymbol{R}$. More specifically, $\boldsymbol{u}$ is the eigenvector of $\boldsymbol{R}_x$ corresponding to its eigenvalue 0. If $\boldsymbol{R}_x$ is rank deficient by 1, $\boldsymbol{u}$ can be uniquely determined up to a scale, which is equivalent to saying that the two-channel SIMO system can be blindly identified. Using what has been proved in [51.28], we know that such a two-channel acoustic SIMO system is blindly identifiable using only the second-order statistics (SOS) of the microphone outputs if and only if the following two conditions hold:

- The polynomials formed from $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ are coprime , i. e., their channel transfer functions share no common zeros ,
- The autocorrelation matrix $\boldsymbol{R}_{ss} = E\{\boldsymbol{s}(k)\boldsymbol{s}^{\mathrm{T}}(k)\}$ of the source signal is of full rank (such that the SIMO system can be fully excited).

In practice, noise always exists and the covariance matrix $\boldsymbol{R}_{xx}$ is positively definite rather than positively semidefinite. As a consequence, $\boldsymbol{u}$ is found as the normalized eigenvector of $\boldsymbol{R}_{xx}$ corresponding to its smallest eigenvalue:

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}} \boldsymbol{u}^{\mathrm{T}}\boldsymbol{R}_{xx}\boldsymbol{u}, \quad \text{subject to } \|\boldsymbol{u}\|^2 = 1\,. \tag{51.20}$$

In the AED algorithm, solving (51.20) is carried out adaptively using a constrained LMS algorithm:

Initialize

$$\hat{\boldsymbol{h}}_n(0) = \left[\tfrac{\sqrt{2}}{2} \;\; 0 \;\cdots\; 0\right]^{\mathrm{T}}, \quad n = 1, 2\,;$$

$$\hat{\boldsymbol{u}}(0) = \left[\hat{\boldsymbol{h}}_2^{\mathrm{T}}(0) \;\; -\hat{\boldsymbol{h}}_1^{\mathrm{T}}(0)\right]^{\mathrm{T}}\,;$$

Compute, $\quad k = 0, 1, \cdots$

$$e(k) = \hat{\boldsymbol{u}}^{\mathrm{T}}(k)\boldsymbol{x}(k),$$

$$\hat{\boldsymbol{u}}(k+1) = \frac{\hat{\boldsymbol{u}}(k) - \mu e(k)\boldsymbol{x}(k)}{\|\hat{\boldsymbol{u}}(k) - \mu e(k)\boldsymbol{x}(k)\|}\,, \tag{51.21}$$

where the adaptation step size μ is a small positive constant.

After the AED algorithm converges, the time difference between the direct paths of the two identified channel impulse responses $\hat{\boldsymbol{h}}_1$ and $\hat{\boldsymbol{h}}_2$ is measured as the TDOA estimate:

$$\hat{\tau}_{12}^{\mathrm{AED}} = \arg\max_l |\hat{\boldsymbol{h}}_{1,l}| - \arg\max_l |\hat{\boldsymbol{h}}_{2,l}|\,. \tag{51.22}$$

51.2.4 Adaptive Blind Multichannel Identification Based Methods

The AED algorithm provides us with a new way to look at the TDE problem, which was found to be particularly useful to combat room reverberation. It applies the more-realistic real-reverberant model to a two-microphone acoustic system at a time and attempts to blindly identify the two-channel impulse responses, from which the embedded TDOA information of interest is then extracted. Clearly the blind two-channel identification technique plays a central role in such an approach. The more accurately the two impulse responses are blindly estimated, the more precisely the TDOA can be inferred. However, for a two-channel system, the zeros of the two channels can be close, especially when their impulse responses are long, which leads to an ill-conditioned system that is difficult to identify. If they share some common zeros, the system becomes unidentifiable (using only second-order statistics) and the AED algorithm may completely fail. It was suggested in [51.29] that this problem can be alleviated by employing more microphones. When more microphones are employed, it is less likely that all channels will share a common zero. As such, blind identification deals with a more well-conditioned SIMO system and the solutions can be globally optimized over all channels. The resulting algorithm is referred to as the adaptive blind multichannel identification (ABMCI)-based TDE algorithm.

The generalization of blind SIMO identification from two channels to multiple (> 2) channels is not straightforward and in [51.30] a systematic way was proposed. Consider a SIMO system with N channels whose outputs are described by (51.3). Each pair of the system outputs has a cross-relation similar to (51.18) in the absence of noise:

$$\boldsymbol{x}_i^{\mathrm{T}}(k)\boldsymbol{h}_j = \boldsymbol{x}_j^{\mathrm{T}}(k)\boldsymbol{h}_i\,, \quad i, j = 1, 2, \cdots, N\,. \tag{51.23}$$

When noise is present or the channel impulse responses are improperly modeled, the cross-relation does not hold and the a priori error signal can be defined as

$$e_{ij}(k+1) = \frac{\boldsymbol{x}_i^{\mathrm{T}}(k+1)\hat{\boldsymbol{h}}_j(k) - \boldsymbol{x}_j^{\mathrm{T}}(k+1)\hat{\boldsymbol{h}}_i(k)}{\|\hat{\boldsymbol{h}}(k)\|}\,, \quad i, j = 1, 2, \cdots, N\,, \tag{51.24}$$

where $\hat{\boldsymbol{h}}_i(k)$ is the model filter for the i-th channel at time k and

$$\hat{\boldsymbol{h}}(k) = \left[\hat{\boldsymbol{h}}_1^{\mathrm{T}}(k) \;\; \hat{\boldsymbol{h}}_2^{\mathrm{T}}(k) \;\cdots\; \hat{\boldsymbol{h}}_N^{\mathrm{T}}(k)\right]^{\mathrm{T}}\,.$$

The model filters are normalized to avoid a trivial solution whose elements are all zeros. Based on the error signal defined here, a cost function at time $k+1$ is given by

$$J(k+1) = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \mathrm{e}_{ij}^2(k+1)\,. \tag{51.25}$$

The multichannel LMS (MCLMS) algorithm updates the estimate of the channel impulse responses as follows:

$$\hat{\boldsymbol{h}}(k+1) = \hat{\boldsymbol{h}}(k) - \mu\nabla J(k+1), \tag{51.26}$$

where μ is again a small positive step size. As shown in [51.30], the gradient of $J(k+1)$ is computed as:

$$\nabla J(k+1) = \frac{\partial J(k+1)}{\partial \hat{\boldsymbol{h}}(k)} = \frac{2\left[\tilde{\boldsymbol{R}}_{x+}(k+1)\hat{\boldsymbol{h}}(k) - J(k+1)\hat{\boldsymbol{h}}(k)\right]}{\|\hat{\boldsymbol{h}}(k)\|^2}\,, \tag{51.27}$$

where

$$\tilde{\boldsymbol{R}}_{x+}(k) \triangleq \begin{pmatrix} \sum_{n\neq 1} \tilde{\boldsymbol{R}}_{x_n x_n}(k) & -\tilde{\boldsymbol{R}}_{x_2 x_1}(k) & \cdots & -\tilde{\boldsymbol{R}}_{x_N x_1}(k) \\ -\tilde{\boldsymbol{R}}_{x_1 x_2}(k) & \sum_{n\neq 2} \tilde{\boldsymbol{R}}_{x_n x_n}(k) & \cdots & -\tilde{\boldsymbol{R}}_{x_N x_2}(k) \\ \vdots & \vdots & \ddots & \vdots \\ -\tilde{\boldsymbol{R}}_{x_1 x_N}(k) & -\tilde{\boldsymbol{R}}_{x_2 x_N}(k) & \cdots & \sum_{n\neq N} \tilde{\boldsymbol{R}}_{x_n x_n}(k) \end{pmatrix}$$

and

$$\tilde{\boldsymbol{R}}_{x_i x_j}(k) \triangleq \boldsymbol{x}_i(k)\boldsymbol{x}_j^{\mathrm{T}}(k)\,, \quad i, j = 1, 2, \cdots, N\,.$$

If the model filters are always normalized after each update, then a simplified MCLMS algorithm is obtained:

$$\hat{\boldsymbol{h}}(k+1) = \frac{\hat{\boldsymbol{h}}(k) - 2\mu\left[\tilde{\boldsymbol{R}}_{x+}(k+1)\hat{\boldsymbol{h}}(k) - J(k+1)\hat{\boldsymbol{h}}(k)\right]}{\left\|\hat{\boldsymbol{h}}(k) - 2\mu\left[\tilde{\boldsymbol{R}}_{x+}(k+1)\hat{\boldsymbol{h}}(k) - J(k+1)\hat{\boldsymbol{h}}(k)\right]\right\|}\,. \tag{51.28}$$

A number of other adaptive blind SIMO identification algorithms with faster convergence and lower computational complexity have also been developed, e.g., [51.31, 32]. Due to space limitations, we choose not to present the development of those algorithms and refer the reader to [51.33] and references therein for more details.

After the adaptive algorithm converges, the TDOA between any two microphones can be inferred as

$$\hat{\tau}_{ij}^{\mathrm{ABMCI}} = \arg\max_l |\hat{h}_{i,l}| - \arg\max_l |\hat{h}_{j,l}|\,, \quad i, j = 1, 2, \cdots, N\,, \tag{51.29}$$

which implies that in every channel the direct path is always dominant. This is generally true for acoustic waves, which would be considerably attenuated by wall reflection. But sometimes two or more reverberant signals via multipaths of equal delay could add coherently such that the direct-path component no longer dominates the impulse response. Therefore a more-robust way to pick the direct-path component is to identify the Q ($Q > 1$) strongest elements in the impulse responses and choose the one with the smallest delay [51.29]:

$$\hat{\tau}_{ij}^{\mathrm{ABMCI}} = \min\left\{\arg\max_l{}^q |\hat{h}_{i,l}|\right\} - \min\left\{\arg\max_l{}^q |\hat{h}_{j,l}|\right\}\,, \quad i, j = 1, 2, \cdots, N\,, \quad q = 1, 2, \cdots, Q\,, \tag{51.30}$$

where $\max^q$ computes the q-th largest element.

51.2.5 Multichannel Spatial Prediction and Interpolation Methods

While the blind SIMO identification-based TDE algorithms cope well with room reverberation, they have relatively poor tracking capability. As a result, in a rapidly varying acoustic environment, they will have a problem. From this perspective, cross-correlation-based methods do have some advantages and so there have been continuous efforts to improve their robustness against noise and reverberation.

In Sect. 51.2, only two microphones was considered. In this section, we explore the possibility of using multiple microphones (greater than two) to improve the estimate accuracy of the TDOA between the first two microphones in adverse acoustic environments. For a three-microphone system, the three TDOAs, namely τ_{12}, τ_{13}, and τ_{23}, are apparently not independent but obey the following relation: $\tau_{13} = \tau_{12} + \tau_{23}$. The idea of exploiting TDOA redundancy in sensor arrays was first conceived in [51.34] and a Kalman-filtering-based two-stage TDE algorithm was proposed. Recently, following a similar line of thoughts, several fusion algorithms have been developed [51.35–37]. In what follows, we present a multichannel TDE algorithm using spatial prediction and interpolation [51.38, 39], which takes advantage of the TDOA redundancy among multiple microphones in a more-intuitive way.

Suppose that there are N ($N > 2$) microphones in an open space to which the ideal free field can be properly applied. The N microphones are positioned in a prespecified geometric pattern such that the TDOA of a microphone pair can be expressed as a function of the TDOAs of a *small* number of other microphone pairs. The number depends on the distance of the source from the microphone array and on whether the microphones constitute a two-dimensional (2-D) or three-dimensional (3-D) array. For ease of presentation, let us consider a linear array and a far-field speech source. Then we

have

$$\tau_{1n} = \tau_1 - \tau_n = f_n(\tau_{12}) , \quad n = 1, 2, \cdots , N , \tag{51.31}$$

where

$$f_1(\tau_{12}) = 0 , \quad f_2(\tau_{12}) = \tau_{12} .$$

Substituting (51.31) into (51.1) gives

$$x_n(k) = \alpha_n s[k - \tau_1 + f_n(\tau_{12})] + b_n(k) . \tag{51.32}$$

In the absence of noise, since it can be easily checked that

$$x_n[k + f_N(\tau_{12}) - f_n(\tau_{12})] = \alpha_n s(k - \tau_N) , \quad \forall n = 1, 2, \cdots , N , \tag{51.33}$$

we know that $x_1[k + f_N(\tau_{12})]$ is aligned with $x_n[k + f_N(\tau_{12}) - f_n(\tau_{12})]$. Therefore, a forward spatial prediction error signal can be defined as:

$$e_1(k, \tau) \triangleq x_1[k + f_N(\tau)] - \boldsymbol{x}_{\mathrm{a},2:N}^{\mathrm{T}}(k, \tau)\boldsymbol{a}_{2:N}(\tau) , \tag{51.34}$$

where τ is a dummy variable for the hypothesized TDOA τ_{12},

$$\boldsymbol{x}_{\mathrm{a},2:N}(k, \tau) = \Big[x_2[k + f_N(\tau) - f_2(\tau)] \ \cdots \ x_N(k) \Big]^{\mathrm{T}} \tag{51.35}$$

is the aligned (subscript 'a') signal vector, and

$$\boldsymbol{a}_{2:N}(\tau) = \Big[a_2(\tau) \ a_3(\tau) \ \cdots \ a_N(\tau) \Big]^{\mathrm{T}}$$

contains the forward spatial linear prediction coefficients. Minimizing the mean-square value of the prediction error signal

$$J_1(\tau) \triangleq E\left\{ \mathrm{e}_1^2(k, \tau) \right\} \tag{51.36}$$

leads to the Wiener–Hopf equation

$$\boldsymbol{R}_{\mathrm{a},2:N}(\tau)\boldsymbol{a}_{2:N}(\tau) = \boldsymbol{r}_{\mathrm{a},2:N}(\tau) , \tag{51.37}$$

where

$$\boldsymbol{R}_{\mathrm{a},2:N}(\tau) \triangleq E\left\{ \boldsymbol{x}_{\mathrm{a},2:N}(k, \tau)\boldsymbol{x}_{\mathrm{a},2:N}^{\mathrm{T}}(k, \tau) \right\} = \begin{pmatrix} \sigma_{x_2}^2 & r_{\mathrm{a},x_2x_3}(\tau) & \cdots & r_{\mathrm{a},x_2x_N}(\tau) \\ r_{\mathrm{a},x_3x_2}(\tau) & \sigma_{x_3}^2 & \cdots & r_{\mathrm{a},x_3x_N}(\tau) \\ \vdots & \vdots & \ddots & \vdots \\ r_{\mathrm{a},x_Nx_2}(\tau) & r_{\mathrm{a},x_Nx_3}(\tau) & \cdots & \sigma_{x_N}^2 \end{pmatrix}$$

is the spatial correlation matrix of the aligned signals with

$$\sigma_{x_n}^2 = E\left\{ x_n^2(k) \right\} , \quad n = 1, 2, \cdots , N ,$$

$$\begin{aligned} r_{\mathrm{a},x_ix_j}(\tau) &= E\big\{ x_i[k + f_N(\tau) - f_i(\tau)] \\ &\quad \times x_j[k + f_N(\tau) - f_j(\tau)] \big\} \\ &= E\big\{ x_i[k - f_i(\tau)] x_j[k - f_j(\tau)] \big\} , \\ i, j &= 1, 2, \cdots , N , \end{aligned}$$

and

$$\boldsymbol{r}_{\mathrm{a},2:N}(\tau) \triangleq \Big[r_{\mathrm{a},x_1x_2}(\tau) \ r_{\mathrm{a},x_1x_3}(\tau) \ \cdots \ r_{\mathrm{a},x_1x_N}(\tau) \Big]^{\mathrm{T}} .$$

Substituting the solution of the Wiener–Hopf (51.37), which is

$$\boldsymbol{a}_{2:N}(\tau) = \boldsymbol{R}_{\mathrm{a},2:N}^{-1}(\tau)\boldsymbol{r}_{\mathrm{a},2:N}(\tau) ,$$

into (51.34) gives the minimum forward prediction error

$$\begin{aligned} &e_{1,\min}(k, \tau) \\ &= x_1[k + f_N(\tau)] - \boldsymbol{x}_{\mathrm{a},2:N}^{\mathrm{T}}(k, \tau)\boldsymbol{R}_{\mathrm{a},2:N}^{-1}(\tau)\boldsymbol{r}_{\mathrm{a},2:N}(\tau) . \end{aligned} \tag{51.38}$$

Accordingly, we have

$$\begin{aligned} J_{1,\min}(\tau) &\triangleq E\left\{ e_{1,\min}^2(k, \tau) \right\} \\ &= \sigma_{x_1}^2 - \boldsymbol{r}_{\mathrm{a},2:N}^{\mathrm{T}}(\tau)\boldsymbol{R}_{\mathrm{a},2:N}^{-1}(\tau)\boldsymbol{r}_{\mathrm{a},2:N}(\tau) . \end{aligned} \tag{51.39}$$

Then we can argue that the time lag τ that induces a minimum $J_{1,\min}(\tau)$ would be the TDOA between the first two microphones:

$$\hat{\tau}_{12}^{\mathrm{FSP}} = \arg\min_{\tau} J_{1,\min}(\tau) , \tag{51.40}$$

where the superscript 'FSP' stands for forward spatial predication .

Similarly, the multichannel TDE algorithm can be developed using backward prediction or interpolation with any one of the N microphone outputs being regarded as the reference signal.

51.2.6 Multichannel Cross-Correlation Coefficient Algorithm

The idea of using the spatial prediction (or interpolation) error to measure the correlation among multiple signals is intuitive and helpful. However in statistics and signal processing, the cross-correlation coefficient is a more widely used concept for this purpose. Since the cross-correlation coefficient is defined traditionally only for

two random processes, it first needs to be generalized to multiple random processes before being employed in the development of a multichannel TDE algorithm.

The definition of a multiple coherence function, derived from the concepts of the ordinary coherence function between two signals and the partial (conditioned) coherence function, was presented in [51.40] to measure the correlation among the output of a multiple-input signal-output (MISO) system and its inputs. In [51.38] and [51.39], the multichannel cross-correlation coefficient (MCCC) was constructed in a new way from the multichannel correlation matrix; this provides a seamless generalization of the classical cross-correlation coefficient to the case where there are multiple random processes.

Following (51.35), we denote

$$\boldsymbol{x}_{\mathrm{a}}(k,\tau) \triangleq \left[x_1[k+f_N(\tau)] \;\; \boldsymbol{x}_{\mathrm{a},2:N}^{\mathrm{T}}(k,\tau)\right]^{\mathrm{T}} , \quad (51.41)$$

and

$$\boldsymbol{R}_{\mathrm{a}}(\tau) \triangleq E\left\{\boldsymbol{x}_{\mathrm{a}}(k,\tau)\boldsymbol{x}_{\mathrm{a}}^{\mathrm{T}}(k,\tau)\right\} = \begin{pmatrix} \sigma_{x_1}^2 & r_{\mathrm{a},x_1x_2}(\tau) & \cdots & r_{\mathrm{a},x_1x_N}(\tau) \\ r_{\mathrm{a},x_2x_1}(\tau) & \sigma_{x_2}^2 & \cdots & r_{\mathrm{a},x_2x_N}(\tau) \\ \vdots & \vdots & \ddots & \vdots \\ r_{\mathrm{a},x_Nx_1}(\tau) & r_{\mathrm{a},x_Nx_2}(\tau) & \cdots & \sigma_{x_N}^2 \end{pmatrix} . \quad (51.42)$$

The spatial correlation matrix $\boldsymbol{R}_{\mathrm{a}}(\tau)$ can be factored as follows:

$$\boldsymbol{R}_{\mathrm{a}}(\tau) = \boldsymbol{\Sigma}\tilde{\boldsymbol{R}}_{\mathrm{a}}(\tau)\boldsymbol{\Sigma}, \quad (51.43)$$

where

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_{x_1} & 0 & \cdots & 0 \\ 0 & \sigma_{x_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \sigma_{x_N} \end{pmatrix}$$

is a diagonal matrix,

$$\tilde{\boldsymbol{R}}_{\mathrm{a}}(\tau) = \begin{pmatrix} 1 & \rho_{\mathrm{a},x_1x_2}(\tau) & \cdots & \rho_{\mathrm{a},x_1x_N}(\tau) \\ \rho_{\mathrm{a},x_2x_1}(\tau) & 1 & \cdots & \rho_{\mathrm{a},x_2x_N}(\tau) \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{\mathrm{a},x_Nx_1}(\tau) & \rho_{\mathrm{a},x_Nx_2}(\tau) & \cdots & 1 \end{pmatrix}$$

is a symmetric matrix, and

$$\rho_{\mathrm{a},x_ix_j}(\tau) \triangleq \frac{r_{\mathrm{a},x_ix_j}(\tau)}{\sigma_i\sigma_j}, \quad i,j = 1,2,\cdots,N ,$$

is the cross-correlation coefficient between the i-th and j-th aligned microphone signals.

Since matrix $\tilde{\boldsymbol{R}}_{\mathrm{a}}(\tau)$ is symmetric, positively semidefinite, and its diagonal elements are all equal to one, it was shown in [51.38] and [51.39] that

$$0 \le \det\left[\tilde{\boldsymbol{R}}_{\mathrm{a}}(\tau)\right] \le 1 , \quad (51.44)$$

where $\det(\cdot)$ stands for determinant.

In the two-channel case, it can easily be checked that the squared cross-correlation coefficient is linked to the normalized spatial correlation matrix by

$$\rho_{\mathrm{a},x_1x_2}^2(\tau) = 1 - \det\left[\tilde{\boldsymbol{R}}_{\mathrm{a},1:2}(\tau)\right] . \quad (51.45)$$

Then by analogy the squared MCCC among the N aligned signals $x_n[k+f_N(\tau)-f_n(\tau)]$, $n=1,2,\cdots,N$, is constructed as:

$$\rho_{\mathrm{a},x_1:x_N}^2(\tau) \triangleq 1 - \det\left[\tilde{\boldsymbol{R}}_{\mathrm{a}}(\tau)\right] = 1 - \frac{\det\left[\boldsymbol{R}_{\mathrm{a}}(\tau)\right]}{\prod_{n=1}^{N}\sigma_{x_n}^2} . \quad (51.46)$$

The MCCC has the following properties (presented without proof) [51.41].

1. $0 \le \rho_{\mathrm{a},x_1:x_N}^2(\tau) \le 1$.
2. If two or more signals are perfectly correlated, then $\rho_{\mathrm{a},x_1:x_N}^2(\tau) = 1$.
3. If all the signals are completely uncorrelated with each other, then $\rho_{\mathrm{a},x_1:x_N}^2(\tau) = 0$.
4. If one of the signals is completely uncorrelated with the $N-1$ other signals, then the MCCC will measure the correlation among those $N-1$ remaining signals.

Using the definition of MCCC, we deduce an estimate of the TDOA between the first two microphone signals as

$$\hat{\tau}_{12}^{\mathrm{MCCC}} = \arg\max_{\tau} \rho_{\mathrm{a},x_1:x_N}^2(\tau) , \quad (51.47)$$

which is equivalent to computing

$$\hat{\tau}_{12}^{\mathrm{MCCC}} = \arg\max_{\tau}\left\{1 - \frac{\det\left[\boldsymbol{R}_{\mathrm{a}}(\tau)\right]}{\prod_{n=1}^{N}\sigma_{x_n}^2}\right\} = \arg\min_{\tau}\det\left[\boldsymbol{R}_{\mathrm{a}}(\tau)\right] . \quad (51.48)$$

To investigate the link of the MCCC method to the FSP method, let us revisit the spatial prediction error function given by (51.39). We define

$$\boldsymbol{a}(\tau) \triangleq \left[a_1(\tau) \;\; a_2(\tau) \;\; \cdots \;\; a_N(\tau)\right]^{\mathrm{T}} = \left[a_1(\tau) \;\; \boldsymbol{a}_{2:N}^{\mathrm{T}}(\tau)\right]^{\mathrm{T}} . \quad (51.49)$$

Then, for $a_1(\tau) = -1$, the forward spatial prediction error signal (51.34) can be written as

$$e_1(k,\tau) = -\boldsymbol{x}_{\mathrm{a}}^{\mathrm{T}}(k,\tau)\boldsymbol{a}(\tau)\,, \tag{51.50}$$

and (51.36) can be expressed as

$$\begin{aligned} J_1(\tau) &= E\left\{\mathrm{e}_1^2(k,\tau)\right\} + \kappa\left[\boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{a}(\tau)+1\right] \\ &= \boldsymbol{a}^{\mathrm{T}}(\tau)\boldsymbol{R}_{\mathrm{a}}(\tau)\boldsymbol{a}(\tau) + \kappa\left[\boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{a}(\tau)+1\right]\,, \end{aligned} \tag{51.51}$$

where κ is a Lagrange multiplier introduced to force $a_1(\tau)$ to have value -1 and

$$\boldsymbol{v}_1 \triangleq \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^{\mathrm{T}}\,.$$

Taking the derivative of (51.51) with respect to $\boldsymbol{a}(\tau)$ and setting the result to zero yields

$$\frac{\partial J_1(\tau)}{\partial \boldsymbol{a}(\tau)} = 2\boldsymbol{R}_{\mathrm{a}}(\tau)\boldsymbol{a}(\tau) + k\boldsymbol{v}_1 = \boldsymbol{0}\,. \tag{51.52}$$

Solving (51.52) for $\boldsymbol{a}(\tau)$ produces

$$\boldsymbol{a}(\tau) = -\frac{\kappa\boldsymbol{R}_{\mathrm{a}}^{-1}(\tau)\boldsymbol{v}_1}{2}\,. \tag{51.53}$$

Substituting (51.53) into (51.51) leads to

$$J_1(\tau) = \kappa\left(1 - \frac{\boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{R}_{\mathrm{a}}^{-1}(\tau)\boldsymbol{v}_1}{4}\kappa\right)\,, \tag{51.54}$$

from which we know that

$$J_{1,\min}(\tau) = \frac{1}{\boldsymbol{v}_1^{\mathrm{T}}\boldsymbol{R}_{\mathrm{a}}^{-1}(\tau)\boldsymbol{v}_1}\,. \tag{51.55}$$

Substituting (51.43) into (51.55) and using the fact that

$$\boldsymbol{\Sigma}^{-1}\boldsymbol{v}_1 = \frac{\boldsymbol{v}_1}{\sigma_{x_1}}\,,$$

we have

$$J_{1,\min}(\tau) = \frac{\sigma_{x_1}^2}{\boldsymbol{v}_1^{\mathrm{T}}\tilde{\boldsymbol{R}}_{\mathrm{a}}^{-1}(\tau)\boldsymbol{v}_1}\,. \tag{51.56}$$

Note that $\boldsymbol{v}_1^{\mathrm{T}}\tilde{\boldsymbol{R}}_{\mathrm{a}}^{-1}(\tau)\boldsymbol{v}_1$ is the (1, 1)-th element of matrix $\tilde{\boldsymbol{R}}_{\mathrm{a}}^{-1}(\tau)$, which is computed using the adjoint method as the (1, 1)-th cofactor of $\tilde{\boldsymbol{R}}_{\mathrm{a}}(\tau)$ divided by the determinant of $\tilde{\boldsymbol{R}}_{\mathrm{a}}(\tau)$, i. e.,

$$\boldsymbol{v}_1^{\mathrm{T}}\tilde{\boldsymbol{R}}_{\mathrm{a}}^{-1}(\tau)\boldsymbol{v}_1 = \frac{\det\left[\tilde{\boldsymbol{R}}_{\mathrm{a,2:N}}(\tau)\right]}{\det\left[\tilde{\boldsymbol{R}}_{\mathrm{a}}(\tau)\right]}\,, \tag{51.57}$$

where $\tilde{\boldsymbol{R}}_{\mathrm{a,2:N}}(\tau)$ is the lower-right submatrix of $\tilde{\boldsymbol{R}}_{\mathrm{a}}(\tau)$ formed by removing the first row and the first column. By substituting (51.57) into (51.56), we get

$$J_{1,\min}(\tau) = \sigma_{x_1}^2 \cdot \frac{\det\left[\tilde{\boldsymbol{R}}_{\mathrm{a}}(\tau)\right]}{\det\left[\tilde{\boldsymbol{R}}_{\mathrm{a,2:N}}(\tau)\right]}\,. \tag{51.58}$$

Therefore, the FSP estimate of τ_{12} is found as

$$\begin{aligned} \hat{\tau}_{12}^{\mathrm{FSP}} &= \arg\min_{\tau} J_{1,\min}(\tau) \\ &= \arg\min_{\tau} \frac{\det\left[\tilde{\boldsymbol{R}}_{\mathrm{a}}(\tau)\right]}{\det\left[\tilde{\boldsymbol{R}}_{\mathrm{a,2:N}}(\tau)\right]}\,. \end{aligned} \tag{51.59}$$

Comparing (51.48) to (51.59) reveals a clear distinction between the two methods in spite of the strong similarity in principle. In practice, the FSP method has a numerical stability problem since the calculation of the FSP cost function (51.58) involves division by $\det\left[\tilde{\boldsymbol{R}}_{\mathrm{a,2:}N}(\tau)\right]$, while the MCCC method is found to be fairly stable.

It is worth pointing out that the microphone outputs can be pre-whitened before computing their MCCC as was done in the SCOT algorithm in the two-channel scenario. Using this approach, the TDE algorithms become more robust to volume variations of the speech source signal.

51.2.7 Minimum-Entropy Method

While the MCCC-based TDE algorithm performs well in the presence of noise and reverberation, the MCCC is by no means the only choice for developing the concept of multichannel TDE. MCCC is a second-order-statistics (SOS) measure of dependence among multiple random variables and is ideal for Gaussian source signals. However for non-Gaussian source signals such as speech, MCCC is not sufficient and higher-order statistics (HOS) have more to say about their dependence.

The concept of entropy, which is a statistical (apparently HOS) measure of randomness or uncertainty of a random variable, was introduced by *Shannon* in the context of communication theory [51.42]. For a random variable x with a probability density function (PDF) $p(x)$ (in this chapter, we choose not to distinguish between random variables and their realizations for the conciseness of presentation), the entropy is defined as [51.43]:

$$\begin{aligned} H(x) &= -\int p(x)\ln p(x)\,\mathrm{d}x \\ &= -E\left\{\ln p(x)\right\}\,. \end{aligned} \tag{51.60}$$

The entropy (in the continuous case) is a measure of the structure contained in the PDF [51.44]. As far as the

multivariate random variable $\boldsymbol{x}_a(k,\tau)$ given by (51.41) is concerned, the joint entropy is:

$$H\left[\boldsymbol{x}_a(k,\tau)\right] = -\int p\left[\boldsymbol{x}_a(k,\tau)\right] \ln p\left[\boldsymbol{x}_a(k,\tau)\right] \mathrm{d}\boldsymbol{x}_a(k,\tau) \,. \tag{51.61}$$

It was then argued in [51.45] that the time lag τ that gives the minimum of $H\left[\boldsymbol{x}_a(k,\tau)\right]$ corresponds to the TDOA between the two microphones:

$$\hat{\tau}_{12}^{\mathrm{ME}} = \arg\min_{\tau} H\left[\boldsymbol{x}_a(k,\tau)\right] , \tag{51.62}$$

where the superscript 'ME' refers to the minimum-entropy method.

Gaussian Source Signal

If the source is Gaussian, so are the microphone outputs in the absence of noise. Suppose that the aligned microphone signals are zero-mean and joint Gaussian random signals. Their joint PDF is then given by

$$p\left[\boldsymbol{x}_a(k,\tau)\right] = \frac{\exp\left[-\eta_a(k,\tau)/2\right]}{\sqrt{(2\pi)^N \det\left[\boldsymbol{R}_a(\tau)\right]}} , \tag{51.63}$$

where

$$\eta_a(k,\tau) \triangleq \boldsymbol{x}_a^{\mathrm{T}}(k,\tau)\boldsymbol{R}_a^{-1}(\tau)\boldsymbol{x}_a(k,\tau) \,. \tag{51.64}$$

By substituting (51.63) into (51.61), the joint entropy can be computed [51.45] as:

$$H\left[\boldsymbol{x}_a(k,\tau)\right] = \frac{1}{2}\ln\left\{(2\pi\,\mathrm{e})^N \det\left[\boldsymbol{R}_a(\tau)\right]\right\} \,. \tag{51.65}$$

Consequently, (51.62) becomes

$$\hat{\tau}_{12}^{\mathrm{ME}} = \arg\min_{\tau} \det\left[\boldsymbol{R}_a(\tau)\right] \,. \tag{51.66}$$

It is clear from (51.48) and (51.66) that minimizing the entropy is equivalent to maximizing the MCCC for Gaussian source signals.

Speech Source Signal

Speech is a complicated random process and there is no rigorous mathematical formula for its entropy. However, in speech research it was found that speech can be fairly well modeled by a Laplace distribution [51.46, 47].

The univariate Laplace distribution with zero mean and variance σ_x^2 is given by

$$p(x) = \frac{\sqrt{2}}{2\sigma_x}\,\mathrm{e}^{-\sqrt{2}|x|/\sigma_x} , \tag{51.67}$$

and the corresponding entropy is [51.43],

$$H(x) = 1 + \ln\left(\sqrt{2}\,\sigma_x\right) . \tag{51.68}$$

Suppose that $\boldsymbol{x}_a(k,\tau)$ has a multivariate Laplace distribution with mean $\boldsymbol{0}$ and covariance matrix $\boldsymbol{R}_a(\tau)$ [51.48], [51.49]:

$$p\left[\boldsymbol{x}_a(k,\tau)\right] = \frac{2\left[\eta_a(k,\tau)/2\right]^{P/2} K_P\left[\sqrt{2\eta_a(k,\tau)}\right]}{\sqrt{(2\pi)^N \det\left[\boldsymbol{R}_a(\tau)\right]}} , \tag{51.69}$$

where $P = (2-N)/2$ and $K_P(\cdot)$ is the modified Bessel function of the third kind (also called the modified Bessel function of the second kind) given by

$$K_P(a) = \frac{1}{2}\left(\frac{a}{2}\right)^P \int_0^\infty z^{-P-1}\exp\left(-z-\frac{a^2}{4z}\right)\mathrm{d}z \,, \quad a>0 \,. \tag{51.70}$$

The joint entropy is

$$H\left[\boldsymbol{x}_a(k,\tau)\right] = \frac{1}{2}\ln\left(\frac{(2\pi)^N \det\left[\boldsymbol{R}_a(\tau)\right]}{4}\right) - \frac{P}{2}E\left\{\ln\left(\frac{\eta_a(k,\tau)}{2}\right)\right\} - E\left\{\ln K_P\left[\sqrt{2\eta_a(k,\tau)}\right]\right\} \,. \tag{51.71}$$

The two quantities $E\left\{\ln\left[\eta_a(k,\tau)/2\right]\right\}$ and $E\left\{\ln K_P\left[\sqrt{2\eta_a(k,\tau)}\right]\right\}$ do not seem to have a closed form. So a numerical scheme needs to be developed to estimate them. One possibility to do this is the following. Assume that all processes are ergodic . As a result, ensemble averages can be replaced by time averages. If there are K samples for each element of the observation vector $\boldsymbol{x}_a(k,\tau)$, the following estimators were proposed in [51.45]:

$$E\left\{\ln\left[\eta_a(k,\tau)/2\right]\right\} \approx \frac{1}{K}\sum_{k=0}^{K-1}\ln\left[\eta_a(k,\tau)/2\right] , \tag{51.72}$$

$$E\left\{\ln K_P\left[\sqrt{2\eta_a(k,\tau)}\right]\right\} \approx \frac{1}{K}\sum_{k=0}^{K-1}\ln K_P\left[\sqrt{2\eta_a(k,\tau)}\right] \,. \tag{51.73}$$

The simulation results presented in [51.45] show that the ME algorithm performs in general comparably to or better than the MCCC algorithm. Apparently the ME algorithm is computationally intensive, but the idea of using entropy expands our horizon of knowledge in pursuit of new TDE algorithms.

Amazon EX1010 Page 1050

51.3 Source Localization

51.3.1 Problem Formulation

The problem addressed here is the estimation of the location of an acoustic point source given the array geometry and the relative TDOA measurements between different microphone pairs. The problem is stated mathematically as follows.

Consider an array that consists of N microphones. As shown in Fig. 51.2, these microphones are located in a 3-D Cartesian coordinate system at positions

$$\boldsymbol{\gamma}_n \triangleq \begin{bmatrix} x_n & y_n & z_n \end{bmatrix}^{\mathrm{T}}, \quad n = 1, 2, \cdots, N . \tag{51.74}$$

Without loss of generality, the first microphone ($n = 1$) is regarded as the reference and is placed at the origin of the coordinate system, i. e., $\boldsymbol{\gamma}_1 = [0\ 0\ 0]^{\mathrm{T}}$. The acoustic source is located at $\boldsymbol{\gamma}_{\mathrm{s}} \triangleq [x_{\mathrm{s}}\ y_{\mathrm{s}}\ z_{\mathrm{s}}]^{\mathrm{T}}$. The distances from the origin to the n-th microphone and the source are denoted by Ω_n and Ω_{s}, respectively, where

$$\Omega_n \triangleq \|\boldsymbol{\gamma}_n\| = \sqrt{x_n^2 + y_n^2 + z_n^2}, \quad n = 1, 2, \cdots, N, \tag{51.75}$$

$$\Omega_{\mathrm{s}} \triangleq \|\boldsymbol{\gamma}_{\mathrm{s}}\| = \sqrt{x_{\mathrm{s}}^2 + y_{\mathrm{s}}^2 + z_{\mathrm{s}}^2} . \tag{51.76}$$

The distance between the source and the n-th microphone is denoted by

$$D_n \triangleq \|\boldsymbol{\gamma}_n - \boldsymbol{\gamma}_{\mathrm{s}}\| = \sqrt{(x_n - x_{\mathrm{s}})^2 + (y_n - y_{\mathrm{s}})^2 + (z_n - z_{\mathrm{s}})^2} . \tag{51.77}$$

The difference in the distances of microphones i and j from the source is given by

$$d_{ij} \triangleq D_i - D_j, \quad i, j = 1, 2, \cdots, N . \tag{51.78}$$

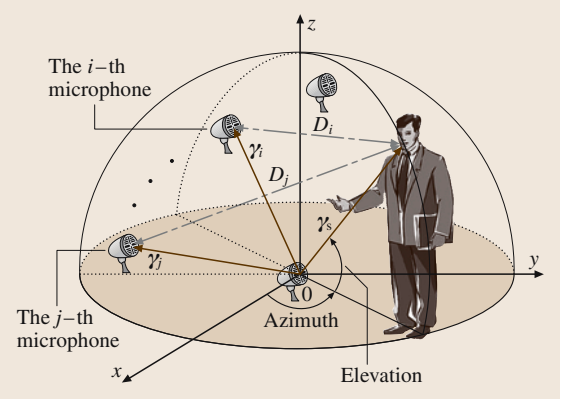


Fig. 51.2 Spatial diagram illustrating variables defined in the source localization problem

This difference is usually termed the range difference; it is proportional to the time difference of arrival τ_{ij}. If the speed of sound is c, then

$$d_{ij} = c \cdot \tau_{ij} . \tag{51.79}$$

The speed of sound (in m/s) can be estimated from the air temperature t_{air} (in degrees Celsius) according to the following approximate (first-order) formula,

$$c \approx 331 + 0.610\, t_{\mathrm{air}} . \tag{51.80}$$

The localization problem is then to estimate $\boldsymbol{\gamma}_{\mathrm{s}}$ given the set of $\boldsymbol{\gamma}_n$ and τ_{ij}. Note that there are $N(N-1)/2$ distinct TDOA estimates τ_{ij}, which exclude the case $i = j$ and counts the $\tau_{ij} = -\tau_{ji}$ pair only once. However, in the absence of noise, the space spanned by these TDOA estimates is $(N-1)$-dimensional. Any $N-1$ linearly independent TDOAs determine all of the others. In a noisy environment, the TDOA redundancy can be used to improve the accuracy of the source localization algorithms, but this would increase their computational complexity. For simplicity and also without loss of generality, we choose $\tau_{n1}, n = 2, \cdots, N$ as the basis for this $\mathbb{R}^{N-1}$ space in this Chapter.

51.3.2 Measurement Model and Cramèr–Rao Lower Bound

When the source localization problem is examined using estimation theory, the measurements of the range differences are modeled by

$$d_{n1} = g_n(\boldsymbol{\gamma}_{\mathrm{s}}) + \epsilon_n, \quad n = 2, \cdots, N , \tag{51.81}$$

where

$$g_n(\boldsymbol{\gamma}_{\mathrm{s}}) = \|\boldsymbol{\gamma}_n - \boldsymbol{\gamma}_{\mathrm{s}}\| - \|\boldsymbol{\gamma}_{\mathrm{s}}\| ,$$

and the ϵ_n are measurement errors. In vector form, such an additive measurement error model becomes,

$$\boldsymbol{d} = \boldsymbol{g}(\boldsymbol{\gamma}_{\mathrm{s}}) + \boldsymbol{\epsilon} , \tag{51.82}$$

where

$$\boldsymbol{d} = \begin{bmatrix} d_{21} & d_{31} & \cdots & d_{N1} \end{bmatrix}^{\mathrm{T}} ,$$

$$\boldsymbol{g}(\boldsymbol{\gamma}_{\mathrm{s}}) = \begin{bmatrix} g_2(\boldsymbol{\gamma}_{\mathrm{s}}) & g_3(\boldsymbol{\gamma}_{\mathrm{s}}) & \cdots & g_N(\boldsymbol{\gamma}_{\mathrm{s}}) \end{bmatrix}^{\mathrm{T}} ,$$

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_2 & \epsilon_3 & \cdots & \epsilon_N \end{bmatrix}^{\mathrm{T}} .$$

Furthermore, it was postulated that the additive measurement errors have zero mean and are independent of

the range difference observation, as well as the source location $\boldsymbol{\gamma}_s$ [51.50]. For a continuous-time estimator, the corrupting noise, as indicated in [51.16], is jointly Gaussian distributed. The PDF of $\boldsymbol{d}$ conditioned on $\boldsymbol{\gamma}_s$ is subsequently given by,

$$p\left(\boldsymbol{d}|\boldsymbol{\gamma}_s\right) = \frac{\exp\left\{-\frac{1}{2}\left[\boldsymbol{d}-\boldsymbol{g}(\boldsymbol{\gamma}_s)\right]^T \boldsymbol{R}_{\epsilon\epsilon}^{-1}\left[\boldsymbol{d}-\boldsymbol{g}(\boldsymbol{\gamma}_s)\right]\right\}}{\sqrt{(2\pi)^N \det(\boldsymbol{R}_{\epsilon\epsilon})}}, \tag{51.83}$$

where $\boldsymbol{R}_{\epsilon\epsilon} \triangleq E\left\{\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T\right\}$ is the covariance matrix of $\boldsymbol{\epsilon}$.

Note that $\boldsymbol{R}_{\epsilon\epsilon}$ is independent of $\boldsymbol{\gamma}_s$ by assumption. Since digital equipment is used to sample the microphone waveforms and estimate the TDOAs, the error introduced by discrete-time processing also has to be taken into account. When this is done, the measurement error is no longer Gaussian and is more-properly modeled as a mixture of Gaussian noise and noise that is uniformly distributed over $[-T_s c/2, T_s c/2]$, where T_s is the sampling period. As an example, for a digital source location estimator with an 8 KHz sampling rate operating at room temperature (25 °C, implying that $c \approx 346.25$ m/s), the maximum error in range difference estimates due to sampling is about ± 2.164 cm, which leads to considerable errors in the location estimate, especially when the source is far from the microphone array.

Under the measurement model given by (51.82), we are now faced with the parameter estimation problem of extracting the source location information from the mismeasured range differences or the equivalent TDOAs. For an *unbiased* estimator, a Cramèr–Rao lower bound (CRLB) can be placed on the variance of each estimated coordinate of the source location. However, since the range difference function $\boldsymbol{g}(\boldsymbol{\gamma}_s)$ in the measurement model is nonlinear in the parameters under estimation, it is very difficult (or even impossible) to find an unbiased estimator that is mathematically simple and attains the CRLB. The CRLB is usually used as a benchmark against which the statistical efficiency of any unbiased estimators can be compared.

In general, without any assumptions made about the PDF of the measurement error $\boldsymbol{\epsilon}$, the CRLB of the i-th ($i = 1, 2, 3$) parameter variance is found as the $[i, i]$ element of the inverse of the Fisher information matrix defined by [51.51]:

$$[\boldsymbol{I}(\boldsymbol{\gamma}_s)]_{ij} \triangleq -E\left(\frac{\partial^2 \ln p\left(\boldsymbol{d}|\boldsymbol{\gamma}_s\right)}{\partial\gamma_{s,i}\partial\gamma_{s,j}}\right), \quad i, j = 1, 2, 3, \tag{51.84}$$

where the three parameters of $\boldsymbol{\gamma}_s$, i. e., $\gamma_{s,1}$, $\gamma_{s,2}$, and $\gamma_{s,3}$, are the x, y, and z coordinates of the source location, respectively.

In the case of a Gaussian measurement error, the Fisher information matrix becomes [51.52]:

$$\boldsymbol{I}(\boldsymbol{\gamma}_s) = \left(\frac{\partial \boldsymbol{g}(\boldsymbol{\gamma}_s)}{\partial \boldsymbol{\gamma}_s}\right)^T \boldsymbol{R}_{\epsilon\epsilon}^{-1}\left(\frac{\partial \boldsymbol{g}(\boldsymbol{\gamma}_s)}{\partial \boldsymbol{\gamma}_s}\right), \tag{51.85}$$

where $\partial \boldsymbol{g}(\boldsymbol{\gamma}_s)/\partial \boldsymbol{\gamma}_s$ is an $(N-1)\times 3$ Jacobian matrix defined as,

$$\frac{\partial \boldsymbol{g}(\boldsymbol{\gamma}_s)}{\partial \boldsymbol{\gamma}_s} = \begin{pmatrix} \frac{\partial g_2(\boldsymbol{\gamma}_s)}{\partial x_s} & \frac{\partial g_2(\boldsymbol{\gamma}_s)}{\partial y_s} & \frac{\partial g_2(\boldsymbol{\gamma}_s)}{\partial z_s} \\ \frac{\partial g_3(\boldsymbol{\gamma}_s)}{\partial x_s} & \frac{\partial g_3(\boldsymbol{\gamma}_s)}{\partial y_s} & \frac{\partial g_3(\boldsymbol{\gamma}_s)}{\partial z_s} \\ \vdots & \vdots & \vdots \\ \frac{\partial g_N(\boldsymbol{\gamma}_s)}{\partial x_s} & \frac{\partial g_N(\boldsymbol{\gamma}_s)}{\partial y_s} & \frac{\partial g_N(\boldsymbol{\gamma}_s)}{\partial z_s} \end{pmatrix} = \begin{pmatrix} (\boldsymbol{u}_{2\to s} - \boldsymbol{u}_{1\to s})^T \\ (\boldsymbol{u}_{3\to s} - \boldsymbol{u}_{1\to s})^T \\ \vdots \\ (\boldsymbol{u}_{N\to s} - \boldsymbol{u}_{1\to s})^T \end{pmatrix}, \tag{51.86}$$

and

$$\boldsymbol{u}_{n\to s} \triangleq \frac{\boldsymbol{\gamma}_s - \boldsymbol{\gamma}_n}{\|\boldsymbol{\gamma}_s - \boldsymbol{\gamma}_n\|} = \frac{\boldsymbol{\gamma}_s - \boldsymbol{\gamma}_n}{D_n}, \quad n = 1, 2, \cdots, N, \tag{51.87}$$

is the normalized vector of unit length pointing from the n-th microphone to the sound source.

51.3.3 Maximum-Likelihood Estimator

The measurement model for the source localization problem is apparently nonlinear; an efficient estimator that attains the CRLB may not exist or otherwise might be impossible to identify. In practice, the maximum-likelihood estimator is often used since it has the well-proven advantage of asymptotic efficiency for a large sample space.

To apply the maximum-likelihood principle, the statistical characteristics of the measurements need to be known or properly assumed prior to any processing. From the central limit theorem and also for mathematical simplicity, the measurement error is usually modeled as Gaussian and the likelihood function is given by (51.83), which is considered as a function of the source position $\boldsymbol{\gamma}_s$ under estimation.

Since the exponential function is monotonically increasing, the MLE is equivalent to minimizing a (log-

likelihood) cost function defined as,

$$\mathcal{L}(\boldsymbol{\gamma}_s) \triangleq \left[\boldsymbol{d} - \boldsymbol{g}(\boldsymbol{\gamma}_s)\right]^T \boldsymbol{R}_{\epsilon\epsilon}^{-1} \left[\boldsymbol{d} - \boldsymbol{g}(\boldsymbol{\gamma}_s)\right] . \quad (51.88)$$

Direct estimation of the minimizer is generally not practical. If the noise signals at different microphones are assumed to be uncorrelated, the covariance matrix is diagonal

$$\boldsymbol{R}_{\epsilon\epsilon} = \text{diag}(\sigma_{\epsilon_2}^2, \sigma_{\epsilon_3}^2, \cdots, \sigma_{\epsilon_N}^2), \quad (51.89)$$

where $\sigma_{\epsilon_n}^2 = E\{\epsilon_n^2\}$ $(n = 2, 3, \cdots, N)$ is the variance of ϵ_n, and the cost function (51.88) becomes,

$$\mathcal{L}(\boldsymbol{\gamma}_s) = \sum_{n=2}^{N} \frac{\left[d_{n1} - g_n(\boldsymbol{\gamma}_s)\right]^2}{\sigma_{\epsilon_n}^2}. \quad (51.90)$$

Among other approaches, the steepest-descent algorithm can be used to find $\hat{\boldsymbol{\gamma}}_s^{\text{MLE}}$ iteratively with

$$\hat{\boldsymbol{\gamma}}_s(k+1) = \hat{\boldsymbol{\gamma}}_s(k) - \frac{1}{2}\mu \nabla \mathcal{L}[\hat{\boldsymbol{\gamma}}_s(k)] , \quad (51.91)$$

where μ is the step size.

The foregoing MLE can be determined and is asymptotically optimal for this problem only if its two assumptions (Gaussian and uncorrelated measurement noise) hold. However, this is not the case in practice, as discussed in Sect. 51.3. Furthermore, the number of microphones in an array for camera pointing or beamformer steering is always limited, which makes the source localization a finite-sample rather than a large-sample problem. In addition, the cost function (51.90) is generally not strictly convex. In order to avoid a local minimum with the steepest-descent algorithm, we need to select a good initial guesstimate of the source location, which is difficult to do in practice, and convergence of the iterative algorithm to the desired solution cannot be guaranteed.

51.3.4 Least-Squares Estimators

Two limitations of the MLE are that probabilistic assumptions have to be made about the measured range differences and that the iterative algorithm to find the solution is computationally intensive. An alternative method is the well-known least-squares estimator (LSE). The LSE makes no probabilistic assumptions about the data and hence can be applied to the source localization problem in which a precise statistical characterization of the data is hard to determine. Furthermore, an LSE usually produces a closed-form estimate, which is desirable in real-time applications.

51.3.5 Least-Squares Error Criteria

In the least-squares (LS) approach, we attempt to minimize a squared error function that is zero in the absence of noise and model inaccuracies. Various error functions can be defined for closeness from the assumed (noiseless) signal based on hypothesized parameters for the observed data. When these are applied, various LSEs can be derived. For the source localization problem two LS error criteria have been proposed.

Hyperbolic LS Error Function

The first LS error function is defined as the difference between the observed range difference and that generated by a signal model depending upon the unknown parameters:

$$\boldsymbol{e}_h(\boldsymbol{\gamma}_s) \triangleq \boldsymbol{d} - \boldsymbol{g}(\boldsymbol{\gamma}_s) , \quad (51.92)$$

and the corresponding LS criterion is given by

$$J_h = \boldsymbol{e}_h^T \boldsymbol{e}_h = [\boldsymbol{d} - \boldsymbol{g}(\boldsymbol{\gamma}_s)]^T [\boldsymbol{d} - \boldsymbol{g}(\boldsymbol{\gamma}_s)] . \quad (51.93)$$

In the source localization problem, an observed range difference d_{n1} defines a hyperboloid in 3-D space. All points lying on such a hyperboloid are potential source locations and all have the same range difference d_{n1} to the two microphones n and 1. Therefore, a sound source that is located by minimizing (51.93) has the shortest distance to all hyperboloids associated with different microphone pairs and specified by the estimated range differences, after which fact the error criterion (51.93) is termed the hyperbolic error criterion (subscript 'h').

In (51.92), the signal model $\boldsymbol{g}(\boldsymbol{\gamma}_s)$ consists of a set of hyperbolic functions. Since they are nonlinear, minimizing (51.93) leads to a mathematically intractable solution as N becomes large. Moreover, the hyperbolic function is very sensitive to noise, especially for far-field sources. As a result, it is rarely used in practice.

When the statistical characteristics of the corrupting noise are unknown, uncorrelated *white* Gaussian noise is one reasonable assumption. In this case, it is not surprising that the hyperbolic LSE and the MLE minimize (maximize) similar criteria.

Spherical LS Error Function

The second LS criterion is based on the errors found in the distances from a hypothesized source location to the microphones. In the absence of measurement errors, the correct source location is preferably at the intersection of a group of spheres centered at the microphones. When

measurement errors are present, the best estimate of the source location would be the point that yields the shortest distance to those spheres defined by the range differences and the hypothesized source range.

Consider the distance D_n from the n-th microphone to the source. From the definition of the range difference (51.78) and the fact that $D_1 = \Omega_s$, we have:

$$\hat{D}_n = \Omega_s + d_{n1} \,, \tag{51.94}$$

where $\hat{D}_n$ denotes an observation based on the measured range difference. From the inner product, we can derive the true value for D_n^2, the square of the noise-free distance generated by a spherical signal model:

$$D_n^2 = \|\boldsymbol{\gamma}_n - \boldsymbol{\gamma}_s\|^2 = \Omega_n^2 - 2\boldsymbol{\gamma}_n^T\boldsymbol{\gamma}_s + \Omega_s^2 \,. \tag{51.95}$$

The spherical (subscript 'sp') LS error function is then defined as the difference between the measured and hypothesized values

$$\begin{aligned} e_{sp,n}(\boldsymbol{\gamma}_s) &\triangleq \frac{1}{2}\left(\hat{D}_n^2 - D_n^2\right) \\ &= \boldsymbol{\gamma}_n^T\boldsymbol{\gamma}_s + d_{n1}\Omega_s - \frac{1}{2}(\Omega_n^2 - d_{n1}^2) \,, \\ n &= 2, 3, \cdots, N \,. \end{aligned} \tag{51.96}$$

Putting the $N-1$ errors together and writing them in vector form gives,

$$\boldsymbol{e}_{sp}(\boldsymbol{r}_s) = \boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{\xi} \,, \tag{51.97}$$

where

$$\boldsymbol{A} \triangleq \left[\boldsymbol{S} \,|\, \boldsymbol{d}\right], \quad \boldsymbol{S} \triangleq \begin{pmatrix} x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ & \vdots & \\ x_N & y_N & z_N \end{pmatrix},$$

$$\boldsymbol{\theta} \triangleq \begin{pmatrix} x_s \\ y_s \\ z_s \\ \Omega_s \end{pmatrix}, \quad \boldsymbol{\xi} \triangleq \frac{1}{2}\begin{pmatrix} \Omega_2^2 - d_{21}^2 \\ \Omega_3^2 - d_{21}^2 \\ \vdots \\ \Omega_N^2 - d_{21}^2 \end{pmatrix},$$

and $[\boldsymbol{S} \,|\, \boldsymbol{d}]$ indicates that $\boldsymbol{S}$ and $\boldsymbol{d}$ are stacked side by side. The corresponding LS criterion is then given by:

$$J_{sp} = \boldsymbol{e}_{sp}^T\boldsymbol{e}_{sp} = [\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{\xi}]^T[\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{\xi}] \,. \tag{51.98}$$

In contrast to the hyperbolic error function (51.92), the spherical error function (51.97) is linear in $\boldsymbol{\gamma}_s$ given Ω_s and vice versa. Therefore, the computational complexity to find a solution will *not* dramatically increase as N gets large.

51.3.6 Spherical Intersection (SX) Estimator

The SX source location estimator employs the spherical error and solves the problem in two steps [51.53]. It first finds the least-squares solution for $\boldsymbol{\gamma}_s$ in terms of Ω_s,

$$\boldsymbol{\gamma}_s = \boldsymbol{S}^\dagger(\boldsymbol{\xi} - \Omega_s\boldsymbol{d}) \,, \tag{51.99}$$

where $\boldsymbol{S}^\dagger \triangleq \left(\boldsymbol{S}^T\boldsymbol{S}\right)^{-1}\boldsymbol{S}^T$ is the pseudo-inverse of the matrix $\boldsymbol{S}$. Then, substituting (51.99) into the constraint $\Omega_s^2 = \boldsymbol{\gamma}_s^T\boldsymbol{\gamma}_s$ yields a quadratic equation as follows

$$\Omega_s^2 = \left[\boldsymbol{S}^\dagger\,(\boldsymbol{\xi} - \Omega_s\boldsymbol{d})\right]^T\left[\boldsymbol{S}^\dagger\,(\boldsymbol{\xi} - \Omega_s\boldsymbol{d})\right] \,. \tag{51.100}$$

After expansion, this becomes

$$\alpha_1\Omega_s^2 + \alpha_2\Omega_s + \alpha_3 = 0 \,, \tag{51.101}$$

where

$$\begin{aligned} &\alpha_1 = 1 - \|\boldsymbol{S}^\dagger\boldsymbol{d}\|^2, \quad \alpha_2 = 2\boldsymbol{\xi}^T\boldsymbol{S}^{\dagger T}\boldsymbol{S}^\dagger\boldsymbol{d} \,, \\ &\alpha_3 = -\|\boldsymbol{S}^\dagger\boldsymbol{\xi}\|^2 \,. \end{aligned}$$

The valid (real, positive) root is taken as an estimate of the source range Ω_s and is then substituted into (51.99) to calculate the SX estimate $\hat{\boldsymbol{\gamma}}_s^{SX}$ of the source location.

In the SX estimation procedure, the solution of the quadratic equation (51.101) for the source range Ω_s is required. This solution must be a positive value. If a real positive root is not available, the SX solution does not *exist*. On the contrary, if both of the roots are real and greater than 0, then the SX solution is not *unique*. In both cases, the SX source location estimator fails to produce a reliable estimate, which is not desirable for a real-time implementation.

51.3.7 Spherical Interpolation (SI) Estimator

In order to overcome this drawback of the SX algorithm, a spherical interpolation estimator was proposed in [51.54], which attempts to relax the restriction $\Omega_s = \|\boldsymbol{\gamma}_s\|$ by estimating Ω_s in the least-squares sense.

To begin, we substitute the least-squares solution (51.99) into the original spherical equation $\boldsymbol{A}\boldsymbol{\theta} = \boldsymbol{\xi}$ to obtain

$$\Omega_s\boldsymbol{P}_{\boldsymbol{S}^\perp}\boldsymbol{d} = \boldsymbol{P}_{\boldsymbol{S}^\perp}\boldsymbol{\xi} \,, \tag{51.102}$$

where

$$\boldsymbol{P}_{\boldsymbol{S}^\perp} \triangleq \boldsymbol{I}_{N\times N} - \boldsymbol{S}\boldsymbol{S}^\dagger \,, \tag{51.103}$$

and $\boldsymbol{I}_{N\times N}$ is the $N\times N$ identity matrix. The matrix $\boldsymbol{P}_{\boldsymbol{S}^\perp}$ is a projection matrix that projects a vector, when multiplied by the matrix, onto a space that is orthogonal

to the column space of $\boldsymbol{S}$. Such a projection matrix is symmetric (i. e., $\boldsymbol{P}_{\boldsymbol{S}^\perp} = \boldsymbol{P}_{\boldsymbol{S}^\perp}^{\mathrm{T}}$) and idempotent (i. e., $\boldsymbol{P}_{\boldsymbol{S}^\perp} = \boldsymbol{P}_{\boldsymbol{S}^\perp} \cdot \boldsymbol{P}_{\boldsymbol{S}^\perp}$). Then the least-squares solution to (51.102) is given by

$$\hat{\Omega}_{\mathrm{s}}^{\mathrm{SI}} = \frac{\boldsymbol{d}^{\mathrm{T}} \boldsymbol{P}_{\boldsymbol{S}^\perp} \boldsymbol{\xi}}{\boldsymbol{d}^{\mathrm{T}} \boldsymbol{P}_{\boldsymbol{S}^\perp} \boldsymbol{d}} \,. \tag{51.104}$$

Substituting this solution into (51.99) yields the SI estimate

$$\hat{\boldsymbol{\gamma}}_{\mathrm{s}}^{\mathrm{SI}} = \boldsymbol{S}^{\dagger} \left[\boldsymbol{I}_{N\times N} - \left(\frac{\boldsymbol{d}\boldsymbol{d}^{\mathrm{T}} \boldsymbol{P}_{\boldsymbol{S}^\perp}}{\boldsymbol{d}^{\mathrm{T}} \boldsymbol{P}_{\boldsymbol{S}^\perp} \boldsymbol{d}} \right) \right] \boldsymbol{\xi} \,. \tag{51.105}$$

In practice, the SI estimator performs better, but is computationally slightly more complex, than the SX estimator.

51.3.8 Linear-Correction Least-Squares Estimator

Finding the LSE based on the spherical error criterion (51.98) is a linear minimization problem, i. e.,

$$\hat{\boldsymbol{\theta}}^{\mathrm{LSE}} = \arg\min_{\boldsymbol{\theta}} (\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{\xi})^{\mathrm{T}} (\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{\xi}) \tag{51.106}$$

subject to a quadratic constraint

$$\boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{\Xi} \boldsymbol{\theta} = 0 \,, \tag{51.107}$$

where $\boldsymbol{\Xi} \triangleq \mathrm{diag}(1, 1, 1, -1)$ is a diagonal and orthonormal matrix.

For such a constrained minimization problem, the technique of Lagrange multipliers will be used and the source location is determined by minimizing the Lagrangian

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}, \kappa) &= J_{\mathrm{sp}} + \kappa \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{\Xi} \boldsymbol{\theta} \\ &= (\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{\xi})^{\mathrm{T}} (\boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{\xi}) + \kappa \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{\Xi} \boldsymbol{\theta} \,, \end{aligned} \tag{51.108}$$

where κ is a Lagrange multiplier. Expanding this expression yields

$$\mathcal{L}(\boldsymbol{\theta}, \kappa) = \boldsymbol{\theta}^{\mathrm{T}} \left(\boldsymbol{A}^{\mathrm{T}} \boldsymbol{A} + \kappa \boldsymbol{\Xi} \right) \boldsymbol{\theta} - 2 \boldsymbol{\xi}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{\theta} + \boldsymbol{\xi}^{\mathrm{T}} \boldsymbol{\xi} \,. \tag{51.109}$$

The necessary conditions for minimizing (51.109) can be obtained by taking the gradient of $\mathcal{L}(\boldsymbol{\theta}, \kappa)$ with respect to $\boldsymbol{\theta}$ and equating the result to zero. This produces:

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}, \kappa)}{\partial \boldsymbol{\theta}} = 2 \left(\boldsymbol{A}^{\mathrm{T}} \boldsymbol{A} + \kappa \boldsymbol{\Xi} \right) \boldsymbol{\theta} - 2 \boldsymbol{A}^{\mathrm{T}} \boldsymbol{\xi} = \boldsymbol{0} \,. \tag{51.110}$$

Solving (51.110) for $\boldsymbol{\theta}$ yields the constrained least-squares estimate

$$\hat{\boldsymbol{\theta}} = \left(\boldsymbol{A}^{\mathrm{T}} \boldsymbol{A} + \kappa \boldsymbol{\Xi} \right)^{-1} \boldsymbol{A}^{\mathrm{T}} \boldsymbol{\xi} \,, \tag{51.111}$$

where κ is yet to be determined.

In order to find κ, we can impose the quadratic constraint directly by substituting (51.111) into (51.107), which leads to

$$\boldsymbol{\xi}^{\mathrm{T}} \boldsymbol{A} \left(\boldsymbol{A}^{\mathrm{T}} \boldsymbol{A} + \kappa \boldsymbol{\Xi} \right)^{-1} \boldsymbol{\Xi} \left(\boldsymbol{A}^{\mathrm{T}} \boldsymbol{A} + \kappa \boldsymbol{\Xi} \right)^{-1} \boldsymbol{A}^{\mathrm{T}} \boldsymbol{\xi} = 0 \,. \tag{51.112}$$

With eigenvalue analysis, the matrix $\boldsymbol{A}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{\Xi}$ can be decomposed as

$$\boldsymbol{A}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{\Xi} = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^{-1} \,, \tag{51.113}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \cdots, \lambda_4)$ and λ_i, $i = 1, \cdots, 4$, are the eigenvalues of the matrix $\boldsymbol{A}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{\Xi}$. Substituting (51.113) into (51.112), we may rewrite the constraint as:

$$\boldsymbol{p}^{\mathrm{T}} (\boldsymbol{\Lambda} + \kappa \boldsymbol{I})^{-2} \boldsymbol{q} = 0, \tag{51.114}$$

where

$$\begin{aligned} \boldsymbol{p} &= \boldsymbol{U}^{\mathrm{T}} \boldsymbol{\Xi} \boldsymbol{A}^{\mathrm{T}} \boldsymbol{\xi}, \\ \boldsymbol{q} &= \boldsymbol{U}^{\mathrm{T}} \boldsymbol{A}^{\mathrm{T}} \boldsymbol{\xi} \,. \end{aligned}$$

Let us define a function of the Lagrange multiplier as follows

$$f(\kappa) \triangleq \boldsymbol{p}^{\mathrm{T}} (\boldsymbol{\Lambda} + \kappa \boldsymbol{I})^{-2} \boldsymbol{q} = \sum_{i=1}^{4} \frac{p_i q_i}{(\kappa + \lambda_i)^2} \,. \tag{51.115}$$

This is a polynomial of degree eight and, because of its complexity, numerical methods need to be used for root searching. Since the root of (51.115) for κ is not unique, a two-step procedure will be followed such that the desired source location could be found.

Unconstrained Spherical Least-Squares Estimator

In the first step, we assume that x_{s}, y_{s}, z_{s}, and Ω_{s} are mutually independent or equivalently we disregard the quadratic constraint (51.107) intentionally. Then the LS solution minimizing (51.98) for $\boldsymbol{\theta}$ (the source location as well as its range) is given by

$$\hat{\boldsymbol{\theta}}_1 = \boldsymbol{A}^{\dagger} \boldsymbol{\xi} \,, \tag{51.116}$$

where $\boldsymbol{A}^{\dagger} = \left(\boldsymbol{A}^{\mathrm{T}} \boldsymbol{A} \right)^{-1} \boldsymbol{A}^{\mathrm{T}}$ is the pseudo-inverse of the matrix $\boldsymbol{A}$.

A good parameter estimator first and foremost needs to be unbiased. For such an unconstrained spherical least-squares estimator, the bias and covariance matrix can be approximated by using the following perturbation analysis method.

When measurement errors are present in the range differences, $\boldsymbol{A}$, $\boldsymbol{\xi}$, and the parameter estimate $\hat{\boldsymbol{\theta}}_1$ deviate from their true values and can be expressed as:

$$\boldsymbol{A} = \boldsymbol{A}^{\mathrm{t}} + \Delta\boldsymbol{A}, \quad \boldsymbol{\xi} = \boldsymbol{\xi}^{\mathrm{t}} + \Delta\boldsymbol{\xi}, \quad \hat{\boldsymbol{\theta}}_1 = \boldsymbol{\theta}^{\mathrm{t}} + \Delta\boldsymbol{\theta}\,, \tag{51.117}$$

where variables with superscript 't' denote the true values, which also satisfy

$$\boldsymbol{\theta}^{\mathrm{t}} = \boldsymbol{A}^{\mathrm{t}\dagger}\boldsymbol{\xi}^{\mathrm{t}}\,. \tag{51.118}$$

If the magnitudes of the perturbations are small, the second-order errors are insignificant compared to their first-order counterparts and therefore can be neglected for simplicity, which then yields:

$$\Delta\boldsymbol{A} = [\boldsymbol{0}\ \ \boldsymbol{\epsilon}]\,, \quad \Delta\boldsymbol{\xi} \approx -\boldsymbol{d}^{\mathrm{t}} \odot \boldsymbol{\epsilon}\,, \tag{51.119}$$

where $\odot$ denotes the Schur (element-by-element) product. Substituting (51.117) into (51.116) gives,

$$\begin{aligned}&\left(\boldsymbol{A}^{\mathrm{t}} + \Delta\boldsymbol{A}\right)^{\mathrm{T}}\left(\boldsymbol{A}^{\mathrm{t}} + \Delta\boldsymbol{A}\right)\left(\boldsymbol{\theta}^{\mathrm{t}} + \Delta\boldsymbol{\theta}\right)\\ &= \left(\boldsymbol{A}^{\mathrm{t}} + \Delta\boldsymbol{A}\right)^{\mathrm{T}}\left(\boldsymbol{\xi}^{\mathrm{t}} + \Delta\boldsymbol{\xi}\right)\,.\end{aligned} \tag{51.120}$$

Retaining only the linear perturbation terms and using (51.118) and (51.119) produces:

$$\Delta\boldsymbol{\theta} \approx -\boldsymbol{A}^{\mathrm{t}\dagger}\boldsymbol{D}\boldsymbol{\epsilon}, \tag{51.121}$$

where $\boldsymbol{D} \triangleq \mathrm{diag}(D_2, D_3, \cdots, D_N)$ is a diagonal matrix. Since the measurement error $\boldsymbol{\epsilon}$ in the range differences has zero mean, $\hat{\boldsymbol{\theta}}_1$ is an unbiased estimate of $\boldsymbol{\theta}^{\mathrm{t}}$ when the small-error assumption holds:

$$E\{\Delta\boldsymbol{\theta}\} \approx E\left\{-\boldsymbol{A}^{t\,\dagger}\boldsymbol{D}\boldsymbol{\epsilon}\right\} = \boldsymbol{0}_{4\times 1}\,. \tag{51.122}$$

The covariance matrix of $\Delta\boldsymbol{\theta}$ is then found as,

$$\boldsymbol{R}_{\Delta\theta\Delta\theta} = E\left\{\Delta\boldsymbol{\theta}\Delta\boldsymbol{\theta}^{\mathrm{T}}\right\} = \boldsymbol{A}^{\mathrm{t}\dagger}\boldsymbol{D}\boldsymbol{R}_{\epsilon\epsilon}\boldsymbol{D}\boldsymbol{A}^{\mathrm{t}\dagger^{\mathrm{T}}}\,, \tag{51.123}$$

where $\boldsymbol{R}_{\epsilon\epsilon}$ is known or is properly assumed a priori. Theoretically, the covariance matrix $\boldsymbol{R}_{\Delta\theta\Delta\theta}$ cannot be calculated since it contains true values. Nevertheless, it can be approximated by using the values in $\hat{\boldsymbol{\theta}}_1$ with sufficient accuracy, as suggested by numerical studies.

In the first unconstrained spherical LS estimate (51.116), the range information is redundant because of the independence assumption on the source location and range. If that information is simply discarded, the source location estimate is the same as the SI estimate but with less computational complexity [51.55]. To demonstrate this, we first write (51.116) in block form as

$$\hat{\boldsymbol{\theta}}_1 = \begin{pmatrix}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{S} & \boldsymbol{S}^{\mathrm{T}}\boldsymbol{d}\\ \boldsymbol{d}^{\mathrm{T}}\boldsymbol{S} & \boldsymbol{d}^{\mathrm{T}}\boldsymbol{d}\end{pmatrix}^{-1}\begin{pmatrix}\boldsymbol{S}^{\mathrm{T}}\\ \boldsymbol{d}^{\mathrm{T}}\end{pmatrix}\boldsymbol{\xi}\,. \tag{51.124}$$

It can easily be shown that:

$$\begin{pmatrix}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{S} & \boldsymbol{S}^{\mathrm{T}}\boldsymbol{d}\\ \boldsymbol{d}^{\mathrm{T}}\boldsymbol{S} & \boldsymbol{d}^{\mathrm{T}}\boldsymbol{d}\end{pmatrix}^{-1} = \begin{pmatrix}\boldsymbol{Q} & \boldsymbol{v}\\ \boldsymbol{v}^{\mathrm{T}} & \varrho\end{pmatrix}\,, \tag{51.125}$$

where

$$\begin{aligned}\boldsymbol{v} &= -\left(\boldsymbol{S}^{\mathrm{T}}\boldsymbol{S} - \frac{\boldsymbol{S}^{\mathrm{T}}\boldsymbol{d}\boldsymbol{d}^{\mathrm{T}}\boldsymbol{S}}{\boldsymbol{d}^{\mathrm{T}}\boldsymbol{d}}\right)^{-1}\frac{\boldsymbol{S}^{\mathrm{T}}\boldsymbol{d}}{\boldsymbol{d}^{\mathrm{T}}\boldsymbol{d}}\,,\\ \boldsymbol{Q} &= \left(\boldsymbol{S}^{\mathrm{T}}\boldsymbol{S}\right)^{-1}\left[\boldsymbol{I} - \left(\boldsymbol{S}^{\mathrm{T}}\boldsymbol{d}\right)\boldsymbol{v}^{\mathrm{T}}\right]\,,\\ \varrho &= \frac{1 - (\boldsymbol{d}^{\mathrm{T}}\boldsymbol{S})\boldsymbol{v}}{\boldsymbol{d}^{\mathrm{T}}\boldsymbol{d}}\,.\end{aligned}$$

Next, we define another projection matrix $\boldsymbol{P}_{\boldsymbol{d}^\perp}$ associated with the $\boldsymbol{d}$-orthogonal space:

$$\boldsymbol{P}_{\boldsymbol{d}^\perp} \triangleq \boldsymbol{I} - \frac{\boldsymbol{d}\boldsymbol{d}^{\mathrm{T}}}{\boldsymbol{d}^{\mathrm{T}}\boldsymbol{d}}\,, \tag{51.126}$$

and find

$$\boldsymbol{v} = -\left(\boldsymbol{S}^{\mathrm{T}}\boldsymbol{P}_{\boldsymbol{d}^\perp}\boldsymbol{S}\right)^{-1}\frac{\boldsymbol{S}^{\mathrm{T}}\boldsymbol{d}}{\boldsymbol{d}^{\mathrm{T}}\boldsymbol{d}}\,, \tag{51.127}$$

$$\boldsymbol{Q} = \left(\boldsymbol{S}^{\mathrm{T}}\boldsymbol{P}_{\boldsymbol{d}^\perp}\boldsymbol{S}\right)^{-1}\,. \tag{51.128}$$

Substituting (51.125) together with (51.127) and (51.128) into (51.124) yields the unconstrained spherical LS estimate for the source coordinates:

$$\hat{\boldsymbol{\gamma}}_{\mathrm{s},1} = \left(\boldsymbol{S}^{\mathrm{T}}\boldsymbol{P}_{\boldsymbol{d}^\perp}\boldsymbol{S}\right)^{-1}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{P}_{\boldsymbol{d}^\perp}\boldsymbol{\xi}\,, \tag{51.129}$$

which is the minimizer of

$$J_1(\boldsymbol{\gamma}_{\mathrm{s}}) = \left\|\boldsymbol{P}_{\boldsymbol{d}^\perp}\boldsymbol{\xi} - \boldsymbol{P}_{\boldsymbol{d}^\perp}\boldsymbol{S}\boldsymbol{\gamma}_{\mathrm{s}}\right\|^2\,, \tag{51.130}$$

or the least-squares solution to the linear equation

$$\boldsymbol{P}_{\boldsymbol{d}^\perp}\boldsymbol{S}\boldsymbol{\gamma}_{\mathrm{s}} = \boldsymbol{P}_{\boldsymbol{d}^\perp}\boldsymbol{\xi}. \tag{51.131}$$

In fact, the first unconstrained spherical LS estimator tries to approximate the projection of the observation vector $\boldsymbol{\xi}$ with the projections of the column vectors of the microphone location matrix $\boldsymbol{S}$ onto the $\boldsymbol{d}$-orthogonal space. The source location estimate is the coefficient vector associated with the *best* approximation. Clearly from (51.131), this estimation procedure is the generalization of the plane intersection (PI) method proposed in [51.56].

By using the Sherman–Morrison formula [51.57]

$$\left(\boldsymbol{A} + \boldsymbol{x}\boldsymbol{y}^{\mathrm{T}}\right)^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{x}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}^{-1}}{1 + \boldsymbol{y}^{\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{x}}\,, \tag{51.132}$$

Amazon EX1010 Page 1056

we can expand the item in (51.129) as

$$\left(\boldsymbol{S}^{\mathrm{T}}\boldsymbol{P}_{\boldsymbol{d}^{\perp}}\boldsymbol{S}\right)^{-1}=\left[\boldsymbol{S}^{\mathrm{T}}\boldsymbol{S}-\left(\frac{\boldsymbol{S}^{\mathrm{T}}\boldsymbol{d}}{\boldsymbol{d}^{\mathrm{T}}\boldsymbol{d}}\right)(\boldsymbol{S}^{\mathrm{T}}\boldsymbol{d})^{\mathrm{T}}\right]^{-1},$$

and finally show that the unconstrained spherical LS estimate (51.129) is equivalent to the SI estimate (51.105), i. e., $\hat{\boldsymbol{\gamma}}_{\mathrm{s},1}\equiv\hat{\boldsymbol{\gamma}}_{\mathrm{s}}^{\mathrm{SI}}$.

Although the unconstrained spherical LS and the SI estimators are mathematically equivalent, they are quite different in computational efficiency due to their different approaches to the source localization problem. The complexities of the SI and unconstrained spherical LS estimators are $\mathcal{O}\left(N^3\right)$ and $\mathcal{O}(N)$, respectively. In comparison, the unconstrained spherical LS estimator reduces the complexity of the SI estimator by a factor of N^2, which is significant when N is large (more microphones are used).

Linear Correction

In the previous subsection, we developed the unconstrained spherical LS estimator (USLSE) for source localization and demonstrated that it is mathematically equivalent to the SI estimator but with less computational complexity. Although the USLSE/SI estimates can be accurate, as indicated in [51.55] among others, it is helpful to exploit the redundancy of source range to improve the statistical efficiency (i. e., to reduce the variance of source location estimates) of the overall estimation procedure. Therefore, in the second step, we intend to correct the USLS estimate $\hat{\boldsymbol{\theta}}_1$ to make a better estimate $\hat{\boldsymbol{\theta}}_2$ of $\boldsymbol{\theta}$. This new estimate should be in the neighborhood of $\hat{\boldsymbol{\theta}}_1$ and should obey the constraint (51.107). We expect that the corrected estimate will still be unbiased and have a smaller variance.

To begin, we substitute $\hat{\boldsymbol{\theta}}_1=\boldsymbol{\theta}^{\mathrm{t}}+\Delta\boldsymbol{\theta}$ into (51.110) and expand the expression to find

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\hat{\boldsymbol{\theta}}_1+\kappa\boldsymbol{\Xi}\hat{\boldsymbol{\theta}}_1-(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}+\kappa\boldsymbol{\Xi})\Delta\boldsymbol{\theta}=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{\xi}\,. \tag{51.133}$$

Combined with (51.116), (51.133) becomes

$$(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}+\kappa\boldsymbol{\Xi})\Delta\boldsymbol{\theta}=\kappa\boldsymbol{\Xi}\hat{\boldsymbol{\theta}}_1\,, \tag{51.134}$$

and hence

$$\Delta\boldsymbol{\theta}=\kappa\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{\Xi}\boldsymbol{\theta}^{\mathrm{t}}\,. \tag{51.135}$$

Substituting (51.135) into $\hat{\boldsymbol{\theta}}_1=\boldsymbol{\theta}^{\mathrm{t}}+\Delta\boldsymbol{\theta}$ yields

$$\hat{\boldsymbol{\theta}}_1=\left[\boldsymbol{I}+\kappa\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{\Xi}\right]\boldsymbol{\theta}^{\mathrm{t}}\,. \tag{51.136}$$

Solving (51.136) for $\boldsymbol{\theta}^{\mathrm{t}}$ produces the corrected estimate $\hat{\boldsymbol{\theta}}_2$ and also the final output of the linear-correction least-squares (LCLS) estimator:

$$\hat{\boldsymbol{\theta}}_2=\left[\boldsymbol{I}+\kappa\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{\Xi}\right]^{-1}\hat{\boldsymbol{\theta}}_1\,. \tag{51.137}$$

Equation (51.137) suggests how the second-step processing updates the source location estimate based on the first unconstrained spherical least squares result, or equivalently the SI estimate. If the regularity condition [51.58]

$$\lim_{i\to\infty}\left[\kappa(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{\Xi}\right]^{i}=\boldsymbol{0} \tag{51.138}$$

is satisfied, then the estimate $\hat{\boldsymbol{\theta}}_2$ can be expanded in a Neumann series :

$$\begin{aligned}\hat{\boldsymbol{\theta}}_2&=\left\{\boldsymbol{I}+\left[-\kappa\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{\Xi}\right]\right.\\&\quad\left.+\left[-\kappa\left[\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right]^{-1}\boldsymbol{\Xi}\right]^{2}+\cdots\right\}\hat{\boldsymbol{\theta}}_1\\&=\hat{\boldsymbol{\theta}}_1+\sum_{i=1}^{\infty}\left[-\kappa\left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{\Xi}\right]^{i}\hat{\boldsymbol{\theta}}_1\,,\end{aligned} \tag{51.139}$$

where the second term is the linear correction. Equation (51.138) implies that, in order to avoid divergence, the Lagrange multiplier κ should be small. In addition, κ needs to be determined carefully such that $\hat{\boldsymbol{\theta}}_2$ obeys the quadratic constraint (51.107).

Because the function $f(\kappa)$ is smooth near $\kappa=0$ (corresponding to the neighborhood of $\hat{\boldsymbol{\theta}}_1$), as suggested by numerical experiments, the secant method [51.59] can be used to determine its desired root. Two reasonable initial points can be chosen as:

$$\kappa_0=0,\quad \kappa_1=\beta\,, \tag{51.140}$$

where the small number β depends on the array geometry. Five iterations should be sufficient to give an accurate approximation to the root.

The idea of exploiting the relationship between a sound source's range and its location coordinates to improve the estimation efficiency of the SI estimator was first suggested by *Chan* and *Ho* [51.52] with a quadratic correction. Accordingly, they constructed a quadratic data model for $\hat{\boldsymbol{\theta}}_1$.

$$\hat{\boldsymbol{\theta}}_1\odot\hat{\boldsymbol{\theta}}_1=\boldsymbol{T}(\boldsymbol{\gamma}_{\mathrm{s}}\odot\boldsymbol{\gamma}_{\mathrm{s}})+\boldsymbol{v}\,, \tag{51.141}$$

where

$$\boldsymbol{T} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

is a constant matrix, and $\boldsymbol{n}$ is the corrupting noise. In contrast to the linear correction technique based on the Lagrange multiplier, the quadratic counterpart needs to know the covariance matrix $\boldsymbol{R}_{\epsilon\epsilon}$ of measurement errors in the range differences a priori. In a real-time digital source localization system, a poorly estimated $\boldsymbol{R}_{\epsilon\epsilon}$ will lead to performance degradation. In addition, the quadratic-correction least-squares estimation procedure uses the perturbation approaches to linearly approximate $\Delta\boldsymbol{\theta}$ and $\boldsymbol{v}$ in (51.117) and (51.141), respectively. Therefore, the approximations of their corresponding covariance matrices $\boldsymbol{R}_{\Delta\theta\Delta\theta}$ and $\boldsymbol{R}_{\nu\nu}$ can be good only when the noise level is low. When noise is at a practically high level, the quadratic-correction least squares estimate has a large bias and a high variance. Furthermore, since the true value of the source location that is necessary for calculating $\boldsymbol{R}_{\Delta\theta\Delta\theta}$ and $\boldsymbol{R}_{\nu\nu}$ cannot be known theoretically, the estimated source location has to be utilized for approximation. It was suggested in [51.52] that several iterations in the second correction stage would improve estimation accuracy. However, while the bias is suppressed after iterations, the estimate is closer to the SI solution and the variance is boosted, as reported in [51.50]. Finally, the direct solutions of the quadratic-correction least-squares estimator are the squares of the source location coordinates $\boldsymbol{\gamma}_s \odot \boldsymbol{\gamma}_s$. In 3-D space, these correspond to eight positions, which introduce decision ambiguities. Other physical criteria, such as the domain of interest, were suggested, but these are hard to define in practical situations, particularly when one of the source coordinates is close to zero.

In comparison, the linear-correction method updates the source location estimate of the first unconstrained spherical LS estimator without making any assumption about the error covariance matrix and without resorting to a linear approximation. Even though we need to find a small root of function (51.115) for the Lagrange multiplier κ that satisfies the regularity condition (51.138), the function $f(\kappa)$ is smooth around zero and the solution can easily be determined using the secant method. The linear-correction method achieves a better balance between computational complexity and estimation accuracy.

51.4 Summary

In this chapter, we have explained why TDE-based two-stage approaches have become the technique of choice for real-time speech source localization using microphone arrays. We presented a comprehensive overview of the research and development on this technology over the last three decades.

The chapter consists of two parts. The first part was devoted to the TDE problem. We began with a close examination of two signal models used in this problem. Their strengths and drawbacks were discussed. We reviewed the well-known generalized cross-correlation (GCC) methods and intended to treat this class of methods as a benchmark for a number of recently developed new algorithms. We made it clear by analysis that the GCC methods cannot cope well with room reverberation and described two strategies to combat this effect. One strategy is to employ the more-realistic real reverberant model in the formulation of the TDE problem and try to blindly identify the acoustic channel impulse responses from which the time difference of arrival (TDOA) is inferred. The adaptive eigenvalue decomposition algorithm and the adaptive blind multichannel identification algorithms were developed. The other strategy is to exploit the TDOA redundancy among different microphone pairs. This leads to the multichannel spatial predication and interpolation methods, the multichannel cross-correlation coefficient method, and the minimum-entropy method, which have all been thoroughly explored.

The second part of the Chapter surveyed the source localization techniques. The localization problem was postulated from the perspective of estimation theory, and the Cramèr–Rao lower bound for unbiased location estimators was derived. After an insightful review of conventional approaches ranging from maximum-likelihood to least-squares estimators, we presented a recently developed linear-correction least-squares algorithm that is more robust to measurement errors and that is computationally as well as statistically more efficient.

Amazon EX1010 Page 1058

References

51.1 D.R. Fischell, C.H. Coker: A speech direction finder, Proc. ICASSP (1984) pp. 19.8.1–19.8.4.

51.2 H.F. Silverman: Some analysis of microphone arrays for speech data analysis, IEEE Trans. ASSP **35**, 1699–1712 (1987)

51.3 J.L. Flanagan, A. Surendran, E. Jan: Spatially selective sound capture for speech and audio processing, Speech Commun. **13**, 207–222 (1993)

51.4 D.B. Ward, G.W. Elko: Mixed nearfield/farfield beamforming: a new technique for speech acquisition in a reverberant environment, Proc. IEEE ASSP Workshop Appl. Signal Process. Audio Acoust. (1997)

51.5 D.V. Rabinkin, R.J. Ranomeron, J.C. French, J.L. Flanagan: A DSP implementation of source location using microphone arrays, Proc. SPIE **2846**, 88–99 (1996)

51.6 H. Wang, P. Chu: Voice source localization for automatic camera pointing system in videoconferencing, Proc. IEEE ASSP Workshop Appl. Signal Process. Audio Acoust. (1997)

51.7 C. Wang, M.S. Brandstein: A hybrid real-time face tracking system, Proc. ICASSP, Vol. 6 (1998) pp. 3737–3741

51.8 Y. Huang, J. Benesty, G.W. Elko: Microphone arrays for video camera steering. In: *Acoustic Signal Processing for Telecommunication*, ed. by S.L. Gay, J. Benesty (Kluwer Academic, Boston 2000) pp. 239–259, chap. 11

51.9 S. Haykin: Radar array processing for angle of arrival estimation. In: *Array Signal Process*, ed. by S. Haykin (Prentice-Hall, Englewood Cliffs 1985)

51.10 H. Krim, M. Viberg: Two decades of array signal processing research: the parametric approach, IEEE Signal Process. Mag. **13**(4), 67–94 (1996)

51.11 R.J. Vaccaro: The past, present; future of underwater acoustic signal processing, IEEE Signal Process. Mag. **15**, 21–51 (1998)

51.12 D.V. Sidorovich, A.B. Gershman: Two-dimensional wideband interpolated root-MUSIC applied to measured seismic data, IEEE Trans. Signal Process. **46**(8), 2263–2267 (1998)

51.13 J. Capon: Maximum-likelihood spectral estimation, Proc. IEEE **57**, 1408–1418 (1969)

51.14 R.O. Schmidt: *A Signal Subspace Approach to Multiple Emitter Location and Spectral Estimation* (Stanford University, Stanford 1981), Ph.D. thesis

51.15 W. Bangs, P. Schultheis: Space–time processing for optimal parameter estimation. In: *Signal Process*, ed. by J. Griffiths, P. Stocklin, C. Van Schooneveld (New York, Academic 1973) pp. 577–590

51.16 W.R. Hahn, S.A. Tretter: Optimum processing for delay-vector estimation in passive signal arrays, IEEE Trans. Inform. Theory **19**, 608–614 (1973)

51.17 M. Wax, T. Kailath: Optimum localization of multiple sources by passive arrays, IEEE Trans. ASSP **31**(5), 1210–1218 (1983)

51.18 M.S. Brandstein, H.F. Silverman: A practical methodology for speech source localization with microphone arrays, Comput. Speech Lang. **2**, 91–126 (1997)

51.19 C.H. Knapp, G.C. Carter: The generalized correlation method for estimation of time delay, IEEE Trans. ASSP **24**, 320–327 (1976)

51.20 G.C. Carter, A.H. Nuttall, P.G. Cable: The smoothed coherence transform, Proc. IEEE **61**, 1497–1498 (1973)

51.21 J.P. Ianniello: Time delay estimation via cross-correlation in the presence of large estimation errors, IEEE Trans. ASSP **30**, 998–1003 (1982)

51.22 B. Champagne, S. Bédard, A. Stéphenne: Performance of time-delay estimation in presence of room reverberation, IEEE Trans. Speech Audio Process. **4**, 148–152 (1996)

51.23 M. Omologo, P. Svaizer: Acoustic event localization using a crosspower-spectrum phase based technique, Proc. ICASSP, Vol. 2 (1994) pp. 273–276

51.24 M.S. Brandstein: A pitch-based approach to time-delay estimation of reverberant speech, Proc. IEEE ASSP Workshop Appl. Signal Process. Audio Acoustics (1997)

51.25 M. Omologo, P. Svaizer: Acoustic source location in noisy and reverberant environment using CSP analysis, ICASSP, Vol. 2 (1996) pp. 921–924

51.26 A. Stéphenne, B. Champagne: Cepstral prefiltering for time delay estimation in reverberant environments, Proc. ICASSP, Vol. 5. (1995) pp. 3055–3058

51.27 J. Benesty: Adaptive eigenvalue decomposition algorithm for passive acoustic source localization, J. Acoust. Soc. Am. **107**, 384–391 (2000)

51.28 G. Xu, H. Liu, L. Tong, T. Kailath: A least-squares approach to blind channel identification, IEEE Trans. Signal Process. **43**, 2982–2993 (1995)

51.29 Y. Huang, J. Benesty: Adaptive multichannel time delay estimation based on blind system identification for acoustic source localization. In: *Adaptive Signal Processing: Application to Real-World Problems*, ed. by J. Benesty, Y. Huang (Springer, Berlin:Heidelberg 2003)

51.30 Y. Huang, J. Benesty: Adaptive multi-channel least mean square and Newton algorithms for blind channel identification, Signal Process. **82**, 1127–1138 (2002)

51.31 Y. Huang, J. Benesty: A class of frequency-domain adaptive approaches to blind multichannel identification, IEEE Trans. Signal Process. **51**, 11–24 (2003)

51.32 Y. Huang, J. Benesty, J. Chen: Optimal step size of the adaptive multichannel LMS algorithm for blind

SIMO identification, IEEE Signal Process. Lett. **12**, 173–176 (2005)
51.33 Y. Huang, J. Benesty, J. Chen: *Acoustic MIMO Signal Process* (Berlin, Springer 2006)
51.34 R.L. Kirlin, D.F. Moore, R.F. Kubichek: Improvement of delay measurements from sonar arrays via sequential state estimation, IEEE Trans. ASSP **29**, 514–519 (1981)
51.35 T. Nishiura, T. Yamada, S. Nakamura, K. Shikano: Localization of multiple sound sources based on a CSP analysis with a microphone array, Proc. ICASSP (2000) pp. 1053–1055
51.36 S.M. Griebel, M.S. Brandstein: Microphone array source localization using realizable delay vectors, Proc. IEEE ASSP Workshop Appl. Signal Process. Audio Acoust. (2001) pp. 71–74
51.37 J. DiBiase, H. Silverman, M. Brandstein: Robust localization in reverberant rooms. In: *Microphone Arrays: Signal Processing Techniques and Applications*, ed. by M. Branstein, D. Ward (Springer, Berlin 2001)
51.38 J. Chen, J. Benesty, Y. Huang: Robust time delay estimation exploiting redundancy among multiple microphones, IEEE Trans. Speech Audio Process. **11**, 549–557 (2003)
51.39 J. Benesty, J. Chen, Y. Huang: Time-delay estimation via linear interpolation and cross-correlation, IEEE Trans. Speech Audio Process. **12**, 509–519 (2004)
51.40 J.S. Bendat, A.G. Piersol: *Random Data Analysis and Measurement Procedures* (Wiley, New York 1986)
51.41 D. Cochran, H. Gish, D. Sinno: A geometric approach to multichannel signal detection, IEEE Trans. Signal Process. **43**, 2049–2057 (1995)
51.42 C.E. Shannon: A mathematical theory of communication, Bell Syst. Tech. J. **27**, 379–423, 623–656 (1948)
51.43 T.M. Cover, J.A. Thomas: *Elements of Information Theory* (Wiley, New York 1991)
51.44 I. Kojadinovic: On the use of mutual information in data analysis: an overview, Int. Symposium on Applied Stochastic Models and Data Analysis (2005)
51.45 J. Benesty, Y. Huang, J. Chen: Time delay estimation via minimum entropy, IEEE Signal Process. Lett. **14**, 157–160 (2006)
51.46 L.R. Rabiner, R.W. Schafer: *Digital Process. of Speech Signals* (Prentice-Hall, Englewood Cliffs 1978)
51.47 S. Gazor, W. Zhang: Speech probability distribution, IEEE Signal Process. Lett. **10**(7), 204–207 (2003)
51.48 S. Kotz, T.J. Kozubowski, K. Podgórski: *An asymmetric multivariate Laplace distribution*, Technical Report No. 367, Department of Statistics and Applied Probaility (Univ. of California, Santa Barbara 2000)
51.49 T. Eltoft, T. Kim, T.-W. Lee: On the multivariate Laplace distribution, IEEE Signal Process. Lett. **13**, 300–303 (2006)
51.50 Y. Huang, J. Benesty, G.W. Elko, R.M. Mersereau: Real-time passive source localization: an unbiased linear-correction least-squares approach, IEEE Trans. Speech Audio Process. **9**, 943–956 (2001)
51.51 S.M. Kay: *Fundamentals of Statistical Signal Process.: Estimation Theory* (Prentice-Hall, Englewood Cliffs 1993)
51.52 Y.T. Chan, K.C. Ho: A simple and efficient estimator for hyperbolic location, IEEE Trans. Signal Process. **42**, 1905–1915 (1994)
51.53 H.C. Schau, A.Z. Robinson: Passive source localization employing intersecting spherical surfaces from time-of-arrival differences, IEEE Trans. ASSP **35**, 1223–1225 (1987)
51.54 J.S. Abel, J.O. Smith: The spherical interpolation method for closed-form passive source localization using range difference measurements echo cancelation, Proc. ICASSP (1987) pp. 471–474
51.55 Y. Huang, J. Benesty, G.W. Elko: Passive acoustic source localization for video camera steering, Proc. IEEE Int. Conf. ACSP, Vol. 2. (2000) pp. 909–912
51.56 R.O. Schmidt: A new approach to geometry of range difference location, IEEE Trans. Aerosp. Electron. **8**, 821–835 (1972)
51.57 T.K. Moon, W.C. Stirling: *Mathematical Methods and Algorithms* (Prentice-Hall, Upper Saddle River 1999)
51.58 C.D. Meyer: *Matrix Analysis and Applied Linear Algebra* (SIAM, Philadelphia 2000)
51.59 W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling: *Numerical Recipes in C: The Art of Scientific Computing* (Cambridge Univ. Press, Cambridge 1988)

Amazon EX1010 Page 1060

52. Convolutive Blind Source Separation Methods

M. S. Pedersen, J. Larsen, U. Kjems, L. C. Parra

In this chapter, we provide an overview of existing algorithms for blind source separation of convolutive audio mixtures. We provide a taxonomy in which many of the existing algorithms can be organized and present published results from those algorithms that have been applied to real-world audio separation tasks.

52.1 **The Mixing Model** 1066
52.1.1 Special Cases 1067
52.1.2 Convolutive Model in the Frequency Domain 1067
52.1.3 Block-Based Model 1068
52.2 **The Separation Model** 1068
52.2.1 Feedforward Structure 1068
52.2.2 Relation Between Source and Separated Signals 1068
52.2.3 Feedback Structure 1069
52.2.4 Example: The TITO System 1070
52.3 **Identification** 1071
52.4 **Separation Principle** 1071
52.4.1 Higher-Order Statistics 1072
52.4.2 Second-Order Statistics 1073
52.4.3 Sparseness in the Time/Frequency Domain 1075
52.4.4 Priors from Auditory Scene Analysis and Psychoacoustics 1076
52.5 **Time Versus Frequency Domain** 1076
52.5.1 Frequency Permutations 1077
52.5.2 Time–Frequency Algorithms 1077
52.5.3 Circularity Problem 1078
52.5.4 Subband Filtering 1078
52.6 **The Permutation Ambiguity** 1078
52.6.1 Consistency of the Filter Coefficients 1078
52.6.2 Consistency of the Spectrum of the Recovered Signals 1081
52.6.3 Global Permutations 1083
52.7 **Results** 1084
52.8 **Conclusion** 1084
References 1084

During the past decades, much attention has been given to the separation of mixed sources, in particular for the *blind* case where both the sources and the mixing process are unknown and only recordings of the mixtures are available. In several situations it is desirable to recover all sources from the recorded mixtures, or at least to segregate a particular source. Furthermore, it may be useful to identify the mixing process itself to reveal information about the physical mixing system.

In some simple mixing models each recording consists of a sum of differently weighted source signals. However, in many real-world applications, such as in acoustics, the mixing process is more complex. In such systems, the mixtures are weighted and delayed, and each source contributes to the sum with multiple delays corresponding to the multiple paths by which an acoustic signal propagates to a microphone. Such filtered sums of different sources are called convolutive mixtures. Depending on the situation, the filters may consist of a few delay elements, as in radio communications, or up to several thousand delay elements as in acoustics. In these situations the sources are the desired signals, yet only the recordings of the mixed sources are available and the mixing process is unknown.

There are multiple potential applications of convolutive blind source separation. In acoustics different sound sources are recorded simultaneously with possibly multiple microphones. These sources may be speech or music, or underwater signals recorded in passive sonar [52.1]. In radio communications, antenna arrays receive mixtures of different communication signals [52.2,3]. Source separation has also been applied to astronomical data or satellite images [52.4]. Finally, convolutive models have been used to interpret functional brain imaging data and biopotentials [52.5–8].

This chapter considers the problem of separating linear convolutive mixtures focusing in particular on acoustic mixtures. The *cocktail-party problem* has come to characterize the task of recovering speech in a room of simultaneous and independent speakers [52.9, 10]. Con-

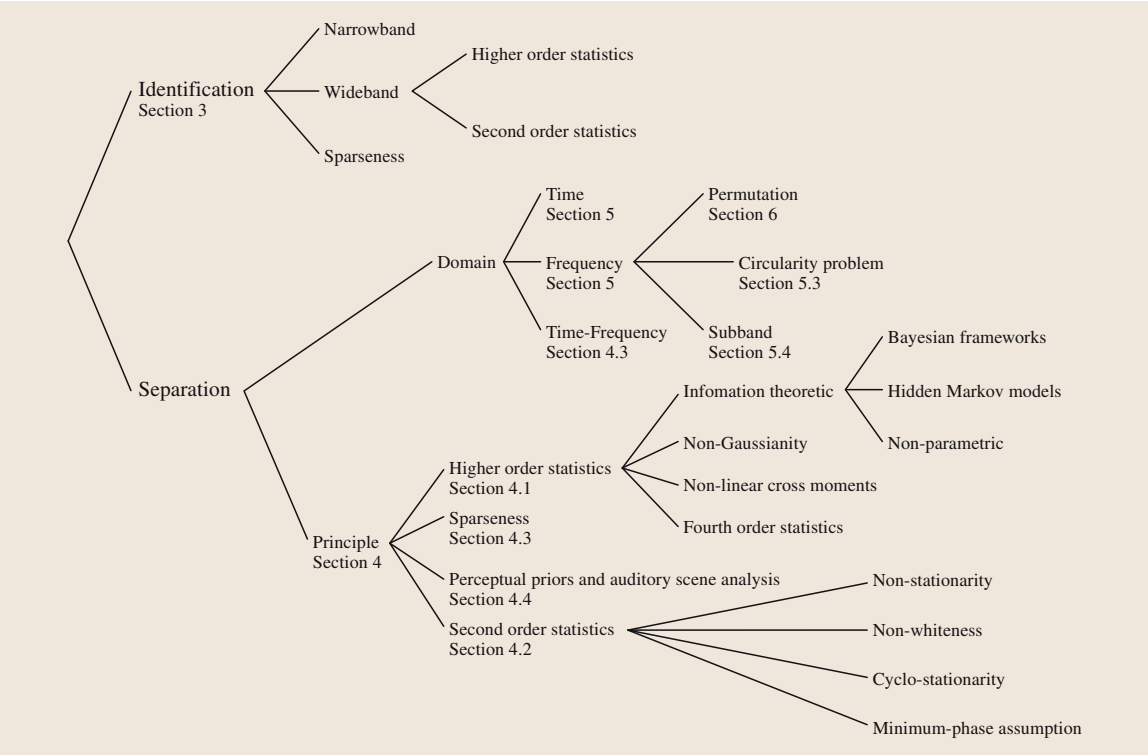


Fig. 52.1 Overview of important areas within blind separation of convolutive sources

volutive blind source separation (BSS) has often been proposed as a possible solution to this problem as it carries the promise to recover the sources exactly. The theory on linear noise-free systems establishes that a system with multiple inputs (sources) and multiple output (sensors) can be inverted under some reasonable assumptions with appropriately chosen multidimensional filters [52.11]. The challenge lies in finding these convolution filters.

There are already a number of partial reviews available on this topic [52.12–22]. The purpose of this chapter is to provide a complete survey of convolutive BSS and identify a taxonomy that can organize the large number of available algorithms. This may help practitioners and researchers new to the area of convolutive source separation obtain a complete overview of the field. Hopefully those with more experience in the field can identify useful tools, or find inspiration for new algorithms. Figure 52.1 provides an overview of the different topics within convolutive BSS and in which section they are covered. An overview of published results is given in Sect. 52.7.

52.1 The Mixing Model

First we introduce the basic model of convolutive mixtures. At the discrete time index t, a mixture of N source signals $\boldsymbol{s}(t) = [s_1(t), \ldots, s_N(t)]^{\mathrm{T}}$ are received at an array of M sensors. The received signals are denoted by $\boldsymbol{x}(t) = [x_1(t), \ldots, x_M(t)]^{\mathrm{T}}$. In many real-world applications the sources are said to be *convolutively* (or dynamically) mixed. The convolutive model introduces the following relation between the m-th mixed signal, the original source signals, and some additive sensor noise $v_m(t)$:

$$x_m(t) = \sum_{n=1}^{N} \sum_{k=0}^{K-1} a_{mnk} s_n(t-k) + v_m(t) \qquad (52.1)$$

The mixed signal is a linear mixture of filtered versions of each of the source signals, and a_{mnk} represent the corresponding mixing filter coefficients. In practice, these coefficients may also change in time, but for simplicity the mixing model is often assumed stationary. In theory the filters may be of infinite length, which may be implemented as infinite impulse response (IIR) systems, however, in practice it is sufficient to assume $K < \infty$. In matrix form the convolutive model can be written:

$$\boldsymbol{x}(t) = \sum_{k=0}^{K-1} \boldsymbol{A}_k \boldsymbol{s}(t-k) + \boldsymbol{v}(t) \,, \tag{52.2}$$

where $\boldsymbol{A}_k$ is an $M \times N$ matrix that contains the k-th filter coefficients. $\boldsymbol{v}(t)$ is the $M \times 1$ noise vector. In the z-domain the convolutive mixture (52.2) can be written:

$$\boldsymbol{X}(z) = \boldsymbol{A}(z)\boldsymbol{S}(z) + \boldsymbol{V}(z) \,, \tag{52.3}$$

where $\boldsymbol{A}(z)$ is a matrix with finite impulse response (FIR) polynomials in each entry [52.23].

52.1.1 Special Cases

There are some special cases of the convolutive mixture which simplify (52.2).

Instantaneous Mixing Model

Assuming that all the signals arrive at the sensors at the same time without being filtered, the convolutive mixture model (52.2) simplifies to

$$\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t) + \boldsymbol{v}(t) \,. \tag{52.4}$$

This model is known as the *instantaneous* or delayless (linear) mixture model. Here, $\boldsymbol{A} = \boldsymbol{A}_0$, is an $M \times N$ matrix containing the mixing coefficients. Many algorithms have been developed to solve the instantaneous mixture problem, see e.g., [52.17, 24].

Delayed Sources

Assuming a reverberation-free environment with propagation delays the mixing model can be simplified to

$$x_m(t) = \sum_{n=1}^{N} a_{mn} s_n(t - k_{mn}) + v_m(t) \,, \tag{52.5}$$

where k_{mn} is the propagation delay between source n and sensor m.

Noise Free

In the derivation of many algorithms, the convolutive model (52.2) is assumed to be noise-free, i. e.,

$$\boldsymbol{x}(t) = \sum_{k=0}^{K-1} \boldsymbol{A}_k \boldsymbol{s}(t-k) \,. \tag{52.6}$$

Over- and Underdetermined Sources

Often it is assumed that the number of sensors equals (or exceeds) the number of sources in which case linear methods may suffice to invert the linear mixing. However, if the number of sources exceeds the number of sensors the problem is underdetermined, and even under perfect knowledge of the mixing system linear methods will not be able to recover the sources.

52.1.2 Convolutive Model in the Frequency Domain

The convolutive mixing process (52.2) can be simplified by transforming the mixtures into the frequency domain. The linear convolution in the time domain can be written in the frequency domain as separate multiplications for each frequency:

$$\boldsymbol{X}(\omega) = \boldsymbol{A}(\omega)\boldsymbol{S}(\omega) + \boldsymbol{V}(\omega) \,. \tag{52.7}$$

At each frequency, $\omega = 2\pi f$, $\boldsymbol{A}(\omega)$ is a complex $M \times N$ matrix, $\boldsymbol{X}(\omega)$ and $\boldsymbol{V}(\omega)$ are complex $M \times 1$ vectors, and similarly $\boldsymbol{S}(\omega)$ is a complex $N \times 1$ vector. The frequency transformation is typically computed using a discrete Fourier transform (DFT) within a time frame of length T starting at time t:

$$\boldsymbol{X}(\omega, t) = \mathrm{DFT}([\boldsymbol{x}(t), \cdots, \boldsymbol{x}(t+T-1)]) \,, \tag{52.8}$$

and correspondingly for $\boldsymbol{S}(\omega, t)$ and $\boldsymbol{V}(\omega, t)$. Often a windowed discrete Fourier transform is used:

$$\boldsymbol{X}(\omega, t) = \sum_{\tau=0}^{T-1} w(\tau)\boldsymbol{x}(t+\tau)\mathrm{e}^{-i\omega\tau/T} \,, \tag{52.9}$$

where the window function $w(\tau)$ is chosen to minimize band overlap due to the limited temporal aperture. By using the fast Fourier transform (FFT) convolutions can be implemented efficiently in the discrete Fourier domain, which is important in acoustics as it often requires long time-domain filters.

52.1.3 Block-Based Model

Instead of modeling individual samples at time t one can also consider a block consisting of T samples. The equations for such a block can be written as:

$$\begin{aligned}
\boldsymbol{x}(t) &= \boldsymbol{A}_0\boldsymbol{s}(t)+\cdots+\boldsymbol{A}_{K-1}\boldsymbol{s}(t-K+1)\,,\\
\boldsymbol{x}(t-1) &= \boldsymbol{A}_0\boldsymbol{s}(t-1)+\cdots+\boldsymbol{A}_{K-1}\boldsymbol{s}(t-K)\,,\\
\boldsymbol{x}(t-2) &= \boldsymbol{A}_0\boldsymbol{s}(t-2)+\cdots+\boldsymbol{A}_{K-1}\boldsymbol{s}(t-K-1)\,,\\
&\vdots \qquad .
\end{aligned}$$

The M-dimensional output sequence can be written as an $MT\times 1$ vector:

$$\hat{\boldsymbol{x}}(t) = [\boldsymbol{x}^{\mathrm{T}}(t), \boldsymbol{x}^{\mathrm{T}}(t-1), \cdots, \boldsymbol{x}^{\mathrm{T}}(t-T+1)]^{\mathrm{T}}\,, \tag{52.10}$$

where $\boldsymbol{x}^{\mathrm{T}}(t) = [x_1(t), \cdots, x_M(t)]$. Similarly, the N-dimensional input sequence can be written as an $N(T+K-1)\times 1$ vector:

$$\hat{\boldsymbol{s}}(t) = [\boldsymbol{s}^{\mathrm{T}}(t), \boldsymbol{s}^{\mathrm{T}}(t-1), \cdots, \boldsymbol{s}^{\mathrm{T}}(t-T-K+2)]^{\mathrm{T}} \tag{52.11}$$

From this the convolutive mixture can be expressed formally as:

$$\hat{\boldsymbol{x}}(t) = \hat{\boldsymbol{A}}\hat{\boldsymbol{s}}(t)+\hat{\boldsymbol{v}}(t), \tag{52.12}$$

where $\hat{\boldsymbol{A}}$ has the following form:

$$\hat{\boldsymbol{A}} = \begin{pmatrix} \boldsymbol{A}_0 & \cdots & \boldsymbol{A}_{K-1} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \ddots & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_0 & \cdots & \boldsymbol{A}_{K-1} \end{pmatrix}. \tag{52.13}$$

The block-Toeplitz matrix $\hat{\boldsymbol{A}}$ has dimensions $MT\times N(T+K-1)$. On the surface, (52.12) has the same structure as an instantaneous mixture given in (52.4), and the dimensionality has increased by a factor T. However, the models differ considerably as the elements within $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{s}}(t)$ are now coupled in a rather specific way.

The majority of the work in convolutive source separation assumes a mixing model with a finite impulse response as in (52.2). A notable exception is the work by Cichocki, which also considers an autoregressive (AR) component as part of the mixing model [52.18]. The autoregressive moving-average (ARMA) mixing system proposed there is equivalent to a first-order Kalman filter with an infinite impulse response.

52.2 The Separation Model

The objective of blind source separation is to find an estimate $\boldsymbol{y}(t)$ that is a model of the original source signals $\boldsymbol{s}(t)$. For this, it may not be necessary to identify the mixing filters $\boldsymbol{A}_k$ explicitly. Instead, it is often sufficient to estimate separation filters $\boldsymbol{W}_l$ that remove the cross-talk introduced by the mixing process. These separation filters may have a feed-back structure with an infinite impulse response, or may have a finite impulse response expressed as feedforward structure.

52.2.1 Feedforward Structure

An FIR separation system is given by

$$y_n(t) = \sum_{m=1}^{M}\sum_{l=0}^{L-1} w_{nml}x_m(t-l) \tag{52.14}$$

or in matrix form

$$\boldsymbol{y}(t) = \sum_{l=0}^{L-1} \boldsymbol{W}_l\boldsymbol{x}(t-l)\,. \tag{52.15}$$

As with the mixing process, the separation system can be expressed in the z-domain as

$$\boldsymbol{Y}(z) = \boldsymbol{W}(z)\boldsymbol{X}(z)\,, \tag{52.16}$$

and can also be expressed in block-Toeplitz form with the corresponding definitions for $\hat{\boldsymbol{y}}(t)$ and $\hat{\boldsymbol{W}}$ [52.25]:

$$\hat{\boldsymbol{y}}(t) = \hat{\boldsymbol{W}}\hat{\boldsymbol{x}}(t)\,. \tag{52.17}$$

Table 52.1 summarizes the mixing and separation equations in the different domains.

52.2.2 Relation Between Source and Separated Signals

The goal in source separation is not necessarily to recover identical copies of the original sources. Instead, the aim is to recover model sources without interferences from other sources, i. e., each separated signal $y_n(t)$ should contain signals originating from a single source only (Fig. 52.3). Therefore, each model source signal can be a filtered version of the original source signals, i. e.,

$$\boldsymbol{Y}(z) = \boldsymbol{W}(z)\boldsymbol{A}(z)\boldsymbol{S}(z) = \boldsymbol{G}(z)\boldsymbol{S}(z)\,, \tag{52.18}$$

Amazon EX1010 Page 1064

Table 52.1 The convolutive mixing equation and its corresponding separation equation for different domains in which blind source separation algorithms have been derived

	Mixing process	Separation model
Time	$x_m(t)=\sum_{n=1}^{N}\sum_{k=0}^{K-1} a_{mnk}s_n(t-k)+v_m(t)$ $\boldsymbol{x}(t)=\sum_{k=0}^{K-1}\boldsymbol{A}_k\boldsymbol{s}(t-k)+\boldsymbol{v}(t)$	$y_n(t)=\sum_{m=1}^{M}\sum_{l=0}^{L-1} w_{nml}x_m(t-l)$ $\boldsymbol{y}(t)=\sum_{l=0}^{L-1}\boldsymbol{W}_l\boldsymbol{x}(t-l)$
z-domain	$\boldsymbol{X}(z)=\boldsymbol{A}(z)\boldsymbol{S}(z)+\boldsymbol{V}(z)$	$\boldsymbol{Y}(z)=\boldsymbol{W}(z)\boldsymbol{X}(z)$
Frequency domain	$\boldsymbol{X}(\omega)=\boldsymbol{A}(\omega)\boldsymbol{S}(\omega)+\boldsymbol{V}(\omega)$	$\boldsymbol{Y}(\omega)=\boldsymbol{W}(\omega)\boldsymbol{X}(\omega)$
Block-Toeplitz form	$\hat{\boldsymbol{x}}(t)=\hat{\boldsymbol{A}}\hat{\boldsymbol{s}}(t)$	$\hat{\boldsymbol{y}}(t)=\hat{\boldsymbol{W}}\hat{\boldsymbol{x}}(t)$

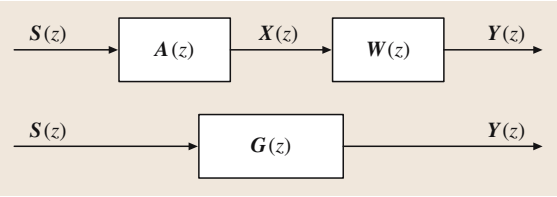


Fig. 52.2 The source signals $\boldsymbol{Y}(z)$ are mixed with the mixing filter $\boldsymbol{A}(z)$. An estimate of the source signals is obtained through an unmixing process, where the received signals $\boldsymbol{X}(z)$ are unmixed with the filter $\boldsymbol{W}(z)$. Each estimated source signal is then a filtered version of the original source, i. e., $\boldsymbol{G}(z)=\boldsymbol{W}(z)\boldsymbol{A}(z)$. Note that the mixing and the unmixing filters do not necessarily have to be of the same order

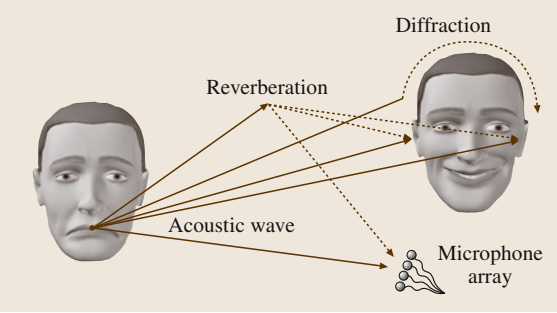


Fig. 52.3 Illustration of a speech source. It is not always clear what the desired acoustic source should be. It could be the acoustic wave as emitted from the mouth. This corresponds to the signal as it would have been recorded in an anechoic chamber in the absence of reverberations. It could be the individual source as it is picked up by a microphone array, or it could be the speech signal as it is recorded on microphones close to the two eardrums of a person. Due to reverberations and diffraction, the recorded speech signal is most likely a filtered version of the signal at the mouth

as illustrated in Fig. 52.2. The criterion for separation, i. e., interference-free signals, is satisfied if the recovered signals are permuted, and possibly scaled and filtered versions of the original signals, i. e.,

$$\boldsymbol{G}(z)=\mathbf{P}\boldsymbol{\Lambda}(z)\,, \tag{52.19}$$

where $\mathbf{P}$ is a permutation matrix, and $\boldsymbol{\Lambda}(z)$ is a diagonal matrix with scaling filters on its diagonal. If one can identify $\boldsymbol{A}(z)$ exactly, and choose $\boldsymbol{W}(z)$ to be its (stable) inverse, then $\boldsymbol{\Lambda}(z)$ is an identity matrix, and one recovers the sources exactly. In source separation, instead, one is satisfied with convolved versions of the sources, i. e., arbitrary diagonal $\boldsymbol{\Lambda}(z)$.

52.2.3 Feedback Structure

The mixing system given by (52.2) is called a feedforward system. Often such FIR filters are inverted by a feedback structure using IIR filters. The estimated sources are then given by the following equation, where the number of sources equals the number of receivers:

$$y_n(t)=x_n(t)+\sum_{l=0}^{L-1}\sum_{m=1}^{M} u_{nml}y_m(t-l)\,, \tag{52.20}$$

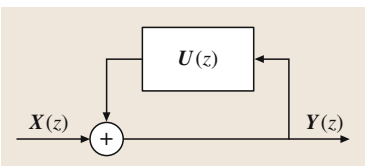


Fig. 52.4 Recurrent unmixing (feedback) network given by equation (52.21). The received signals are separated by an IIR filter to achieve an estimate of the source signals

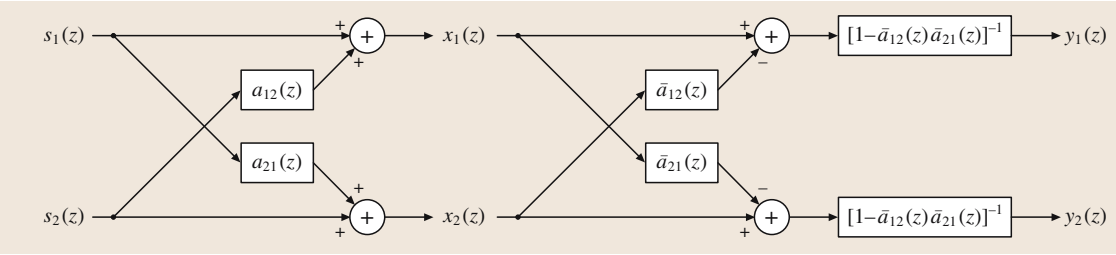


Fig. 52.5 The two mixed sources s_1 and s_2 are mixed by an FIR mixing system. The system can be inverted by an alternative system if the estimates $\bar{a}_{12}(z)$ and $\bar{a}_{21}(z)$ of the mixing filters $a_{12}(z)$ and $a_{12}(z)$ are known. Furthermore, if the filter $[1-\bar{a}_{12}(z)\bar{a}_{21}(z)]^{-1}$ is stable, the sources can be perfectly reconstructed as they were recorded at the microphones

and u_{nml} are the IIR filter coefficients. This can also be written in matrix form

$$\boldsymbol{y}(t) = \boldsymbol{x}(t) + \sum_{l=0}^{L-1} \boldsymbol{U}(l)\boldsymbol{y}(t-l)\,. \tag{52.21}$$

The architecture of such a network is shown in Fig. 52.4. In the z-domain, (52.21) can be written as [52.26]

$$\boldsymbol{Y}(z) = [\boldsymbol{I} + \boldsymbol{U}(z)]^{-1}\boldsymbol{X}(z)\,, \tag{52.22}$$

provided $[\boldsymbol{I} + \boldsymbol{U}(z)]^{-1}$ exists and all poles are within the unit circle. Therefore,

$$\boldsymbol{W}(z) = [\boldsymbol{I} + \boldsymbol{U}(z)]^{-1}\,. \tag{52.23}$$

The feedforward and the feedback network can be combined to a so-called hybrid network, where a feedforward structure is followed by a feedback network [52.27,28].

52.2.4 Example: The TITO System

A special case, which is often used in source separation work is the two-input-two-output (TITO) system [52.29]. It can be used to illustrate the relationship between the mixing and unmixing system, feedforward and feedback structures, and the difference between recovering sources versus generating separated signals.

Figure 52.5 shows a diagram of a TITO mixing and unmixing system. The signals recorded at the two microphones are described by the following equations:

$$x_1(z) = s_1(z) + a_{12}(z)s_2(z)\,, \tag{52.24}$$

$$x_2(z) = s_2(z) + a_{21}(z)s_1(z)\,. \tag{52.25}$$

The mixing system is thus given by

$$\boldsymbol{A}(z) = \begin{pmatrix} 1 & a_{12}(z) \\ a_{21}(z) & 1 \end{pmatrix}\,, \tag{52.26}$$

which has the following inverse

$$[\boldsymbol{A}(z)]^{-1} = \frac{1}{1 - a_{12}(z)a_{21}(z)} \begin{pmatrix} 1 & -a_{12}(z) \\ -a_{21}(z) & 1 \end{pmatrix}\,. \tag{52.27}$$

If the two mixing filters $a_{12}(z)$ and $a_{21}(z)$ can be identified or estimated as $\bar{a}_{12}(z)$ and $\bar{a}_{21}(z)$, the separation system can be implemented as

$$y_1(z) = x_1(z) - \bar{a}_{12}(z)x_2(z) \tag{52.28}$$

$$y_2(z) = x_2(z) - \bar{a}_{21}(z)x_1(z)\,. \tag{52.29}$$

A sufficient FIR separating filter is

$$\boldsymbol{W}(z) = \begin{pmatrix} 1 & -a_{12}(z) \\ -a_{21}(z) & 1 \end{pmatrix}\,. \tag{52.30}$$

However, the exact sources are not recovered until the model sources $\boldsymbol{y}(t)$ are filtered with the IIR filter $[1-\bar{a}_{12}(z)\bar{a}_{21}(z)]^{-1}$. Thus, the mixing process is invertible, provided that the inverse IIR filter is stable. If a filtered version of the separated signals is acceptable, we may disregard the potentially unstable recursive filter in (52.27) and limit separation to the FIR inversion of the mixing system with (52.30).

Amazon EX1010 Page 1066

52.3 Identification

Blind identification deals with the problem of estimating the coefficients in the mixing process $\boldsymbol{A}_k$. In general, this is an ill-posed problem, and no unique solution exists. In order to determine the conditions under which the system is blindly identifiable, assumptions about the mixing process and the input data are necessary. Even though the mixing parameters are known, this does not imply that the sources can be recovered. Blind identification of the sources refers to the exact recovery of sources. Therefore one should distinguish between the conditions required to identify the mixing system and the conditions necessary to identify the sources. The limitations for the exact recovery of sources when the mixing filters are known are discussed in [52.11,30,31]. For a recent review on identification of acoustic systems see [52.32]. This review considers systems with single and multiple inputs and outputs for the case of completely known sources as well as blind identification, where both the sources and the mixing channels are unknown.

52.4 Separation Principle

Blind source separation algorithms are based on different assumptions on the sources and the mixing system. In general, the sources are assumed to be *independent* or at least uncorrelated. The separation criteria can be divided into methods based on higher-order statistics (HOS), and methods based on second-order statistics (SOS). In convolutive separation it is also assumed that sensors receive N linearly independent versions of the sources. This means that the sources should originate from different locations in space (or at least emit signals into different orientations) and that there are at least as many sources as sensors for separation, i. e., $M \geq N$.

Instead of spatial diversity a series of algorithms make strong assumptions on the statistics of the sources. For instance, they may require that sources do not overlap in the time–frequency domain, utilizing therefore a form of *sparseness* in the data. Similarly, some algorithms for acoustic mixtures exploit regularity in the sources such as common onset, harmonic structure, etc. These methods are motivated by the present understanding on the grouping principles of auditory perception commonly referred to as *auditory scene analysis*. In radio communications a reasonable assumption on the sources is cyclo-stationarity or the fact that source signals take on only discrete values. By using such strong assumptions on the source statistics it is sometimes possible to relax the conditions on the number of sensors, e.g., $M < N$. The different criteria for separation are summarized in Table 52.2.

Table 52.2 Assumptions made for separation

$N < M$	$N = M$	$N > M$
• Subspace methods [52.25]	• Asymmetric sources by second- and third-order cumulants [52.33]	• Nonstationary, column-wise co-prime sources [52.34]
• Reduction of problem to instantaneous mixture [52.25, 38–43]	• Separation criteria based on SOS and HOS for 2×2 system [52.35]	• Cross-cumulants [52.36, 37]
	• Uncorrelated sources with distinct power spectra [52.44]	• Sparseness in time and frequency [52.45–47]
	• 2×2, temporally colored sources [52.48]	
	• Cumulants of order > 2, ML principle [52.49]	
	• Known cross filters [52.35]	
	• 2×2, each with different correlation [52.50, 51], extended to $M \times M$ in [52.52]	
	• Nonlinear odd functions [52.26, 53–58]	
	• Nonlinearity approximating the cumulative distribution function (CDF), see [52.59]	

52.4.1 Higher-Order Statistics

Source separation based on higher-order statistics is based on the assumption that the sources are statistically independent. Many algorithms are based on minimizing second and fourth order dependence between the model signals. A way to express independence is that all the cross-moments between the model sources are zero, i. e.,

$$E\left[y_n(t)^\alpha,\, y_{n'}(t-\tau)^\beta\right] = 0\,, \tag{52.31}$$

for all τ, $\alpha, \beta = \{1, 2, \dots\}$, and $n \neq n'$. Here $E[\cdot]$ denotes the statistical expectation. Successful separation using higher-order moments requires that the underlying sources are non-Gaussian (with the exception of at most one), since Gaussian sources have zero higher cumulants [52.60] and therefore equations (52.31) are trivially satisfied without providing useful conditions.

Fourth-Order Statistics

It is not necessary to minimize all cross-moments in order to achieve separation. Many algorithms are based on minimization of second- and fourth-order dependence between the model source signals. This minimization can either be based on second and fourth order cross-moments or second- and fourth-order cross-cumulants. Whereas off-diagonal elements of cross-cumulants vanish for independent signals the same is not true for all cross-moments [52.61]. Source separation based on cumulants has been used by several authors. Separation of convolutive mixtures by means of fourth-order cumulants has also been addressed [52.35, 61–71]. In [52.72–74], the joint approximate diagonalization of eigenmatrices (JADE) algorithm for complex-valued signals [52.75] was applied in the frequency domain in order to separate convolved source signals. Other cumulant-based algorithms in the frequency domain are given in [52.76, 77]. Second- and third-order cumulants have been used by Ye et al. [52.33] for separation of asymmetric signals. Other algorithms based on higher-order cumulants can be found in [52.78, 79]. For separation of more sources than sensors, cumulant-based approaches have been proposed in [52.70, 80]. Another popular fourth-order measure of non-Gaussianity is *kurtosis*. Separation of convolutive sources based on kurtosis has been addressed in [52.81–83].

Nonlinear Cross-Moments

Some algorithms apply higher-order statistics for separation of convolutive sources indirectly using nonlinear functions by requiring:

$$E[f(y_n(t)),\, g(y_{n'}(t-\tau))] = 0\,, \tag{52.32}$$

where $f(\cdot)$ and $g(\cdot)$ are odd, nonlinear functions. The Taylor expansion of these functions captures higher-order moments and this is found to be sufficient for separation of convolutive mixtures. This approach was among of the first for separation of convolutive mixtures [52.53] extending an instantaneous blind separation algorithm by *Herault* and *Jutten* (H–J) [52.84]. In *Back* and *Tsoi* [52.85], the H–J algorithm was applied in the frequency domain, and this approach was further developed in [52.86]. In the time domain, the approach of using nonlinear odd functions has been used by *Nguyen Thi* and *Jutten* [52.26]. They present a group of TITO (2×2) algorithms based on fourth-order cumulants, nonlinear odd functions, and second- and fourth-order cross-moments. This algorithm has been further examined by *Serviere* [52.54], and also been used by *Ypma* et al. [52.55]. In *Cruces* and *Castedo* [52.87] a separation algorithm can be found, which can be regarded as a generalization of previous results from [52.26, 88]. In *Li* and *Sejnowski* [52.89], the H–J algorithm has been used to determine the delays in a beamformer. The H–J algorithm has been investigated further by *Charkani* and *Deville* [52.57, 58, 90]. They extended the algorithm further to colored sources [52.56, 91]. Depending on the distribution of the source signals, optimal choices of nonlinear functions were also found. For these algorithms, the mixing process is assumed to be minimum-phase, since the H–J algorithm is implemented as a feedback network. A natural gradient algorithm based on the H–J network has been applied in *Choi* et al. [52.92]. A discussion of the H–J algorithm for convolutive mixtures can be found in *Berthommier* and *Choi* [52.93]. For separation of two speech signals with two microphones, the H–J model fails if the two speakers are located on the same side, as the appropriate separating filters can not be implemented without delaying one of the sources and the FIR filters are constrained to be causal. HOS independence obtained by applying antisymmetric nonlinear functions has also been used in [52.94, 95].

Information-Theoretic Methods

Statistical independence between the source signals can also be expressed in terms of the joint probability function (PDF). If the model sources $\boldsymbol{y}$ are independent, the joint PDF can be written as

$$p(\boldsymbol{y}) = \prod_n p(y_n)\,. \tag{52.33}$$

This is equivalent to stating that model sources y_n do not carry mutual information. Information-theoretic methods for source separation are based on maximizing the entropy in each variable. Maximum entropy is obtained when the sum of the entropy of each variable y_n equals the total joint entropy in $\boldsymbol{y}$. In this limit variables do not carry any mutual information and are hence mutually independent [52.96]. A well-known algorithm based on this idea is the Infomax algorithm by *Bell* and *Sejnowski* [52.97], which was significantly improved in convergence speed by the natural gradient method of *Amari* [52.98]. The Infomax algorithm can also be derived directly from model equation (52.33) using maximum likelihood [52.99], or equivalently, using the Kullback–Leibler divergence between the empirical distribution and the independence model [52.100].

In all instances it is necessary to assume, or model, the probability density functions $p_s(s_n)$ of the underlying sources s_n. In doing so, one captures higher-order statistics of the data. In fact, most information-theoretic algorithms contain expressions rather similar to the nonlinear cross-statistics in (52.32) with $f(y_n) = \partial \ln p_s(y_n)/\partial y_n$, and $g(y_n) = y_n$. The PDF is either assumed to have a specific form or it is estimated directly from the recorded data, leading to *parametric* and *nonparametric* methods, respectively [52.16]. In nonparametric methods the PDF is captured implicitly through the available data. Such methods have been addressed [52.101–103]. However, the vast majority of convolutive algorithms have been derived based on explicit parametric representations of the PDF.

Infomax, the most common parametric method, was extended to the case of convolutive mixtures by *Torkkola* [52.59] and later by *Xi* and *Reilly* [52.104,105]. Both feedforward and feedback networks were used. In the frequency domain it is necessary to define the PDF for complex variables. The resulting analytic nonlinear functions can be derived as [52.106, 107]

$$f(Y) = -\frac{\partial \ln p(|Y|)}{\partial |Y|} e^{j \arg(Y)} , \qquad (52.34)$$

where $p(Y)$ is the probability density of the model source $Y \in \mathbb{C}$. Some algorithms assume circular sources in the complex domain, while other algorithms have been proposed that specifically assume noncircular sources [52.108, 109].

The performance of the algorithm depends to a certain degree on the selected PDF. It is important to determine if the data has super-Gaussian or sub-Gaussian distributions. For speech commonly a Laplace distribution is used. The nonlinearity is also known as the Bussgang nonlinearity [52.110]. A connection between the Bussgang blind equalization algorithms and the Infomax algorithm is given in *Lambert* and *Bell* [52.111]. Multichannel blind deconvolution algorithms derived from the Bussgang approach can be found in [52.23, 111, 112]. These learning rules are similar to those derived in *Lee* et al. [52.113].

Choi et al. [52.114] have proposed a *nonholonomic* constraint for multichannel blind deconvolution. Nonholonomic means that there are some restrictions related to the direction of the update. The nonholonomic constraint has been applied for both a feedforward and a feedback network. The nonholonomic constraint was applied to allow the natural gradient algorithm by *Amari* et al. [52.98] to cope with overdetermined mixtures. The nonholonomic constraint has also been used in [52.115–122]. Some drawbacks in terms of stability and convergence in particular when there are large power fluctuations within each signal (e.g., for speech) have been addressed in [52.115].

Many algorithms have been derived from (52.33) directly using maximum likelihood (ML) [52.123]. The ML approach has been applied in [52.99, 124–132]. Closely related to the ML are the maximum a posteriori (MAP) methods. In MAP methods, prior information about the parameters of the model are taken into account. MAP has been used in [52.23, 133–141].

The convolutive blind source separation problem has also been expressed in a Bayesian formulation [52.142]. The advantage of a Bayesian formulation is that one can derive an optimal, possibly nonlinear, estimator of the sources enabling the estimation of more sources than the number of available sensors. The Bayesian framework has also been applied [52.135, 137, 143–145].

A strong prior on the signal can also be realized via hidden Markov models (HMMs). HMMs can incorporate state transition probabilities of different sounds [52.136]. A disadvantage of HMMs is that they require prior training and they carry a high computational cost [52.146]. HMMs have also been used in [52.147, 148].

52.4.2 Second-Order Statistics

In some cases, separation can be based on second-order statistics (SOS) by requiring only uncorrelated sources rather then the stronger condition of independence. Instead of assumptions on higher-order statistics these methods make alternate assumptions such as the nonstationarity of the sources [52.149], or a minimum-phase mixing system [52.50]. By itself, however, second-order

conditions are not sufficient for separation. Sufficient conditions for separation are given in [52.15, 150]. The main advantage of SOS is that they are less sensitive to noise and outliers [52.13], and hence require less data for their estimation [52.34, 50, 150–152]. The resulting algorithms are often also easier to implement and computationally efficient.

Minimum-Phase Mixing

Early work by *Gerven* and *Compernolle* [52.88] had shown that two source signals can be separated by decorrelation if the mixing system is minimum-phase. The FIR coupling filters have to be strictly causal and their inverses stable. The condition for stability is given as $|a_{12}(z)a_{21}(z)| < 1$, where $a_{12}(z)$ and $a_{21}(z)$ are the two coupling filters (Fig. 52.5). These conditions are not met if the mixing process is non-minimum-phase [52.153]. Algorithms based on second-order statistic assuming minimum-phase mixing can be found in [52.41, 42, 50–52, 154–158].

Nonstationarity

The fact that many signals are nonstationary has been successfully used for source separation. Speech signals in particular can be considered non-stationary on time scales beyond 10 ms [52.159, 160]). The temporally varying statistics of nonstationarity sources provides additional information for separation. Changing locations of the sources, on the other hand, generally complicate source separation as the mixing channel changes in time. Separation based on decorrelation of nonstationary signals was proposed by *Weinstein* et al. [52.29], who suggested that minimizing cross-powers estimated during different stationarity times should give sufficient conditions for separation. *Wu* and *Principe* proposed a corresponding joint diagonalization algorithm [52.103, 161] extending an earlier method developed for instantaneous mixtures [52.162]. *Kawamoto* et al. extend an earlier method [52.163] for instantaneous mixtures to the case of convolutive mixtures in the time domain [52.153, 164] and frequency domain [52.165]. This approach has also been employed in [52.166–169] and an adaptive algorithm was suggested by *Aichner* et al. [52.170]. By combining this approach with a constraint based on whiteness, the performance can be further improved [52.171].

Note that not all of these papers have used simultaneous decorrelation, yet, to provide sufficient second-order constraints it is necessary to minimize multiple cross-correlations simultaneously. An effective frequency-domain algorithm for simultaneous diagonalization was proposed by *Parra* and *Spence* [52.149]. Second-order statistics in the frequency domain is captured by the cross-power spectrum,

$$\boldsymbol{R}_{yy}(\omega, t) = E[\boldsymbol{Y}(\omega, t)\boldsymbol{Y}^{\mathrm{H}}(\omega, t)] \tag{52.35}$$

$$= \boldsymbol{W}(\omega)\boldsymbol{R}_{xx}(\omega, t)\boldsymbol{W}^{\mathrm{H}}(\omega)\,, \tag{52.36}$$

where the expectations are estimated around some time t. The goal is to minimize the cross-powers represented by the off-diagonal elements of this matrix, e.g., by minimizing:

$$J = \sum_{t,\omega} \|\boldsymbol{R}_{yy}(\omega, t) - \boldsymbol{\Lambda}_y(\omega, t)\|^2\,, \tag{52.37}$$

where $\boldsymbol{\Lambda}_y(\omega, t)$ is an estimate of the cross-power spectrum of the model sources and is assumed to be diagonal. This cost function simultaneously captures multiple times and multiple frequencies, and has to be minimized with respect to $\boldsymbol{W}(\omega)$ and $\boldsymbol{\Lambda}_y(\omega, t)$ subject to some normalization constraint. If the source signals are nonstationary the cross-powers estimated at different times t differ and provide independent conditions on the filters $\boldsymbol{W}(\omega)$. This algorithm has been successfully used on speech signals [52.172, 173] and investigated further by *Ikram* and *Morgan* [52.174–176] to determine the trade-offs between filter length, estimation accuracy, and stationarity times. Long filters are required to cope with long reverberation times of typical room acoustics, and increasing filter length also reduces problems associated with the circular convolution in (52.36) (see Sect. 52.5.3). However, long filters increase the number of parameters to be estimated and extend the effective window of time required for estimating cross-powers, thereby potentially losing the benefit of the nonstationarity of speech signals. A number of variations of this algorithm have been proposed subsequently, including time-domain implementations [52.177–179], and other methods that incorporate additional assumptions [52.174, 180–187]. A recursive version of the algorithm was given in *Ding* et al. [52.188]. In *Robeldo-Arnuncio* and *Juang* [52.189], a version with noncausal separation filters was suggested. Based on a different way to express (52.36), *Wang* et al. [52.148, 190–192] proposed a slightly different separation criterion that leads to a faster convergence than the original algorithm by *Parra* and *Spence* [52.149].

Other methods that exploit nonstationarity have been derived by extending the algorithm of *Molgedey* and *Schuster* [52.193] to the convolutive case [52.194, 195] including a common two-step approach of *sphering* and rotation [52.159, 196–199]. (Any matrix, for instance matrix $\mathbf{W}$, can be represented as a concatenation of a rotation with subsequent scaling, which can be used to

Amazon EX1010 Page 1070

remove second-order moments, i. e., sphering, and an additional rotation.)

In *Yin* and *Sommen* [52.160] a source separation algorithm was presented based on nonstationarity and a model of the direct path. The reverberant signal paths are considered as noise. A time-domain decorrelation algorithm based on different cross-correlations at different time lags is given in *Ahmed* et al. [52.200]. In *Yin* and *Sommen* [52.201] the cost function is based on minimization of the power spectral density between the source estimates. The model is simplified by assuming that the acoustic transfer function between the source and closely spaced microphones is similar. The simplified model requires fewer computations. An algorithm based on joint diagonalization is suggested in *Rahbar* and *Reilly* [52.152, 152]. This approach exploits the spectral correlation between the adjacent frequency bins in addition to nonstationarity. Also in [52.202, 203] a diagonalization criterion based on nonstationarity was used.

In *Olsson* and *Hansen* [52.138, 139] the nonstationary assumption has been included in a state-space Kalman filter model.

In *Buchner* et al. [52.204], an algorithm that uses a combination of non-stationarity, non-Gaussianity, and nonwhiteness has been suggested. This has also been applied in [52.205–207]. In the case of more source signals than sensors, an algorithm based on nonstationarity has also been suggested [52.70]. In this approach, it is possible to separate three signals: a mixture of two nonstationary source signals with short-time stationarity and one signal that is long-term stationary. Other algorithms based on the nonstationary assumptions can be found in [52.208–214].

Cyclo-Stationarity

If a signal is assumed to be cyclo-stationary, its cumulative distribution is invariant with respect to time shifts of some period T multiples thereof. Further, a signal is said to be wide-sense cyclo-stationary if the signals mean and autocorrelation is invariant to shifts of some period T [52.215], i. e.,

$$E[\boldsymbol{s}(t)] = E[\boldsymbol{s}(t+\alpha T)]\,, \tag{52.38}$$

$$E[\boldsymbol{s}(t_1), \boldsymbol{s}(t_2)] = E[\boldsymbol{s}(t_1+\alpha T), \boldsymbol{s}(t_2+\alpha T)]\,. \tag{52.39}$$

An example of a cyclo-stationary signal is a random-amplitude sinusoidal signal. Many communication signals have the property of cyclo-stationarity, and voiced speech is sometimes considered approximately cyclo-stationary [52.216]. This property has been used explicitly to recover mixed sources in, e.g., [52.34, 55, 118, 216–222]. In [52.220] cyclo-stationarity is used to solve the frequency permutation problem (see Sect. 52.5.1) and in [52.118] it is used as additional criteria to improve separation performance.

Nonwhiteness

Many natural signals, in particular acoustic signals, are temporally correlated. Capturing this property can be beneficial for separation. For instance, capturing temporal correlations of the signals can be used to reduce a convolutive problem to an instantaneous mixture problem, which is then solved using additional properties of the signal [52.25, 38–43]. In contrast to instantaneous separation where decorrelation may suffice for nonwhite signals, for convolutive separation additional conditions on the system or the sources are required. For instance, *Mei* and *Yin* [52.223] suggest that decorrelation is sufficient provided the sources are an ARMA process.

52.4.3 Sparseness in the Time/Frequency Domain

Numerous source separation applications are limited by the number of available microphones. It is in not always guaranteed that the number of sources is less than or equal to the number of sensors. With linear filters it is in general not possible to remove more than $M-1$ interfering sources from the signal. By using nonlinear techniques, in contrast, it may be possible to extract a larger number of source signals. One technique to separate more sources than sensors is based on sparseness. If the source signals do not overlap in the time–frequency (T–F) domain it is possible to separate them. A mask can be applied in the T–F domain to attenuate interfering signal energy while preserving T–F bins where the signal of interest is dominant. Often a binary mask is used giving perceptually satisfactory results even for partially overlapping sources [52.224, 225]. These methods work well for anechoic (delay-only) mixtures [52.226]. However, under reverberant conditions, the T–F representation of the signals is less sparse. In a mildly reverberant environment ($T_{60} \leq 200$ ms) underdetermined sources have been separated with a combination of independent component analysis (ICA) and T–F masking [52.47]. The first $N-M$ signals are removed from the mixtures by applying a T–F mask estimated from the direction of arrival of the signal (Sect. 52.6.1). The remaining M sources are separated by conventional BSS techniques. When a binary mask is applied to a signal, artifacts (musical noise) are often introduced. In order to reduce the musical noise, smooth masks have been proposed [52.47, 227].

Sparseness has also been used as a postprocessing step. In [52.77], a binary mask has been applied as post-processing to a standard BSS algorithm. The mask is determined by comparison of the magnitude of the outputs of the BSS algorithm. Hereby a higher signal-to-interference ratio is obtained. This method was further developed by *Pedersen* et al. in order to segregate underdetermined mixtures [52.228, 229]. Because the T–F mask can be applied to a single microphone signal, the segregated signals can be maintained as, e.g., in stereo signals.

Most T–F masking methods do not effectively utilize information from more than two microphones because the T–F masks are applied to a single microphone signal. However, some methods have been proposed that utilize information from more than two microphones [52.225, 230].

Clustering has also been used for sparse source separation [52.140, 141, 230–236]. If the sources are projected into a space where each source groups together, the source separation problem can be solved with clustering algorithms. In [52.45, 46] the mask is determined by clustering with respect to amplitude and delay differences.

In particular when extracting sources from single channels sparseness becomes an essential criterion. *Pearlmutter* and *Zador* [52.237] use strong prior information on the source statistic in addition to knowledge of the head-related transfer functions (HRTF). An a priori dictionary of the source signals as perceived through a HRTF makes it possible to separate source signals with only a single microphone. In [52.238], a priori knowledge is used to construct basis functions for each source signals to segregate different musical signals from their mixture. Similarly, in [52.239, 240] sparseness has been assumed in order to extract different music instruments.

Techniques based on sparseness are further discussed in the survey by *O'Grady* et al. [52.21].

52.4.4 Priors from Auditory Scene Analysis and Psychoacoustics

Some methods rely on insights gained from studies of the auditory system. The work by *Bergman* [52.241] on auditory scene analysis characterized the cues used by humans to segregate sound sources. This motivated computational algorithms that are referred to as computational auditory scene analysis (CASA). For instance, the phenomenon of auditory masking (the dominant perception of the signal with largest power) has motivated the use of T–F masking for many year [52.242]. In addition to the direct T–F masking methods outlined above, separated sources have been enhanced by filtering based on perceptual masking and auditory hearing thresholds [52.191, 243].

Another important perceptual cue that has been used in source separation is pitch frequency, which typically differs for simultaneous speakers [52.135, 137, 138, 147, 244, 245]. In *Tordini* and *Piazza* [52.135] pitch is extracted from the signals and used in a Bayesian framework. During unvoiced speech, which lacks a well-defined pitch they use an ordinary blind algorithm. In order to separate two signals with one microphone, *Gandhi* and *Hasegawa-Johnson* [52.137] proposed a state-space separation approach with strong *a priori* information. Both pitch and mel-frequency cepstral coefficients (MFCC) were used in their method. A pitch codebook as well as an MFCC codebook have to be known in advance. *Olsson* and *Hansen* [52.138] used an HMM, where the sequence of possible states is limited by the pitch frequency that is extracted in the process. As a preprocessing step to source separation, *Furukawa* et al. [52.245] use pitch in order to determine the number of source signals.

A method for separation of more sources than sensors is given in *Barros* et al. [52.244]. They combined ICA with CASA techniques such as pitch tracking and auditory filtering. Auditory filter banks are used in order to model the cochlea. In [52.244] wavelet filtering has been used for auditory filtering. Another commonly used auditory filter bank is the Gammatone filter-bank (see, e.g., *Patterson* [52.246] or [52.247, 248]). In *Roman* et al. [52.248] binaural cues have been used to segregate sound sources, whereby interaural time and interaural intensity differences (ITD and IID) have been used to group the source signals.

52.5 Time Versus Frequency Domain

The blind source separation problem can either be expressed in the time domain

$$\boldsymbol{y}(t) = \sum_{l=0}^{L-1} \boldsymbol{W}_l \boldsymbol{x}(t-l) \tag{52.40}$$

or in the frequency domain

$$\boldsymbol{Y}(\omega, t) = \boldsymbol{W}(\omega)\boldsymbol{X}(\omega, t)\,. \tag{52.41}$$

A survey of frequency-domain BSS is provided in [52.22]. In *Nishikawa* et al. [52.249] the advantages and disadvantages of the time and frequency-domain approaches are compared. This is summarized in Table 52.3.

An advantage of blind source separation in the frequency domain is that the separation problem can be decomposed into smaller problems for each frequency bin in addition to the significant gains in computational efficiency. The convolutive mixture problem is reduced to *instantaneous* mixtures for each frequency. Although this simplifies the task of convolutive separation a set of new problems arise: the frequency-domain signals obtained from the DFT are complex-valued. Not all instantaneous separation algorithms are designed for complex-valued signals. Consequently, it is necessary to modify existing algorithms correspondingly [52.5, 250–252]. Another problem that may arise in the frequency domain is that there are no longer enough data points available to evaluate statistical independence [52.131]. For some algorithms [52.149] the frame size T of the DFT has to be much longer than the length of the room impulse response K (Sect. 52.5.3). Long frames result in fewer data samples per frequency [52.131], which complicates the estimation of the independence criteria. A method that addresses this issue has been proposed by *Servière* [52.253].

52.5.1 Frequency Permutations

Another problem that arises in the frequency domain is the permutation and scaling ambiguity. If separation is treated for each frequency bin as a separate problem, the source signals in each bin may be estimated with an arbitrary permutation and scaling, i. e.,

$$\boldsymbol{Y}(\omega, t) = \boldsymbol{P}(\omega)\boldsymbol{\Lambda}(\omega)\boldsymbol{S}(\omega, t)\,. \tag{52.42}$$

If the permutation $\boldsymbol{P}(\omega)$ is not consistent across frequency then converting the signal back to the time domain will combine contributions from different sources into a single channel, and thus annihilate the separation achieved in the frequency domain. An overview of the solutions to this permutation problem is given in Sect. 52.6. The scaling indeterminacy at each frequency – the arbitrary solution for $\boldsymbol{\Lambda}(\omega)$ – will result in an overall filtering of the sources. Hence, even for perfect separation, the separated sources may have a different frequency spectrum than the original sources.

52.5.2 Time–Frequency Algorithms

Algorithms that define a separation criteria in the time domain do typically not exhibit frequency permutation problems, even when computations are executed in the frequency domain. A number of authors have therefore used time-domain (TD) criteria combined with frequency-domain implementations that speed up computations [52.101, 113, 121, 171, 179, 254–257]. However, note that second-order criteria may be sus-

Table 52.3 Advantages and disadvantages of separation in the time and frequency domain

Time domain		**Frequency domain**	
Advantages	Disadvantages	Advantages	Disadvantages
The independence assumption holds better for full-band signals	Degradation of convergence in a strongly reverberant environment	The convolutive mixture can be transformed into instantaneous mixture problems for each frequency bin	For each frequency band, there is a permutation and a scaling ambiguity which needs to be solved
Possible high convergence near the optimal point	Many parameters need to be adjusted for each iteration step	Due to the FFT, computations are saved compared to an implementation in the time domain	Problem with too few samples in each frequency band may cause the independence assumption to fail
		Convergence is faster	Circular convolution deteriorates the separation performance.
			Inversion of $\boldsymbol{A}$ is not guaranteed

ceptible to the permutation problem even if they are formulated in the time domain [52.184].

52.5.3 Circularity Problem

When the convolutive mixture in the time domain is expressed in the frequency domain by the DFT, the convolution becomes separate multiplications, i. e.,

$$\boldsymbol{x}(t) = \boldsymbol{A} * \boldsymbol{s}(t) \leftrightarrow \boldsymbol{X}(\omega, t) \approx \boldsymbol{A}(\omega)\boldsymbol{S}(\omega, t)\,. \quad (52.43)$$

However, this is only an approximation which is exact only for periodic $\boldsymbol{s}(t)$ with period T, or equivalently, if the time convolution is *circular*:

$$\boldsymbol{x}(t) = \boldsymbol{A} \circledast \boldsymbol{s}(t) \longleftrightarrow \boldsymbol{X}(\omega) = \boldsymbol{A}(\omega)\boldsymbol{S}(\omega)\,. \quad (52.44)$$

For a *linear convolution* errors occur at the frame boundary, which are conventionally corrected with the overlap-save method. However, a correct overlap-save algorithm is difficult to implement when computing cross-powers such as in (52.36) and typically the approximate expression (52.43) is assumed.

The problem of linear/circular convolution has been addressed by several authors [52.62, 121, 149, 171, 258]. *Parra* and *Spence* [52.149] note that the frequency-domain approximation is satisfactory provided that the DFT length T is significantly larger than the length of the un-mixing channels. In order to reduce the errors due to the circular convolution, the filters should be at least two times the length of the un-mixing filters [52.131, 176].

To handle long impulse responses in the frequency domain, a frequency model which is equivalent to the time-domain linear convolution has been proposed in [52.253]. When the time-domain filter extends beyond the analysis window the frequency response is undersampled [52.22, 258]. These errors can be mitigated by spectral smoothing or equivalently by windowing in the time domain. According to [52.259] the circularity problem becomes more severe when the number of sources increases.

Time-domain algorithms are often derived using Toeplitz matrices. In order to decrease the complexity and improve computational speed, some calculations involving Toeplitz matrices are performed using the fast Fourier transform. For that purpose, it is necessary to express the Toeplitz matrices in circulant Toeplitz form [52.23, 121, 171, 195, 260, 261]. A method that avoids the circularity effects but maintains the computational efficiency of the FFT has been presented in [52.262]. Further discussion on the circularity problem can be found in [52.189].

52.5.4 Subband Filtering

Instead of the conventional linear Fourier domain some authors have used subband processing. In [52.142] a long time-domain filter is replaced by a set of short independent subband filters, which results in faster convergence as compared to the full-band methods [52.214]. Different filter lengths for each subband filter have also been proposed, motivated by the varying reverberation time of different frequencies (typically low frequencies have a longer reverberation time) [52.263].

52.6 The Permutation Ambiguity

The majority of algorithms operate in the frequency domain due to the gains in computational efficiency, which are important in particular for acoustic mixtures that require long filters. However, in frequency-domain algorithms the challenge is to solve the permutation ambiguity, i. e., to make the permutation matrix $\boldsymbol{P}(\omega)$ independent of frequency. Especially when the number of sources and sensors is large, recovering consistent permutations is a severe problem. With N model sources there are $N!$ possible permutations in each frequency bin. Many frequency-domain algorithms provide ad hoc solutions, which solve the permutation ambiguity only partially, thus requiring a combination of different methods. Table 52.4 summarizes different approaches. They can be grouped into two categories:

1. Consistency of the filter coefficients
2. Consistency of the spectrum of the recovered signals

The first exploits prior knowledge about the mixing filters, and the second uses prior knowledge about the sources. Within each group the methods differ in the way consistency across frequency is established, varying sometimes in the metric they use to measure *distance* between solutions at different frequencies.

52.6.1 Consistency of the Filter Coefficients

Different methods have been used to establish consistency of filter coefficients across frequency, such as

Amazon EX1010 Page 1074

Table 52.4 Categorization of approaches to solve the permutation problem in the frequency domain

Class	Metric	Reference
Consistency of the filter coefficients	Smooth spectrum	[52.149, 264]
	Source locations	[52.265]
	Directivity pattern	[52.73, 175, 266]
	Location vectors	[52.267]
	DOA	[52.72, 184, 268]
	Adjacent matrix distance	[52.269]
	Invariances	[52.48]
	Split spectrum	[52.270]
	Frequency link in update process	[52.127]
	Initialization	[52.250, 271]
	Moving sources	[52.167]
	Vision	[52.148]
Consistency of the spectrum of the recovered signals	Amplitude modulation	[52.126, 159, 197, 203, 272]
	Pitch	[52.135, 147]
	Psychoacoustics	[52.243, 243]
	Fundamental frequency	[52.244]
	Cyclo-stationarity	[52.273]
	Periodic signals	[52.221]
	Cross-correlation	[52.62, 209, 274]
	Cross-cumulants	[52.275]
	Kurtosis	[52.86]
	Source distribution	[52.134, 276]
	Multidimensional prior	[52.277, 278]
	Clustering	[52.230, 279]
Time-frequency information	FIR polynomial	[52.23, 113, 254, 255]
	TD cost function	[52.178]
	Apply ICA to whole spectrogram	[52.280]
Combined approaches		[52.106, 258, 281, 282]

constraints on the length of the filters, geometric information, or consistent initialization of the filter weights.

Consistency across frequency can be achieved by requiring continuity of filter values in the frequency domain. One may do this directly by comparing the filter values of neighboring frequencies after adaptation, and pick the permutation that minimize the Euclidean distance between neighboring frequencies [52.74, 269]. Continuity (in a discrete frequency domain) is also expressed as smoothness, which is equivalent with a limited temporal support of the filters in the time domain. The simplest way to implement such a smoothness constraint is by zero-padding the time-domain filters prior to performing the frequency transformation [52.264]. Equivalently, one can restrict the frequency-domain updates to have a limited support in the time domain. This method is explained in *Parra* et al. [52.149] and has been used extensively [52.119, 122, 161, 174, 188, 190, 192, 201, 269, 283]. *Ikram* and *Morgan* [52.174, 176] evaluated this constraint and point out that there is a trade-off between the permutation alignment and the spectral resolution of the filters. Moreover, restricting the filter length may be problematic in reverberant environments where long separation filters are required. As a solution they have suggest to relax the constraint on filter length after the algorithm converges to satisfactory solutions [52.176].

Another suggestion is to assess continuity after accounting for the arbitrary scaling ambiguity. To do so, the separation matrix can be normalized as proposed in [52.265]:

$$\boldsymbol{W}(\omega) = \widetilde{\boldsymbol{W}}(\omega)\boldsymbol{\Lambda}(\omega)\,, \tag{52.45}$$

where $\boldsymbol{\Lambda}(\omega)$ is a diagonal matrix and $\widetilde{\boldsymbol{W}}(\omega)$ is a matrix with unit diagonal. The elements of $\widetilde{\boldsymbol{W}}(\omega)$, $\widetilde{W}_{mn}(\omega)$ are the ratios between the filters and these are used to assess continuity across frequencies [52.48, 220].

Instead of restricting the *unmixing* filters, *Pham* et al. [52.202] have suggested to require continuity in the *mixing* filters, which is reasonable as the mixing process will typically have a shorter time constant. A specific distance measure has been proposed by *Asano* et al. [52.267, 284]. They suggest to use the cosine between the filter coefficients of different frequencies ω_1 and ω_2:

$$\cos\alpha_n = \frac{\boldsymbol{a}_n^{\mathrm{H}}(\omega_1)\boldsymbol{a}_n(\omega_2)}{\left\|\boldsymbol{a}_n^{H}(\omega_1)\right\|\left\|\boldsymbol{a}_n(\omega_2)\right\|}\,, \tag{52.46}$$

where $\boldsymbol{a}_n(\omega)$ is the n-th column vector of $\boldsymbol{A}(\omega)$, which is estimated as the pseudo-inverse of $\boldsymbol{W}(\omega)$. Measuring distance in the space of separation filters rather than mixing filters was also suggested because these may better reflect the spacial configuration of the sources [52.285].

In fact, continuity across frequencies may also be assessed in terms of the estimated spatial locations of sources. Recall that the mixing filters are impulse responses between the source locations and the microphone locations. Therefore, the parameters of the separation filters should account for the position of the source in space. Hence, if information about the sensor location is available it can be used to address the permutation problem.

To understand this, consider the signal that arrives at an array of sensors. Assuming a distant source in an reverberation-free environment the signal approximates a plane wave. If the plane wave arrives at an angle to the microphone array it will impinge on each microphone with a certain delay (Fig. 52.6). This delay τ is given by the microphone distance d, the velocity of the wave c, and the direction-of-arrival (DOA) angle θ:

$$\tau = \frac{d}{c}\sin\theta\,, \tag{52.47}$$

Filters that compensate for this delay can add the microphone signals constructively (or destructively) to produce a maximum (or minimum) response in the DOA. Hence, the precise delay in filters (which in the frequency domain correspond to precise phase relationships) establishes a relationship between different frequencies that can be used to identify correct permutations. This was first considered by *Soon* et al. [52.286].

To be specific, each row in the separation matrix $\mathbf{W}(\omega)$ defines a *directivity pattern*, and therefore each row can be thought of as a separate beamformer. This

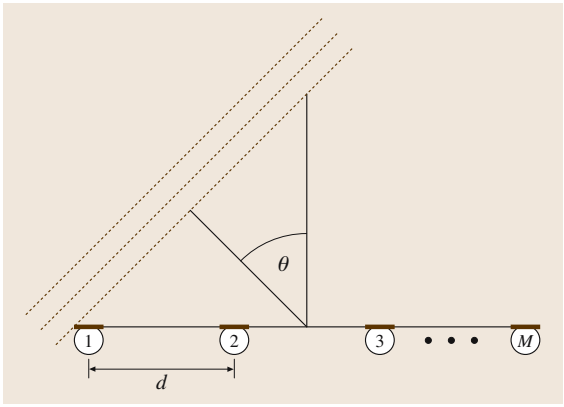


Fig. 52.6 Linear array with M sensors separated by distance d. The sensors are placed in a free field. A source signal is considered coming from a point source a distance r away from the sensor array. The source signal is placed in the far field, i. e., $r \gg d$. Therefore the incident wave is planar and the arrival angle θ is the same for all the sensors

directivity pattern is determined by the transfer function between the source and the filter output. The magnitude response of the n-th output is given by

$$\boldsymbol{r}_n(\omega,\boldsymbol{\theta}) = \left|\boldsymbol{w}_n^{\mathrm{H}}(\omega)\boldsymbol{a}(\omega,\boldsymbol{\theta})\right|^2\,, \tag{52.48}$$

where $\boldsymbol{a}(\omega)$ is an $M\times 1$ vector representing the propagation of a distant source with DOA θ to the sensor array. When M sensors are available, it is possible to place $M-1$ nulls in each of the M *directivity patterns*, i. e., directions from which the arriving signal is canceled out. In an ideal, reverberation-free environment separation is achieved if these nulls point to the directions of the interfering sources. The locations of these nulls, as they may be identified by the separation algorithm, can be used to resolve the permutation ambiguity [52.77, 81, 131, 266, 287–290]. These techniques draw strong parallels between source separation solutions and *beamforming*. The DOAs do not have to be known in advance and can instead be extracted from the resulting separation filters. Note, however, that the ability to identify source locations is limited by the physics of wave propagation and sampling: distant microphones will lead to grading lobes which will confuse the source locations, while small aperture limits spatial resolution at low frequencies. *Ikram* and *Morgan* [52.175] extend the idea of *Kurita* et al. [52.266] to the case where the sensor space is wider than one half of the wavelength. Source locations are estimated at lower frequencies, which do not exhibit grating lobes. These estimates are then used to determine the correct nulls for the higher fre-

Amazon EX1010 Page 1076

quencies and hereby the correct permutations. In order to resolve permutations when sources arrive from the same direction, *Mukai* et al. [52.291] use a near-field model. *Mitianoudis* and *Davies* [52.268] suggested frequency alignment based on DOA estimated with the multiple signal classification (MuSIC) algorithm [52.292]. A subspace method has been used in order to avoid constraints on the number of sensors. *Knaak* et al. [52.222] include DOA information as a part of the BSS algorithm in order to avoid permutation errors. Although all these methods assume a reverberation-free environment they give reasonable results in reverberant environments as long as the source has a strong direct path to the sensors.

Two other methods also utilize geometry. In the case of moving sources, where only one source is moving, the permutation can be resolved by noting that only one of the parameters in the separation matrix changes [52.167]. If visual cues are available, they may also be used to solve the permutation ambiguity [52.148].

Instead of using geometric information as a separate step to solve the permutation problem *Parra* and *Alvino* include geometric information directly into the cost function [52.184, 185]. This approach has been applied to microphone arrays under reverberant conditions [52.187]. *Baumann* et al. [52.72] have also suggested a cost function, which includes the DOA estimation. The arrival angles of the signals are found iteratively and are included in the separation criterion. *Baumann* et al. [52.73] also suggest a maximum-likelihood approach to solve the permutation problem. Given the probability of a permuted or unpermuted solution as function of the estimated zero directions, the most likely permutation is found.

Gotanda et al. [52.270] proposed a method to reduce the permutation problem based on the split spectral difference, and the assumption that each source is closer to one microphone. The split spectrum is obtained when each of the separated signals are filtered by the estimated mixing channels.

Finally, for iterative update algorithms a proper initialization of the separation filters can result in consistent permutations across frequencies. *Smaragdis* [52.250] proposed to estimate filter values sequentially starting with low frequencies and initializing filter values with the results of the previous lower frequency. This will tend to select solutions with filters that are smooth in the frequency domain, or equivalently, filters that are short in the time domain. Filter values may also be initialized to simple beamforming filters that point to estimated source locations. The separation algorithm will then tend to converge to solutions with the same target source across all frequencies [52.184, 271].

52.6.2 Consistency of the Spectrum of the Recovered Signals

Some solutions to the permutation ambiguity are based on the properties of speech. Speech signals have strong correlations across frequency due to a common amplitude modulation.

At the coarsest level the power envelope of the speech signal changes depending on whether there is speech or silence, and within speech segments the power of the carrier signal induces correlations among the amplitude of different frequencies. A similar argument can be made for other natural sounds. Thus, it is fair to assume that natural acoustic signals originating from the same source have a correlated amplitude envelope for neighboring frequencies. A method based on this comodulation property was proposed by *Murata* et al. [52.159, 196]. The permutations are sorted to maximize the correlation between different envelopes. This is illustrated in Fig. 52.7. This method has also been used in [52.198, 199, 203, 263, 287, 293]. *Rahbar* and *Reilly* [52.152, 209] suggest efficient methods for finding the correct permutations based on cross-frequency correlations.

Asano and *Ikeda* [52.294] report that the method sometimes fails if the envelopes of the different source signals are similar. They propose the following function to be maximized in order to estimate the permutation

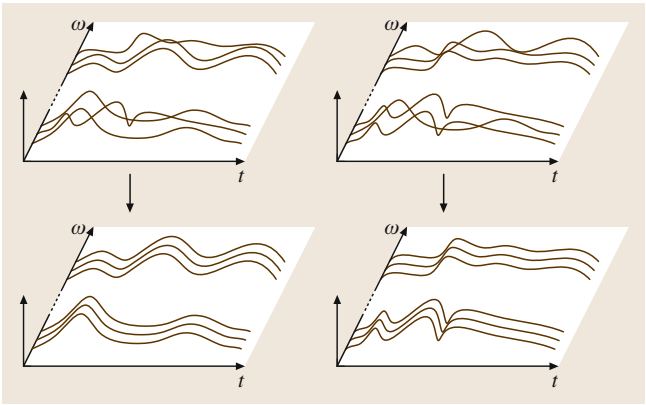


Fig. 52.7 For speech signals, it is possible to estimate the permutation matrix by using information on the envelope of the speech signal (amplitude modulation). Each speech signal has a particular envelope. Therefore, by comparison with the envelopes of the nearby frequencies, it is possible to order the permuted signals

matrix:

$$\hat{\boldsymbol{P}}(\omega) = \arg\max_{\boldsymbol{P}(\omega)} \sum_{t=1}^{T} \sum_{j=1}^{\omega-1} [\boldsymbol{P}(\omega)\bar{\boldsymbol{y}}(\omega, t)]^{\mathrm{H}} \bar{\boldsymbol{y}}(j, t) \,, \tag{52.49}$$

where $\bar{\boldsymbol{y}}$ is the power envelope of $\boldsymbol{y}$ and $\boldsymbol{P}(\omega)$ is the permutation matrix. This approach has also been adopted by *Peterson* and *Kadambe* [52.232]. *Kamata* et al. [52.282] report that the correlation between envelopes of different frequency channels may be small, if the frequencies are too far from each other. *Anemüller* and *Gramms* [52.127] avoid the permutations since the different frequencies are linked in the update process. This is done by serially switching from low to high-frequency components while updating.

Another solution based on amplitude correlation is the so-called amplitude modulation decorrelation (AMDecor) algorithm presented by *Anemüller* and *Kollmeier* [52.126, 272]. They propose to solve the source separation problem and the permutation problems simultaneously. An amplitude modulation correlation is defined, where the correlation between the frequency channels ω_k and ω_l of the two spectrograms $\boldsymbol{Y}_\mathrm{a}(\omega, t)$ and $\boldsymbol{Y}_\mathrm{b}(\omega, t)$ is calculated as

$$\begin{aligned} c(\boldsymbol{Y}_\mathrm{a}(\omega, t), \boldsymbol{Y}_\mathrm{b}(\omega, t)) = {} & E[|\boldsymbol{Y}_\mathrm{a}(\omega, t)||\boldsymbol{Y}_\mathrm{b}(\omega, t)|] \\ & - E[|\boldsymbol{Y}_\mathrm{a}(\omega, t)|]E[|\boldsymbol{Y}_\mathrm{b}(\omega, t)|] \,. \end{aligned} \tag{52.50}$$

This correlation can be computed for all combinations of frequencies. This results in a square matrix $\boldsymbol{C}(\boldsymbol{Y}_\mathrm{a}, \boldsymbol{Y}_\mathrm{b})$ with sizes equal to the number of frequencies in the spectrogram, whose k, l-th element is given by (52.50). Since the unmixed signals $\boldsymbol{y}(t)$ have to be independent, the following decorrelation property must be fulfilled

$$C_{kl}(\boldsymbol{Y}_\mathrm{a}, \boldsymbol{Y}_\mathrm{b}) = 0 \quad \forall a \neq b, \forall k, l. \tag{52.51}$$

This principle also solves the permutation ambiguity. The source separation algorithm is then based on the minimization of a cost function given by the Frobenius norm of the amplitude-modulation correlation matrix.

A priori knowledge about the source distributions has also been used to determine the correct permutations. Based on assumptions of Laplacian distributed sources, *Mitianopudis* and *Davies* [52.134, 251, 276] propose a likelihood ratio test to test which permutation is most likely. A time-dependent function that imposes frequency coupling between frequency bins is also introduced. Based on the same principle, the method has been extended to more than two sources by *Rahbar* and *Reilly* [52.152]. A hierarchical sorting is used in order to avoid errors introduced at a single frequency. This approach has been adopted in *Mertins* and *Russel* [52.212].

Finally, one of the most effective convolutive BSS methods to date (Table 52.5) uses the statistical relationship of signal powers across frequencies. Rather than solving separate *instantaneous* source separation problems in each frequency band *Lee* et al. [52.277,278,280] propose a multidimensional version of the density estimation algorithms. The density function captures the power of the entire model source rather than the power at individual frequencies. As a result, the joint statistics across frequencies are effectively captured and the algorithm converges to satisfactory permutations in each frequency bin.

Other properties of speech have also been suggested in order to solve the permutation indeterminacy. A *pitch*-based method has been suggested by *Tordini* and *Piazza* [52.135]. Also *Sanei* et al. [52.147] use the property of different pitch frequency for each speaker. The pitch and formants are modeled by a coupled HMM. The model is trained based on previous time frames.

Motivated by psychoacoustics, *Guddeti* and *Mulgrew* [52.243] suggest to disregard frequency bands that are perceptually masked by other frequency bands. This simplifies the permutation problem as the number of frequency bins that have to be considered is reduced. In *Barros* et al. [52.244], the permutation ambiguity is avoided due to a priori information of the phase associated with the fundamental frequency of the desired speech signal.

Nonspeech signals typically also have properties which can be exploited. Two proposals for solving the

Table 52.5 An overview of algorithms applied in real rooms, where the SIR improvement has been reported

Room size (approx.) [m]	T_{60} [ms]	N	M	SIR [dB]	Reference
6×3×3	300	2	2	13	[52.169,170][1]
6×3×3	300	2	2	8–10	[52.271][1]
6×3×3	300	2	2	12	[52.249]
6×3×3	300	2	2	5.7	[52.290]
6×3×3	300	2	2	18–20	[52.132,295][1]
	50	2	2	10	[52.207]
	250	2	2	16	[52.262]
6×6×3	200	2	2	< 16	[52.205][2]
6×6×3	150	2	2	< 15	[52.206]
6×6×3	150	2	2	< 20	[52.171]

Amazon EX1010 Page 1078

Table 52.5 (continued)

Room size (approx.) [m]	T_{60} [ms]	N	M	SIR [dB]	Reference
	500	2	2	6	[52.262]
4×4×3	130	3	2	4–12	[52.296]
4×4×3	130	3	2	14.3	[52.227]
4×4×3	130	3	2	< 12	[52.47]
4×4×3	130	2	2	7–15	[52.130]
4×4×3	130	2	2	4–15	[52.22,297][2]
4×4×3	130	2	2	12	[52.291]
4×4×3	130	6	8	18	[52.298]
4×4×3	130	4	4	12	[52.259]
	130	3	2	10	[52.140,141]
Office		2	2	5.5–7.6	[52.142]
6×5	130	2	8	1.6–7.0	[52.269]
8×7	300	2	2	4.2–6.0	[52.73]
15×10	300	2	2	5–8.0	[52.72]
		2	2	< 10	[52.57,91]
Office		2	2	6	[52.122]
Many rooms		2	2	3.1–27.4	[52.115]
Small room		2	2	4.7–9.5	[52.252]
4×3×2		2	2	< 10	[52.181]
4×3×2		2	2	14.4	[52.183]
4×3×2		2	2	4.6	[52.182]
		2	2	< 15	[52.245]
6×7	580	2	3	< 73	[52.31][3]
	810	2	2	< 10	[52.167][2]
Conf. room		4	4	14	[52.278]
	150	3	3	10	[52.222]
15×10×4	300	2	2	10	[52.77]
	360	2	2	5	[52.266]
5×5	200	2	2	6–21	[52.299]
	300	2	2–12	8–12	[52.300]
3×6		3	8	10	[52.184]
4×3×2		2	2	15	[52.149]
5×5×3		2	2	5	[52.187]
8×4	700	2	4	16	[52.152]
7×4×3	250	2	2	9.3	[52.253][1]
4×4	200	2	2	15	[52.301]
Office	500	3	2	4.3–10.6	[52.45]
	300	2	6	< 15	[52.213]

[1] Sources convolved with real impulse responses

[2] Moving sources

[3] This method is not really blind as it requires that sources are on one at a time

permutation in the case of cyclo-stationary signals can be found in *Antoni* et al. [52.273]. For machine acoustics, the permutations can be solved easily since machine signals are (quasi)periodic. This can be employed to find the right component in the output vector [52.221].

Continuity of the frequency spectra has been used by *Capdevielle* et al. [52.62] to solve the permutation ambiguity. The idea is to consider the sliding Fourier transform with a delay of one point. The cross-correlation between different sources are zero due to the independence assumption. Hence, when the cross-correlation is maximized, the output belongs to the same source. This method has also been used by *Servière* [52.253]. A disadvantage of this method is that it is computationally very expensive since the frequency spectrum has to be calculated with a window shift of one. A computationally less expensive method based on this principle has been suggested by *Dapena* and *Servière* [52.274]. The permutation is determined from the solution that maximizes the correlation between only two frequencies. If the sources have been whitened as part of separation, the approach by *Capdevielle* et al. [52.62] does not work. Instead, *Kopriva* et al. [52.86] suggest that the permutation can be solved by independence tests based on kurtosis. For the same reason, *Mejuto* et al. [52.275] consider fourth-order cross-cumulants of the outputs at all frequencies. If the extracted sources belong to the same sources, the cross-cumulants will be nonzero. Otherwise, if the sources belong to different sources, the cross-cumulants will be zero.

Finally, *Hoya* et al. [52.302] use pattern recognition to identify speech pauses that are common across frequencies, and in the case of overcomplete source separation, k-means clustering has been suggested. The clusters with the smallest variance are assumed to correspond to the desired sources [52.230]. *Dubnov* et al. [52.279] also address the case of more sources than sensors. Clustering is used at each frequency and Kalman tracking is performed in order to link the frequencies together.

52.6.3 Global Permutations

In many applications only one of the source signals is desired and the other sources are only considered as interfering noise. Even though the local (frequency) permutations are solved, the global (external) permutation problem still exists. Few algorithms address the problem of selecting the desired source signal from the available outputs. In some situations, it can be assumed that the desired signal arrives from a certain direction (e.g., the

speaker of interest is in front of the array). Geometric information can determine which of the signals is the target [52.171, 184]. In other situations, the desired speaker is selected as the most dominant speaker. In *Low* et al. [52.289], the most dominant speaker is determined on a criterion based on kurtosis. The speaker with the highest kurtosis is assumed to be the dominant. In separation techniques based on clustering, the desired source is assumed to be the cluster with the smallest variance [52.230]. If the sources are moving it is necessary to maintain the global permutation by tracking each source. For block-based algorithm the global permutation might change at block boundaries. This problem can often be solved by initializing the filter with the estimated filter from the previous block [52.186].

52.7 Results

The overwhelming majority of convolutive source separation algorithms have been evaluated on simulated data. In the process, a variety of simulated room responses have been used. Unfortunately, it is not clear whether any of these results transfer to real data. The main concerns are the sensitivity to microphone noise (often not better than −25 dB), nonlinearity in the sensors, and strong reverberations with a possibly weak direct path. It is suggestive that only a small subset of research teams evaluate their algorithms on actual recordings. We have considered more than 400 references and found results on real room recordings in only 10% of the papers. Table 52.5 shows a complete list of those papers. The results are reported as signal-to-interference ratio (SIR), which is typically averaged over multiple output channels. The resulting SIR are not directly comparable as the results for a given algorithm are very likely to depend on the recording equipment, the room that was used, and the SIR in the recorded mixtures. A state-of-the art algorithm can be expected to improve the SIR by 10–20 dB for two stationary sources. Typically a few seconds of data (2–10 s) will be sufficient to generate these results. However, from this survey nothing can be said about moving sources. Note that only eight (of over 400) papers reported separation of more than two sources, indicating that this remains a challenging problem.

52.8 Conclusion

We have presented a taxonomy for blind separation of convolutive mixtures with the purpose of providing a survey and discussion of existing methods. Further, we hope that this might stimulate the development of new models and algorithms which more efficiently incorporate specific domain knowledge and useful prior information.

In the title of the BSS review by *Torkkola* [52.13], it was asked: *Are we there yet?* Since then numerous algorithms have been proposed for blind separation of convolutive mixtures. Many convolutive algorithms have shown good performance when the mixing process is stationary, but still few methods work in real-world, time-varying environments. In these challenging environments, there are too many parameters to update in the separation filters, and too little data available in order to estimate the parameters reliably, while the less complicated methods such as null beamformers may perform just as well. This may indicate that the long demixing filters are not the solution for real-world, time-varying environments such as the cocktail-party situation.

References

52.1 A. Mansour, N. Benchekroun, C. Gervaise: Blind separation of underwater acoustic signals, ICA'06 (2006) pp. 181–188

52.2 S. Cruces-Alvarez, A. Cichocki, L. Castedo-Ribas: An iterative inversion approach to blind source separation, IEEE Trans. Neural Netw. **11**(6), 1423–1437 (2000)

52.3 K.I. Diamantaras, T. Papadimitriou: MIMO blind deconvoluition using subspace-based filter deflation, Proc. ICASSP'04, Vol. IV (2004) pp. 433–436

52.4 D. Nuzillard, A. Bijaoui: Blind source separation and analysis of multispectral astronomical images, Astron. Astrophys. Suppl. Ser. **147**, 129–138 (2000)

52.5 J. Anemüller, T.J. Sejnowski, S. Makeig: Complex independent component analysis of frequency-domain electroencephalographic data, Neural Netw. **16**(9), 1311–1323 (2003)

52.6 M. Dyrholm, S. Makeig, L.K. Hansen: Model structure selection in convolutive mixtures, ICA'06 (2006) pp. 74–81

52.7 C. Vayá, J.J. Rieta, C. Sánchez, D. Moratal: Performance study of convolutive BSS algorithms applied to the electrocardiogram of atrial fibrillation, ICA'06 (2006) pp. 495–502

52.8 L.K. Hansen: ICA of fMRI based on a convolutive mixture model, Ninth Annual Meeting of the Organization for Human Brain Mapping (2003)

52.9 E.C. Cherry: Some experiments on the recognition of speech, with one and two ears, J. Acoust. Soc. Am. **25**(5), 975–979 (1953)

52.10 S. Haykin, Z. Chen: The cocktail party problem, Neural Comput. **17**, 1875–1902 (2005)

52.11 M. Miyoshi, Y. Kaneda: Inverse filtering of room acoustics, IEEE Trans. Acoust. Speech. Signal Proc. **36**(2), 145–152 (1988)

52.12 K.J. Pope, R.E. Bogner: Blind signal separation II. Linear convolutive combinations, Digital Signal Process. **6**, 17–28 (1996)

52.13 K. Torkkola: Blind separation for audio signals – are we there yet?, ICA'99 (1999) pp. 239–244

52.14 K. Torkkola: Blind separation of delayed and convolved sources. In: *Unsupervised Adaptive Filtering, Blind Source Separation*, Vol. 1, ed. by S. Haykin (Wiley, New York 2000) pp. 321–375, Chap. 8

52.15 R. Liu, Y. Inouye, H. Luo: A system-theoretic foundation for blind signal separation of MIMO-FIR convolutive mixtures – a review, ICA'00 (2000) pp. 205–210

52.16 K.E. Hild: Blind Separation of Convolutive Mixtures Using Renyi's Divergence. Ph.D. Thesis (University of Florida, Gainesville 2003)

52.17 A. Hyvärinen, J. Karhunen, E. Oja: *Independent Component Analysis* (Wiley, New York 2001)

52.18 A. Cichocki, S. Amari: *Adaptive Blind Signal and Image Processing* (Wiley, New York 2002)

52.19 S.C. Douglas: Blind separation of acoustic signals. In: *Microphone Arrays*, ed. by M.S. Brandstein, D.B. Ward (Springer, Berlin, Heidelberg 2001) pp. 355–380, Chap. 16

52.20 S.C. Douglas: Blind signal separation and blind deconvolution. In: *The Neural Networks for Signal Processing Handbook*, Ser. Electrical Engineering and Applied Signal Processing, ed. by Y.H. Hu, J.-N. Hwang (CRC Press, Boca Raton 2002), Chap. 7

52.21 P.D. O'Grady, B.A. Pearlmutter, S.T. Rickard: Survey of sparse and non-sparse methods in source separation, IJIST **15**, 18–33 (2005)

52.22 S. Makino, H. Sawada, R. Mukai, S. Araki: Blind source separation of convolutive mixtures of speech in frequency domain, IEICE Trans. Fundamentals **E88-A**(7), 1640–1655 (2005)

52.23 R. Lambert: Multichannel Blind Deconvolution: FIR Matrix Algebra and Separation of Multipath Mixtures. Ph.D. Thesis (University of Southern California Department of Electrical Engineering, Los Angeles 1996)

52.24 S. Roberts, R. Everson: *Independent Components Analysis: Principles and Practice* (Cambridge University Press, Cambridge 2001)

52.25 A. Gorokhov, P. Loubaton: Subspace based techniques for second order blind separation of convolutive mixtures with temporally correlated sources, IEEE Trans. Circ. Syst. **44**(9), 813–820 (1997)

52.26 H.-L. Nguyen Thi, C. Jutten: Blind source separation for convolutive mixtures, Signal Process. **45**(2), 209–229 (1995)

52.27 S. Choi, A. Cichocki: Adaptive blind separation of speech signals: Cocktail party problem, ICSP'97 (1997) pp. 617–622

52.28 S. Choi, A. Cichocki: A hybrid learning approach to blind deconvolution of linear MIMO systems, Electron. Lett. **35**(17), 1429–1430 (1999)

52.29 E. Weinstein, M. Feder, A. Oppenheim: Multichannel signal separation by decorrelation, IEEE Trans. Speech Audio Process. **1**(4), 405–413 (1993)

52.30 S.T. Neely, J.B. Allen: Invertibility of a room impulse response, J. Acoust. Soc. Am. **66**(1), 165–169 (1979)

52.31 Y.A. Huang, J. Benesty, J. Chen: Blind channel identification-based two-stage approach to separation and dereverberation of speech signals in a reverberant environment, IEEE Trans. Speech Audio Process. **13**(5), 882–895 (2005)

52.32 Y. Huang, J. Benesty, J. Chen: Identification of acoustic MIMO systems: Challenges and opportunities, Signal Process. **86**, 1278–1295 (2006)

52.33 Z. Ye, C. Chang, C. Wang, J. Zhao, F.H.Y. Chan: Blind separation of convolutive mixtures based on second order and third order statistics, Proc. ICASSP'03, Vol. 5 (2003) pp. V–305, –308

52.34 K. Rahbar, J.P. Reilly, J.H. Manton: Blind identification of MIMO FIR systems driven by quasistationary sources using second-order statistics: A frequency domain approach, IEEE Trans. Signal Process. **52**(2), 406–417 (2004)

52.35 D. Yellin, E. Weinstein: Multichannel signal separation: Methods and analysis, IEEE Trans. Signal Process. **44**(1), 106–118 (1996)

52.36 B. Chen, A.P. Petropulu: Frequency domain blind MIMO system identification based on second- and higher order statistics, IEEE Trans. Signal Process. **49**(8), 1677–1688 (2001)

52.37 B. Chen, A.P. Petropulu, L.D. Lathauwer: Blind identification of complex convolutive MIMO systems with 3 sources and 2 sensors, Proc. ICASSP'02, Vol. II (2002) pp. 1669–1672

Amazon EX1010 Page 1081

52.38 A. Mansour, C. Jutten, P. Loubaton: Subspace method for blind separation of sources and for a convolutive mixture model, EUSIPCO 96, Signal Processing VIII, Theories and Applications (Elsevier, Amsterdam 1996) pp. 2081–2084

52.39 W. Hachem, F. Desbouvries, P. Loubaton: On the identification of certain noisy FIR convolutive mixtures, ICA'99 (1999)

52.40 A. Mansour, C. Jutten, P. Loubaton: Adaptive subspace algorithm for blind separation of independent sources in convolutive mixture, IEEE Trans. Signal Process. **48**(2), 583–586 (2000)

52.41 N. Delfosse, P. Loubaton: Adaptive blind separation of convolutive mixtures, Proc. ICASSP'96 (1996) pp. 2940–2943

52.42 N. Delfosse, P. Loubaton: Adaptive blind separation of independent sources: A second-order stable algorithm for the general case, IEEE Trans. Circ. Syst.–I: Fundamental Theory Appl. **47**(7), 1056–1071 (2000)

52.43 L.K. Hansen, M. Dyrholm: A prediction matrix approach to convolutive ICA, NNSP'03 (2003) pp. 249–258

52.44 Y. Hua, J.K. Tugnait: Blind identifiability of FIR-MIMO systems with colored input using second order statistics, IEEE Signal Process. Lett. **7**(12), 348–350 (2000)

52.45 O. Yilmaz, S. Rickard: Blind separation of speech mixtures via time-frequency masking, IEEE Trans. Signal Process. **52**(7), 1830–1847 (2004)

52.46 N. Roman: Auditory-based algorithms for sound segregatoion in multisource and reverberant environments. Ph.D. Thesis (The Ohio State University, Columbus 2005)

52.47 A. Blin, S. Araki, S. Makino: Underdetermined blind separation og convolutive mixtures of speech using time-frequency mask and mixing matrix estimation, IEICE Trans. Fundamentals **E88-A**(7), 1693–1700 (2005)

52.48 K.I. Diamantaras, A.P. Petropulu, B. Chen: Blind Two-Input-TwoOutput FIR Channel Identification based on frequency domain second- order statistics, IEEE Trans. Signal Process. **48**(2), 534–542 (2000)

52.49 E. Moulines, J.-F. Cardoso, E. Cassiat: Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models, Proc. ICASSP'97, Vol. 5 (1997) pp. 3617–3620

52.50 U.A. Lindgren, H. Broman: Source separation using a criterion based on second-order statistics, IEEE Trans. Signal Process. **46**(7), 1837–1850 (1998)

52.51 H. Broman, U. Lindgren, H. Sahlin, P. Stoica: Source separation: A TITO system identification approach, Signal Process. **73**(1), 169–183 (1999)

52.52 H. Sahlin, H. Broman: MIMO signal separation for FIR channels: A criterion and performance analysis, IEEE Trans. Signal Process. **48**(3), 642–649 (2000)

52.53 C. Jutten, L. Nguyen Thi, E. Dijkstra, E. Vittoz, J. Caelen: Blind separation of sources: An algorithm for separation of convolutive mixtures. In: *Higher Order Statistics. Proceedings of the International Signal Processing Workshop*, ed. by J. Lacoume (Elsevier, Amsterdam 1992) pp. 275–278

52.54 C. Servière: Blind source separation of convolutive mixtures, SSAP'96 (1996) pp. 316–319

52.55 A. Ypma, A. Leshem, R.P. Duina: Blind separation of rotating machine sources: Bilinear forms and convolutive mixtures, Neurocomputing **49**(1–4), 349–368 (2002)

52.56 N. Charkani, Y. Deville, J. Herault: Stability analysis and optimization of time-domain convolutive source separation algorithms, SPAWC'97 (1997) pp. 73–76

52.57 N. Charkani, Y. Deville: A convolutive source separation method with self-optimizing non-linearities, Proc. ICASSP'99, Vol. 5 (1999) pp. 2909–2912

52.58 N. Charkani, Y. Deville: Self-adaptive separation of convolutively mixed signals with a recursive structure. Part I: Stability analysis and optimization of asymptotic behaviour, Signal Process. **73**(3), 225–254 (1999)

52.59 K. Torkkola: Blind separation of convolved sources based on information maximization, NNSP'96 (1996) pp. 423–432

52.60 P. Comon: Independent component analysis, a new concept?, Signal Process. **36**(3), 287–314 (1994), special issue on Higher-Order Statistics

52.61 P. Comon, L. Rota: Blind separation of independent sources from convolutive mixtures, IEICE Trans. Fundamentals **E86-A**(3), 542–549 (2003)

52.62 V. Capdevielle, C. Servière, J.L. Lacoume: Blind separation of wide-band sources in the frequency domain, ICASSP'95, Vol. III (Detroit 1995) pp. 2080–2083

52.63 S. Icart, R. Gautier: Blind separation of convolutive mixtures using second and fourth order moments, Proc. ICASSP'96, Vol. 5 (1996) pp. 3018–3021

52.64 M. Girolami, C. Fyfe: A temporal model of linear anti-hebbian learning, Neural Process. Lett. **4**(3), 139–148 (1996)

52.65 J.K. Tugnait: On blind separation of convolutive mixtures of independent linear signals in unknown additive noise, IEEE Trans. Signal Process. **46**(11), 3117–3123 (1998)

52.66 C. Simon, P. Loubaton, C. Vignat, C. Jutten, G. d'Urso: Separation of a class of convolutive mixtures: A contrast function approach, Proc. ICASSP'99 (1999) pp. 1429–1432

52.67 Y. Su, L. He, R. Yang: An improved cumulant-based blind speech separation method, Proc. ICASSP'00 (2000) pp. 1867–1870

52.68 P. Baxter, J. McWhirter: Blind signal separation of convolutive mixtures, AsilomarSSC, Vol. 1 (2003) pp. 124–128

Amazon EX1010 Page 1082

52.69 S. Hornillo-Mellado, C.G. Puntonet, R. Martin-Clemente, M. Rodríguez-Álvarez: Characterization of the sources in convolutive mixtures: A cumulant-based approach, ICA'04 (2004) pp. 586–593

52.70 Y. Deville, M. Benali, F. Abrard: Differential source separation for underdetermined instantaneous or convolutive mixtures: Concept and algorithms, Signal Process. **84**(10), 1759–1776 (2004)

52.71 M. Ito, M. Kawamoto, N. Ohnishi, Y. Inouye: Eigenvector algorithms with reference signals for frequency domain BSS, ICA'06 (2006) pp. 123–131

52.72 W. Baumann, D. Kolossa, R. Orglmeister: Beamforming-based convolutive source separation, Proc. ICASSP'03, Vol. V (2003) pp. 357–360

52.73 W. Baumann, D. Kolossa, R. Orglmeister: Maximum likelihood permutation correction for convolutive source separation, ICA'03 (2003) pp. 373–378

52.74 M.S. Pedersen, C.M. Nielsen: Gradient flow convolutive blind source separation, MLSP'04 (2004) pp. 335–344

52.75 J.-F. Cardoso, A. Souloumiac: Blind beamforming for non Gaussian signals, IEEE Proc. **140**(6), 362–370 (1993)

52.76 D. Yellin, E. Weinstein: Criteria for multichannel signal separation, IEEE Trans. Signal Process. **42**(8), 2158–2168 (1994)

52.77 D. Kolossa, R. Orglmeister: Nonlinear postprocessing for blind speech separation, ICA'04 (2004) pp. 832–839

52.78 P. Comon, E. Moreau, L. Rota: Blind separation of convolutive mixtures: A contrast-based joint diagonalization approach, ICA'01 (2001) pp. 686–691

52.79 E. Moreau, J. Pesquet: Generalized contrasts for multichannel blind deconvolution of linear systems, IEEE Signal Process. Lett. **4**(6), 182–183 (1997)

52.80 Y. Li, J. Wang, A. Cichocki: Blind source extraction from convolutive mixtures in ill-conditioned multi-input multi-output channels, IEEE Trans. Circ. Syst. I: Regular Papers **51**(9), 1814–1822 (2004)

52.81 R.K. Prasad, H. Saruwatari, K. Shikano: Problems in blind separation of convolutive speech mixtures by negentropy maximization, IWAENC'03 (2003) pp. 287–290

52.82 X. Sun, S. Douglas: Adaptive paraunitary filter banks for contrast-based multichannel blind deconvolution, Proc. ICASSP'01, Vol. 5 (2001) pp. 2753–2756

52.83 J. Thomas, Y. Deville, S. Hosseini: Time-domain fast fixed-point algorithms for convolutive ICA, IEEE Signal Process. Lett. **13**(4), 228–231 (2006)

52.84 C. Jutten, J. Herault: Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture, Signal Process. **24**(1), 1–10 (1991)

52.85 A.D. Back, A.C. Tsoi: Blind deconvolution of signals using a complex recurrent network, NNSP'94 (1994) pp. 565–574

52.86 I. Kopriva, Ž. Devčić, H. Szu: An adaptive short-time frequency domain algorithm for blind separation of nonstationary convolved mixtures, IJCNN'01 (2001) pp. 424–429

52.87 S. Cruces, L. Castedo: A Gauss-Newton method for blind source separation of convolutive mixtures, Proc. ICASSP'98, Vol. IV (1998) pp. 2093–2096

52.88 S.V. Gerven, D.V. Compernolle: Signal separation by symmetric adaptive decorrelation: Stability, convergence, and uniqueness, IEEE Trans. Signal Process. **43**(7), 1602–1612 (1995)

52.89 S. Li, T.J. Sejnowski: Adaptive separation of mixed broadband sound sources with delays by a beamforming Herault-Jutten network, IEEE J. Ocean. Eng. **20**(1), 73–79 (1995)

52.90 N. Charkani, Y. Deville: Optimization of the asymptpotic preformance of time-domain convolutive source separation algorithms, ESANN'97 (1997) pp. 273–278

52.91 N. Charkani, Y. Deville: Self-adaptive separation of convolutively mixed signals with a recursive structure. Part II: Theoretical extensions and application to synthetic and real signals, Signal Process. **75**(2), 117–140 (1999)

52.92 S. Choi, H. Hong, H. Glotin, F. Berthommier: Multichannel signal separation for cocktail party speech recognition: A dynamic recurrent network, Neurocomputing **49**(1–4), 299–314 (2002)

52.93 F. Berthommier, S. Choi: Several improvements of the Hérault-Jutten model for speech segregation, ICA'03 (2003) pp. 1089–1094

52.94 M. Cohen, G. Cauwenbergh: Blind separation of linear convolutive mixtures through parallel stochastic optimization, ISCAS'98, Vol. 3 (1998) pp. 17–20

52.95 M. Stanacevic, M. Cohen, G. Cauwenberghs: Blind separation of linear convolutive mixtures using orthogonal filter banks, ICA'01 (2001) pp. 260–265

52.96 G. Deco, D. Obradovic: *An Information-Theoretic Approach to Neural Computing* (Springer, New York 1996)

52.97 A.J. Bell, T.J. Sejnowski: An information-maximization approach to blind separation and blind deconvolution, Neural Comput. **7**(6), 1129–1159 (1995)

52.98 S. Amari, S. Douglas, A. Cichocki, H.H. Yang: Multichannel blind deconvolution and equalization using the natural gradient, IEEE International Workshop on Wireless Communication (1997) pp. 101–104

52.99 B.A. Pearlmutter, L.C. Parra: Maximum likelihood blind source separation: A context-sensitive generalization of ICA, NIPS'97 (1997) pp. 613–619

Amazon EX1010 Page 1083

52.100 J.-F. Cardoso: Blind signal separation: Statistical principles, Proc. IEEE **9**(10), 2009–2025 (1998)

52.101 K. Kokkanikis, A.K. Nandi: Optimal blind separation of convolutive audio mixtures without temporal constraints, Proc. ICASSP'04, Vol. I (2004) pp. 217–220

52.102 K. Kokkinakis, A.K. Nandi: Multichannel speech separation using adaptive parameterization of source pdfs, ICA'04 (2004) pp. 486–493

52.103 H.-C. Wu, J.C. Principe: Generalized anti-hebbian learning for source separation, Proc. ICASSP'99, Vol. 2 (1999) pp. 1073–1076

52.104 J. Xi, J.P. Reilly: Blind separation and restoration of signals mixed in convolutive environment, Proc. ICASSP'97 (1997) pp. 1327–1330

52.105 J.P. Reilly, L.C. Mendoza: Blind signal separation for convolutive mixing environments using spatial-temporal processing, Proc. ICASSP'99 (1999) pp. 1437–1440

52.106 H. Sawada, R. Mukai, S. Araki, S. Makino: A robust and precise method for solving the permutation problem of frequency-domain blind source separation, IEEE Trans. Speech Audio Process. **12**(5), 530–538 (2004)

52.107 H. Sawada, R. Mukai, S. Araki, S. Makino: Polar coordinate based nonlinear function for frequency domain blind source separation, Proc. ICASSP'02 (2002) pp. 1001–1004

52.108 J.-F. Cardoso, T. Adali: The maximum likelihood approach to complex ICA, Proc. ICASSP, Vol. V (2006) pp. 673–676

52.109 M. Novey, T. Adali: Adaptable nonlinearity for complex maximization of nongaussianity and a fixed-point algorithm, MLSP (2006)

52.110 M. Joho, H. Mathis, G.S. Moschytz: An FFT-based algorithm for multichannel blind deconvolution, ISCAS'99, Vol. 3 (1999) pp. 203–206

52.111 R.H. Lambert, A.J. Bell: Blind separation of multiple speakers in a multipath environment, Proc. ICASSP'97, Vol. 1 (1997) pp. 423–426

52.112 R.H. Lambert: A new method for source separation, Proc. ICASSP'95, Vol. 3 (1995) pp. 2116–2119

52.113 T.-W. Lee, A.J. Bell, R. Orglmeister: Blind source separation of real world signals, ICNN'97 (1997) pp. 2129–2135

52.114 S. Choi, S. Amari, A. Cichocki, R. Liu: Natural gradient learning with a nonholonomic constraint for blind deconvolution of multiple channels, ICA'99 (1999) pp. 371–376

52.115 S.C. Douglas, X. Sun: Convolutive blind separation of speech mixtures using the natural gradient, Speech Commun. **39**, 65–78 (2003)

52.116 K. Matsuoka, Y. Ohba, Y. Toyota, S. Nakashima: Blind separation for convolutive mixture of many voices, IWAENC'03 (2003) pp. 279–282

52.117 K. Matsuoka, S. Nakashima: Minimal distortion principle for blind source separation, ICA'01 (2001) pp. 722–727

52.118 W. Wang, M.G. Jafari, S. Sanei, J.A. Chambers: Blind separartion of convolutive mixtures of cyclostationary signals, Int. J. Adapt. Contr. Signal Process. **18**, 279–298 (2004)

52.119 G.-J. Jang, C. Choi, Y. Lee, Y.-H. Oh: Adaptive cross-channel interference cancellation on blind signal separation outputs using source absence/presence detection and spectral subtraction, ICLSP'04, Vol. IV (2004) pp. 2865–2868

52.120 S.H. Nam, S. Beack: A frequency-domain normalized multichannel blind deconvolution algorithm for acoustical signals, ICA'04 (2004) pp. 524–531

52.121 M. Joho, P. Schniter: Frequency domain realization of a multichannel blind deconvolution algorithm based on the natural gradient, ICA'03 (2003) pp. 543–548

52.122 C. Choi, G. Jang, Y. Lee, S.R. Kim: Adaptive cross-channel interference cancellation on blind source separation outputs, ICA'04 (2004) pp. 857–864

52.123 L. Parra, C. Spence, B. de Vries: Convolutive source separation and signal modeling with ML, ISIS'97 (1997)

52.124 L.C. Parra: Temporal models in blind source separation, Lect. Notes Comput. Sci. **1387**, 229–247 (1998)

52.125 K. Yamamoto, F. Asano, W. van Rooijen, E. Ling, T. Yamada, N. Kitawaki: Estimation of the number of sound sources using support vector machines and its application to sound source separation, Proc. ICASSP'03, Vol. 5 (2003) pp. 485–488

52.126 J. Anemüller: Across-frequency processing in convolutive blind source separation. Ph.D. Thesis (Univ. Oldenburg, Oldenburg 2001)

52.127 J. Anemüller, T. Gramms: On-line blind separation of moving sound sources, ICA'99 (1999) pp. 331–334

52.128 S. Deligne, R. Gopinath: An EM algorithm for convolutive independent component analysis, Neurocomputing **49**, 187–211 (2002)

52.129 J. Rosca, C. Borss, R. Balan: Generalized sparse signal mixing model and application to noisy blind source separation, Proc. ICASSP'04, Vol. III (2004) pp. 877–880

52.130 S.C. Douglas, H. Sawada, S. Makino: Natural gradient multichannel blind deconvolution and speech separation using causal FIR filters, IEEE Trans. Speech Audio Process. **13**(1), 92–104 (2005)

52.131 S. Araki, R. Mukai, S. Makino, T. Nishikawa, H. Saruwatari: The fundamental limitation of frequency domain blind source separation for convolutive mixtures of speech, IEEE Trans. Speech Audio Process. **11**(2), 109–116 (2003)

52.132 S. Ukai, T. Takatani, T. Nishikawa, H. Saruwatari: Blind source separation combining SIMO-model-based ICA and adaptive beamforming, Proc. ICASSP'05, Vol. III (2005) pp. 85–88

Amazon EX1010 Page 1084

52.133 R.H. Lambert, C.L. Nikias: Polynomial matrix whitening and application to the multichannel blind deconvolution problem, MILCOM'95, Vol. 3 (1995) pp. 988–992

52.134 N. Mitianoudis, M. Davies: Audio source separation of convolutive mixtures, IEEE Trans. Speech Audio Process. **11**(5), 489–497 (2002)

52.135 F. Tordini, F. Piazza: A semi-blind approach to the separation of real world speech mixtures, IJCNN'02, Vol. 2 (2002) pp. 1293–1298

52.136 H. Attias: Source separation with a sensor array using graphical models and subband filtering, NIPS'02, Vol. 15 (2002) pp. 1229–1236

52.137 M.A. Gandhi, M.A. Hasegawa-Johnson: Source separation using particle filters, ICLSP'04, Vol. III (2004) pp. 2449–2452

52.138 R.K. Olsson, L.K. Hansen: A harmonic excitation state-space approach to blind separation of speech, NIPS (2004)

52.139 R.K. Olsson, L.K. Hansen: Probabilistic blind deconvolution of non-stationary sources, EUSIPCO'04 (2004) pp. 1697–1700

52.140 S. Winter, H. Sawada, S. Araki, S. Makino: Hierarchical clustering applied to overcomplete BSS for convolutive mixtures, SAPA'04 (2004)

52.141 S. Winter, H. Sawada, S. Araki, S. Makino: Overcomplete BSS for convolutive mixtures based on hierarchical clustering, ICA'04 (2004) pp. 652–660

52.142 H. Attias: New EM algorithms for source separation and deconvolution, Proc. ICASSP'03, Vol. V (2003) pp. 297–300

52.143 C. Andrieu, S. Godsill: A particle filter for model based audio source separation, ICA'00 (2000) pp. 381–386

52.144 J.R. Hopgood: Bayesian blind MIMO deconvolution of nonstationary subband autoregressive sources mixed through subband all-pole channels, SSP'03 (2003) pp. 422–425

52.145 S.J. Godsill, C. Andrieu: Bayesian separation and recovery of convolutively mixed autoregressive sources, Proc. ICASSP'99, Vol. III (1999) pp. 1733–1736

52.146 K. Abed-Meraim, W. Qiu, Y. Hua: Blind system identification, Proc. IEEE **85**(8), 1310–1322 (1997)

52.147 S. Sanei, W. Wang, J.A. Chambers: A coupled HMM for solving the permutation problem in frequency domain BSS, Proc. ICASSP'04, Vol. V (2004) pp. 565–568

52.148 W. Wang, D. Cosker, Y. Hicks, S. Sanei, J. Chambers: Video assisted speech source seperation, Proc. ICASSP'05, Vol. V (2005) pp. 425–428

52.149 L. Parra, C. Spence: Convolutive blind separation of non-stationary sources, IEEE Trans. Speech Audio Process. **8**(3), 320–327 (2000)

52.150 D.W.E. Schobben, P.C.W. Sommen: On the indeterminacies of convolutive blind signal separation based on second-order statistics, ISSPA'99 (1999) pp. 215–218

52.151 J. Liang, Z. Ding: Blind MIMO system identification based on cumulant subspace decomposition, IEEE Trans. Signal Process. **51**(6), 1457–1468 (2003)

52.152 K. Rahbar, J.P. Reilly: A frequency domain method for blind source separation of convolutive audio mixtures, IEEE Trans. Speech Audio Process. **13**(5), 832–844 (2005)

52.153 M. Kawamoto, K. Matsuoka, N. Ohnishi: A method of blind separation for convolved non-stationary signals, Neurocomputing **22**(1–3), 157–171 (1998)

52.154 A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, E. Moulines: A blind source separation technique using second-order statistics, IEEE Trans. Signal Process. **45**(2), 434–444 (1997)

52.155 A.G. Lindgren, T.P. von Hoff, A.N. Kaelin: Stability and performance of adaptive algorithms for multichannel blind source separation and deconvolution, EUSIPCO'00, Vol. 2 (2000) pp. 861–864

52.156 H. Sahlin, H. Broman: Separation of real-world signals, Signal Process. **64**, 103–113 (1998)

52.157 D.C.B. Chan, P.J.W. Rayner, S.J. Godsill: Multichannel blind signal separation, Proc. ICASSP'96 (1996) pp. 649–652

52.158 C. Simon, C. Vignat, P. Loubaton, C. Jutten, G. d'Urso: On the convolutive mixture – source separation by the decorrelation approach, Proc. ICASSP'98, Vol. 4 (1998) pp. 2109–2112

52.159 N. Murata, S. Ikeda, A. Ziehe: *An Approach to Blind Source Separation Based on Temporal Structure of Speech Signals* (RIKEN Brain Science Institute, Japan 1998), BSIS Technical Reports 98-2

52.160 B. Yin, P. Sommen: Adaptive blind signal separation using a new simplified mixing model, ProRISC'99, ed. by J. Veen (1999) pp. 601–606

52.161 H.-C. Wu, J.C. Principe: Simultaneous diagonalization in the frequency domain (SDIF) for source separation, ICA'99 (1999) pp. 245–250

52.162 A. Souloumiac: Blind source detection and separation using second order non-stationarity, Proc. ICASSP'95, Vol. III (1995) pp. 1912–1915

52.163 K. Matsuoka, M. Ohya, M. Kawamoto: A neural net for blind separation of nonstationary signals, Neural Netw. **8**(3), 411–419 (1995)

52.164 M. Kawamoto, A.K. Barros, A. Mansour, K. Matsuoka, N. Ohnishi: Blind separation for convolutive mixtures of non-stationary signals, Int. Conf. Neural Inf. Process. (1998) pp. 743–746

52.165 M. Kawamoto, A.K. Barros, K. Matsuoka, N. Ohnishi: A method of real-world blind separation implemented in frequency domain, ICA'00 (2000) pp. 267–272

52.166 M. Ito, M. Maruyoshi, M. Kawamoto, T. Mukai, N. Ohneshi: Effectiveness of directional microphones and utilization of source arriving directions in source separation, ICONIP'02 (2002) pp. 523–526

Amazon EX1010 Page 1085

52.167 M. Ito, Y. Takeuchi, T. Matsumoto, H. Kudo, M. Kawamoto, T. Mukai, N. Ohnishi: Moving-source separation using directional microphones, ISSPIT'02 (2002) pp. 523–526
52.168 Y. Katayama, M. Ito, Y. Takeuchi, T. Matsumoto, H. Kudo, N. Ohnishi, T. Mukai: Reduction of source separation time by placing microphones close together, ISSPIT'02 (2002) pp. 540–544
52.169 R. Aichner, S. Araki, S. Makino, T. Nishikawa, H. Saruwatari: Time domain blind source separation of non-stationary convolved signals by utilizing geometric beamforming, NNSP'02 (2002) pp. 445–454
52.170 R. Aichner, H. Buchner, S. Araki, S. Makino: On-line time-domain blind source separation of nonstationary convolved signals, ICA'03 (2003) pp. 987–992
52.171 H. Buchner, R. Aichner, W. Kellermann: A generalization of blind source separation algorithms for convolutive mixtures based on second-order statistics, IEEE Trans. Speech Audio Process. **13**(1), 120–134 (2005)
52.172 E. Visser, T.-W. Lee: Speech enhancement using blind source separation and two-channel energy based speaker detection, Proc. ICASSP'03, Vol. 1 (2003) pp. 884–887
52.173 E. Visser, K. Chan, S. Kim, T.-W. Lee: A comparison of simultaneous 3-channel blind source separation to selective separation on channel pairs using 2-channel BSS, ICLSP'04, Vol. IV (2004) pp. 2869–2872
52.174 M.Z. Ikram, D.R. Morgan: A multiresolution approach to blind separation of speech signals in a reverberant environment, Proc. ICASSP'01, Vol. V (2001)
52.175 M. Ikram, D. Morgan: A beamforming approach to permutation alignment for multichannel frequency-domain blind speech separation, Proc. ICASSP'02 (2002) pp. 881–884
52.176 M.Z. Ikram, D.R. Morgan: Permutation inconsistency in blind speech separation: Investigation and solutions, IEEE Trans. Speech Audio Process. **13**(1), 1–13 (2005)
52.177 L. Parra, C. Spence: On-line convolutive source separation of non-stationary signals, IEEE J. VLSI Signal Process. **26**(1,2), 39–46 (2000)
52.178 M. Joho: Blind signal separation of convolutive mixtures: A time-domain joint-diagonalization approach, ICA'04 (2004) pp. 578–585
52.179 M. Joho: Convolutive blind signal separation in acoustics by joint approximate diagonalization of spatiotemporal correlation matrices, Asilomar SSC (2004)
52.180 S. Araki, S. Makino, R. Mukai, H. Saruwatari: Equivalence between frequency domain blind source separation and frequency domain adaptive beamformers, CRAC'01 (2001)
52.181 C.L. Fancourt, L. Parra: The coherence function in blind source separation of convolutive mixtures of non-stationary signals, NNSP (2001) pp. 303–312
52.182 C.L. Fancourt, L. Parra: The generalized sidelobe decorrelator, WASPAA'01 (2001)
52.183 C. Fancourt, L. Parra: A comparison of decorrelation criteria for the blind source separation of non-stationary signals, SAM'02 (2002)
52.184 L. Parra, C. Alvino: Geometric source separation: Merging convolutive source separation with geometric beamforming, IEEE Trans. Speech Audio Process. **10**(6), 352–362 (2002)
52.185 L. Parra, C. Fancourt: An Adaptive Beamforming Perspective on Convolutive Blind Source Separation. In: *Noise Reduction in Speech Applications*, ed. by G. Davis (CRC, Boca Raton 2002)
52.186 E. Visser, T.-W. Lee: Blind source sepoaration in mobile environments usinbg a priori knowledge, Proc. ICASSP'04, Vol. III (2004) pp. 893–896
52.187 M.S. Pedersen, L.K. Hansen, U. Kjems, K.B. Rasmussen: Semi-blind source separation using head-related transfere functions, Proc. ICASSP'04, Vol. V (2004) pp. 713–716
52.188 S. Ding, T. Hikichi, T. Niitsuma, M. Hamatsu, K. Sugai: Recursive method for blind source separation and its applications to real-time separations of acoustic signals, ICA'03 (2003) pp. 517–522
52.189 E. Robledo-Arnuncio, B.F. Juang: Issues in frequency domain blind source separation – a critical revisit, Proc. ICASSP'05, Vol. V (2005) pp. 281–284
52.190 W. Wang, J.A. Chambers, S. Sanei: A joint diagonalization method for convolutive blind separation of nonstationary sources in the frequency domain, ICA'03 (2003) pp. 939–944
52.191 W. Wang, J.A. Chambers, S. Sanei: A novel hybrid approach to the permutation problem of frequency domain blind source separation, ICA'04 (2004) pp. 532–539
52.192 W. Wang, S. Sanei, J.A. Chambers: Penalty function-based joint diagonalization approach for convolutive blind separation of nonstationary sources, IEEE Trans. Signal Process. **53**(5), 1654–1669 (2005)
52.193 L. Molgedey, H. Schuster: Separation of independent signals using time-delayed correlations, Phys. Rev. Lett. **72**(23), 3634–3637 (1994)
52.194 T.J. Ngo, N. Bhadkamkar: Adaptive blind separation of audio sources by a physically compact device using second order statistics, ICA'99 (1999) pp. 257–260
52.195 D.W. Schobben, P.C.W. Sommen: A frequency domain blind signal separation method based on decorrelation, IEEE Trans. Signal Process. **8**(50), 1855–1865 (2002)
52.196 S. Ikeda, N. Murata: An approach to blind source separation of speech signals, ICANN'98, Vol. 2 (1998) pp. 761–766

Amazon EX1010 Page 1086

52.197 S. Ikeda, N. Murata: A method of blind separation on temporal structre of signals, ICONIP'98, Vol. 2 (1998) pp. 737–742

52.198 S. Ikeda, N. Murata: A method of ICA in time-frequency domain, ICA'99 (1999) pp. 365–371

52.199 N. Murata, S. Ikeda, A. Ziehe: An approach to blind source separation based on temporal structure of speech signals, Neurocomputing **41**(1–4), 1–24 (2001)

52.200 A. Ahmed, P.J.W. Rayner, S.J. Godsill: Considering non-stationarity for blind signal separation, WASPAA'99 (1999) pp. 111–114

52.201 B. Yin, P. Sommen: A new convolutive blind signal separation algorithm based on second order statistics using a simplified mixing model, EUSIPCO'00, Vol. 4 (2000) pp. 2049–2053

52.202 D.-T. Pham, C. Servière, H. Boumaraf: Blind separation of convolutive audio mixtures using nonstationarity, ICA'03 (2003) pp. 981–986

52.203 C. Servière, D. Pham: A novel method for permutation correction in frequency-domain in blind separation of speech mixtures, ICA'04 (2004) pp. 807–815

52.204 H. Buchner, R. Aichner, W. Kellermann: Blind source separation for convolutive mixtures exploiting nongaussianity, nonwhiteness, and nonstationarity, IWAENC'03 (2003) pp. 275–278

52.205 R. Aichner, H. Buchner, F. Yan, W. Kellermann: Real-time convolutive blind source separation based on a broadband approach, ICA'04 (2004) pp. 840–848

52.206 H. Buchner, R. Aichner, W. Kellermann: Trinicon: A versatile framework for multichannel blind signal processing, Proc. ICASSP'04, Vol. III (2004) pp. 889–892

52.207 R. Aichner, H. Buchner, W. Kellermann: On the causality problem in time-domain blind source separation and deconvolution algorithms, Proc. ICASSP'05, Vol. V (2005) pp. 181–184

52.208 B.S. Krongold, D.L. Jones: Blind source separation of nonstationary convolutively mixed signals, SSAP'00 (2000) pp. 53–57

52.209 K. Rahbar, J.P. Reilly: Blind source separation algorithm for MIMO convolutive mixtures, ICA'01 (2001)

52.210 A. Holobar, D. Zazula: A novel approach to convolutive blind separation of close-to-orthogonal pulse sources using second-order statistics, EUSIPCO'04 (2004) pp. 381–384

52.211 K.-C. Yen, Y. Zhao: Adaptive co-channel speech separation and recognition, IEEE Trans Speech Audio Process. **7**(2), 138–151 (1999)

52.212 A. Mertins, I. Russel: An extended ACDC algorithm for the blind estimation of convolutive mixing systems, ISSPA'03, Vol. 2 (2003) pp. 527–530

52.213 Y. Zhao, R. Hu: Fast convergence speech source separation in reverberant acoustic environment, Proc. ICASSP'04, Vol. III (2004) pp. 897–900

52.214 I. Russell, J. Xi, A. Mertins, J. Chicharo: Blind source separation of nonstationary convolutively mixed signals in the subband domain, Proc. ICASSP'04, Vol. V (2004) pp. 484–484

52.215 A. Leon-Garcia: *Probability, Random Processes for Electrical Engineering*, 2nd edn. (Addison Wesley, Reading 1994)

52.216 S. Shamsunder, G.B. Giannakis: Multichannel blind signal separation and reconstruction, IEEE Trans. Speech, Audio Process. **5**(6), 515–528 (1997)

52.217 L. Deneire, D.T. Slock: A Schur method for multiuser multichannel blind identification, Proc. ICASSP'99 (1999) pp. 2905–2908

52.218 C.T. Ma, Z. Ding, S.F. Yau: A two-stage algorithm for MIMO blind deconvolution of nonstationary colored signals, IEEE Trans. Signal Process. **48**(4), 1187–1192 (2000)

52.219 I. Bradaric, A.P. Petropulu, K.I. Diamantaras: On blind identifiability of FIR-MIMO systems with cyclostationary inputs using second order statistics, Proc. ICASSP'02, Vol. II (2002) pp. 1745–1748

52.220 I. Bradaric, A.P. Petropulu, K.I. Diamantaras: On blind identifiability of FIR-MIMO systems with cyclostationary inputs using second order statistics, IEEE Trans. Signal Process. **51**(2), 434–441 (2003)

52.221 M. Knaak, M. Kunter, D. Filbert: Blind source separation for acoustical machine diagnosis, DSP'02 (2002)

52.222 M. Knaak, S. Araki, S. Makino: Geometrically constrained ICA for robust separation of sound mixtures, ICA'03 (2003) pp. 951–956

52.223 T. Mei, F. Yin: Blind separation of convolutive mixtures by decorrelation, Signal Process. **84**(12), 2297 (2004)

52.224 S. Rickard, O. Yilmaz: On the approximate W-disjoint orthogonality of speech, Proc. ICASSP'02, Vol. I (2002) pp. 529–532

52.225 S. Rickard, T. Melia, C. Fearon: DESPRIT – histogram based blind source separation of more sources than sensors using subspace methods, WASPAA'05 (2005) pp. 5–8

52.226 A. Jourjine, S. Rickard, O. Yilmaz: Blind separation of disjoint orthogonal signals: Demixing N sources from 2 mixtures, Proc. ICASSP'00, Vol. V (2000) pp. 2985–2988

52.227 S. Araki, S. Makino, H. Sawada, R. Mukai: Reducing musical noise by a fine-shift overlap-add method applied to source separation using a time-frequency mask, Proc. ICASSP'05, Vol. III (2005) pp. 81–84

52.228 M.S. Pedersen, D. Wang, J. Larsen, U. Kjems: Overcomplete blind source separation by combining ICA and binary time-frequency masking, MLSP'05 (2005)

52.229 M.S. Pedersen, D.L. Wang, J. Larsen, U. Kjems: Separating underdetermined convolutive speech mixtures, ICA'06 (2006) pp. 674–681

Amazon EX1010 Page 1087

52.230 H. Sawada, S. Araki, R. Mukai, S. Makino: Blind extraction of a dominant source signal from mixtures of many sources, Proc. ICASSP'05, Vol. III (2005) pp. 61–64
52.231 H.-C. Wu, J.C. Principe, D. Xu: Exploring the time-frequency microstructure of speech for blind source separation, Proc. ICASSP'98, Vol. 2 (1998) pp. 1145–1148
52.232 J.M. Peterson, S. Kadambe: A probabilistic approach for blind source separation of underdetermined convolutive mixtures, Proc. ICASSP'03, Vol. 6 (2003) pp. 581–584
52.233 D. Luengo, I. Santamaria, L. Vielva, C. Pantaleon: Underdetermined blind separation of sparse sources with instantaneous and convolutive mixtures, NNSP'03 (2003) pp. 279–288
52.234 S.A. Abdallah, M.D. Plumbley: Application of geometric dependency analysis to the seperation of convolved mixtures, ICA'04 (2004) pp. 540–547
52.235 M. Babaie-Zadeh, A. Mansour, C. Jutten, F. Marvasti: A geometric approach for separating several speech signals, ICA'04 (2004) pp. 798–806
52.236 Y. Li, A. Cichocki, L. Zhang: Blind source estimation of FIR channels for binary sources: A grouping decision approach, Signal Process. **84**(12), 2245–2263 (2004)
52.237 B.A. Pearlmutter, A.M. Zador: Monaural source separation using spectral cues, ICA'04 (2004) pp. 478–485
52.238 P. Smaragdis: Non negative matrix factor deconvolution, extraction of multiple sound sources from monophonic inputs, ICA'04 (2004) pp. 494–499
52.239 T. Virtanen: Separation of sound sources by convolutive sparse coding, SAPA'04 (2004)
52.240 M.S. Pedersen, T. Lehn-Schiøler, J. Larsen: BLUES from music: BLind Underdetermined Extraction of Sources from Music, ICA'06 (2006) pp. 392–399
52.241 A.S. Bregman: *Auditory Scene Analysis*, 2nd edn. (MIT Press, Cambridge 1990)
52.242 M. Weintraub: The GRASP sound separation system, Proc. ICASSP'84 (1984) pp. 69–72
52.243 R.R. Guddeti, B. Mulgrew: Perceptually motivated blind source separation of convolutive mixtures, Proc. ICASSP'05, Vol. V (2005) pp. 273–276
52.244 A.K. Barros, T. Rutkowski, F. Itakura, N. Ohnishi: Estimation of speech embedded in a reverberant and noisy environment by independent component analysis and wavelets, IEEE Trans. Neural Netw. **13**(4), 888–893 (2002)
52.245 M. Furukawa, Y. Hioka, T. Ema, N. Hamada: Introducing new mechanism in the learning process of FDICA-based speech separation, IWAENC'03 (2003) pp. 291–294
52.246 R.D. Patterson: The sound of a sinusoid: Spectral models, J. Acoust Soc. Am. **96**, 1409–1418 (1994)
52.247 T. Rutkowski, A. Cichocki, A.K. Barros: Speech enhancement from interfering sounds using CASA techniques and blind source separation, ICA'01 (2001) pp. 728–733
52.248 N. Roman, D. Wang, G.J. Brown: Speech segregation based on sound localization, J. Acoust. Soc. Am. **114**(4), 2236–2252 (2003)
52.249 T. Nishikawa, H. Saruwatari, K. Shikano: Blind source separation of acoustic signals based on multistage ICA combining frequency-domain ICA and time-domain ICA, IEICE Trans. Fundamentals **E86-A**(4), 846–858 (2003)
52.250 P. Smaragdis: Efficient blind separation of convolved sound mixtures, WASPAA'97 (1997)
52.251 M. Davies: Audio source separation. In: *Mathematics in Signal Processing V*, ed. by J.G. McWhirter, I.K. Proudler (Oxford Univ. Press, Oxford 2001)
52.252 F. Duplessis-Beaulieu, B. Champagne: Fast convolutive blind speech separation via subband adaptation, Proc. ICASSP'03, Vol. 5 (2003) pp. 513–516
52.253 C. Servière: Separation of speech signals under reverberant conditions, EUSIPCO'04 (2004) pp. 1693–1696
52.254 T.-W. Lee, A.J. Bell, R.H. Lambert: Blind separation of delayed and convolved sources, NIPS, Vol. 9 (1997) pp. 758–764
52.255 T.-W. Lee, A. Ziehe, R. Orglmeister, T.J. Sejnowski: Combining time-delayed decorrelation and ICA: towards solving the cocktail party problem, Proc. ICASSP'98, Vol. 2 (1998) pp. 1249–1252
52.256 A. Westner, V.M. Bove Jr.: Blind separation of real world audio signals using overdetermined mixtures, ICA'99 (1999)
52.257 K. Kokkinakis, A.K. Nandi: Multichannel blind deconvolution for source separation in convolutive mixtures of speech, IEEE Trans. Audio Speech Lang. Process. **14**(1), 200–212 (2006)
52.258 H. Sawada, R. Mukai, S. de la Kethulle, S. Araki, S. Makino: Spectral smoothing for frequency-domain blind source separation, IWAENC'03 (2003) pp. 311–314
52.259 H. Sawada, R. Mukai, S. Araki, S. Makino: Convolutive blind source separation for more than two sources in the frequency domain, Proc. ICASSP'04, Vol. III (2004) pp. 885–888
52.260 D.W.E. Schobben, P.C.W. Sommen: A new blind signal separation algorithm based on second-order statistics, IASTED SIP'06 (1998) pp. 564–569
52.261 H. Attias, J.C. Platt, A. Acero, L. Deng: Speech denoising and dereverberation using probabilistic models, NIPS'01, Vol. 13 (2001)
52.262 R. Aichner, H. Buchner, W. Kellermann: A novel normalization and regularization scheme for broadband convolutive blind source separation, ICA'06 (2006) pp. 527–535
52.263 H. Sawada, S. Araki, R. Mukai, S. Makino: Blind source separation with different sensor spacing and filter length for each frequency range, NNSP'02 (2002) pp. 465–474

Amazon EX1010 Page 1088

52.264 P. Smaragdis: Blind separation of convolved mixtures in the frequency domain, Neurocomputing **22**(1–3), 21–34 (1998)
52.265 V.C. Soon, L. Tong, Y.F. Huang, R. Liu: A wideband blind identification approach to speech acquisition using a microphone array, Proc. ICASSP'92, Vol. 1 (1992) pp. 293–296
52.266 S. Kurita, H. Saruwatari, S. Kajita, K. Takeda, F. Itakura: Evaluation of frequency-domain blind signal separation using directivity pattern under reverberant conditions, Proc. ICASSP'00 (2000) pp. 3140–3143
52.267 F. Asano, S. Ikeda, M. Ogawa, H. Asoh, N. Kitawaki: Combined approach of array processing and independent component analysis for blind separation of acoustic signals, IEEE Trans. Speech Audio Process. **11**(3), 204–215 (2003)
52.268 N. Mitianoudis, M.E. Davies: Permutation alignment for frequency domain ICA using subspace beamforming methods, ICA'04 (2004) pp. 669–676
52.269 W. Baumann, B.-U. Köhler, D. Kolossa, R. Orglmeister: Real time separation of convolutive mixtures, ICA'01 (2001) pp. 65–69
52.270 H. Gotanda, K. Nobu, T. Koya, K. Kaneda, T. Ishibashi, N. Haratani: Permutation correction and speech extraction based on split spectrum through fastICA, ICA'03 (2003) pp. 379–384
52.271 S. Araki, S. Makino, R. Aichner, T. Nishikawa, H. Saruwatari: Subband based blind source separation with appropriate processing for each frequency band, ICA'03 (2003) pp. 499–504
52.272 J. Anemüller, B. Kollmeier: Amplitude modulation decorrelation for convolutive blind sourc separation, ICA'00 (2000) pp. 215–220
52.273 J. Antoni, F. Guillet, M. El Badaoui, F. Bonnardot: Blind separation of convolved cyclostationary processes, Signal Process. **85**(1), 51–66 (2005)
52.274 A. Dapena, C. Serviere: A simplified frequency-domain approach for blind separation of convolutive mixtures, ICA'01 (2001) pp. 569–574
52.275 C. Mejuto, A. Dapena, L. Castedo: Frequency-domain infomax for blind separation of convolutive mixtures, ICA'00 (2000) pp. 315–320
52.276 N. Mitianoudis, M. Davies: New fixed-point ICA algorithms for convolved mixtures, ICA'01 (2001) pp. 633–638
52.277 I. Lee, T. Kim, T.-W. Lee: Complex fastiva: A robust maximum likelihood approach of mica for convolutive bss, ICA'06 (2006) pp. 625–632
52.278 T. Kim, H. Attias, S.-Y. Lee, T.-W. Lee: Blind source separation exploiting higher-order frequency dependencies, IEEE Trans. Audio Speech Lang. Process. **15**(1), 70–79 (2006)
52.279 S. Dubnov, J. Tabrikain, M. Arnon-Targan: A method for directionally-disjoint source separation in convolutive environment, Proc. ICASSP'04, Vol. V (2004) pp. 489–492
52.280 A. Hiroe: Solution of permutation problem in frequency domain ica, using multivariate probability density functions, ICA'06 (2006) pp. 601–608
52.281 D. Kolossa, B.-U. Köhler, M. Conrath, R. Orgelmeister: Optimal permutation correlation by multiobjective genetic algorithms, ICA'01 (2001) pp. 373–378
52.282 K. Kamata, X. Hu, H. Kobataka: A new approach to the permutation problem in frequency domain blind source separation, ICA'04 (2004) pp. 849–856
52.283 H. Attias, C.E. Schreiner: Blind source separation and deconvolution: The dynamic component analysis algorithm, Neural Comput. **11**, 803–852 (1998)
52.284 F. Asano, S. Ikeda, M. Ogawa, H. Asoh, N. Kitawaki: Blind source separation in reflective sound fields, HSC'01 (2001)
52.285 H. Sawada, S. Araki, R. Mukai, S. Makino: On calculating the inverse of separation matrix in frequency-domain blind source separation, ICA'06 (2006) pp. 691–699
52.286 V.C. Soon, L. Tong, Y.F. Huang, R. Liu: A robust method for wideband signal separation, ISCAS'93 (1993) pp. 703–706
52.287 R. Mukai, S. Araki, H. Sawada, S. Makino: Removal of residual cross-talk components in blind source separation using LMS filters, NNSP'02 (2002) pp. 435–444
52.288 H. Saruwatari, S. Kurita, K. Takeda, F. Itakura, K. Shikano: Blind source separation based on subband ICA and beamforming, ICSLP'00, Vol. III (2000) pp. 94–97
52.289 S.Y. Low, S. Nordholm, R. Togneri: Convolutive blind signal separation with post-processing, IEEE Trans. Speech, Audio Process. **12**(5), 539–548 (2004)
52.290 H. Saruwatari, T. Kawamura, T. Nishikawa, A. Lee, K. Shikano: Blind source separation based on a fast-convergence algorithm combining ICA and beamforming, IEEE Trans. Audio, Speech, Lang. Process. **14**(2), 666–678 (2006)
52.291 R. Mukai, H. Sawada, S. Araki, S. Makino: Near-field frequency domain blind source separation for convolutive mixtures, Proc. ICASSP'04, Vol. IV (2004) pp. 49–52
52.292 R.O. Schmidt, R.E. Franks: Multiple source DF signal processing: an experimental system, IEEE Trans. Ant. Prop. **4**(3), 281–290 (1986)
52.293 N. Murata, S. Ikeda: An on-line algorightm for blind source separation on speech signals, International Symposium on Theory and its Applications, Vol. 3 (1998) pp. 923–926
52.294 F. Asano, S. Ikeda: Evaluation and real-time implementation of blind source separation system using time-delayed decorrelation, ICA'00 (2000) pp. 411–416
52.295 S. Ukai, H. Saruwatari, T. Takatani, R. Mukai: Multistage SIMO-model-based blind source sep-

Amazon EX1010 Page 1089

aration combining frequenct-domain ICA and time-domain ICA, Proc. ICASSP'04, Vol. IV (2004) pp. 109–112

52.296 S. Araki, S. Makino, A. Blin, R. Mukai, H. Sawada: Blind separation of more speech than sensors with less distortion by combining sparseness and ICA, IWAENC'03 (2003) pp. 271–274

52.297 R. Mukai, H. Sawada, S. Arakiand, S. Makino: Blind source separation for moving speech signals using blockwise ICA and residual crosstalk subtraction, IEICE Trans. Fundamentals **E87-A**(8), 1941–1948 (2004)

52.298 R. Mukai, H. Sawada, S. Araki, S. Makino: Frequency domain blind source separation for many speech signals, ICA'04 (2004) pp. 461–469

52.299 Y. Mori, H. Saruwatari, T. Takatani, K. Shikano, T. Hiekata, T. Morita: ICA and binary-mask-based blind source separation with small directional microphones, ICA'06 (2006) pp. 649–657

52.300 T. Nishikawa, H. Saruwatari, K. Shikano: Stable learning algorithm for blind separation of temporally correlated signals combining multistage ICA and linear prediction, ICA'03 (2003) pp. 337–342

52.301 T. Takatani, T. Nishikawa, H. Saruwatari, K. Shikano: Blind separation of binaural sound mixtures using SIMO-model-based independent component analysis, Proc. ICASSP'04, Vol. IV (2004) pp. 113–116

52.302 T. Hoya, A.K. Barros, T. Rutkowski, A. Cichocki: Speech extraction based upon a combined subband independent component analysis and neural memory, ICA'03 (2003) pp. 355–360

Amazon EX1010 Page 1090

53. Sound Field Reproduction

R. Rabenstein, S. Spors

Multichannel sound field reproduction aims at the physically correct synthesis of acoustical wave fields with a large number of loudspeakers. It goes beyond stereophony by extending or eliminating the so-called sweet spot. This chapter presents the physical and mathematical description of various methods for this purpose and presents the physical and mathematical background of several methods for multichannel sound field reproduction.

Section 53.2 introduces the mathematical representation of sound fields. The panning laws of stereophony are given in Sect. 53.3 as an introduction to vector-based amplitude panning in Sect. 53.4. Then mathematically more-involved methods are described in the Sections on ambisonics (Sect. 53.5) and wave field synthesis (Sect. 53.6).

53.1 **Sound Field Synthesis** 1095

53.2 **Mathematical Representation of Sound Fields** 1096
- 53.2.1 Coordinate Systems 1096
- 53.2.2 Wave Equation 1096
- 53.2.3 Plane Waves 1097
- 53.2.4 General Wave Fields in Two Dimensions 1098
- 53.2.5 Spherical Waves 1099
- 53.2.6 The Kirchhoff–Helmholtz Integral .. 1099

53.3 **Stereophony** 1100
- 53.3.1 Sine Law 1100
- 53.3.2 Tangent Law 1101
- 53.3.3 Application of Amplitude Panning . 1102
- 53.3.4 The Sweet Spot 1102

53.4 **Vector-Based Amplitude Panning** 1103
- 53.4.1 Two-Dimensional Vector-Based Amplitude Panning .. 1103
- 53.4.2 Three-Dimensional Vector-Based Amplitude Panning .. 1104
- 53.4.3 Perception of Vector-Based Amplitude Panning 1104

53.5 **Ambisonics** 1104
- 53.5.1 Two-Dimensional Ambisonics 1105
- 53.5.2 Three-Dimensional Ambisonics 1108
- 53.5.3 Extensions of Ambisonics 1109

53.6 **Wave Field Synthesis** 1109
- 53.6.1 Description of Acoustical Scenes by the Kirchhoff–Helmholtz Integral 1109
- 53.6.2 Monopole and Dipole Sources 1110
- 53.6.3 Reduction to Two Spatial Dimensions 1111
- 53.6.4 Spatial Sampling 1112
- 53.6.5 Determination of the Loudspeaker Driving Signals 1112

References 1113

53.1 Sound Field Synthesis

Sound field synthesis encompasses a number of techniques for the creation of acoustical events using electroacoustical transducers. The most widespread application is the reproduction of acoustical performances for human listeners. Another emerging application is the reproduction of spatially distributed environmental noise (engines, streets, car interior, etc.) in laboratory settings for testing of speech communication equipment. Since all techniques for sound field synthesis have been initially developed for human entertainment, only the first application is used as an example below.

The spatial sound impression perceived by human listeners depends on two major factors

- the sound pressure at the listener's ears
- the psychoacoustics of spatial hearing

The first factor is a physical quantity that can be produced with electroacoustical equipment. The second factor requires precise knowledge about how our hearing organs translate physical excitations into spatial perception. Various reproduction methods make use of these factors in different ways. They can be roughly classified

into binaural techniques, stereophony, and sound field synthesis.

Binaural techniques use head phones to create the required sound pressure immediately at the listener's ears. There is no need for a detailed analysis of acoustical waves. Instead the design of binaural techniques is based mostly on the psychoacoustics of spatial hearing.

Stereophony uses two or more loudspeakers to deliver the sound pressure to the listener. The sound field around the listener's head can be described as a superposition of the sound waves emitted by the loudspeakers subject to scattering at the head. Then psychoacoustics plays a similar role as for binaural techniques.

Sound field synthesis methods use a large number of loudspeakers (tens to hundreds) to reproduce a sound field not only at the ears of one listener but in a larger space enclosing multiple listeners. Here the focus is on the technology for the correct reproduction of a physical field quantity. Psychoacoustical effects play a minor role as long as it can be assumed the listeners respond to a synthesized sound field in the same way as to the original one.

53.2 Mathematical Representation of Sound Fields

Sound fields are represented by scalar- and vector-valued quantities that depend on two or three spatial and one time coordinate. This section discusses various coordinate systems and different representations of sound fields with respect to these coordinates.

53.2.1 Coordinate Systems

The evolution of sound fields with time is denoted by the scalar time coordinate t. The spatial distribution of sound fields requires in general a three-dimensional (3-D) coordinate system. However, many arrangements of microphones and loudspeakers are restricted to positions in a plane, e.g., a horizontal plane in the height of the listener's ears. Then two spatial dimensions are sufficient although sound propagation is still a three-dimensional process. In these cases, the term two-dimensional (2-D) sound field is used to denote a special case of a three-dimensional sound field that exhibits no dependence on one of the three spatial coordinates. Different representations of two- and three-dimensional sound fields are presented next.

In Cartesian coordinates the three-dimensional space coordinates are denoted by

$$\boldsymbol{z} = \begin{pmatrix} \boldsymbol{x} \\ z \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \tag{53.1}$$

where x and y are chosen for a two-dimensional representation of a three-dimensional function independent of z. These functions are also written $u(\boldsymbol{x}) = v(\boldsymbol{z})$.

For two spatial dimensions, also circular or polar coordinates are used here. Two-dimensional Cartesian coordinates $\boldsymbol{x}$ and polar coordinates $\boldsymbol{r}$

$$\boldsymbol{x} = \begin{pmatrix} x \\ y \end{pmatrix}, \qquad \boldsymbol{r} = \begin{pmatrix} r \\ \alpha \end{pmatrix} \tag{53.2}$$

are related by

$$\boldsymbol{x} = r \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix}, \qquad \boldsymbol{r} = \begin{pmatrix} \sqrt{x^2 + y^2} \\ \arctan\left(\frac{y}{x}\right) \end{pmatrix}. \tag{53.3}$$

Figure 53.1 shows the coordinate systems discussed above. Functions of two-dimensional Cartesian and polar coordinates are distinguished by the subscripts 'c' and 'p', such that $u_\mathrm{c}(\boldsymbol{x}) = u_\mathrm{p}(\boldsymbol{r})$ holds.

53.2.2 Wave Equation

Physical descriptions of sound fields are usually formulated in terms of the scalar-valued sound pressure $p(\boldsymbol{z}, t)$ and the vector-valued particle velocity $\boldsymbol{v}(\boldsymbol{z}, t)$. Their mutual dependencies are described by certain partial differential equations: Newton's law of motion

$$\operatorname{grad} p(\boldsymbol{z}, t) + \rho \frac{\partial}{\partial t} \boldsymbol{v}(\boldsymbol{z}, t) = 0 \tag{53.4}$$

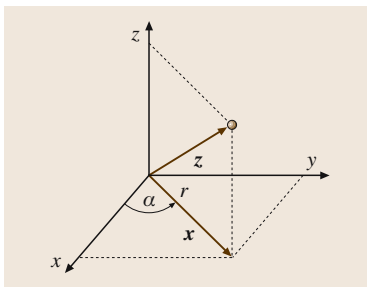


Fig. 53.1 Cartesian and polar coordinates in two and three dimensions

and the equation of continuity

$$\frac{\partial}{\partial t} p(\boldsymbol{z}, t) + \rho c^2 \operatorname{div} \boldsymbol{v}(\boldsymbol{z}, t) = 0 \,, \tag{53.5}$$

where ρ denotes the static density and c the speed of sound. Elimination of the particle velocity leads to the wave equation

$$\frac{\partial^2}{\partial t^2} p(\boldsymbol{z}, t) + c^2 \nabla^2 p(\boldsymbol{z}, t) = 0 \,, \tag{53.6}$$

which constitutes the basic mathematical representation of acoustic phenomena [53.1–4]. Solutions of the wave equation are called *sound fields* or *wave fields*.

The wave equation holds for different coordinate systems with the appropriate definition of the nabla operator ∇. Unique solutions also require physically meaningful boundary and initial conditions. Boundary conditions are determined by the acoustical properties of reflecting surfaces. Enclosures where reflections can be neglected are often modeled by free field conditions. Arbitrary solutions of the wave equation may be represented as a superposition of simple wave forms [53.4], e.g., plain or spherical waves as discussed below.

53.2.3 Plane Waves

A simple solution of the wave equation are the so-called plane waves. Their mathematical description in two and three spatial dimensions is now introduced.

Plane Waves in Three Dimensions

Consider the function

$$p(\boldsymbol{z}, t) = f(ct + \boldsymbol{n}^{\mathrm{T}} \boldsymbol{z}) = g\big(t + \frac{1}{c} \boldsymbol{n}^{\mathrm{T}} \boldsymbol{z}\big) \tag{53.7}$$

with $f(ct) = g(t)$ and a vector $\boldsymbol{n}$ of unit length. Vector transposition is indicated by the superscript 'T'. Equation (53.7) is a solution of the wave equation, as can be proven by inserting it into (53.6).

The arbitrary function $g(t)$ – and similarly $f(t)$ – describes the waveform of $p(\boldsymbol{z}, t)$ at the origin of the spatial coordinate system. The argument of f represents a plane in space with normal vector $\boldsymbol{n}$. As time increases, this plane propagates through space from the direction of $\boldsymbol{n}$ with speed c. Therefore the solution (53.7) is called a *plane-wave solution*.

Fourier transformation [53.5–7]

$$G(\omega) = \mathcal{F}\{g(t)\} = \int_{-\infty}^{\infty} g(t)\, \mathrm{e}^{-\mathrm{i}\omega t}\, \mathrm{d}t \tag{53.8}$$

of $p(\boldsymbol{z}, t)$ with respect to t gives

$$P(\boldsymbol{z}, \omega) = G(\omega)\, \mathrm{e}^{\mathrm{i}\boldsymbol{k}^{\mathrm{T}} \boldsymbol{z}} \tag{53.9}$$

with the wave vector $\boldsymbol{k} = k\boldsymbol{n}$ and the wave number $k = \omega/c$.

In this form, the influence of the temporal variation of the waveform $g(t)$ and the propagation through space are separated into two multiplicative factors. The spectral content of the waveform is given by $G(\omega)$ and the propagation of the plane wave is given by the exponential term.

Plane Waves in Two Dimensions

A plane wave in the xy-plane is characterized by the angle θ of the normal vector $\boldsymbol{n}$ from (53.7) in this plane

$$\boldsymbol{n} = \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} . \tag{53.10}$$

To indicate the direction from which the plane wave emanates, the angle θ is added to the list of arguments of the sound pressure $p_{\mathrm{c}}(\boldsymbol{x}, t, \theta)$ and the corresponding waveform $g(t, \theta)$ and likewise for their Fourier transforms $P_{\mathrm{c}}(\boldsymbol{x}, \omega, \theta)$ and $G(\omega, \theta)$.

To express two-dimensional plane waves in Cartesian coordinates

$$P_{\mathrm{c}}(\boldsymbol{x}, \omega, \theta) = G(\omega, \theta)\, \mathrm{e}^{\mathrm{i}\boldsymbol{k}^{\mathrm{T}} \boldsymbol{x}} \tag{53.11}$$

in polar form, it is convenient to separate the spectral properties $G(\omega, \theta)$ from the spatial properties as

$$P_{\mathrm{c}}(\boldsymbol{x}, \omega, \theta) = G(\omega, \theta)\, \hat{p}_{\mathrm{c}}(\boldsymbol{x}, \theta) \tag{53.12}$$

with

$$\hat{p}_{\mathrm{c}}(\boldsymbol{x}, \theta) = \hat{p}_{\mathrm{c}}(x, y, \theta) = \mathrm{e}^{\mathrm{i}\boldsymbol{k}^{\mathrm{T}} \boldsymbol{x}} \,. \tag{53.13}$$

A change of the spatial coordinate affects only the spatial term $\hat{p}_{\mathrm{c}}(x, y, \theta)$ but not $G(\omega, \theta)$ in (53.11). Therefore, only this term is considered for a transition to polar coordinates.

In polar coordinates, the wavevector $\boldsymbol{k}$ takes the form

$$\boldsymbol{k} = k\boldsymbol{n} = k \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} . \tag{53.14}$$

The polar representation $\hat{p}_{\mathrm{c}}(x, y, \theta) = \hat{p}_{\mathrm{p}}(r, \alpha, \theta)$ follows by expressing $\boldsymbol{k}^{\mathrm{T}} \boldsymbol{x}$ through r and α

$$\boldsymbol{k}^{\mathrm{T}} \boldsymbol{x} = kr \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} = kr \cos(\theta - \alpha) \tag{53.15}$$

as

$$\hat{p}_{\mathrm{p}}(r, \alpha, \theta) = \mathrm{e}^{\mathrm{i}kr \cos(\theta - \alpha)} \,. \tag{53.16}$$

Since $\hat{p}_\mathrm{p}(r, \alpha, \theta)$ is a periodic function of α, it can be expanded into a Fourier series. From the properties of the Bessel functions $J_\nu(kr)$ of the first kind and order ν [53.8–11] follows the Fourier series expansion (Jacobi–Anger expansion) [53.9]

$$\hat{p}_\mathrm{p}(r, \alpha, \theta) = \sum_{\nu=-\infty}^{\infty} \mathrm{i}^\nu J_\nu(kr)\, \mathrm{e}^{\mathrm{i}\nu(\alpha-\theta)} \,. \tag{53.17}$$

This can be regarded as an expansion of a plane wave into the angular modes $\mathrm{e}^{\mathrm{i}\nu\alpha}$ with coefficients $\mathrm{i}^\nu J_\nu(kr)\mathrm{e}^{-\mathrm{i}\nu\theta}$. This series can also be written in terms of sine and cosine functions. From the symmetry relation of the Bessel functions $J_\nu(kr)$ in the form

$$\mathrm{i}^{-\nu} J_{-\nu}(kr) = \mathrm{i}^\nu J_\nu(kr) \tag{53.18}$$

it follows that

$$\hat{p}_\mathrm{p}(r, \alpha, \theta) = J_0(kr) + 2\sum_{\nu=1}^{\infty} \mathrm{i}^\nu J_\nu(kr)\, \cos[\nu(\alpha-\theta)] \,. \tag{53.19}$$

A similar representation can be derived on the basis of sine functions.

With any of the representations (53.16), (53.17), or (53.19) and

$$\hat{p}_\mathrm{c}(\boldsymbol{x}, \theta) = \hat{p}_\mathrm{c}(x, y, \theta) = \hat{p}_\mathrm{p}(r, \alpha, \theta) = \hat{p}_\mathrm{p}(\boldsymbol{r}, \theta) \tag{53.20}$$

follows the two-dimensional plane wave in polar coordinates

$$P_\mathrm{p}(\boldsymbol{r}, \omega, \theta) = G(\omega, \theta)\, \hat{p}_\mathrm{p}(\boldsymbol{r}, \theta) \,, \tag{53.21}$$

$$P_\mathrm{p}(r, \alpha, \omega, \theta) = G(\omega, \theta)\, \hat{p}_\mathrm{p}(r, \alpha, \theta) \,. \tag{53.22}$$

The notation $P_\mathrm{p}(r, \alpha, \omega, \theta)$ is used to address the different components of $\boldsymbol{r}$ individually.

53.2.4 General Wave Fields in Two Dimensions

Representation by Angular Modes

Real wave fields rarely exhibit the ideal plane-wave property presented above. However, general wave fields can be represented as a superposition of plane waves from different directions θ and with different spectral weighting $G(\omega, \theta)$. This superposition follows from (53.22) by integration with respect to all possible directions $0 \le \theta < 2\pi$ as [53.4, 11]

$$P_\mathrm{p}(\boldsymbol{r}, \omega) = \frac{1}{2\pi} \int_0^{2\pi} G(\omega, \theta)\, \hat{p}_\mathrm{p}(r, \alpha, \theta)\, \mathrm{d}\theta \,, \tag{53.23}$$

where $P_\mathrm{p}(\boldsymbol{r}, \omega)$ denotes a general wave field with contributions from all directions θ.

Replacing $\hat{p}_\mathrm{p}(r, \alpha, \theta)$ by the expansion (53.17) gives

$$P_\mathrm{p}(\boldsymbol{r}, \omega) = \sum_{\nu=-\infty}^{\infty} \mathrm{i}^\nu J_\nu(kr)\, G^\circ(\omega, \nu)\, \mathrm{e}^{\mathrm{i}\nu\alpha} \,, \tag{53.24}$$

where $G^\circ(\omega, \nu)$ are the Fourier series coefficients

$$G^\circ(\omega, \nu) = \frac{1}{2\pi} \int_0^{2\pi} G(\omega, \theta)\, \mathrm{e}^{-\mathrm{i}\nu\theta}\, \mathrm{d}\theta \,. \tag{53.25}$$

Similar as for the plane wave in (53.17), (53.25) is an expansion into the angular modes with the coefficients $\mathrm{i}^\nu J_\nu(kr)\, G^\circ(\omega, \nu)$ and similar to (53.19) there is a representation in terms of the real and imaginary part of $G^\circ(\omega, \nu) = G^\circ_\mathrm{R}(\omega, \nu) + \mathrm{i}G^\circ_\mathrm{I}(\omega, \nu)$

$$\begin{aligned} P_\mathrm{p}(\boldsymbol{r}, \omega) = {} & J_0(kr)\, G^\circ(\omega, 0) \\ & + 2\sum_{\nu=1}^{\infty} \mathrm{i}^\nu J_\nu(kr)\, G^\circ_\mathrm{R}(\omega, \nu) \cos\nu\alpha \\ & - 2\sum_{\nu=1}^{\infty} \mathrm{i}^\nu J_\nu(kr)\, G^\circ_\mathrm{I}(\omega, \nu) \sin\nu\alpha \,, \end{aligned} \tag{53.26}$$

since the angular coefficients have conjugate symmetry $G^\circ(\omega, \pm\nu) = G^\circ_\mathrm{R}(\omega, \nu) \pm \mathrm{i}G^\circ_\mathrm{I}(\omega, \nu)$.

Physical Interpretation of the Angular Coefficients

The low-order angular coefficients have a physical interpretation in terms of sound pressure and particle velocity for $r = 0$. This is derived from (53.26) with the following relations for the Bessel functions

$$\begin{aligned} & J_0(0) = 1, \quad J_0'(0) = 0 \\ & J_1(0) = 0, \quad J_1'(0) = \tfrac{1}{2} \\ & J_\nu(0) = 0, \quad J_\nu'(0) = 0, \quad \nu > 1 \\ & \tfrac{\mathrm{d}}{\mathrm{d}r} J_\nu(kr) = k J_\nu'(kr) \qquad , \end{aligned} \tag{53.27}$$

where the prime denotes the derivative of a function with respect to its argument. Then the Fourier transform of the pressure $P_\mathrm{p}(\boldsymbol{r}, \omega)$ and its radial derivative $\partial/\partial r P_\mathrm{p}(\boldsymbol{r}, \omega)$ at $r = 0$ are given by

$$P_\mathrm{p}(\boldsymbol{r}, \omega)\big|_{r=0} = G^\circ(\omega, 0) \tag{53.28}$$

$$\begin{aligned} \frac{\partial}{\partial r} P_\mathrm{p}(\boldsymbol{r}, \omega)\Big|_{r=0} = {} & \mathrm{i}k\, G^\circ_\mathrm{R}(\omega, 1) \cos\alpha \\ & - \mathrm{i}k\, G^\circ_\mathrm{I}(\omega, 1) \sin\alpha \,. \end{aligned} \tag{53.29}$$

The real and imaginary part of the angular coefficient $G^\circ(\omega, 1)$ can also be expressed by the components of the

Amazon EX1010 Page 1094

particle velocity in the x- and y-directions. The connection between sound pressure and particle velocity is established by transforming Newton's law of motion (53.4) for two spatial coordinates into the frequency domain

$$\operatorname{grad} P_\mathrm{p}(\boldsymbol{r},\omega) + \mathrm{i}kz_0 \boldsymbol{V}_\mathrm{p}(\boldsymbol{r},\omega) = 0 \tag{53.30}$$

with the free-field impedance $z_0 = \rho c$ and the Fourier transform of the particle velocity $\boldsymbol{V}_\mathrm{p}(\boldsymbol{r},\omega)$. Considering only the component in the radial direction gives

$$\frac{\partial}{\partial r} P_\mathrm{p}(\boldsymbol{r},\omega) + \mathrm{i}kz_0 V_{\mathrm{p},r}(\boldsymbol{r},\omega) = 0\,, \tag{53.31}$$

where $V_{\mathrm{p},r}(\boldsymbol{r},\omega)$ is the radial component of $\boldsymbol{V}_\mathrm{p}(\boldsymbol{r},\omega)$.

For $r = 0$ and $\alpha = 0,\ \frac{\pi}{2}$, it follows [using the notation from (53.22)]

$$\left.\frac{\partial}{\partial r} P_\mathrm{p}(r,0,\omega)\right|_{r=0} = -\mathrm{i}kz_0 V_{\mathrm{p},r}(0,0,\omega)\,,$$
$$\left.\frac{\partial}{\partial r} P_\mathrm{p}(r,\frac{\pi}{2},\omega)\right|_{r=0} = -\mathrm{i}kz_0 V_{\mathrm{p},r}(0,\frac{\pi}{2},\omega)\,.$$

Figure 53.1 shows that the radial components of the particle velocity $V_{\mathrm{p},r}(0,\alpha,\omega)$ in the direction $\alpha = 0$ and $\alpha = \frac{\pi}{2}$ are equal to the x- and y-components of $\boldsymbol{V}_\mathrm{c}(\boldsymbol{x},\omega)$ at the origin of the Cartesian coordinate system

$$\left.\frac{\partial}{\partial r} P_\mathrm{p}(r,0,\omega)\right|_{r=0} = -\mathrm{i}kz_0 V_{\mathrm{c},x}(0,0,\omega) \tag{53.32}$$
$$\left.\frac{\partial}{\partial r} P_\mathrm{p}(r,\frac{\pi}{2},\omega)\right|_{r=0} = -\mathrm{i}kz_0 V_{\mathrm{c},y}(0,0,\omega)\,. \tag{53.33}$$

Now the real and imaginary part of $G^\circ(\omega,1)$ can be identified from (53.29) with the x- and y-components of $\boldsymbol{V}_\mathrm{c}(0,0,\omega)$ as

$$z_0 V_{\mathrm{c},x}(0,0,\omega) = -G^\circ_\mathrm{R}(\omega,1) \tag{53.34}$$
$$z_0 V_{\mathrm{c},y}(0,0,\omega) = G^\circ_\mathrm{I}(\omega,1)\,. \tag{53.35}$$

Equations (53.28), (53.34), and (53.35) show the following physical interpretation of the angular coefficients for $\nu = 0, 1$: $G^\circ(\omega,0)$ is equal to the Fourier transform of the sound pressure and the real and imaginary parts of $G^\circ(\omega,1)$ are equal (up to a sign change) to the Fourier transforms of the x- or y-components of the particle velocity, where both sound pressure and particle velocity are measured at the origin of the coordinate system.

53.2.5 Spherical Waves

Arbitrary solutions of the wave equation with homogeneous boundary conditions are described in terms of Green's functions, which can be regarded as the response of a sound field to an impulse in time and space. They are the equivalent to impulse responses for one-dimensional (1-D) systems.

The Green's function $G_\mathrm{3D}(\boldsymbol{z}|\boldsymbol{z}',\omega)$ describes the contribution of a point source at position $\boldsymbol{z}'$ to the sound field at position $\boldsymbol{z}$. The position $\boldsymbol{z}$ is also referred to as the listener position. In the three-dimensional free field it is given by [53.12]

$$G^\mathrm{f}_\mathrm{3D}(\boldsymbol{z}|\boldsymbol{z}',\omega) = \frac{1}{4\pi}\,\frac{\mathrm{e}^{-\mathrm{i}\omega\|\boldsymbol{z}-\boldsymbol{z}'\|/c}}{\|\boldsymbol{z}-\boldsymbol{z}'\|}\,, \tag{53.36}$$

where $\|\boldsymbol{z}-\boldsymbol{z}'\|$ denotes the Euclidian distance between $\boldsymbol{z}$ and $\boldsymbol{z}'$. This Green's function describes a spherical wave and is also called the free-field Green's function. The denominator accounts for the decay of the amplitude over distance and the exponential term accounts for the time delay of the propagating spherical wave. Other forms of Green's functions exist for other types of spatial impulses, e.g., line sources.

53.2.6 The Kirchhoff–Helmholtz Integral

The *Kirchhoff–Helmholtz integral* or *Helmholtz integral equation* provides the relation between the sound field inside a volume of arbitrary shape and on the enclosing boundary. It is presented here because it is the key element of the wave field synthesis principle.

The Kirchhoff–Helmholtz integral expresses the sound pressure $P(\boldsymbol{z},\omega)$ inside a volume V by an integral

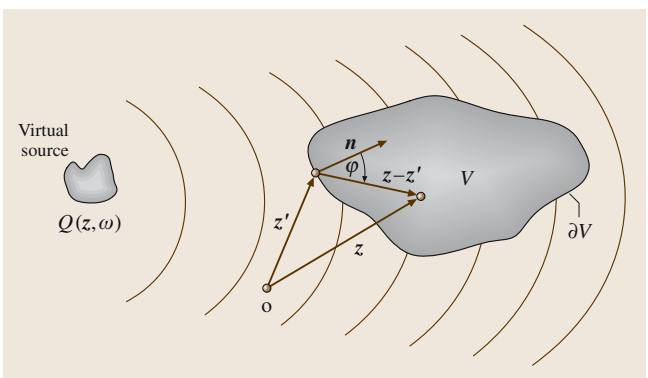


Fig. 53.2 Reproduction of the spatial wave field emitted by the virtual source $Q(\boldsymbol{z},\omega)$. The coordinate vector $\boldsymbol{z}$ addresses any arbitrary point inside the volume V, while $\boldsymbol{z}'$ denotes points on the boundary ∂V. $\boldsymbol{n}$ is the normal vector on ∂V and φ the angle between $\boldsymbol{n}$ and $\boldsymbol{z}-\boldsymbol{z}'$

Amazon EX1010 Page 1095

on the surface ∂V [53.2,4,12,13]

$$\oint_{\partial V} P(z',\omega)\,\frac{\partial}{\partial \boldsymbol{n}} G_{3\mathrm{D}}(z|z',\omega)\,\mathrm{d}z' - \oint_{\partial V} G_{3\mathrm{D}}(z|z',\omega)\,\frac{\partial}{\partial \boldsymbol{n}} P(z',\omega)\,\mathrm{d}z' = \begin{cases} P(z,\omega)\,, & z \in V \\ 0\,, & z \notin V \end{cases} . \tag{53.37}$$

$G_{3\mathrm{D}}(z|z',\omega)$ is a Green's function that satisfies suitable homogeneous boundary conditions on ∂V. Figure 53.2 explains the spatial coordinates.

The Kirchhoff–Helmholtz integral states that at any point within the source-free region V the sound pressure $P(z,\omega)$ can be calculated if both the sound pressure $P(z',\omega)$ and its directional gradient $\frac{\partial}{\partial \boldsymbol{n}} P(z',\omega)$ are known on the boundary ∂V enclosing the volume. The direction $\boldsymbol{n}$ of the gradient points inward from the boundary ∂V. This boundary does not necessarily have to be a real physical existing surface.

53.3 Stereophony

Stereophony comprises a number of techniques for reproduction with two or more loudspeakers. They describe how to feed the loudspeakers with appropriate signals to create suitable spatial cues at the ears of a listener. These cues are expressed as interaural level and time differences and by other psychoacoustic principles [53.14]. Overviews on stereophonic techniques are given, e.g., in [53.15–17].

Stereophonic sound reproduction is covered only briefly here. The positioning of a phantom source between two loudspeakers is addressed on the basis of a purely physical description of the created sound field. Such an approach is not capable of describing all psychoacoustic effects encountered in practice. It merely serves as an introduction to the multichannel methods in the subsequent Sections.

Figure 53.4 shows a listener in the center of a coordinate system and two loudspeakers in the direction of the angles θ_0 and $-\theta_0$ to the right and left. The distance of the loudspeakers is not given explicitly, since it is assumed to be large enough that their sound fields at the position of the listener can be well approximated by a plane wave.

To reproduce the image of a sound source at some angle θ with $|\theta| < \theta_0$, a technique called *amplitude panning* is applied. The same driving signal is fed to both loudspeakers, but with different weighting factors g_r and g_l. These are selected such that the superposition of the sound fields of both loudspeakers makes the listener perceive a single sound source at the desired angle θ. This perception is called a *phantom source* or *virtual source*. The functional dependency of the weighting factors $g_\mathrm{r}(\theta)$ and $g_\mathrm{l}(\theta)$ on θ is called a *panning law*.

The choice of a suitable panning law depends not only on the acoustics of the sound field but also on human spatial perception. Therefore, a unique determination of the panning law is not possible and different proposals exist in the literature [53.17]. Two of these panning laws are presented here briefly: the sine law and the tangent law.

53.3.1 Sine Law

The sine panning law is based on the model of a locally plane wave. The weighting factors are determined such that the wave field in the close vicinity of the listener can be approximated by a plane wave with a normal vector in the desired direction.

To this end consider first only the right loudspeaker. Feeding it with an arbitrary time signal produces a spherical wave. If the loudspeaker is sufficiently far away from the listener, then the spherical wave can be approximated by a plane wave at the origin of the form of (53.9). Further it is assumed that the origin of the time axis is adjusted such that any delay terms due to the propagation delay from the loudspeaker to the listener vanish. The component of $P(z,\omega)$ in the direction of the unit vector $\boldsymbol{e}_x = [1\,0\,0]^\mathrm{T}$ in the x-direction is given by

$$P_{\mathbf{c},x}(x,\omega) = G(\omega,\theta_0)\,\mathrm{e}^{\mathrm{i}\boldsymbol{k}^\mathrm{T}x\boldsymbol{e}_x} = G(\omega,\theta_0)\,\mathrm{e}^{\mathrm{i}k_x x}\,. \tag{53.38}$$

where $k_x = k\sin\theta_0$ is the component of the wave vector $\boldsymbol{k}$ in the x-direction. Equation (53.38) represents a plane wave since the argument of the exponential function is linear in x.

Now consider the superposition of the sound fields of the two speakers fed by the same signal $G(\omega,\theta_0) = G(\omega,-\theta_0)$, but with an individual weighting factor g_r and g_l for the right and left loudspeaker, respectively.

Amazon EX1010 Page 1096

The resulting sound field is

$$P_{\mathbf{c},x}(x,\omega) = G(\omega,\theta_0)\hat{p}_{\mathbf{c},x}(x) \tag{53.39}$$

with

$$\begin{aligned}\hat{p}_{\mathbf{c},x}(x) &= g_\mathrm{r}\,\mathrm{e}^{\mathrm{i}k_x x} + g_\mathrm{l}\,\mathrm{e}^{-\mathrm{i}k_x x} \\ &= (g_\mathrm{r}+g_\mathrm{l})\cos k_x x + j(g_\mathrm{r}-g_\mathrm{l})\sin k_x x \\ &= |\hat{p}_{\mathbf{c},x}(x)|\cdot \mathrm{e}^{\mathrm{i}\varphi(x)}\end{aligned} \tag{53.40}$$

The phase term $\varphi(x)$ of $\hat{p}(x)$ is not linear in x

$$\varphi(x) = \arctan\left[\frac{g_\mathrm{r}-g_\mathrm{l}}{g_\mathrm{r}+g_\mathrm{l}}\tan(k_x x)\right]. \tag{53.41}$$

However, close to the origin the arguments of both the tangent and the arc tangent are small and both functions may be approximated by their arguments. Then $\varphi(x)$ becomes, with $k_x = k\sin\theta_0$,

$$\varphi(x,\omega) \approx kx\left(\frac{g_\mathrm{r}-g_\mathrm{l}}{g_\mathrm{r}+g_\mathrm{l}}\sin\theta_0\right). \tag{53.42}$$

This expression is linear in x and thus represents a plane wave in the region where the approximation holds. According to Fig. 53.3 this is the area around the listeners head. Denoting the normal direction of this plane wave with θ, its corresponding angle is given by

$$\sin\theta = \frac{g_\mathrm{r}-g_\mathrm{l}}{g_\mathrm{r}+g_\mathrm{l}}\sin\theta_0 \tag{53.43}$$

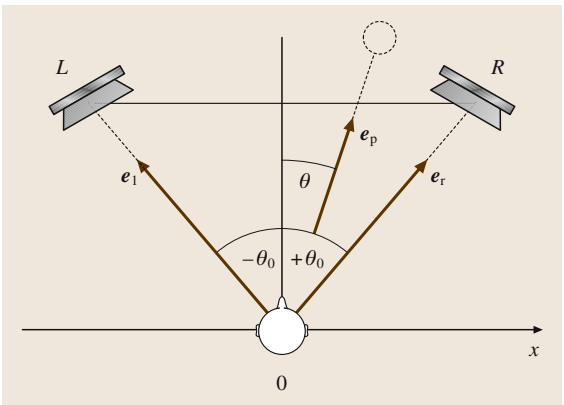


Fig. 53.3 Listener in front of two loudspeakers for two-channel stereophonic reproduction. The angle between the loudspeakers and the look direction of the listener is denoted by $\pm\theta_0$. The angle of a phantom source (*dashed lines*) relative to the listener is given by θ. The unit vectors in the direction of the loudspeakers and the phantom source are $\boldsymbol{e}_\mathrm{l}$, $\boldsymbol{e}_\mathrm{r}$, and $\boldsymbol{e}_\mathrm{p}$, respectively

for gain values $0 \le g_\mathrm{r}, g_\mathrm{l} \le 1$. The resulting panning law

$$\frac{g_\mathrm{r}-g_\mathrm{l}}{g_\mathrm{r}+g_\mathrm{l}} = \frac{1-\frac{g_\mathrm{l}}{g_\mathrm{r}}}{1+\frac{g_\mathrm{l}}{g_\mathrm{r}}} = \frac{\sin\theta}{\sin\theta_0} \tag{53.44}$$

is called the *sine law of stereophony* [53.16], and allows one to calculate the relationship between the gain factors g_r, g_l.

53.3.2 Tangent Law

The tangent panning law is based on a projection of the phantom source at angle θ to a coordinate system represented by the unit vectors $\boldsymbol{e}_\mathrm{r}$ and $\boldsymbol{e}_\mathrm{l}$ in the direction of the right and left loudspeaker (Fig. 53.3). These unit vectors do not need to be orthogonal but they must not be collinear. The gain factors for the left and right loudspeaker are then found as the corresponding projection coefficients.

The panning law follows from representing the unit vector in the direction of the virtual source $\boldsymbol{e}_\mathrm{p}$ by $\boldsymbol{e}_\mathrm{r}$ and $\boldsymbol{e}_\mathrm{l}$

$$\begin{aligned}\begin{pmatrix}\sin\theta\\ \cos\theta\end{pmatrix} &= g_\mathrm{l}\boldsymbol{e}_\mathrm{l} + g_\mathrm{r}\boldsymbol{e}_\mathrm{r} \\ &= g_\mathrm{l}\begin{pmatrix}\sin(-\theta_0)\\ \cos(-\theta_0)\end{pmatrix} + g_\mathrm{r}\begin{pmatrix}\sin\theta_0\\ \cos\theta_0\end{pmatrix} \\ &= \begin{pmatrix}(g_\mathrm{r}-g_\mathrm{l})\sin\theta_0\\ (g_\mathrm{r}+g_\mathrm{l})\cos\theta_0\end{pmatrix}\end{aligned} \tag{53.45}$$

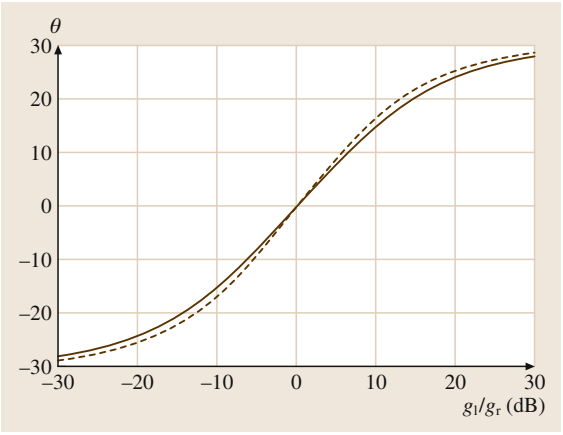


Fig. 53.4 Location θ of a phantom source versus the relation of the gain values $g_\mathrm{l}/g_\mathrm{r}$ for the sine (*solid line*) and tangent (*dashed line*) panning laws and for $\theta_0 = 30°$ (Fig. 53.3)

Amazon EX1010 Page 1097

and by division of $\sin\theta$ through $\cos\theta$

$$\frac{g_\mathrm{r}-g_\mathrm{l}}{g_\mathrm{r}+g_\mathrm{r}}=\frac{1-\frac{g_\mathrm{l}}{g_\mathrm{r}}}{1+\frac{g_\mathrm{l}}{g_\mathrm{r}}}=\frac{\tan\theta}{\tan\theta_0}. \tag{53.46}$$

Figure 53.4 shows the angle θ of a phantom source depending on the relation of the gain coefficients $g_\mathrm{l}/g_\mathrm{r}$ according to the sine and tangent panning laws and a loudspeaker angle $\theta_0 = 30°$. Obviously, there are only minor differences between the panning laws.

53.3.3 Application of Amplitude Panning

The derivation of these two panning laws is based on idealized physical arguments and does not take into account the human spatial perception. However, it has been confirmed by experiments that not only does amplitude panning lead to the perception of a phantom source between the loudspeakers, but also that the relation between the perceived source angle and the gain factors is described reasonably well by the panning laws (see [53.16, Fig. 3.4]). For a discussion of the application of these panning laws in two- or three-channel stereo see [53.18].

There are two basic methods for the determination of the weighting factors g_r and g_l, either by the sound engineer at the mixing console or through an appropriate microphone setup for the recording of an auditory event.

Mixing consoles usually contain a device called panorama potentiometer (panpot). It is fed with a mono signal $g(t)$ and creates two weighted outputs $g_\mathrm{r}g(t)$ and $g_\mathrm{l}g(t)$. The values of g_r and g_l are not determined numerically. Rather the sound engineer varies the position of the panorama potentiometer until the position of the virtual source meets his intentions.

In live recording of auditory events the gain factors are created by the directional sensitivity of microphones. Consider placing two directional microphones almost at the same spot in such a way that the direction of their maximum sensitivity points in different directions. Then both microphones will record the same sound signal, however with different amplitude, i. e., with different gain factors. A certain spacing of the microphones introduces a delay between the microphone signals which can augment the spatial impression at the side of the listener. In practice, the choice of the directional microphones and their placement and orientation depends not on analytical formulas but on the art of the Tonmeister [53.17].

The panning laws presented here for two-channel stereo apply also to other stereophonic reproduction formats, such as surround sound or the so-called 5.1 format. However, not all additional loudspeakers are used for enhancing the spatial localization of virtual sound sources. The 5.1 format uses one channel for low-frequency effects with no spatial localization capabilities and the two rear loudspeakers for ambiance effects. The localization properties of the remaining three front loudspeakers are similar to two-channel stereo, except that the center speaker enhances the stability of the front image [53.16].

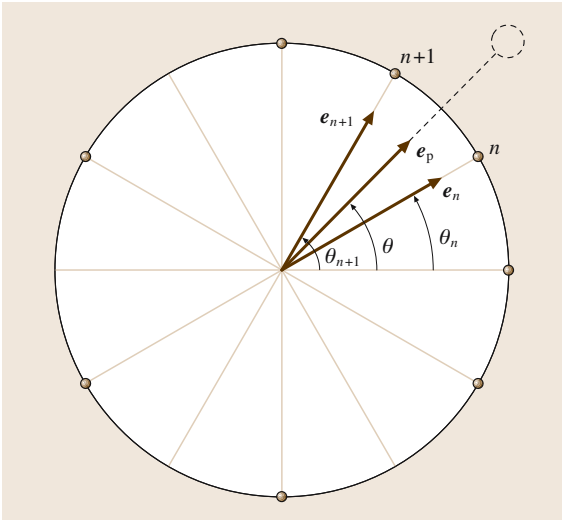


Fig. 53.5 Loudspeaker arrangement for two-dimensional vector-based amplitude panning. The angle between the loudspeaker n and an arbitrary reference direction is denoted by θ_n. The angle of a phantom source (*dashed lines*) is given by θ

53.3.4 The Sweet Spot

The derivation of the panning laws relies on some idealizations. It has been assumed that there is only one listener and that the position of the listener does not change. This ideal position which corresponds the origin of the coordinate system in Fig. 53.3 is called the *sweet spot*. Outside of this distinguished position, the assumptions for the panning laws do not hold. In addition, reflections in the listening rooms have been neglected.

For listeners outside of the sweet spot, the localization of virtual sources degrades. Close to the left or right loudspeaker, all sources are perceived at the location of this loudspeaker. The only way to avoid this problem is to increase the number of independent reproduction channels significantly. The multichannel audio reproduction methods to be discussed next, aim at either enlarging the sweet spot or extending the correct localization to the whole listening area.

Amazon EX1010 Page 1098

53.4 Vector-Based Amplitude Panning

Vector-based amplitude panning (VBAP) is a multichannel audio reproduction method that uses amplitude panning just like in stereophony. The difference is that panning is not only applied to two loudspeakers, but to two or three adjacent loudspeakers out of an arbitrary number distributed on a circle or a sphere around the listener. In this way, virtual sources can appear on a full horizontal circle or anywhere in space. The theory and application of vector-based amplitude panning was first described in [53.19–23].

Vector-based amplitude panning extends the tangent panning law for two loudspeakers to panning between adjacent speakers of a one- or two-dimensional loudspeaker array. To express this extension in mathematical terms, reconsider the tangent panning law from Fig. 53.3. The projection of the unit vector $\boldsymbol{e}_\mathrm{p}$ in the direction of the virtual source can be written in vector form as

$$\boldsymbol{e}_\mathrm{p} = \boldsymbol{M}\boldsymbol{g} \tag{53.47}$$

with

$$\boldsymbol{e}_\mathrm{p} = \begin{pmatrix} \sin\theta \\ \cos\theta \end{pmatrix}, \qquad \boldsymbol{g} = \begin{pmatrix} g_\mathrm{l} \\ g_\mathrm{r} \end{pmatrix}, \tag{53.48}$$

$$\boldsymbol{M} = (\boldsymbol{e}_\mathrm{l}\ \ \boldsymbol{e}_\mathrm{r}) = \begin{pmatrix} -\sin\theta_0 & \sin\theta_0 \\ \cos\theta_0 & \cos\theta_0 \end{pmatrix}. \tag{53.49}$$

The gain factors follow from

$$\boldsymbol{g} = \boldsymbol{M}^{-1}\boldsymbol{e}_\mathrm{p} \tag{53.50}$$

after some trigonometric manipulations as

$$g_\mathrm{l} = \frac{\sin(\theta-\theta_0)}{\sin(2\theta_0)}, \quad g_\mathrm{r} = \frac{\sin(\theta+\theta_0)}{\sin(2\theta_0)}. \tag{53.51}$$

Thus the gain factors are the relation of the sine values of two angles: the angle between the virtual source and the respective loudspeaker ($\theta \pm \theta_0$) and the angle between both loudspeakers ($2\theta_0$).

53.4.1 Two-Dimensional Vector-Based Amplitude Panning

The determination of the gain factors for two-dimensional vector-based amplitude panning can be obtained in the same way. Figure 53.5 shows a general loudspeaker setup in a plane. Each loudspeaker is characterized by its angle θ_n relative to the center. The distance of the loudspeakers from the center does not matter since it is assumed that it is large enough that the wavefront from each loudspeaker at the center can be approximated by a plane wave. Consequently the virtual source is also only determined by its angle θ.

Now assume that the loudspeakers n and $n+1$ shall reproduce weighted signals to create a virtual source in between, i. e., $\theta_n \le \theta \le \theta_{n+1}$. The weighting factors g_n and g_{n+1} are obtained from a set of equations similar to (53.47)

$$\boldsymbol{e}_\mathrm{p} = \boldsymbol{M}_n\boldsymbol{g}_n \tag{53.52}$$

with

$$\boldsymbol{M}_n = \begin{pmatrix} \sin\theta_n & \sin\theta_{n+1} \\ \cos\theta_n & \cos\theta_{n+1} \end{pmatrix}, \ \boldsymbol{g}_n = \begin{pmatrix} g_n \\ g_{n+1} \end{pmatrix} \tag{53.53}$$

as

$$g_n = \frac{\sin(\theta_{n+1}-\theta)}{\sin(\theta_{n+1}-\theta_n)}, \quad g_{n+1} = \frac{\sin(\theta-\theta_n)}{\sin(\theta_{n+1}-\theta_n)}. \tag{53.54}$$

Now consider the case that the position of the virtual source moves without being restricted to certain loudspeaker positions, i. e., $0 \le \theta < 2\pi$. Then only those two loudspeakers are active which enclose the direction θ of the virtual source. For $\theta = \theta_n$ only one loudspeaker is active ($g_n = 1$).

In detail, the weighting factors g_ν for two-dimensional vector-based amplitude panning with N loudspeakers are given by

$$g_\nu(\theta) = \begin{cases} \frac{\sin(\theta_{n+1}-\theta)}{\sin(\theta_{n+1}-\theta_n)} & \nu = n \\ \frac{\sin(\theta-\theta_n)}{\sin(\theta_{n+1}-\theta_n)} & \nu = n+1 \\ 0 & \text{otherwise} \end{cases}. \tag{53.55}$$

Here, n denotes the current position of the virtual source such that $\theta_n \le \theta \le \theta_{n+1}$.

Just as the angle θ is computed modulo 2π, so the indices n of the adjacent loudspeakers are computed modulo N. For more than one virtual source the loudspeaker signals are a superposition of the panned signals for each single source.

Figure 53.6 shows the weighting functions for a two-dimensional vector-based amplitude panning system with five loudspeakers. Obviously, two-dimensional vector-based amplitude panning is the same as stereo panning, only that the position of the pair of active loudspeakers moves with the sound source.

Amazon EX1010 Page 1099

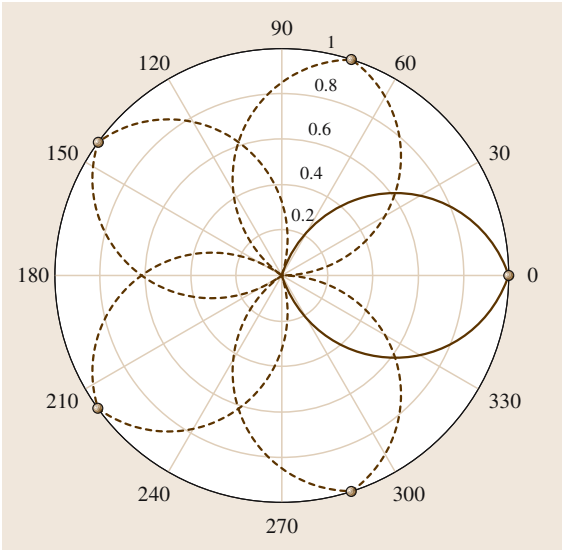


Fig. 53.6 Weighting coefficients for a two-dimensional vector-based amplitude panning system with five equally spaced loudspeakers

53.4.2 Three-Dimensional Vector-Based Amplitude Panning

The extension of vector-based amplitude panning to three spatial dimensions is straightforward. At first, a suitable loudspeaker configuration has to be chosen. As in the two-dimensional case, there is no special spatial distribution and no fixed number of loudspeakers. In [53.19], it is proposed to triangulate the sphere around the listener and to put one loudspeaker at each vertex. For any direction of a desired virtual source, there will be three loudspeakers forming a triangle which contains the source directions. As special cases the virtual source may lie between two loudspeakers or directly in the direction of one loudspeaker.

The virtual source is created by amplitude panning between the three loudspeakers of the corresponding triangle. The three weighting factors are again determined from a projection of the unit vector $\boldsymbol{e}_\mathrm{p}$ in the direction of the virtual source to the unit vectors $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, $\boldsymbol{e}_3$ in the direction of the three loudspeakers. Solving the linear system of equations

$$\boldsymbol{e}_\mathrm{p} = \boldsymbol{M}\boldsymbol{g} \tag{53.56}$$

with

$$\boldsymbol{M} = \left(\boldsymbol{e}_1 \;\; \boldsymbol{e}_2 \;\; \boldsymbol{e}_3\right) \tag{53.57}$$

for the vector $\boldsymbol{g}$ yields the unknown weighting factors. The unit vectors $\boldsymbol{e}_\mathrm{p}$, $\boldsymbol{e}_1$, $\boldsymbol{e}_2$, and $\boldsymbol{e}_3$ have to be specified in a suitable three-dimensional coordinate system.

53.4.3 Perception of Vector-Based Amplitude Panning

So far, the virtual source has been considered as a well-defined subject. This assumption was useful to derive the various panning laws on a mathematical basis. Whether human listeners actually perceive a virtual source at the intended location can only be inferred from listening tests. Results reported in [53.22, 23] indicate that the panning laws for two- and three-dimensional vector-based amplitude panning do indeed correlate well with the human perception of the virtual source.

However, the localization of a virtual source depends on its targeted position relative to the adjacent speakers. The localization is most precise if the virtual source direction coincides with the position of a loudspeaker. In this case the loudspeaker is perceived as a real source. For virtual source position in between adjacent loudspeakers, a certain spread of the localization occurs. Methods to achieve a uniform spreading of amplitude panned virtual sources have been presented in [53.20].

53.5 Ambisonics

Ambisonics describes a family of methods that have been developed since the 1970s [53.24]. Like vector-based amplitude panning, Ambisonics uses a flexible number of loudspeakers in some distance of the listener such that the superposition of their wave fields can be regarded as a superposition of plane waves. In contrast to vector-based amplitude panning a virtual source is not only panned between two or three loudspeakers but by all loudspeakers in the arrangement. Ambisonics is also suitable for the reproduction of wave fields with a more-general structure than a superposition of a finite number of sources. In combination with suitable recording methods, Ambisonics allows the reproduction of real wave fields within the limits of certain approximations. Detailed descriptions of Ambisonics can be found in [53.25, 26].

Amazon EX1010 Page 1100

53.5.1 Two-Dimensional Ambisonics

This section discusses the spatial reproduction with Ambisonics for planar loudspeaker arrays. First, panning of a plane wave from a certain direction is discussed, just as for stereophony and for vector based amplitude panning before. Then the reproduction of general wave fields is presented.

Panning Functions for the Reproduction of Plane Waves

Consider a two-dimensional loudspeaker setup as in Fig. 53.6. It is assumed that the loudspeakers produce sound waves from the directions θ_n that can be approximated by plane waves in the vicinity of the listeners head. A suitable superposition of these plane waves can approximate a virtual plane wave from an arbitrary direction θ. In contrast to vector-based amplitude panning not only two but all N loudspeakers contribute simultaneously to the superposition with appropriate weighting factors $g_n(\theta)$.

With the designation of the plane waves from different angles as in (53.11), this superposition is written as

$$P_\mathrm{p}(\boldsymbol{r}, \omega, \theta) = \sum_{n=0}^{N-1} P_\mathrm{p}(\boldsymbol{r}, \omega, \theta_n) \,. \tag{53.58}$$

The driving signal of each loudspeaker at θ_n is the waveform of the desired plane wave $g(t, \theta)$ multiplied by the weighting factor $g_n(\theta)$. The corresponding Fourier transforms are then related by

$$G(\omega, \theta_n) = g_n(\theta)\, G(\omega, \theta) \,. \tag{53.59}$$

Inserting (53.22) for θ and θ_n and (53.59) into (53.58) shows that $G(\omega, \theta)$ cancels and thus the panning functions $g_n(\theta)$ can be determined independently of the waveform $g(t, \theta)$

$$\hat{p}_\mathrm{p}(r, \alpha, \theta) = \sum_{n=0}^{N-1} g_n(\theta)\, \hat{p}_\mathrm{p}(r, \alpha, \theta_n) \,. \tag{53.60}$$

To derive conditions for the determination of the weighting factors, the functions in (53.60) are expanded into angular modes as in (53.17). Equating the terms for each mode ν on both sides gives the conditions

$$\mathrm{e}^{-\mathrm{i}\nu\theta} = \sum_{n=0}^{N-1} g_n(\theta)\, \mathrm{e}^{-\mathrm{i}\nu\theta_n} \quad \forall \nu \,. \tag{53.61}$$

This approach is called *mode matching* [53.27]. Solving (53.61) for the weighting factors yields the panning functions for a single source. A special case is discussed next.

Closed-Form Solution for the Panning Function

For special loudspeaker setups, the panning functions can be determined in closed form [53.27]. As an example, consider a setup of N loudspeakers with equidistant angular spacing

$$\theta_n = n\frac{2\pi}{N} \,, \qquad n = 0, \ldots, N-1 \,. \tag{53.62}$$

For convenience in the following derivation, N is assumed to be odd. A similar result is obtained for an even number of loudspeakers with a slightly different indexing of the loudspeakers (n) and modes (ν).

Now determine the N unknown weighting factors $g_n(\theta)$ for the reproduction of a plane wave from the N equations (53.61)

$$\mathrm{e}^{-\mathrm{i}\nu\theta} = \sum_{n=0}^{N-1} g_n(\theta)\, \mathrm{e}^{-\mathrm{i}\nu n \frac{2\pi}{N}} \,, \quad -N' \le \nu \le N' \tag{53.63}$$

where $N' = (N-1)/2$. Since N is odd, N' is an integer. Taking the sum of the terms on the left side

$$\frac{1}{N} \sum_{\nu=-N'}^{N'} \mathrm{e}^{-\mathrm{i}\nu\theta} \cdot \mathrm{e}^{\mathrm{i}\nu m \frac{2\pi}{N}} \tag{53.64}$$

and in the same way on the right side and considering the orthogonality of the complex exponentials

$$\frac{1}{N} \sum_{\nu=-N'}^{N'} \mathrm{e}^{\mathrm{i}\nu(m-n)\frac{2\pi}{N}} = \begin{cases} 1 & m = n \\ 0 & \text{otherwise} \end{cases} \tag{53.65}$$

gives the weighting factors $g_n(\theta)$ in closed form

$$g_n(\theta) = \frac{1}{N} \sum_{\nu=-N'}^{N'} \mathrm{e}^{\mathrm{i}\nu(\theta-\theta_n)} = \frac{\sin\left[N(\theta-\theta_n)/2\right]}{N \sin\left[(\theta-\theta_n)/2\right]} \,. \tag{53.66}$$

This result can also be obtained by expressing (53.63) as

$$\mathrm{e}^{-\mathrm{i}\nu\theta} = \mathrm{DFT}_N\left\{g_n(\theta)\right\} \,, \tag{53.67}$$

where DFT_N denotes the discrete Fourier transform of length N. Considering both $\mathrm{e}^{-\mathrm{i}\nu\theta}$ and $g_n(\theta)$ as periodic sequences of length N and taking the inverse of (53.67) gives the same result as (53.66).

Note from (53.66) that the panning function is nonzero not only for two or three but for all N loudspeakers. The panning functions may also be negative

Amazon EX1010 Page 1101

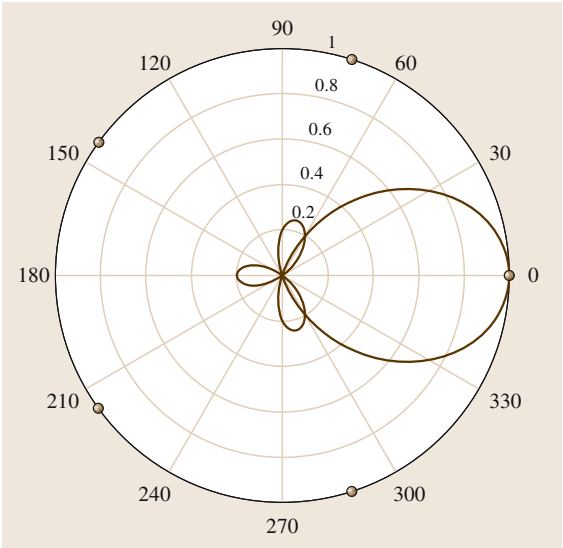


Fig. 53.7 Weighting function $g_0(\theta)$ for the loudspeaker channel $n = 0$ of a two-dimensional ambisonics system with five equally spaced loudspeakers

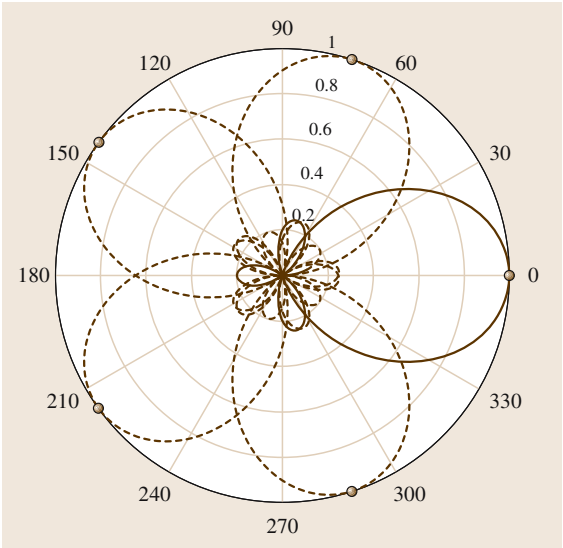


Fig. 53.8 Superposition of the weighting functions for a two-dimensional ambisonics system with five equally spaced loudspeakers

for some loudspeakers. Figure 53.7 shows the weighting factor $g_0(\theta)$ for $N = 5$. The superposition of all weighting factors for $n = 0, \ldots, 4$ is given in Fig. 53.8, which corresponds to Fig. 53.6 for vector-based amplitude panning.

Weighting Functions for General Wave Fields

Now a superposition of plane waves from the directions θ_n is used to represent a general solution of the wave equation

$$P_\mathrm{p}(\boldsymbol{r}, \omega) = \sum_{n=0}^{N-1} P_\mathrm{p}(\boldsymbol{r}, \omega, \theta_n)\,. \tag{53.68}$$

In contrast to (53.58), $P_\mathrm{p}(\boldsymbol{r}, \omega)$ is not constrained to be a plane wave. Still, mode matching can be applied here. Expanding $P_\mathrm{p}(\boldsymbol{r}, \omega)$ as in (53.24) and the plane waves $P_\mathrm{p}(\boldsymbol{r}, \omega, \theta_n)$ as in (53.17) and matching the factors of the modes $\mathrm{e}^{\mathrm{i}\nu\alpha}$ gives

$$G^\circ(\omega, \nu) = \sum_{n=0}^{N-1} G(\omega, \theta_n)\, \mathrm{e}^{\mathrm{i}\nu\theta}\,. \tag{53.69}$$

The angular coefficients $G^\circ(\omega, \nu)$ represent the spatial structure of the sound field to be reproduced. The real and imaginary parts of the corresponding time functions $g^\circ(t, \nu)$ are called *Ambisonics signals*. Their determination is discussed in later. The time functions $g_n(t) = g(t, \theta_n)$ corresponding to $G(\omega, \theta_n)$ are the driving functions of the loudspeakers. Theoretically, they are found by solving (53.69).

However, in practical applications, it is neither possible nor desirable to perform mode matching for a large number modes. If only few loudspeakers are available, then only a few of the Ambisonics signals are required to solve (53.69). If a large number of loudspeakers is available, then the modes of the sound field would have to be known with a sufficient accuracy. Therefore Ambisonics is usually restricted to a low number of modes.

First-Order Ambisonics

The most simple implementation of Ambisonics considers only the terms for $|\nu| \le 1$ of the expansion (53.26)

$$\begin{aligned} P_\mathrm{p}(\boldsymbol{r}, \omega) \approx\; & J_0(kr)\, G^\circ(\omega, 0) \\ & + 2\mathrm{i} J_1(kr)\, G^\circ_\mathrm{R}(\omega, 1) \cos\alpha \\ & - 2\mathrm{i} J_1(kr)\, G^\circ_\mathrm{I}(\omega, 1) \sin\alpha\,, \end{aligned} \tag{53.70}$$

Thus the first-order approximation of the sound field $P_\mathrm{p}(\boldsymbol{r}, \omega)$ can now be represented by the real values $G^\circ(\omega, 0)$, $G^\circ_\mathrm{R}(\omega, 1)$, and $G^\circ_\mathrm{I}(\omega, 1)$.

Second- and Higher-Order Ambisonics

Truncating the series (53.26) for values $|\nu| \le 2$ leads to an expansion similar to (53.70) but with additional terms involving the real and imaginary part of

Amazon EX1010 Page 1102

$G^\circ(\omega, 2)$ and the trigonometric terms $\cos 2\alpha$ and $\sin 2\alpha$. These additional terms provide higher angular resolution but only for larger distances of the origin, since the higher-order Bessel functions have small values for small values of kr. This method is called *second-order Ambisonics*.

The effect of increased angular resolution is more pronounced if terms for $2 < |\nu|$ are added. The corresponding *higher-order Ambisonics* methods are usually combined with a three-dimensional approach discussed in Sect. 53.5.2.

Determination of the Ambisonics Signals

The Ambisonics signals are the inverse Fourier transforms of the angular components $G^\circ(\omega, 0)$, $G^\circ_\mathrm{R}(\omega, \nu)$, and $G^\circ_\mathrm{I}(\omega, \nu)$ for $\nu > 1$. They contain the necessary temporal and spatial information about the sound field subject to the approximation induced by the series truncation discussed in Sect. 53.5.1. There are two principal methods for their determination, a model-based method and a data-based method relying on microphone recordings.

In the model-based method, the angular coefficients $G^\circ(\omega, \nu)$ are determined from a model of the desired sound field through an evaluation of (53.25). For example, modeling the sound field as a superposition of K plane waves from different directions θ_k, $k = 0, \ldots, K-1$ results in

$$G^\circ(\omega, \nu) = \sum_{k=0}^{K-1} G(\omega, \theta_k) \mathrm{e}^{-\mathrm{i}\nu\theta_k} \,. \tag{53.71}$$

This process is called *spatial encoding*.

The data-based method obtains the Ambisonics signals directly from suitable microphone recordings. As an example consider first-order Ambisonics according to (53.70). The terms $G^\circ_\mathrm{R}(\omega, 1)$ and $G^\circ_\mathrm{I}(\omega, 1)$ each correspond to a spatial *figure-of-eight* characteristic of pressure gradient microphones, differing by an angle of 90°. The term $G^\circ(\omega, 0)$ is not associated with any function of α and corresponds to the spatial characteristic of an omnidirectional microphone. Thus the first-order Ambisonics signals for an arbitrary sound field can be recorded by an arrangement of an omnidirectional and two pressure gradient microphones, where the look direction of the latter differs by 90°. The signals delivered by these microphones correspond directly to the sound pressure and the particle velocity components and thus to $G^\circ(\omega, 0)$, $G^\circ_\mathrm{R}(\omega, 1)$, and $G^\circ_\mathrm{I}(\omega, 1)$ according to (53.34) and (53.35).

The data-based method is not easily extended to second- or higher-order Ambisonics, due to the lack of higher-order microphones. Recording approaches with approximations of multipole microphones and with circular microphone arrays are discussed in [53.27–30].

Determination of the Loudspeaker Driving Signals

Ambisonics does not require that the loudspeaker configuration is known when determining the Ambisonics signals. Different loudspeaker configuration can be used for the same set of Ambisonics signals. However, the determination of the loudspeaker driving signals $g_n(t)$ is particularly simple if the loudspeakers are distributed at equal angular spacing, i. e., if the loudspeakers are positioned at $\theta_n = n\,2\pi/N$ for $n = 0, \ldots, N-1$. If the number of Ambisonics signals matches the number of loudspeakers then (53.69) turns into

$$\begin{aligned} G^\circ(\omega, \nu) &= \sum_{n=0}^{N-1} G(\omega, \theta_n) \, \mathrm{e}^{-\mathrm{i}\nu n \frac{2\pi}{N}} \\ &= \mathrm{DFT}_N\{G(\omega, \theta_n)\} \,. \end{aligned} \tag{53.72}$$

This relation is similar to (53.67), but now it is valid for general wave fields and not only for one plane wave. The Fourier transforms $G(\omega, \theta_n)$ of the driving signals then follow from the inverse DFT

$$G(\omega, \theta_n) = \mathrm{DFT}_N^{-1}\{G^\circ(\omega, \nu)\}. \tag{53.73}$$

The determination of the loudspeaker driving signals and the wave field they produce is now discussed in detail for first-order Ambisonics.

First-order Ambisonics for two spatial dimensions uses only the Ambisonics signals $G^\circ(\omega, 0)$, $G^\circ_\mathrm{R}(\omega, 1)$, and $G^\circ_\mathrm{I}(\omega, 1)$ and thus requires at least three loudspeakers for reproduction. A popular choice for simple planar Ambisonics is $N = 4$ [53.16]. In this case, (53.73) becomes

$$\begin{aligned} G(\omega, \theta_n) &= \mathrm{DFT}_4^{-1}\{G^\circ(\omega, \nu)\} \\ &= \frac{1}{4} G^\circ(\omega, 0) + \frac{1}{2} \mathrm{Re}\left\{ G^\circ(\omega, 1) \mathrm{e}^{\mathrm{i}n\frac{2\pi}{N}} \right\} \\ &= \frac{1}{4} G^\circ(\omega, 0) + \frac{1}{2} G^\circ_\mathrm{R}(\omega, 1) \cos\theta_n \\ &\quad - \frac{1}{2} G^\circ_\mathrm{I}(\omega, 1) \sin\theta_n \end{aligned} \tag{53.74}$$

or

$$
\begin{aligned}
G(\omega,\theta_0) &= \frac{1}{4}G^{\circ}(\omega,0)+\frac{1}{2}G^{\circ}_{\mathrm{R}}(\omega,1),\\
G(\omega,\theta_1) &= \frac{1}{4}G^{\circ}(\omega,0)-\frac{1}{2}G^{\circ}_{\mathrm{I}}(\omega,1),\\
G(\omega,\theta_2) &= \frac{1}{4}G^{\circ}(\omega,0)-\frac{1}{2}G^{\circ}_{\mathrm{R}}(\omega,1),\\
G(\omega,\theta_3) &= \frac{1}{4}G^{\circ}(\omega,0)+\frac{1}{2}G^{\circ}_{\mathrm{I}}(\omega,1)\,.
\end{aligned}
\tag{53.75}
$$

These driving signals for the loudspeakers actually reproduce those properties of the original wave field which have been encoded in the Ambisonics signals.

To verify this statement, consider the wave field $Q_{\mathrm{p}}(r,\alpha,\omega)$ that is produced as a superposition of $N=4$ plane waves from the directions $\theta_n = n\,\pi/2$ for $n=0,\ldots,3$. According to (53.19) and (53.22), $Q_{\mathrm{p}}(r,\alpha,\omega)$ is given by

$$
\begin{aligned}
&Q_{\mathrm{p}}(r,\alpha,\omega)\\
&=\sum_{n=0}^{3} G(\omega,\theta_n)\,\hat{p}_{\mathrm{p}}(r,\alpha,\theta_n)\\
&= J_0(kr)\sum_{n=0}^{3} G(\omega,\theta_n)\\
&\quad +2\sum_{\nu=1}^{\infty}\mathrm{i}^{\nu}J_{\nu}(kr)\sum_{n=0}^{3} G(\omega,\theta_n)\cos\left[\nu(\alpha-\theta_n)\right]\,.
\end{aligned}
\tag{53.76}
$$

Now consider the special values $Q_{\mathrm{p}}(0,\alpha,\omega)$ and $\frac{\partial}{\partial r}Q_{\mathrm{p}}(r,\alpha,\omega)\big|_{r=0}$. They follow from (53.76) with (53.27) as

$$
Q_{\mathrm{p}}(0,\alpha,\omega)=\sum_{n=0}^{3} G(\omega,\theta_n)
\tag{53.77}
$$

and

$$
\begin{aligned}
&\frac{\partial}{\partial r}Q_{\mathrm{p}}(r,\alpha,\omega)\bigg|_{r=0}\\
&=\mathrm{i}k\sum_{n=0}^{3} G(\omega,\theta_n)\cos(\alpha-\theta_n)\\
&=\mathrm{i}k[G(\omega,\theta_0)-G(\omega,\theta_2)]\cos\alpha\\
&\quad +\mathrm{i}k[G(\omega,\theta_1)-G(\omega,\theta_3)]\sin\alpha\,.
\end{aligned}
\tag{53.78}
$$

Inserting the loudspeaker signals from (53.75) gives

$$
Q_{\mathrm{p}}(0,\alpha,\omega)=G^{\circ}(\omega,0)=P_{\mathrm{p}}(0,\alpha,\omega)
\tag{53.79}
$$

and

$$
\begin{aligned}
&\frac{\partial}{\partial r}Q_{\mathrm{p}}(r,\alpha,\omega)\bigg|_{r=0}\\
&=\mathrm{i}k\left[G^{\circ}_{\mathrm{R}}(\omega,1)\cos\alpha-G^{\circ}_{\mathrm{I}}(\omega,1)\sin\alpha\right]
\end{aligned}
\tag{53.80}
$$

such that

$$
\frac{\partial}{\partial r}Q_{\mathrm{p}}(r,0,\omega)\bigg|_{r=0}=\frac{\partial}{\partial r}P_{\mathrm{p}}(r,0,\omega)\bigg|_{r=0}\,,
\tag{53.81}
$$

$$
\frac{\partial}{\partial r}Q_{\mathrm{p}}(r,\frac{\pi}{2},\omega)\bigg|_{r=0}=\frac{\partial}{\partial r}P_{\mathrm{p}}(r,\frac{\pi}{2},\omega)\bigg|_{r=0}\,.
\tag{53.82}
$$

Comparing (53.28) and (53.79) and also (53.32) and (53.33) with (53.81) and (53.82) shows that the sound pressure and the components of the particle velocity are correctly reproduced at the position $r=0$.

For loudspeaker arrangements other than equal angular spacing, the system of equations (53.69) cannot be formulated conveniently as a DFT. However it is still possible to solve (53.69) for the loudspeaker signals g_n, either exact or in the least-squares sense depending on the relation between the number of Ambisonics signals and the number of loudspeakers [53.31].

53.5.2 Three-Dimensional Ambisonics

Three-dimensional Ambisonics systems are based on a three-dimensional loudspeaker arrangement, which produces approximately plane waves from arbitrary spatial directions. Three-dimensional Ambisonics is often considered in the context of higher-order Ambisonics [53.31–33]. The mathematical basis is a modal expansion similar to (53.24) and (53.26), where the angular expansion coefficients are defined on a sphere described in terms of an azimuth and an elevation angle. These modal expansion functions are called *spherical harmonics* and are more involved than the two-dimensional expansion from (53.24). Therefore only a short description is given here; for mathematical details see [53.3, 4, 34, 35].

The three-dimensional Ambisonics signals are the coefficients of the expansion into spherical harmonics. Similar to Sect. 53.5.1 they can be obtained by model- and data-based methods.

In the model-based approach, the Ambisonics signals are calculated from the geometrical and acoustical description of a virtual scene [53.36].

The simplest method for the data-based approach consists of an arrangement of four microphones, one omnidirectional microphone, and three velocity microphones at the same spot. The look directions of the

velocity microphones are orthogonal so that they cover all three spatial dimensions. This way only first-order Ambisonics signals can be recorded.

Ambisonics signals of higher order can be recorded with spherical microphone arrays, i. e., an arrangement of pressure microphones distributed evenly over a spherical baffle. Such arrangements can be built as compact as a single microphone and are sometimes called *sound field microphones*. The theory of spherical microphone arrays and applications to spatial sound recording are presented in [53.33, 37–39].

53.5.3 Extensions of Ambisonics

So far two- and three-dimensional Ambisonics has been described as a superposition of plane waves from a finite number of distant loudspeakers. Now, two extensions are briefly discussed.

The sound field of a loudspeaker is modeled more correctly as a spherical wave rather than a plane wave. Taking the curvature of the wavefront into account, not only the direction but also distance of a sound source can be encoded. A higher-order Ambisonics format based also on the near field of a source has been introduced [53.32] under the name near-field-coding higher-order Ambisonics (NFC-HOA).

Even for higher-order Ambisonics, the number of modes is usually rather moderate. A spherical array takes spatial samples of the sound field at the microphone positions. Therefore the number of modes that can be uniquely determined is restricted by spatial aliasing. In practical microphone arrangements, orders of up to five have been reported [53.38].

However, theoretical investigations have shown [53.40, 41] that infinite-order Ambisonics is a particular case of an approach called *holophony* or *acoustic holography*. A technique for an approximate realization of acoustic holography is described in the next section.

53.6 Wave Field Synthesis

Wave field synthesis is a sound reproduction technique that overcomes certain limitations of conventional surround sound methods. It is based on a physical description of the propagation of acoustic waves. Wave field synthesis uses loudspeaker array technology to correctly reproduce sound fields without the *sweet spot* limitation discussed above. Wave field synthesis techniques are formulated in terms of the acoustic wave equation and their Green's functions [53.13, 42, 43]. The following description is an abridged version of a more-detailed account in [53.44].

53.6.1 Description of Acoustical Scenes by the Kirchhoff–Helmholtz Integral

Consider the sound reproduction scenario depicted in Fig. 53.2. The wave field emitted by an arbitrary virtual source $Q(\boldsymbol{z}, \omega)$ shall be reproduced in the bounded region V. This region is called the *listening region*, since the listeners reside there. The virtual source $Q(\boldsymbol{z}, \omega)$ shall lie outside of the listening region V. The limitation to one virtual source poses no constraints on the wave field to be reproduced, since this source may have arbitrary shape and frequency characteristics. Additionally, multiple sources can be reproduced by the principle of superposition.

For this sound reproduction scenario the Green's function $G_{3D}(\boldsymbol{z}|\boldsymbol{z}', \omega)$ and its directional gradient can be understood as the field emitted by sources placed on the boundary ∂V. These sources are called *secondary sources*. The strength of these sources is determined by the pressure $P(\boldsymbol{z}', \omega)$ and the directional pressure gradient $\frac{\partial}{\partial \boldsymbol{n}} P(\boldsymbol{z}', \omega)$ of the virtual source field $Q(\boldsymbol{x}', \omega)$ on ∂V.

Now the Kirchhoff–Helmholtz integral (53.37) can be interpreted as follows: if appropriately chosen secondary sources are driven by the values of the sound pressure and the directional pressure gradient caused by the virtual source $Q(\boldsymbol{x}', \omega)$ on the boundary ∂V, then the wave field within the region V is equivalent to the wave field which would have been produced by the virtual source inside V. Thus, the theoretical basis of sound reproduction is described by the Kirchhoff–Helmholtz integral.

The three-dimensional free-field Green's function is given by (53.36). In the context of sound reproduction it can be interpreted as the field of a monopole point source distribution on the surface ∂V. The Kirchhoff–Helmholtz integral (53.37) also involves the directional gradient of the Green's function. The directional gradi-

ent of the three-dimensional free-field Green's function can be interpreted as the field of a dipole source whose main axis lies in direction of the normal vector $\boldsymbol{n}$. Thus, the Kirchhoff–Helmholtz integral states in this case, that the acoustic pressure inside the volume V can be controlled by a monopole and a dipole point source distribution on the surface ∂V enclosing the volume V.

This interpretation of the Kirchhoff–Helmholtz integral sketches a first draft of a technical system for spatial sound reproduction. In rough terms, such a system would consist of technical approximations of acoustical monopoles and dipoles by appropriate loudspeakers. These loudspeakers cover the surface of a suitably chosen volume around the possible listener positions. They are excited by appropriate driving functions to reproduce the desired sound field inside the volume. However, there remain a number of fundamental questions to be resolved on the way to a technical realization.

53.6.2 Monopole and Dipole Sources

Technical approximations of acoustical monopoles and dipoles consist of loudspeakers with different types of enclosures. A restriction to only one type of sources would be of advantage for a technical realization. For example the exclusive use of monopole sources facilitates a technical solution with small loudspeakers in closed cabinets.

Reconsidering (53.37) shows that the use of monopole and dipole sources allows a very precise reproduction of the desired wave field: it is recreated as $P(z, \omega)$ for all positions inside V and it is zero outside. Such a restriction is usually not required for spatial sound reproduction. As long as the reproduction is correct inside of V, almost arbitrary sound fields outside may be tolerated, as long as their sound intensity is moderate. This situation suggests the following trade-off: the use of one type of sound sources only and tolerate some sound pressure outside of V.

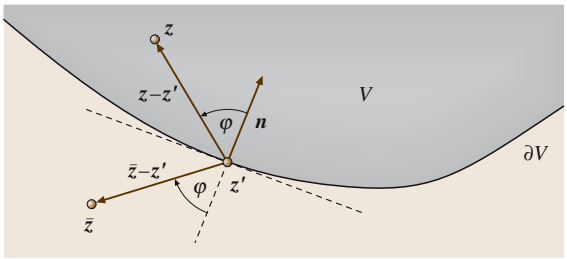


Fig. 53.9 Illustration of the geometry used for the derivation of the modified Green's function for a sound reproduction system using monopole secondary sources only

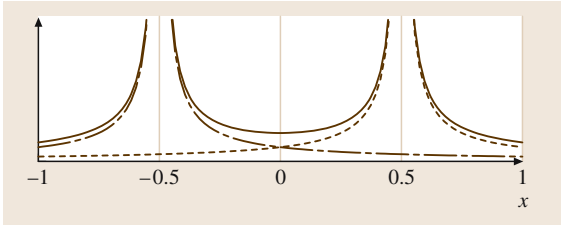


Fig. 53.10 Construction of a Green's function with zero derivative at the boundary $x = 0$ by superposition of the Green's functions of two point sources (only component in x-direction). Free-field source at $x = 0.5$ (*dashed*), mirrored free-field source at $x = -0.5$ (*dash-dotted*), superposition of two sources (*solid*). The *vertical axis* shows the absolute value of the Green's functions in arbitrary units

To realize this trade-off, a Green's function $G_{3D}(z|z', \omega)$ that satisfies boundary conditions of the first or second kind on the surface ∂V is constructed. Since it is desirable to drop the dipoles and to keep the monopole sources, the Green's function of a point source $G^{\mathrm{f}}_{3D}(z|z', \omega)$ according to (53.36) is chosen as the basic building block. Then for each position z' on the boundary, the Green's function $G^{\mathrm{f}}_{3D}(z|z', \omega)$ for a position z inside V and the Green's function $G^{\mathrm{f}}_{3D}(\bar{z}(z)|z', \omega)$ for a position $\bar{z}(z)$ outside V are superposed

$$G_{3D}(z|z', \omega) = G^{\mathrm{f}}_{3D}(z|z', \omega) + G^{\mathrm{f}}_{3D}(\bar{z}(z)|z', \omega)\,. \tag{53.83}$$

The position $\bar{z}(z)$ is chosen as the mirror image of z with respect to the tangent plane in z' on the surface ∂V (see Fig. 53.9). The tangent plane is characterized by the unit vector $\boldsymbol{n}$. The notation $\bar{z}(z)$ indicates that $\bar{z}$ depends on z.

Figure 53.10 exemplifies the construction of $G_{3D}(z|z', \omega)$. It shows a one-dimensional cut in the direction of $\boldsymbol{n}$. The superposition of two free-field Green's functions results in a zero normal derivative of $G_{3D}(z|z', \omega)$ at the boundary (Neumann boundary conditions).

Inserting $G_{3D}(z|z', \omega)$ from (53.83) as the Green's function in the Kirchhoff–Helmholtz integral (53.37) leads to

$$P(z, \omega) = -\oint_{\partial V} G_{3D}(z|z', \omega)\, \frac{\partial}{\partial \boldsymbol{n}} P(z', \omega)\, \mathrm{d}z' \tag{53.84}$$

for $z \in V$. Outside V the wave field consists of a mirrored version of the wave field inside V. An example for a cir-

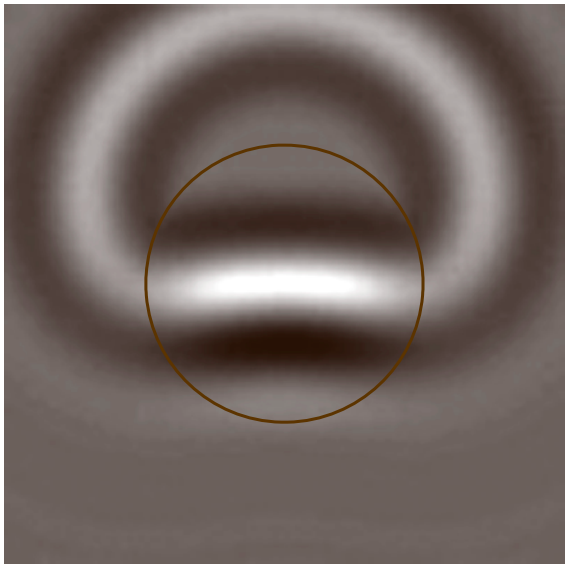


Fig. 53.11 Simulated sound field created by a distribution of monopoles on a *circle* (shown in *black*). Inside the circle: approximation of a plane wave, outside of the circle: mirrored sound field caused by a reflection of the plane wave from inside with respect to the *black circle*

cular boundary is shown in Fig. 53.11. The result (53.84) is also known as the type I Raleigh integral [53.35]. On the boundary ∂V, the following holds:

$$G_{3D}(z|z',\omega) = 2\,G^{f}_{3D}(z|z',\omega)\,. \tag{53.85}$$

The factor of two follows from the superposition shown in Fig. 53.10. Thus the sound field $P(z,\omega)$ inside a volume V can be generated by a distribution of monopole sources on the surface ∂V as

$$P(z,\omega) = -\oint_{\partial V} 2\,G^{f}_{3D}(z|z',\omega)\,\frac{\partial}{\partial \boldsymbol{n}}\,P(z',\omega)\,\mathrm{d}z'\,. \tag{53.86}$$

53.6.3 Reduction to Two Spatial Dimensions

The requirement of creating a distribution of sources on a whole surface around a listening space may be impractical for many sound reproduction systems. This Section shows how to reduce the source distribution from a surface around the listeners to a closed curve in a horizontal plane preferably at the height of the listeners' ears. For convenience this height is denoted by $z = 0$.

The Kirchhoff–Helmholtz integral is now specialized to sound fields that do not depend on the

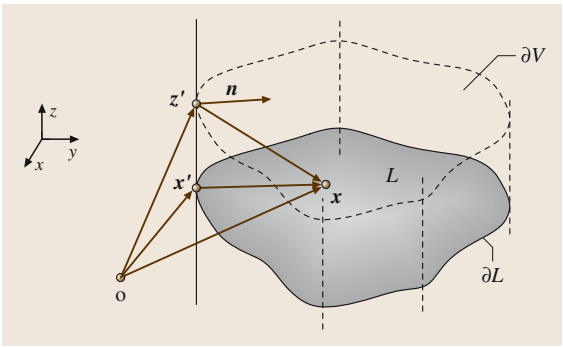


Fig. 53.12 Illustration of a sound field in a prism oriented parallel to the z-axis. For this special geometry, the intersection with any plane parallel to the xy-plane is independent of z. If the source strength on the boundary ∂V is also independent of z, the sound field within the prism does not depend on the z-coordinate

z-coordinate. Then the shape of the volume for the integration in (53.37) turns into a prism oriented parallel to the z-axis (Fig. 53.12). This rather special spatial arrangement allows the transition to a 2-D description of the Kirchhoff–Helmholtz integral.

Since the sound field is assumed to be independent of z, $P(z,\omega)$ depends only on x and y. Furthermore, any vector normal to the surface ∂V has no component in the z-direction and also the normal derivative of $P(z,\omega)$ is independent of z. Thus the surface integration with respect to z' in (53.37) can be split into a contour integration with respect to $\boldsymbol{x}'$ and an integration with respect to z'. The contour ∂L is defined by the intersection of the prism with the xy-plane. This procedure turns the surface integral in (53.86) into

$$\begin{aligned}
&\oint_{\partial V} G^{f}_{3D}(z|z',\omega)\,\frac{\partial}{\partial \boldsymbol{n}}\,P(z',\omega)\,\mathrm{d}z' \\
&= \oint_{\partial L}\int_{-\infty}^{\infty} G^{f}_{3D}(z|z',\omega)\,\frac{\partial}{\partial \boldsymbol{n}}\,P(z',\omega)\,\mathrm{d}z'\,\mathrm{d}\boldsymbol{x}' \\
&= \oint_{\partial L}\left[\int_{-\infty}^{\infty} G^{f}_{3D}(z|z',\omega)\,\mathrm{d}z'\right]\frac{\partial}{\partial \boldsymbol{n}}\,P(z',\omega)\,\mathrm{d}\boldsymbol{x}'\,.
\end{aligned} \tag{53.87}$$

Since $P(z',\omega)$ on the contour ∂L is described by a two-dimensional function $P(z',\omega) = P_c(\boldsymbol{x}',\omega)$, a two-dimensional version of the Raleigh I integral is given by

$$P_c(\boldsymbol{x},\omega) = -\oint_{\partial L} 2G^{f}_{2D}(\boldsymbol{x}|\boldsymbol{x}',\omega)\,\frac{\partial}{\partial \boldsymbol{n}}\,P_c(\boldsymbol{x}',\omega)\,\mathrm{d}\boldsymbol{x}' \tag{53.88}$$

Amazon EX1010 Page 1107

with

$$G^{\mathrm{f}}_{2\mathrm{D}}(\boldsymbol{x}|\boldsymbol{x}',\omega) = \int_{-\infty}^{\infty} G^{\mathrm{f}}_{3\mathrm{D}}(\boldsymbol{x}|z',\omega)\,\mathrm{d}z' \,. \tag{53.89}$$

Evaluating this integral gives [53.9, 3.876]

$$G^{\mathrm{f}}_{2\mathrm{D}}(\boldsymbol{x}|\boldsymbol{x}',\omega) = \tilde{G}^{\mathrm{f}}_{2\mathrm{D}}(\rho,\omega) = -\frac{\mathrm{i}}{4} H_0^{(2)}\Big(\frac{\omega}{c}\,\rho\Big) \tag{53.90}$$

with the Hankel function of the second kind and order zero [53.9] being

$$H_0^{(2)}(u) = J_0(u) - \mathrm{i}N_0(u) \,, \tag{53.91}$$

where $J_0(u)$ and $N_0(u)$ are, respectively, the Bessel and Neumann functions of the first kind and order zero.

Due to the circular symmetry, $G^{\mathrm{f}}_{2\mathrm{D}}(\boldsymbol{x}|\boldsymbol{x}',\omega)$ depends only on the distance ρ between the listener position $\boldsymbol{x}$ and the line source at $\boldsymbol{x}'$

$$\rho = \|\boldsymbol{x}-\boldsymbol{x}'\| = \sqrt{(x-x')^2+(y-y')^2} \tag{53.92}$$

and can thus be replaced by $\tilde{G}^{\mathrm{f}}_{2\mathrm{D}}(\rho)$ as in (53.90). With the far-field approximation of the Hankel function for large values of ρ [53.8]

$$H_0^{(2)}\Big(\frac{\omega}{c}\,\rho\Big) \approx \sqrt{\frac{2\mathrm{i}}{\pi\left(\frac{\omega}{c}\,\rho\right)}}\;\mathrm{e}^{-\mathrm{i}\left(\frac{\omega}{c}\,\rho\right)} \tag{53.93}$$

it follows that

$$\tilde{G}^{\mathrm{f}}_{2\mathrm{D}}(\rho,\omega) = H(\omega)\,A(\rho)\,G^{\mathrm{f}}_{3\mathrm{D}}(\boldsymbol{x}|\boldsymbol{x}',\omega)\,, \tag{53.94}$$

where

$$H(\omega) = \sqrt{\frac{c}{\mathrm{i}\omega}} = \sqrt{\frac{1}{\mathrm{i}k}} \tag{53.95}$$

causes a spectral shaping and

$$A(\rho) = \sqrt{2\pi\,\rho} \tag{53.96}$$

causes an amplitude modification of $G^{\mathrm{f}}_{2\mathrm{D}}(\boldsymbol{x}|\boldsymbol{x}',\omega)$.

Inserting (53.94) into (53.88) gives, with (53.92),

$$P_{\mathrm{c}}(\boldsymbol{x},\omega) = -\oint_{\partial L} G^{\mathrm{f}}_{3\mathrm{D}}(\boldsymbol{x}|\boldsymbol{x}',\omega)\,D(\boldsymbol{x}|\boldsymbol{x}',\omega)\,\mathrm{d}\boldsymbol{x}' \tag{53.97}$$

for $\boldsymbol{x} \in L$ where

$$D(\boldsymbol{x}|\boldsymbol{x}',\omega) = 2A\big(\|\boldsymbol{x}-\boldsymbol{x}'\|\big)\,H(\omega)\,\frac{\partial}{\partial \boldsymbol{n}}\,P_{\mathrm{c}}(\boldsymbol{x}',\omega) \,. \tag{53.98}$$

Here, $G^{\mathrm{f}}_{3\mathrm{D}}(\boldsymbol{x}|\boldsymbol{x}',\omega)$ denotes the Green's function of the monopole sources on the contour ∂L in the xy-plane. It describes the wave propagation in 3-D space, however, the receiver locations $\boldsymbol{x}$ (the listeners' ears) are assumed to reside in the xy-plane as well. $D(\boldsymbol{x}|\boldsymbol{x}',\omega)$ denotes the source signal of the monopoles.

53.6.4 Spatial Sampling

The previous sections showed how the rather general statement of the Kirchhoff–Helmholtz integral can be narrowed down to a model for a spatial reproduction system. A hypothetical distribution of monopole and dipole sources on a 2-D surface around the listener has been replaced by a distribution of monopoles on a 1-D contour in a horizontal plane in the height of the listeners' ears.

For a technical solution, this spatially continuous source distribution has to be replaced by an arrangement of a finite number of loudspeakers with a monopole-like source directivity. The resulting wave field is given by a modification of (53.97), where the integral is substituted by a sum over the discrete loudspeaker positions $\boldsymbol{x}'_n$

$$P_{\mathrm{c}}(\boldsymbol{x},\omega) \approx -\sum_n G^{\mathrm{f}}_{3\mathrm{D}}(\boldsymbol{x}|\boldsymbol{x}'_n,\omega)\,D(\boldsymbol{x}|\boldsymbol{x}'_n,\omega)\,\Delta x'_n \,, \tag{53.99}$$

where $\Delta x'_n$ is the length of the spatial increment between the samples $\boldsymbol{x}'_n$ and is not required to be equidistant. Replacing the continuous-space source distribution by a discrete-space arrangement of loudspeakers can be described as a spatial sampling process. The resulting aliasing effects are discussed, e.g., in [53.45, 46].

53.6.5 Determination of the Loudspeaker Driving Signals

The driving signals for the loudspeakers at positions $\boldsymbol{x}'_n$ follow from (53.98) by inverse Fourier transformation with respect to ω. They are independent of the position of the listener except for the amplitude modification $A(\|\boldsymbol{x}-\boldsymbol{x}'\|)$. In practical situations, the amplitude modification is set to a fixed position inside the volume V. Then the loudspeaker signals are completely independent of the listener's position. The result is an approximate correction of the amplitude deviations for different sources with different distances. However, all other components of $D(\boldsymbol{x}|\boldsymbol{x}'_n,\omega)$ do not depend on any information about listeners at all. So especially frequency compensation by $H(\omega)$ and the computation

Amazon EX1010 Page 1108

of $\partial/\partial \boldsymbol{n} P_c(\boldsymbol{x}'_n, \omega)$ can be performed correctly for all positions inside V.

The crucial point in the determination of the loudspeaker signals is the gradient $\frac{\partial}{\partial \boldsymbol{n}} P_c(\boldsymbol{x}'_n, \omega)$ at the loudspeaker positions. A naive approach is to measure the original sound field with properly positioned and oriented velocity microphones. However, those measurements would be only valid for a reproduction with exactly the same number and position of loudspeakers. More-versatile approaches are again model- and data-based methods.

In the model-based approach, the pressure gradient at the loudspeaker positions is determined from a model of the acoustic scene similar to Fig. 53.2. The simplest model is free-field propagation, where the loudspeaker signals follow from delayed and attenuated versions of the source signals. More-elaborate models take the room acoustics of the recording room into account. Details of the model-based determination of the loudspeaker signals are given, e.g., in [53.44, 47].

In the data-based approach, the spatial characteristics of the room acoustics in the recording room are determined by microphone measurements. From these, a set of room impulse responses is derived for use in wave field synthesis reproduction. Appropriate recording techniques are described, e.g., in [53.28, 48, 49].

References

53.1 L.J. Ziomek: *Fundamentals of Acoustic Field Theory and Space–Time Signal Processing* (CRC, Boca Raton 1995)

53.2 A.D. Pierce: *Acoustics. An Introduction to its Physical Principles and Applications* (Acoustical Society of America, Washington 1991)

53.3 P. Filippi, D. Habault, J.-P. Lefebvre, A. Bergassoli: *Acoustics: Basic Physics, Theory and Methods* (Academic, New York 1999)

53.4 E.G. Williams: *Fourier Acoustics: Sound Radiation and Nearfield Acoustical Holography* (Academic, New York 1999)

53.5 R.N. Bracewell: *The Fourier Transform and its Applications* (McGraw-Hill, New York 1978)

53.6 S. Haykin, B. VanVeen: *Signals and Systems* (Wiley, New York 1999)

53.7 B. Girod, R. Rabenstein, A. Stenger: *Signals and Systems* (Wiley, New York 2001)

53.8 M. Abramowitz, I.A. Stegun: *Handbook of Mathematical Functions* (Dover, London 1972)

53.9 I.S. Gradshteyn, I.M. Ryzhik: *Tables of Integrals, Series, and Products* (Academic, New York 1965)

53.10 I.N. Sneddon: *The Use of Integral Transforms* (McGraw-Hill, New York 1972)

53.11 R. Rabenstein, P. Steffen, S. Spors: Representation of two-dimensional wave fields by multidimensional signals, Signal Process. **86**, 1341–1351 (2006)

53.12 P.M. Morse, H. Feshbach: *Methods of Theoretical Physics. Part I* (McGraw-Hill, New York 1953)

53.13 S. Spors: *Active Listening Room Compensation for Spatial Sound Reproduction Systems* (University Erlangen-Nuremberg 2006) Ph.D. thesis http://www.lnt.de/lms/publications

53.14 J. Blauert: *Spatial Hearing: The Psychophysics of Human Sound Localization* (MIT, Cambridge 1996)

53.15 W.B. Snow: Basic principles of stereophonic sound, IRE Trans. Audio **March-April**, 42–53 (1955)

53.16 F. Rumsey: *Spatial Audio* (Focal, Oxford 2001)

53.17 C. Hugonnet, P. Walder: *Stereophonic Sound Recording* (Wiley, New York 1998)

53.18 D. Griesinger: Stereo and surround panning in practice, 112th Convention (AES 2002)

53.19 V. Pulkki: Virtual sound source positioning using vector base amplitude panning, J. Audio Eng. Soc. **45**(6), 456–466 (1997)

53.20 V. Pulkki: Uniform spreading of amplitude panned virtual sources, Proc. Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE, 1999)

53.21 V. Pulkki, M. Karjalainen, J. Huopaniemi: Analyzing sound source attributes using a binaural auditory model, J. Audio Eng. Soc. **47**(4), 203–217 (1999)

53.22 V. Pulkki, M. Karjalainen: Localization of amplitude-panned virtual sources I: Stereophonic panning, J. Audio Eng. Soc. **49**(9), 739–752 (2001)

53.23 V. Pulkki: Localization of amplitude-panned virtual sources II: Two- and three-dimensional panning, J. Audio Eng. Soc. **49**(9), 753–767 (2001)

53.24 M. Gerzon: Periphony: With-height sound reproduction, J. Audio Eng. Soc. **21**, 2–10 (1973)

53.25 J. Daniel: *Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia*, Ph.D. thesis, Université Paris 6 (2000) http://gyronymo.free.fr/audio3D/download_Thesis_PwPt.html

53.26 F. Hollerweger: Periphonic sound spatialization in multi-user virtual environments, M.S. thesis, University of Music and Dramatic Arts Graz (2005) http://iem.at/projekte/acoustic/awt/periphonic/periphonic.pdf

53.27 M.A. Poletti: A unified theory of horizontal holographic sound systems, J. Audio Eng. Soc. **48**(12), 1155–1182 (2000)

53.28 E. Hulsebos, T. Schuurmanns, D. de Vries, R. Boone: Circular microphone array recording for discrete

Amazon EX1010 Page 1109

multichannel audio recording, 114th AES Convention (AES, Amsterdam 2003)
53.29 H. Teutsch, W. Kellermann: On the estimation of the number of acoustic wideband sources using eigen-beam processing for circular apertures, Workshop on Applications of Signal Processingto Audio and Acoustics (IEEE, 2005)
53.30 H. Teutsch: *Modal Array Signal Processing: Principles and Applications of Acoustic Wavefield Decomposition* (Springer, Berlin 2007)
53.31 J. Daniel, R. Nicol, S. Moreau: Further investigations of high order Ambisonics and wave field synthesis for holophonic sound imaging, 114th AES Convention (AES, 2003), Convention Paper 5788
53.32 J. Daniel: Spatial sound encoding including near field effect: Introducing distance coding filters and a viable, new Ambisonics format, 23rd International Conference (AES, 2003)
53.33 M.A. Poletti: Three-dimensional surround sound systems based on spherical harmonics, J. Audio Eng. Soc. **11**(11), 1004–1025 (2005)
53.34 A.D. Pierce: Mathematical theory of wave propagation. In: *Handbook of Acoustics*, ed. by M.J. Crocker (Wiley, New York 1998)
53.35 N.A. Gumerov, R. Duraiswami: *Fast Multipole Methods for the Helmholtz Equation in three Dimensions* (Elsevier, Amsterdam 2004)
53.36 T. Betlehem, T.D. Abhayapala: Theory and design of sound field reproduction in reverberant rooms, J. Acoust. Soc. Am. **117**(4), 2100–2111 (2005)
53.37 T.D. Abhayapala, D.B. Ward: Theory and design of high order sound field microphones using spherical micrphone array, Proc. ICASSP (2002)
53.38 J. Meyer, G.W. Elko: Spherical microphone arrays for 3D sound recording. In: *Audio Signal Processing For Next-Generation Multimedia Communication Systems*, ed. by Y. Huang, J. Benesty (Kluwer Academic, Dordrecht 2004) pp. 67–89
53.39 B. Rafaely: Analysis and design of spherical microphone arrays, IEEE Trans. Speech Audio Process. **13**(1), 135–143 (2005)
53.40 R. Nicol, M. Emerit: Reproducing 3D-sound for videoconferencing: A comparison between holophony and ambisonic, Proc. Digital Audio Effects (DAFx) (1998) pp. 17–20
53.41 R. Nicol, M. Emerit: 3D-sound reproduction over an extensive listening area: A hybrid method derived from holophony and ambisonic, 16th International Conference (AES, 1999)
53.42 A.J. Berkhout: A holographic approach to acoustic control, J. Audio Eng. Soc. **36**, 977–995 (1988)
53.43 D. de Vries, E.W. Start, V.G. Valstar: The Wave Field Synthesis concept applied to sound reinforcement: Restrictions and solutions, 96th AES Convention (AES, Amsterdam 1994)
53.44 R. Rabenstein, S. Spors, P. Steffen: Wave field synthesis techniques for spatial sound reproduction. In: *Topics in Acoustic Echo and Noise Control*, ed. by E. Hänsler, G. Schmidt (Springer, Berlin, Heidelberg 2006) pp. 517–545
53.45 S. Spors: Spatial aliasing artefacts produced by linear loudspeaker arrays used for wave field synthesis, Proc. 2nd International Symposium on Communications, Control and Signal Processing (ISCCSP'06) (Marrakech, 2006)
53.46 S. Spors, R. Rabenstein: Spatial aliasing artifacts produced by linear and circular loudspeaker arrays used for wave field synthesis, 120th Convention (AES, Paris 2006)
53.47 A.J. Berkhout, D. de Vries: Acoustic holography for sound control, 86th Convention (AES, 1989), Preprint 2801 (F-3)
53.48 E. Hulsebos, D. de Vries, E. Bourdillat: Improved microphone array configurations for auralization of sound fields by Wave Field Synthesis, J. AES **50**(10), 779–790 (2002)
53.49 E. Hulsebos, D. de Vries: Parameterization and reproduction of concert hall acoustics with a circular microphone array, 112th AES Convention (AES, Munich 2002)

Acknowledgements

C.16 Low-Bit-Rate Speech Coding
by A. V. McCree

This work was sponsored by the Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

D.25 Expressive/Affective Speech Synthesis
by N. Campbell

The model arose from extended discussions during and after SP2006, and the author wishes to acknowledge contributions from Sacha Fagel of the TU Berlin. This work is partly supported by the Ministry of Public Management, Home Affairs, Posts and Telecommunications, Japan under the SCOPE funding initiative. The ESP corpus was collected over a period of five years with support from the Japan Science & Technology Corporation core research for evolutional science & technology (JST/CREST) funding initiative. The paper was written while the author was employed by the National Institute of Information and Communications Technology. The author also wishes to thank the management of the Spoken Language Communication Research Laboratory and the Advanced Telecommunications Research Institute International for their continuing support and encouragement of this work.

E.26 Historical Perspective of the Field of ASR/NLU
by L. Rabiner, B.-H. Juang

The authors thank Dr. John Makhoul and Prof. Victor Zue for their comments and insights on the material presented in this paper.

E.28 Speech Recognition with Weighted Finite-State Transducers
by M. Mohri, F. Pereira, M. Riley

We thank Andrej Ljolje for providing the acoustic models, and Don Hindle and Richard Sproat for providing the language models used in our experiments.

The first author's work was partially funded by the New York State Office of Science Technology and Academic Research (NYSTAR). The second author's work was partly funded by NSF grants EIA 0205456, EIA 0205448, and IIS 0428193.

E.30 Towards Superhuman Speech Recognition
by M. Picheny, D. Nahamoo

We would like to thank Brian Kingsbury, Lidia Mangu, Mukund Padmanabhan, and Geoff Zweig for their contributions to the research and many of the ideas in this chapter, and the entire human language technologies group at the IBM TJ Watson Research Center for creating a truly *Superhuman* environment for speech recognition research.

E.35 Spoken Dialogue Systems
by V. Zue, S. Seneff

The research described in this chapter is the result of collaboration with many past and current students and staff of the Spoken Language Systems Group at the MIT Laboratory for Computer Science, especially Jim Glass. Their contributions are gratefully acknowledged. This research is sponsored by the T-Party Project, a joint research program between MIT and Quanta Computer Inc.

F.38 Text-Independent Speaker Recognition
by D. A. Reynolds, W. M. Campbell

This work was sponsored by the Department of Defense under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States government.

G.41 Automatic Language Recognition Via Spectral and Token Based Approaches
by D. A. Reynolds, W. M. Campbell, W. Shen, E. Singer

This work was sponsored by the Department of Defense under Air Force contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States government.

I.52 Convolutive Blind Source Separation Methods
by M. S. Pedersen, J. Larsen, U. Kjems, L. C. Parra

M.S.P. was supported by the Oticon Foundation. M.S.P. and J.L. are partly also supported by the European Commission through the sixth framework IST Network of Excellence: pattern analysis, statistical modeling and computational learning (PASCAL).

About the Authors



Alex Acero

Microsoft Research
Redmond, WA, USA
alexac@microsoft.com

Chapter E.33

Dr. Acero is Research Area Manager at Microsoft Research, overseeing research in speech technology, natural language, computer vision, communication and multimedia collaboration. Dr. Acero is a Fellow of IEEE, VP for the IEEE Signal Processing Society and was Distinguished Lecturer. He is author of the two books, and over 140 technical papers. He holds 29 US patents. He obtained his Ph.D. from Carnegie Mellon, Pittsburgh, PA.

Jont B. Allen

University of Illinois
ECE
Urbana, IL, USA
JontAllen@ieee.org



Chapter A.3

Jont B. Allen received his BS in Electrical Engineering from the University of Illinois, Urbana-Champaign, in 1966, and his MS and Ph.D. in Electrical Engineering from the University of Pennsylvania in 1968 and 1970, respectively. After graduation he joined Bell Laboratories, and was in the Acoustics Research Department in Murray Hill NJ from 1974 to 1996, as a Distinguished member of Technical Staff. Since 1996 Dr. Allen was a Technology Leader at AT&T Labs-Research. Since Aug. 2003 Allen is an Associate Professor in ECE, at the University of Illinois, and on the research staff of the Beckman Inst., Urbana IL. During his 32 year AT&T career Prof. Allen has specialized in cochlear and middle ear modeling and auditory signal processing. In the last 10 years, he has concentrated on the problem of human speech recognition. His expertise spans the areas of signal processing, physical acoustics, acoustic power flow and impedance measurements, cochlear modeling, auditory neurophysiology, auditory psychophysics, and human speech recognition. Professor Allen is a Fellow (May 1981) of the Acoustical Society of America (ASA) and Fellow (January 1985) of the Institute of Electrical and Electronic Engineers (IEEE). In 1986 he was awarded the IEEE Acoustics Speech and Signal Processing (ASSP) Society Meritorious Service Award, and in 2000 received an IEEE Third Millennium Medal.

Jacob Benesty

University of Quebec
INRS-EMT
Montreal, Quebec, Canada
benesty@emt.inrs.ca



Chapters 1, B.6, B.7, B.13, H.43, H.46, I.51

Jacob Benesty received the Master's degree from Pierre & Marie Curie University, France, in 1987, and the Ph.D. degree in control and signal processing from Orsay University, France, in 1991. After positions at Telecom Paris University as a consultant and a member of the technical staff at Bell Laboratories, Murray Hill, Dr. Benesty joined the University of Québec (INRS-EMT) in Montreal, Canada in May 2003 as a professor. His research interests are in signal processing, acoustic signal processing, and multimedia communications.



Frédéric Bimbot

IRISA (CNRS & INRIA) – METISS
Rennes, France
bimbot@irisa.fr

Chapter F.36

After receiving his Ph.D. in 1988, Frédéric Bimbot joined CNRS in 1990 as a permanent researcher, working initially with ENST-Paris and then with IRISA in Rennes. He also repeatedly visited AT&T – Bell Laboratories between 1990 and 1999. His research is focused on audio signal analysis, speech modelling, speaker characterization and audio source separation. Since 2002, he is heading the METISS research group dedicated to selected topics in speech and audio processing.

Amazon EX1010 Page 1112



Thomas Brand

Carl von Ossietzky Universität Oldenburg
Sektion Medizinphysik
Oldenburg, Germany
thomas.brand@uni-oldenburg.de

Chapter A.4

Dr. Thomas Brand studied physics in Göttingen and Oldenburg where he received his Ph.D. for his work on analysis and optimization of psychophysical procedures in audiology. He is now scientist at the University of Oldenburg, Germany, where his research activities focus on psychoacoustics in audiology and measurement and prediction of speech intelligibility.

Nick Campbell

Knowledge Creating Communication
Research Centre
Acoustics & Speech Research Project,
Spoken Language Communication Group
Keihanna Science City, Japan
nick@nict.go.jp



Chapter D.25

Nick Campbell received his Ph.D. in Experimental Psychology from the University of Sussex in the U.K. and is currently engaged as a Chief Researcher in the Department of Acoustics and Speech Research at the Advanced Telecommunications Research Institute International (ATR) in Kyoto, Japan, where he also serves as Research Director for the JST/CREST Expressive Speech Processing and the SCOPE *Robot's Ears* projects. He was first invited as a Research Fellow at the IBM UK Scientific Centre, where he developed algorithms for speech synthesis, and later at the AT&T Bell Laboratories where he worked on the synthesis of Japanese. He served as Senior Linguist at the Edinburgh University Centre for Speech Technology Research before joining ATR in 1990. His research interests are based on large speech databases and include nonverbal speech processing, concatenative speech synthesis, and prosodic information modelling. He spends his spare time working with postgraduate students as Visiting Professor at the Nara Institute of Science and Technology (NAIST) and at Kobe University in Japan.

William M. Campbell

MIT Lincoln Laboratory
Information Systems Technology Group
Lexington, MA, USA
wcampbell@ll.mit.edu



Chapters F.38, G.41

Dr. Campbell is a technical staff member in the Information Systems Technology group at MIT Lincoln Laboratory. He received his Ph.D. in Applied Mathematics from Cornell University in 1995. Prior to joining MIT Lincoln Laboratory, he worked at Motorola on biometrics, speech interfaces, wearable computing, and digital communications. His current research interests include machine learning, speech processing, and plan recognition.



Rolf Carlson

Royal Institute of Technology (KTH)
Department of Speech, Music and Hearing
Stockholm, Sweden
rolf@speech.kth.se

Chapter D.20

Rolf Carlson is Professor in Speech Technology at KTH since 1995. Early research together with Björn Granström resulted in a multi-lingual text-to-speech system in 1976. This formed the base of the speech technology company Infovox in 1982. He has served on the board of ISCA and has published numerous papers in the speech research and technology area, including work on multimodal spoken dialog systems.



Jingdong Chen

Bell Laboratories
Alcatel-Lucent
Murray Hill, NJ, USA
jingdong@research.bell-labs.com

Chapters B.6, B.7, B.13, H.43, H.46, I.51

Dr. Chen holds a Ph.D in pattern recognition and intelligence control, an M.S. and a B.S. in electrical engineering. From 1998 to 1999, he worked at ATR Interpreting Telecommunications Research Laboratories, Japan. From 1999 to 2000, he worked at Griffith University, Australia and from 2000 to 2001, he was with ATR Spoken Language Translation Research Laboratories. He joined Bell Laboratories in 2001 as a Member of Technical Staff. He has extensively published in the area of acoustic signal processing for speech communications and authored/edited two books. He has helped organizing several international conferences and workshops, and serves as a member of the IEEE Signal Processing Society Technical Committee on Audio & Electroacoustics.

Juin-Hwey Chen

Broadcom Corp.
Irvine, CA, USA
rchen@broadcom.com



Chapter C.17

Juin-Hwey Chen is a Senior Technical Director at Broadcom Corporation. He received his Ph.D. in Electrical Engineering from UC Santa Barbara in 1987. He is the primary inventor of the ITU-T G.728 speech coding standard and the BroadVoice speech coding standards in PacketCable, SCTE, ANSI, and ITU-T J.161. He was elected an IEEE Fellow in 1995 and a Broadcom Fellow in 2006.

Israel Cohen

Technion–Israel Institute of Technology
Department of Electrical Engineering
Haifa, Israel
icohen@ee.technion.ac.il



Chapters H.44, H.47

Dr. Israel Cohen is a Professor of Electrical Engineering at the Technion–Israel Institute of Technology. He obtained his Ph.D. degree in electrical engineering from the Technion in 1998. From 1990 to 1998, he was a Research Scientist with RAFAEL research laboratories, Israel Ministry of Defence. From 1998 to 2001, he was a Postdoctoral Research Associate with the Computer Science Department at Yale University. He is a senior member of IEEE.



Jordan Cohen

SRI International
Menlo Park, CA, USA
jrc@speech.sri.com

Chapter E.34

Jordan Cohen is a Senior Scientist at SRI International, specializing in language based applications. He is the Principal Investigator for the DARPA GALE program for SRI, coordinating the activities of 14 subcontractors to harness speech recognition, language translation, and information annotation and distillation resources for the government. Jordan received his Ph.D. in Linguistics from the University of Connecticut, preceeded by a Masters' Degree in Electrical Engineering from the University of Illinois. He has worked in the Department of Defense, IBM, and at the Institute for Defense Analyses in Princeon, NJ. Most recently, Jordan was the CTO of Voice Signal Technologies, a company which produces multimodal speech-centric interfaces for mobile devices. He is a member of the Acoustical Society of America and the IEEE, and has published in both the classified and unclassified literature of speech and language technologies.



Corinna Cortes

Google, Inc.
Google Research
New York, NY, USA
corinna@google.com

Chapter E.29

Corinna Cortes is the Head of Google Research, NY, where she is working on a broad range of theoretical and applied large-scale machine learning problems. Prior to Google, Corinna spent more than ten years at AT&T Labs - Research, formerly AT&T Bell Labs, where she held a distinguished research position. Corinna's research work is well-known in particular for her contributions to the theoretical foundations of support vector machines (SVMs) and her work on data-mining in very large data sets for which she was awarded the AT&T Science and Technology Medal in the year 2000. Corinna received her Ph.D. in computer science from the University of Rochester in 1993.

Eric J. Diethorn

Avaya Labs Research
Multimedia Technologies Research
Department
Basking Ridge, NJ, USA
ejd@avaya.com



Chapter H.43

Dr. Diethorn received his Ph.D. in Electrical Engineering from the University of Illinois, Urbana-Champaign, in 1987. His professional focus is acoustical signal processing for speech communications, and he has extensive experience in the fields of acoustic- and electrical-echo cancellation, noise reduction and speech enhancement. Prior to joining Avaya in 2002, Dr. Diethorn worked for AT&T Bell Laboratories, Lucent Technologies, and Agere Systems.

Simon Doclo

Katholieke Universiteit Leuven
Department of Electrical Engineering
(ESAT-SCD)
Leuven, Belgium
simon.doclo@esat.kuleuven.be



Chapter H.48

Simon Doclo received the Ph.D. degree in electrical engineering from the Katholieke Universiteit Leuven, Belgium, in 2003. His research interests are in microphone array processing for speech enhancement and source localization, adaptive filtering, computational auditory scene analysis and hearing aid processing. Dr. Doclo received a Best Student Paper Award at the International Workshop on Acoustic Echo and Noise Control in 2001, and the EURASIP Signal Processing Best Paper Award in 2003.



Jasha Droppo

Microsoft Research
Speech Technology Group
Redmond, WA, USA
jdroppo@microsoft.com

Chapter E.33

Jasha Droppo received his Ph.D. from the University of Washington, Seattle, in 2000. His thesis developed a discrete theory of time-frequency distributions and non-stationary signal classification, with an application to speech recognition. Since graduation, he has worked at Microsoft Research, Redmond, on noise robust speech recognition, robust audio capture, speech enhancement, and acoustic modelling for automatic speech recognition.



Thierry Dutoit

Faculté Polytechnique de Mons FPMs
TCTS Laboratory
Mons, Belgium
thierry.dutoit@fpms.ac.be

Chapter D.21

Thierry Dutoit graduated as an electrical engineer and earned his Ph.D. in 1988 and 1993 from the Faculté Polytechnique de Mons, Belgium, respectively, where he is now a full professor. He spent 16 months as a consultant for AT&T Labs Research in Murray Hill and Florham Park, NJ, from July, 1996 to September, 1998. He is the author of two books on speech processing and text-to-speech synthesis, and the coordinator of the MBROLA project for free multilingual speech synthesis.

Gary W. Elko

mh acoustics LLC
Summit, NJ, USA
gwe@mhacoustics.com



Chapter I.50

Gary W. Elko studied electrical engineering at Cornell University. He continued his studies in the area of acoustics and signal processing earning Masters and Doctoral degrees at Penn State University. After graduation, he was employed in the Acoustics Research Department at Bell Labs for more than 15 years. In 2002 he cofounded mh acoustics LLC along with three colleagues.

Sadaoki Furui

Tokyo Institute of Technology Street
Department of Computer Science
Tokyo, Japan
furui@cs.titech.ac.jp



Chapter E.32

Dr. Sadaoki Furui is engaged in a wide range of research on speech analysis, speech recognition, speaker recognition, speech synthesis, and multimodal human – computer interaction and has authored or co-authored over 800 published articles. He has received Paper Awards and Achievement Awards from the IEEE, the IEICE, the ASJ, the Minister of Science and Technology, and the Minister of Education, and the Purple Ribbon Medal from the Japanese Emperor.



Sharon Gannot

Bar-Ilan University
School of Electrical Engineering
Ramat-Gan, Israel
gannot@eng.biu.ac.il

Chapters B.8, H.44, H.47

Sharon Gannot received his B.Sc. from the Technion, Israel in 1986 and the M.Sc. and Ph.D. from Tel-Aviv University, Israel, in 1995 and 2000, respectively, all in Electrical Engineering. In 2001 he held a post-doctoral position at K.U. Leuven, Belgium. From 2002 to 2003 he was a research fellow at the Technion, Israel. Currently, he is a lecturer in the School of Engineering, Bar-Ilan University, Israel. His research interests include statistical signal processing and speech processing using either single- or multi-microphone arrays.



Mazin E. Gilbert

AT&T Labs, Inc., Research
Florham Park, NJ, USA
mazin@research.att.com

Chapter E.34

Mazin Gilbert is a world expert in the area of spoken language technologies. He has over 18 years of experience in industrial research at Bell Labs and AT&T Labs and in academia at Rutgers University, Liverpool University and Princeton University. He is currently Executive Director at AT&T Labs-Research. He is the author of a book entitled, Artificial Neural Networks for Speech Analysis/Synthesis. He holds 16 US patents, published over 90 papers, and is a recipient of several national and international awards. He is currently pursuing an MBA for Executives at Wharton Business School.

Michael M. Goodwin

Creative Advanced Technology Center
Audio Research
Scotts Valley, CA, USA
mgoodwin@atc.creative.com



Chapter B.12

Michael Goodwin received the B.S. and M.S. degrees in Electrical Engineering and Computer Science (EECS) from the Massachusetts Institute of Technology and the Ph.D. in EECS from the University of California, Berkeley, and is now with the Audio Research Department of the Creative Advanced Technology Center in Scotts Valley, California. His research interests include signal modeling and enhancement, audio coding, spatial audio, and array processing, and he is currently chair of the IEEE Signal Processing Society's Technical Committee on Audio and Electroacoustics.

Volodya Grancharov

Multimedia Technologies
Ericsson Research, Ericsson AB
Stockholm, Sweden
volodya.grancharov@ericsson.com



Chapter A.5

Dr. Grancharov is a research engineer in the Multimedia Technologies, Ericsson Research, Stockholm, Sweden. He holds a Ph.D. in telecommunication from the Sound and Image Processing (SIP) Laboratory at the Royal Institute of Technology (KTH), Stockholm, Sweden. His research interests are in the fields of audio and video compression and quality assessment.



Björn Granström

Royal Institute of Technology (KTH)
Department for Speech, Music and Hearing
Stockholm, Sweden
bjorn@speech.kth.se

Chapter D.20

Björn Granström is Professor in Speech Communication at the Royal Institute of Technology (KTH) since 1987. Early research together with Rolf Carlson resulted in a multi-lingual text-to-speech system in 1976. This formed the base of the speech technology company Infovox in 1982. He has published numerous papers in the speech research and technology area, including work in audio-visual speech technology.



Patrick Haffner

AT&T Labs-Research
IP and Voice Services
Middletown, NJ, USA
haffner@research.att.com

Chapter E.29

Patrick Haffner is Lead Member of Technical Staff at AT&T Labs-Research. With Yann LeCun and Vladimir Vapnik, he pioneered the use of Machine Learning (including SVMs and FSMs) for document and image processing applications and is one of the inventors of the DjVu platform. He currently works on algorithms and software solutions for speech, language and sequence processing, with a particular focus on global and large-scale solutions.

Roar Hagen

Global IP Solutions
Stockholm, Sweden
roar.hagen@gipscorp.com



Chapter C.15

Dr. Roar Hagen is co-founder and Chief Technology Officer for Global IP Solutions, a company developing media processing software for IP Communication. Roar began R&D within speech processing and coding in 1989. He previously held positions at AT&T Bell Labs and Ericsson Research. He holds a doctorate in Electrical Engineering from Chalmers University of Technology, Sweden, and an M.Sc. in Physics from the Norwegian Institute of Technology, Norway. Roar Hagen has filed more than 10 patents.

Mary P. Harper

University of Maryland
Center for Advanced Study of Language
College Park, MD, USA
mharper@casl.umd.edu



Chapter G.40

Dr. Mary Harper obtained her Ph.D. in Computer Science from Brown University in 1990. She currently holds the rank of Professor in the School of Electrical and Computer Engineering at Purdue University and is a Senior Research Scientist at the Center for Advanced Study of Language at the University of Maryland. Dr. Harper's research focuses on computer modeling of human communication with a focus on methods for incorporating multiple types of knowledge sources, including lexical, syntactic, prosodic, and most recently visual sources.



Jürgen Herre

Fraunhofer Institute for Integrated Circuits (Fraunhofer IIS)
Audio and Multimedia
Erlangen, Germany
hrr@iis.fraunhofer.de

Chapter C.18

Dr. Jürgen Herre is the Chief Scientist for the Audio/Multimedia activities at Fraunhofer IIS, Erlangen, Germany. He has been working on the development of advanced multimedia technologies and, especially, perceptual audio coding algorithms since 1989 and contributed to many well-known audio coding algorithms, including MPEG-1 Layer 3 (MP3), Advanced Audio Coding (AAC), MPEG-4 Audio and MPEG Surround.



Wolfgang J. Hess

University of Bonn
Institute for Communication Sciences,
Dept. of Communication, Language,
and Speech
Bonn, Germany
wgh@ifk.uni-bonn.de

Chapter B.10

Wolfgang Hess has been a professor of phonetics and speech communication at the University of Bonn since 1986. He received his Dr.-Ing. degree in Electrical Engineering in 1972 and the venia legendi for digital signal processing in 1980, both from the Technical University of Munich. His research areas are speech analysis, especially pitch determination, speech synthesis, and phonetics.

Kiyoshi Honda

Université de la Sorbonne
Nouvelle-Paris III
Laboratoire de Phonétique
et de Phonologie,
ATR Cognitive Information Laboratories
Paris, France
honda@atr.jp



Chapter A.2

Kiyoshi Honda is a specialist in speech production with medical background and research career at the University of Tokyo, Haskins Labs, University of Wisconsin, and ATR. His main interest is the discovery of speech production mechanisms using medical techniques such as EMG and MRI. His major works include mechanisms of vocal control and vowel articulation. Since recently he works on vocal-tract modelling and phonetic instrumentation techniques.

Yiteng (Arden) Huang

Bell Laboratories
Alcatel-Lucent
Murray Hill, NJ, USA
arden_huang@ieee.org



Chapters 1, B.6, B.7, B.13, H.43, H.46, I.51

Dr. Yiteng (Arden) Huang received his Ph.D. in Electrical and Computer Engineering from Georgia Tech in 2001. Upon graduation, he joined Bell Laboratories, and has been, so far, working there as a member of Technical Staff. He has extensively published in acoustic MIMO signal processing for speech and has authored/edited 3 books. He received the 2002 Young Author Best Paper Award from the IEEE Signal Processing Society, has served on several editorial boards/technical committees, and has helped organize a couple of international IEEE workshops.



Matthieu Hébert

Network ASR Core Technology
Nuance Communications
Montréal, Québec, Canada
hebert@nuance.com

Chapter F.37

Dr. Hébert received a Ph.D. in theoretical physics from the Université de Sherbrooke in 1995. After a post-doctoral appointment at Collège de France in Professor de Gennes' laboratory, he was hired by Nortel Networks in the Speech Division. In 1999, he joined Nuance Communications' R+D team. In 2002, he assumed the R+D lead for Nuance Verifier. His research interests are in speaker and speech recognition as well as natural language understanding.

Amazon EX1010 Page 1117



Biing-Hwang Juang

Georgia Institute of Technology
School of Electrical
& Computer Engineering
Atlanta, GA, USA
juang@ece.gatech.edu

Chapter E.26

Biing Hwang (Fred) Juang received his Ph.D. from the University of California, Santa Barbara. After over two decades of career at Bell Laboratories, he joined Georgia Institute of Technology in 2001, holding the Motorola Foundation Chair Professorship. Professor Juang has received numerous technical awards, including the Technical Achievement Award from the Signal Processing Society of the IEEE, and the IEEE Third Millennium Medal. He is a Fellow of the IEEE, a Fellow of Bell Laboratories, a member of the National Academy of Engineering and an Academician of Academia Sinica.



Tatsuya Kawahara

Kyoto University
Academic Center
for Computing and Media Studies
Kyoto, Japan
kawahara@i.kyoto-u.ac.jp

Chapter E.32

Tatsuya Kawahara received the B.E. degree in 1987, the M.E. degree in 1989, and the Ph.D. degree in 1995, all in information science from Kyoto University, Japan. Currently, he is a Professor in the Academic Center for Computing and Media Studies and an Adjunct Professor in the School of Informatics, Kyoto University. He has published more than 100 technical papers covering speech recognition, confidence measures, and spoken dialogue systems.



Ulrik Kjems

Oticon A/S
Smørum, Denmark
uk@oticon.dk

Chapter I.52

Ulrik Kjems received his Ph.D. in 1998 and worked two years as a post doc researcher at the Technical University of Denmark, Department of Mathematical Modelling, on topics of statistical analysis of brain scan images. Since 2000 he is a DSP and algorithm designer at hearing aid manufacturer Oticon, working with noise suppression, speech enhancement and beam-forming systems.



Esther Klabbers

Oregon Health & Science University
Center for Spoken Language
Understanding,
OGI School of Science and Engineering
Beaverton, OR, USA
klabbers@cslu.ogi.edu

Chapter D.23

Dr. Esther Klabbers is an expert in the speech synthesis field. She has a Ph.D. from the Eindhoven University of Technology, the Netherlands. Dr. Klabbers currently works as an Assistant Scientist at the Center for Spoken Language Understanding. Her expertise involves many areas of speech synthesis, such as prosody modeling, corpus design, perceptual evaluations and health-related applications of speech synthesis.



W. Bastiaan Kleijn

Royal Institute of Technology (KTH)
School of Electrical Engineering,
Sound and Image Processing Lab
Stockholm, Sweden
bastiaan.kleijn@ee.kth.se

Chapters A.5, C.14, C.15

Bastiaan Kleijn is a Professor at the School of Electrical Engineering at KTH (Royal Institute of Technology) in Stockholm, Sweden and heads the Sound and Image Processing Laboratory. He is also a founder and former Chairman of Global IP Solutions where he remains Chief Scientist. He holds a Ph.D. in Electrical Engineering from Delft University of Technology (Netherlands), a Ph.D. in Soil Science and an M.S. in Physics, both from the University of California, and an M.S. in Electrical Engineering from Stanford University. He worked on speech processing at AT&T Bell Laboratories from 1984 to 1996, first in development and later in research. He is a Fellow of the IEEE.



Birger Kollmeier

Universität Oldenburg
Medizinische Physik
Oldenburg, Germany
birger.kollmeier@uni-oldenburg.de

Chapter A.4

Birger Kollmeier obtained his Ph.D. in physics and his Ph.D. in medicine in Göttingen, Germany. Since 1993 he is a full professor in physics and head of the medical physics group at the Universität Oldenburg performing fundamental and applied research in psychoacoustics, speech processing, hearing aids and auditory neuroscience. He is the chairman of the translational research institutions Hörzentrum Oldenburg and HörTech, i.e. national center of excellence in hearing technology and audioloy, and was awarded several prizes.

Ermin Kozica

Royal Institute of Technology (KTH)
School of Electrical Engineering,
Sound and Image Processing Laboratory
Stockholm, Sweden
ermin.kozica@ee.kth.se



Chapter C.15

Ermin Kozica received his M.Sc. degree in Electrical Engineering from KTH in January 2006. Since then, he is a graduate student in the Sound and Image Processing Lab., School of Electrical Engineering at KTH. The working title of his thesis is *Methods for Robust Video Coding*. His research interests include video processing, image processing and signal processing in general.



Sen M. Kuo

Northern Illinois University
Department of Electrical Engineering
DeKalb, IL, USA
kuo@ceet.niu.edu

Chapter H.49

Dr. Kuo is a Professor and Chair of Electrical Engineering at the Northern Illinois University, DeKalb, IL. He received the B.S. degree from the National Taiwan Normal University, Taipei, Taiwan, in 1976 and the M.S. and Ph.D. degrees from the University of New Mexico in 1983 and 1985, respectively. In 1993, he was with Texas Instruments, Houston, TX.He serves as an associate editor for IEEE Transactions on Audio, Speech and Language Processing since June 2006 to present. Dr. Kuo is the leading author of five books, has been awarded seven US patents and has published over 190 technical papers. His research focuses on active noise and vibration control, real-time DSP systems, adaptive echo and noise cancellation, biomedical signal processing, and digital communication and audio applications.



Jan Larsen

Technical University of Denmark
Informatics and Mathematical Modelling
Kongens Lyngby, Denmark
jl@imm.dtu.dk

Chapter I.52

Dr. Jan Larsen received Ph.D. degree from the Technical University of Denmark (DTU) in 1994 and is currently Associate Professor at Informatics and Mathematical Modelling, DTU. He has authored and co-authored more than 100 papers and book chapters in the areas of machine learning for signal processing, and has been involved in many IEEE conference organizations.

Chin-Hui Lee

Georgia Institute of Technology
School of Electrical
and Computer Engineering
Atlanta, GA, USA
chl@ece.gatech.edu



Chapters G.39, G.42

Dr. Chin-Hui Lee received his Ph.D. degree from University of Washington in 1981. His current research interests include multimedia communication, speech and language processing, biometric authentication, and information retrieval. In 2007 he received a Technical Achievement Award from the IEEE Signal Processing Society for "Exceptional Contributions to the Field of Automatic Speech Recognition". Dr. Lee is a fellow of the IEEE.

Haizhou Li

Institute for Infocomm Research
Department of Human Language
Technology
Singapore
hli@i2r.a-star.edu.sg



Chapter G.42

Dr. Haizhou Li received his Ph.D degree from the South China University of Technology in 1990. He is now the Head of Human Language Technology Department at the Institute for Infocomm Research, Singapore. His research interests include automatic speech recognition, speaker and language recognition, and natural language processing. Dr Li is a recipient of the National Infocomm Award 2001 in Singapore. He is the Vice President of COLIPS and a Senior Member of IEEE.



Jan Linden

Global IP Solutions
San Francisco, CA, USA
jan.linden@gipscorp.com

Chapter C.15

Jan Linden is the Vice President of Engineering at Global IP Solutions. He has been conducting research and development in speech processing and communications for more than 15 years. Prior to joining Global IP Solutions he was with the University of California, Santa Barbara and SignalCom. He holds a Ph.D. and an M.Sc. in Electrical Engineering from Chalmers University of Technology, Sweden.



Manfred Lutzky

Fraunhofer Integrated Circuits (IIS)
Multimedia Realtime Systems
Erlangen, Germany
manfred.lutzky@iis.fraunhofer.de

Chapter C.18

Manfred Lutzky received his M.S. (Diplom) degree in Electrical Engineering from Erlangen University and joined the Fraunhofer Institute for Integrated Circuits (IIS) in Erlangen, Germany, in 1997. He was engaged in the implementation of the audio coding schemes mp3 and MPEG-4 AAC on embedded platforms. Since 2001 he is in the position of a group manager now heading the Audio for Communcation group that deals with research, implementation and optimization of low-delay audio codecs such as MPEG-4 AAC-LD and ULD and related technology for communication applications. He is actively participating in the standardization of audio codecs in ETSI/NGDect body.



Bin Ma

Human Language Technology
Institute for Infocomm Research
Singapore
mabin@i2r.a-star.edu.sg

Chapter G.42

Dr. Ma is a Research Scientist in the Speech and Dialogue Processing group, Institute for Infocomm Research (I^2R), Singapore. Prior to joining I^2R, he worked in speech recognition at the Chinese Academy of Sciences, Lernout & Hauspie Speech Products, and InfoTalk Corporation. He received his Ph.D. degree from The University of Hong Kong in speech recognition. He is a Senior Member of IEEE.



Michael Maxwell

University of Maryland
Center for Advanced Study of Language
College Park, MD, USA
mmaxwell@casl.umd.edu

Chapter G.40

Dr. Maxwell is a Senior Research Scientist at the Center for Advanced Study of Language at the University of Maryland. He obtained his Ph. D. in Linguistics from the University of Washington in 1984. Prior to joining the University of Maryland, his research included describing endangered languages of Ecuador and Colombia (under the auspices of the Summer Institute of Linguistics), computational syntax, morphology and phonology, and the building of resources for low-density languages (at the Linguistic Data Consortium of the University of Pennsylvania).



Alan V. McCree

MIT Lincoln Laboratory
Department of Information Systems Technology
Lexington, MA, USA
mccree@ll.mit.edu

Chapter C.16

Dr. Alan McCree has been with MIT Lincoln Laboratory since 2004, where he conducts research in speech coding, noise suppression, and speaker recognition. He has also worked at Texas Instruments, AT&T Bell Laboratories, and M/A-COM Linkabit. He is an IEEE Fellow. Dr. McCree has published 40 technical papers and has been awarded 30 U. S. patents.



Bernd Meyer

Carl von Ossietzky Universität Oldenburg
Medical Physics Section, Haus des Hörens
Oldenburg, Germany
bernd.meyer@uni-oldenburg.de

Chapter A.4

Bernd Meyer is a Ph.D. student at the Medical Physics Section at the University of Oldenburg. His research activities focus on the speech recognition performance of human listeners compared to machines and the improvement of feature extraction of automatic speech recognition systems.



Jens Meyer

mh acoustcis
Summit, NJ, USA
jmm@mhacoustics.com

Chapter I.50

Jens Meyer studied electrical engineering at Darmstadt University of Technology. He continued there with his Dr.-Ing where he specialized in acoustic signal processing. He stayed in this field ever since, focusing on multi-microphone application for communication and sound recording. In 2002 together with 3 colleagues, he co-founded mh acoustics.

Taniya Mishra

Oregon Health and Science University
Center for Spoken Language
Understanding, Computer Science
and Electrical Engineering,
OGI School of Science and Engineering
Beaverton, OR, USA
mishra@cslu.ogi.edu



Chapter D.23

Taniya Mishra is a doctoral student at the Center for Spoken Language Understanding at the OGI School of Science &Engineering at Oregon Health and Science University. Her main research focus is intonation modelling and speech synthesis. For her dissertation, she is working on developing an algorithm for decomposing pitch contours into their component curves, using the general assumptions of the superpositional model of intonation.



Mehryar Mohri

Courant Institute of Mathematical
Sciences
New York, NY, USA
mohri@cs.nyu.edu

Chapters E.28, E.29

Mehryar Mohri is a Professor of Computer Science at the Courant Institute of Mathematical Sciences and a Research Consultant at Google Research. Prior to these positions, he spent about ten years at AT&T Bell Labs or AT&T Labs - Research (1995-2004) where, in the last four years, he served as the Head of the Speech Algorithms Department and as a Technology Leader, overseeing research projects in machine learning, text and speech processing, and the design of general algorithms. Mehryar Mohri graduated from Ecole Polytechnique in France in 1987, received his M.S. degree in Mathematics and Computer Science from Ecole Normale Superieure d'Ulm (1988), and his M.S. degree (1989) and Ph.D. (1993) in Computer Science from the University of Paris 7. His current topics of interest are machine learning, computational biology, and text and speech processing.



Marc Moonen

Katholieke Universiteit Leuven
Electrical Engineering Department ESAT/
SISTA
Leuven, Belgium
marc.moonen@esat.kuleuven.be

Chapter H.48

Marc Moonen is a Professor of Signal Processing at the Electrical Engineering Department of Katholieke Universiteit Leuven He is a Fellow of the IEEE. He is currently President of the European Association for Signal Processing (EURASIP). He has served as Editor-in-Chief for the EURASIP Journal on Applied Signal Processing (2003-2005), and is/has been a member of the editorial board of five other journals.

Dennis R. Morgan

Bell Laboratories, Alcatel-Lucent
Murray Hill, NJ, USA
drrm@bell-labs.com



Chapter H.49

Dr. Morgan received the B.S. degree in 1965 from the University of Cincinnati, OH, and the M.S. and Ph.D. degrees from Syracuse University, Syracuse, NY, in 1968 and 1970, respectively, all in electrical engineering. He is a Distinguished Member of Technical Staff with Bell Laboratories, Alcatel-Lucent where he has been employed since 1984, and is involved in research on adaptive signal processing applied to RF and optical communication systems. Before that, he was with General Electric Company, Electronics Laboratory, Syracuse, NY, specializing in the analysis and design of signal processing systems used in radar, sonar, and communications. He has authored numerous journal publications and is coauthor of two books.

David Nahamoo

IBM Thomas J. Watson Research Center
Yorktown Heights, NY, USA
nahamoo@us.ibm.com



Chapter E.30

David Nahamoo is the Chief Technical Officer and Business Strategist for Speech at the IBM Thomas J. Watson Research Center. He has worked in the Speech Recognition area since 1982, joining IBM after finishing his doctorate at Purdue. He is a Fellow of the IEEE and has received two best paper awards from the IEEE Signal Processing Society.

Amazon EX1010 Page 1121



Douglas O'Shaughnessy

Université du Québec
INRS Énergie, Matériaux
et Télécommunications (INRS-EMT)
Montréal, Québec, Canada
dougo@emt.inrs.ca

Chapter B.11

Douglas O'Shaughnessy is Professor at INRS-EMT and adjunct Professor at McGill University. His interests include automatic speech synthesis, analysis, coding, enhancement, and recognition. A graduate of MIT, he is a Fellow of the IEEE and the Acoustical Society of America (ASA). He is a member of the IEEE Technical Comittee for Speech Processing, and was the General Chair of the 2004 International Conference on Acoustics, Speech and Signal Processing. He is the author of *Speech Communications: Human and Machine.*



Lucas C. Parra

Steinman Hall,
The City College of New York
Department of Biomedical Engineering
New York, NY, USA
parra@ccny.cuny.edu

Chapter I.52

Dr. Parra is Associate Professor of Biomedical Engineering at The City College of New York (CCNY). Previously he was at Sarnoff Corporation and at Siemens Corporate Research. He is a physicist with expertise in acoustic array processing, emission tomography, electroencephalography, machine learning and pattern recognition. His current research focuses on functional brain imaging and computational models of the central nervous system.

Sarangarajan Parthasarathy

Yahoo!, Applied Research
Sunnyvale, CA, USA
parthas@yahoo-inc.com



Chapter F.36

Dr. Sarangarajan Parthasarathy has spent much of his research career at AT&T Bell Labs and its successors, contributing to advances in many areas of speech processing including articulatory modelling, automatic speech and speaker recognition, and multimedia processing. His responsibilities ranged from basic research to the implementation of algorithms and system prototypes. He was a member of the Speech Technical Committee of the IEEE Signal Processing Society from 2002 to 2006. He recently joined the applied research group at Yahoo to work on statistical modelling applied to web/sponsored search.

Michael Syskind Pedersen

Oticon A/S
Smørum, Denmark
msp@oticon.dk



Chapter I.52

Michael Syskind Pedersen obtained his Ph.D. in 2006 from the department of Informatics and Mathematical Modelling (IMM) at the Technical University of Denmark (DTU). In 2005 he was a visiting scholar at the department of Computer Science and Engineering at The Ohio State University, Columbus OH. His main areas of research are multi-microphone audio processing and noise reduction. Since 2001 Michael has been employed with the hearing aid company Oticon.



Fernando Pereira

University of Pennsylvania
Department of Computer
and Information Science
Philadelphia, PA, USA
pereira@cis.upenn.edu

Chapter E.28

Fernando Pereira is the Andrew and Debra Rachleff Professor and chair of the department of Computer and Information Science, University of Pennsylvania. He received a Ph.D. in Artificial Intelligence from the University of Edinburgh in 1982. Before joining Penn, he held industrial research and management positions at SRI International, at AT&T Labs, and at WhizBang Labs, a Web information extraction company. His main research interests are in machine-learnable models of language and biological sequences. He has over 100 research publications and several patents. He was elected Fellow of the American Association for Artificial Intelligence in 1991 for his contributions to computational linguistics and logic programming, and he was president of the Association for Computational Linguistics.

Amazon EX1010 Page 1122



Michael Picheny

IBM Thomas J. Watson Research Center
Yorktown Heights, NY, USA
Picheny@us.ibm.com

Chapter E.30

Michael Picheny is the Senior Manager of the Speech and Language Algorithms Group at the IBM Thomas J. Watson Research Center. He has worked in the Speech Recognition area since 1981, joining IBM after finishing his doctorate at MIT. He is a Fellow of the IEEE and a member of the board of ISCA.

Rudolf Rabenstein

University Erlangen-Nuremberg
Electrical Engineering, Electronics,
and Information Technology
Erlangen, Germany
rabe@LNT.de



Chapter I.53

Rudolf Rabenstein studied electrical engineering in Erlangen, Germany and Boulder, Colorado, USA. He worked with the Physics Department of the University of Siegen, Germany and currently with the Telecommunications Laboratory of the University Erlangen-Nuremberg, Germany. Here he obtained his doctoral degree and the "Habilitation" degree in signal processing. His research interests are in the fields of multidimensional systems theory and multimedia signal processing.

Lawrence Rabiner

Rutgers University
Department of Electrical
and Computer Engineering
Piscataway, NJ, USA
lrr@caip.rutgers.edu



Chapter E.26

Lawrence Rabiner received the S.B., and S.M. degrees simultaneously in June 1964, and the Ph.D. degree in Electrical Engineering in June 1967, all from MIT. Dr. Rabiner joined AT&T Bell Labs in 1967 as a Member of the Technical Staff and served in various positions. In 1996 he joined AT&T Labs in 1996 as Director of the Speech and Image Processing Services Research Lab, and was promoted to Vice President of Research in 1998. Dr. Rabiner retired at the end of March 2002 and is now a Professor of Electrical and Computer Engineering at Rutgers University, and the Associate Director of the Center for Advanced Information Processing (CAIP). He also has a joint appointment as a Professor of Electrical and Computer Engineering at the University of California at Santa Barbara.



Douglas A. Reynolds

Massachusetts Institute of Technology
Lincoln Laboratory,
Information Systems Technology Group
Lexington, MA, USA
dar@ll.mit.edu

Chapters F.38, G.41

Dr. Douglas Reynolds is a Senior Member of Technical Staff at MIT Lincoln Laboratory with over 20 years experience in research, development, evaluation and deployment of speaker, language and speech recognition technologies for commercial and governmental applications. He invented several widely used algorithms in robust speaker and language recognition and pioneered approaches to exploit high-level information for improved speaker recognition accuracy. His current research efforts are focused on robust, portable speaker and language recognition and synthesis of paralinguistic information for characterizing large audio data sets.



Michael Riley

Google, Inc., Research
New York, NY, USA
riley@google.com

Chapter E.28

Michael Riley received his Ph.D. in computer science from MIT in 1987. He joined Bell Labs in Murray Hill, NJ in 1987 and moved to AT&T Labs in Florham Park, NJ in 1996. He is currently a member of the research staff at Google, Inc. in New York City. His interests include speech and natural language processing, text analysis, information retrieval, and machine learning.

Amazon EX1010 Page 1123

Aaron E. Rosenberg

Rutgers University
Center for Advanced Information Processing
Piscataway, NJ, USA
aer@caip.rutgers.edu



Chapter F.36

Aaron Rosenberg is a Research Professor at the Center for Advanced Information Processing (CAIP) at Rutgers University. He retired from AT&T Labs Research as a Technology Leader in the Speech and Image Processing Services Laboratory. Prior to his service at AT&T, he was a Distinguished Member of the Technical Staff at Bell Laboratories. His research activities have included auditory psychophysics, speech perception, speech quality, and speech and speaker recognition. He has authored or co-authored some 100 papers and been granted 12 patents in these fields. He is a Fellow of the Acoustical Society of America and the IEEE.

Salim Roukos

IBM T. J. Watson Research Center
Multilingual NLP Technologies
Yorktown Heights, NY, USA
roukos@us.ibm.com



Chapter E.31

Salim Roukos is Senior Manager and CTO for Translation Technologies at IBM. His research areas at IBM have been in statistical machine translation, information extraction, statistical parsing, and statistical language understanding for conversational systems. Roukos received his B.E. from the American University of Beirut, in 1976, his M.Sc. and Ph. D. from the University of Florida, in 1978 and 1980, respectively. Roukos has served as Chair of the IEEE Digital Signal Processing Committee in 1988. Roukos lead the group that created IBM's ViaVoice Telephony product, the first commercial software to support full natural language understanding for dialog systems in 2000, and more recently the first statistical machine translation product for Arabic-English translation in 2003.



Jan van Santen

Oregon Health And Science University
OGI School of Science and Engineering,
Department of Computer Science
and Electrical Engineering
Beaverton, OR, USA
vansanten@cslu.ogi.edu

Chapter D.23

Dr. van Santen's current responsibilities consist of teaching, conducting research, and managing the Center for Spoken Language Understanding (CSLU) as well as the Computer Science and Electrical Engineering (CSEE) Department, both at the OGI School of Science and Engineering at the Oregon Health and Science University in Portland, Oregon. His background includes mathematical modeling of prosody, signal processing, and computational linguistics. He has seven patents and over 100 publications. Dr. van Santen is currently the Director of CSLU, the Chair of the CSEE Department, serves on the board of directors of SVOX AG, and is president of BioSpeech Inc. He has a Ph.D. degree in Mathematical Psychology from the University of Michigan.



Ronald W. Schafer

Hewlett-Packard Laboratories
Palo Alto, CA, USA
ron.schafer@hp.com

Chapter B.9

Ronald W. Schafer is Professor Emeritus at Georgia Tech and a HP Fellow in the Mobile and Media Systems Laboratory at Hewlett-Packard Laboratories in Palo Alto, CA. He has co-authored six widely used textbooks in the DSP field, and has received numerous awards for teaching and research including the IEEE Education Medal and membership in the National Academy of Engineering.

Juergen Schroeter

AT&T Labs – Research
Department of Speech Algorithms and Engines
Florham Park, NJ, USA
schroeter@att.com



Chapter D.19

Juergen Schroeter earned his Dr.-Ing. in Electrical Engineering from Ruhr-University in Bochum, Germany, in 1983. As a Member of Technical Staff at AT&T Bell Laboratories in Murray Hill, NJ, he worked on speech coding and speech synthesis methods that employ computational models of the vocal tract and glottis. At AT&T Labs – Research in Florham Park, NJ, he is responsible for research in speech algorithms and engines. His group created the AT&T Natural Voices® Text-to-Speech system in 2001. In the same year, he received the AT&T Science and Technology Medal. He is a Fellow of the Acoustical Society of America and a Fellow of the IEEE.

Amazon EX1010 Page 1124

Stephanie Seneff

Massachusetts Institute of Technology
Computer Science and Artificial
Intelligence Laboratory
Cambridge, MA, USA
senef@csail.mit.edu



Chapter E.35

Dr. Stephanie Seneff's research interests encompass a broad range of topics centered around spoken conversational interfaces to computers, including speech recognition, natural language understanding, discourse and dialogue modelling, natural language generation, and language translation. She has served as a member of the Permanent Council for the International Conference on Spoken Language Systems (ICSLP).

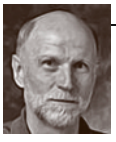


Wade Shen

Massachusetts Institute of Technology
Communication Systems, Information
Systems Technology, Lincoln Laboratory
Lexington, MA, USA
swade@ll.mit.edu

Chapter G.41

Wade Shen received his Master's degree in Computer Science from the University of Maryland, College Park in 1997, and his Bachelor's degree in Electrical Engineering and Computer Science from the University of California, Berkeley in 1994. His current areas of research involve machine translation and machine translation evaluation, speech, speaker, and language recognition for small-scale and embedded applications, named-entity extraction, and prosodic modeling. Prior to joining Lincoln Laboratory in 2003, Wade helped found and served as Chief Techology Officer for Vocentric Corporation, a company specializing in speech technologies for small devices.



Elliot Singer

Massachusetts Institute of Technology
Information Systems Technology Group,
Lincoln Laboratory
Lexington, MA, USA
es@ll.mit.edu

Chapter G.41

Elliot Singer received an S.M. degree in electrical engineering from MIT in 1974 and is currently a staff member of the MIT Lincoln Laboratory Information Systems Technology Group. Since 1977 he has worked in a wide variety of speech related areas, including low rate speech coding, custom hardware development, keyword spotting, pattern recognition, and speaker and language recognition.

Jan Skoglund

Global IP Solutions
San Francisco, CA, USA
jan.skoglund@gipscorp.com



Chapter C.15

Jan Skoglund received his Ph.D. degree from Chalmers University of Technology, Sweden. From 1999 to 2000, he worked on low bit rate speech coding as a consultant at AT&T Labs-Research, Florham Park, NJ. Since 2000, he has been with Global IP Solutions, San Francisco, CA, where he is working on speech and audio processing tailored for packet-switched networks.

M. Mohan Sondhi

Avayalabs Research
Basking Ridge, NJ, USA
mms@research.avayalabs.com



Chapters 1, H.45

M. Mohan Sondhi is a consultant at Avaya Research Labs, Basking Ridge, NJ. Prior to joining Avaya he spent 39 years at Bell Labs, from where he retired in 2001. He holds undergraduate degrees in Physics and Electrical Communication Engineering, and M.S and Ph.D. degrees in Electrical Engineering. At Bell Labs he conducted research in speech signal processing, echo cancellation, acoustical inverse problems, speech recognition, articulatory models for analysis and synthesis of speech and modeling of auditory and visual processing by humans.



Sascha Spors

Deutsche Telekom AG, Laboratories
Berlin, Germany
Sascha.Spors@telekom.de

Chapter I.53

Dr. Sascha Spors is senior research scientist at the Deutsche Telekom Laboratories. He obtained his doctoral degree in Electrical Engineering from the University Erlangen-Nuremberg in January 2006. His research background is on wavefield analysis, multichannel sound reproduction and efficient algorithms for massive multichannel adaptation problems. His special interest is on wave field synthesis, where he has implemented several laboratory systems.

Amazon EX1010 Page 1125



Ann Spriet

ESAT-SCD/SISTA, K.U. Leuven
Department of Electrical Engineering
Leuven, Belgium
ann.spriet@esat.kuleuven.be

Chapter H.48

Dr. A. Spriet is a Postdoctoral Fellow of the Fund for Scientific Research-Flanders, affiliated with the Electrical Engineering Department (ESAT) and the Department of Neurosciences (ExpORL) of the Katholieke Universiteit Leuven. Her research interests are in digital signal processing for speech enhancement in hearing aids and cochlear implants. She received the 2001 Lapperre Award and a Best Student Paper Award at IWAENC in 2003 and ICASSP in 2005.



Richard Sproat

University of Illinois
at Urbana-Champaign
Department of Linguistics
Urbana, IL, USA
rws@uiuc.edu

Chapter D.22

Richard Sproat received his Ph.D. in Linguistics from the Massachusetts Institute of Technology in 1985. Since then he has worked at AT&T Bell Labs, at Lucent's Bell Labs and at AT&T Labs – Research, before joining the faculty of the University of Illinois. Sproat has worked in numerous areas relating to language and computational linguistics, including syntax, morphology, computational morphology, articulatory and acoustic phonetics, text processing, text-to-speech synthesis, writing systems, and text-to-scene conversion, and has published widely in these areas. His most recent work includes work on the prediction of emotion from text for TTS, multilingual named entity transliteration, and the effects of script layout on readers' phonological awareness.



Yannis Stylianou

Institute of Computer Science
Heraklion, Crete, Greece
yannis@csd.uoc.gr

Chapter D.24

Yannis Stylianou is Associate Professor at University of Crete, Department of Computer Science. He received the Diploma of Electrical Engineering from the National Technical University, NTUA, of Athens in 1991 and the M.Sc. and Ph.D. degrees in Signal Processing from the Ecole National Superieure des Telecommunications, ENST, Paris, France in 1992 and 1996, respectively. From 1996 until 2001 he was with AT&T Labs Research (Murray Hill and Florham Park, NJ, USA) as a Senior Technical Staff Member. In 2001 he joined Bell-Labs Lucent Technologies, in Murray Hill, NJ, USA. Since 2002 he is with the Computer Science Department at the University of Crete.



Jes Thyssen

Broadcom Corporation
Irvine, CA, USA
jthyssen@broadcom.com

Chapter C.17

Jes Thyssen received his Ph.D. from the Technical University of Denmark. He is a Senior Principle Scientist with Broadcom working in the area of speech and audio processing. He holds numerous patents in the field and has contributed to a number of speech coding standards. Current research interests include packetloss concealment, noise suppression, blind source separation, and speech coding.



Jay Wilpon

Research AT&T Labs
Voice and IP Services
Florham Park, NJ, USA
jgw@research.att.com

Chapter E.34

Jay Wilpon is Executive Director of Speech Research at AT&T Labs. He has authored over 90 publications. His research led to the first nationwide deployment of both wordspotting and spoken language understanding technologies. Jay was named an IEEE Fellow for his leadership in the development of automatic speech recognition algorithms. He served as Member at Large to the IEEE Signal Processing Society (SPS) Board of Governors, and as Chair of the SPS Speech Processing Technical Committee. For pioneering leadership in the creation and deployment of speech recognition-based services in the telephone network, he was awarded the distinguished honor of AT&T Fellow.

Amazon EX1010 Page 1126

Jan Wouters

ExpORL, Department of Neurosciences,
K.U. Leuven
Leuven, Belgium
jan.wouters@med.kuleuven.be



Chapter H.48

Jan Wouters is a professor at the Department of Neurosciences of the Katholieke Universiteit Leuven, Belgium. He obtained his PhD in Physics in 1989. His research activities are mainly focused on audiology and the signal processing in the auditory system, signal processing for cochlear implants and hearing aids, and electrophysiological and psychophysical measures for assessment. He is author of about 120 articles in peer-review international journals.

Arie Yeredor

Tel-Aviv University
Electrical Engineering - Systems
Tel-Aviv, Israel
arie@eng.tau.ac.il



Chapter B.8

Arie Yeredor received his Ph.D. from Tel-Aviv University (TAU) in 1997. He is currently a Senior Lecturer at TAU, teaching courses in statistical and digital signal processing, and serving as Academic Director of the DSP Labs. His research interests include estimation theory, statistical signal processing and blind source separation.



Steve Young

Cambridge University Engineering Dept
Cambridge, UK
sjy@eng.cam.ac.uk

Chapter E.27

Steve Young is Professor of Information Engineering and Head of the Information Engineering Division at Cambridge University, UK. His main research interests lie in the area of spoken language systems including speech recognition, speech synthesis and dialogue management. He is a Fellow of the Royal Academy of Engineering and the Institution of Electrical Engineers (IEE). Professor Young is a member of the British Computer Society (BCS) and a Senior Member of the IEEE. In 2004, he was a recipient of an IEEE Signal Processing Society Technical Achievement Award.



Victor Zue

Massachusetts Institute of Technology
CSAI Laboratory
Cambridge, MA, USA
zue@csail.mit.edu

Chapter E.35

Victor Zue is the Delta Electronics Professor of Electrical EE&CS at MIT and the Director of the Computer Science and Artificial Intelligence Laboratory. Earlier in his career, Victor conducted research in acoustic phonetics and phonology in American English. Subsequently, his research interest shifted to the development of spoken language interfaces to make human – computer interactions easier and more natural.

Amazon EX1010 Page 1127

Detailed Contents

List of Abbreviations XXXI

1 **Introduction to Speech Processing**
J. Benesty, M. M. Sondhi, Y. Huang 1
1.1 A Brief History of Speech Processing 1
1.2 Applications of Speech Processing 2
1.3 Organization of the Handbook 4
References 4

Part A Production, Perception, and Modeling of Speech

2 **Physiological Processes of Speech Production**
K. Honda 7
2.1 Overview of Speech Apparatus 7
2.2 Voice Production Mechanisms 8
2.2.1 Regulation of Respiration 8
2.2.2 Structure of the Larynx 9
2.2.3 Vocal Fold and its Oscillation 10
2.2.4 Regulation of Fundamental Frequency (F_0) 12
2.2.5 Methods for Measuring Voice Production 13
2.3 Articulatory Mechanisms 14
2.3.1 Articulatory Organs 14
2.3.2 Vocal Tract and Nasal Cavity 18
2.3.3 Aspects of Articulation in Relation to Voicing 19
2.3.4 Articulators' Mobility and Coarticulation 22
2.3.5 Instruments for Observing Articulatory Dynamics 23
2.4 Summary 24
References 25

3 **Nonlinear Cochlear Signal Processing and Masking in Speech Perception**
J. B. Allen 27
3.1 Basics 27
3.1.1 Function of the Inner Ear 28
3.1.2 History of Cochlear Modeling 31
3.2 The Nonlinear Cochlea 35
3.2.1 Cochlear Modeling 35
3.2.2 Outer-Hair-Cell Transduction 41
3.2.3 Micromechanics 42
3.3 Neural Masking 45
3.3.1 Basic Definitions 47
3.3.2 Empirical Models 51

3.3.3 Models of the JND 51
3.3.4 A Direct Estimate of the Loudness JND 52
3.3.5 Determination of the Loudness SNR 54
3.3.6 Weber–Fraction Formula 54
3.4 Discussion and Summary 55
3.4.1 Model Validation 55
3.4.2 The Noise Model 55
References 56

4 **Perception of Speech and Sound**
B. Kollmeier, T. Brand, B. Meyer 61
4.1 Basic Psychoacoustic Quantities 62
4.1.1 Mapping of Intensity into Loudness 62
4.1.2 Pitch 64
4.1.3 Temporal Analysis and Modulation Perception 65
4.1.4 Binaural Hearing 67
4.1.5 Binaural Noise Suppression 68
4.2 Acoustical Information Required for Speech Perception 70
4.2.1 Speech Intelligibility and Speech Reception Threshold (SRT) 70
4.2.2 Measurement Methods 71
4.2.3 Factors Influencing Speech Intelligibility 72
4.2.4 Prediction Methods 72
4.3 Speech Feature Perception 74
4.3.1 Formant Features 75
4.3.2 Phonetic and Distinctive Feature Sets 76
4.3.3 Internal Representation Approach and Higher-Order Temporal-Spectral Features 77
4.3.4 Man–Machine Comparison 80
References 81

5 **Speech Quality Assessment**
V. Grancharov, W. B. Kleijn 83
5.1 Degradation Factors Affecting Speech Quality 84
5.2 Subjective Tests 85
5.2.1 Single Metric (Integral Speech Quality) 85
5.2.2 Multidimensional Metric (Diagnostic Speech-Quality) 87
5.2.3 Assessment of Specific Quality Dimensions 87
5.2.4 Test Implementation 88
5.2.5 Discussion of Subjective Tests 89
5.3 Objective Measures 90
5.3.1 Intrusive Listening Quality Measures 90
5.3.2 Non-Intrusive Listening Quality Measures 93
5.3.3 Objective Measures for Assessment of Conversational Quality 94
5.3.4 Discussion of Objective Measures 94

5.4 Conclusions 95
References 96

Part B Signal Processing for Speech

6 Wiener and Adaptive Filters
J. Benesty, Y. Huang, J. Chen 103
6.1 Overview 103
6.2 Signal Models 104
6.2.1 SISO Model 104
6.2.2 SIMO Model 105
6.2.3 MISO Model 105
6.2.4 MIMO Model 106
6.3 Derivation of the Wiener Filter 106
6.4 Impulse Response Tail Effect 107
6.5 Condition Number 108
6.5.1 Decomposition of the Correlation Matrix 108
6.5.2 Condition Number with the Frobenius Norm 108
6.5.3 Fast Computation of the Condition Number 110
6.6 Adaptive Algorithms 110
6.6.1 Deterministic Algorithm 110
6.6.2 Stochastic Algorithm 112
6.6.3 Variable-Step-Size NLMS Algorithm 113
6.6.4 Proportionate NLMS Algorithms 114
6.6.5 Sign Algorithms 116
6.7 MIMO Wiener Filter 116
6.7.1 Conditioning of the Covariance Matrix 117
6.8 Conclusions 119
References 120

7 Linear Prediction
J. Benesty, J. Chen, Y. Huang 121
7.1 Fundamentals 121
7.2 Forward Linear Prediction 122
7.3 Backward Linear Prediction 123
7.4 Levinson–Durbin Algorithm 124
7.5 Lattice Predictor 126
7.6 Spectral Representation 127
7.7 Linear Interpolation 128
7.8 Line Spectrum Pair Representation 129
7.9 Multichannel Linear Prediction 130
7.10 Conclusions 133
References 133

8 **The Kalman Filter**
S. Gannot, A. Yeredor 135
8.1 Derivation of the Kalman Filter 136
8.1.1 The Minimum Mean Square Linear Optimal Estimator 136
8.1.2 The Estimation Error: Necessary and Sufficient Conditions for Optimality 137
8.1.3 The Kalman Filter 138
8.2 Examples: Estimation of Parametric Stochastic Process from Noisy Observations 141
8.2.1 Autoregressive (AR) Process 142
8.2.2 Moving-Average (MA) Process 143
8.2.3 Autoregressive Moving-Average (ARMA) Process 143
8.2.4 The Case of Temporally Correlated Noise 143
8.3 Extensions of the Kalman Filter 144
8.3.1 The Kalman Predictor 144
8.3.2 The Kalman Smoother 145
8.3.3 The Extended Kalman Filter 148
8.4 The Application of the Kalman Filter to Speech Processing 149
8.4.1 Literature Survey 149
8.4.2 Speech Enhancement 151
8.4.3 Speaker Tracking 154
8.5 Summary 157
References 157

9 **Homomorphic Systems and Cepstrum Analysis of Speech**
R. W. Schafer 161
9.1 Definitions 161
9.1.1 Definition of the Cepstrum 162
9.1.2 Homomorphic Systems 162
9.1.3 Numerical Computation of Cepstra 163
9.2 Z-Transform Analysis 164
9.3 Discrete-Time Model for Speech Production 165
9.4 The Cepstrum of Speech 166
9.4.1 Short-Time Cepstrum of Speech 166
9.4.2 Homomorphic Filtering of Speech 168
9.5 Relation to LPC 169
9.5.1 LPC Versus Cepstrum Smoothing 169
9.5.2 Cepstrum from LPC Model 170
9.5.3 Minimum Phase and Recursive Computation 170
9.6 Application to Pitch Detection 171
9.7 Applications to Analysis/Synthesis Coding 172
9.7.1 Homomorphic Vocoder 173
9.7.2 Homomorphic Formant Vocoder 174
9.7.3 Analysis-by-Synthesis Vocoder 175
9.8 Applications to Speech Pattern Recognition 176
9.8.1 Compensation for Linear Filtering 177
9.8.2 Weighted Distance Measures 177

9.8.3 Group Delay Spectrum 178
9.8.4 Mel-Frequency Cepstrum Coefficients (MFCC) 179
9.9 Summary 180
References 180

10 **Pitch and Voicing Determination of Speech with an Extension Toward Music Signals**
W. J. Hess 181
10.1 Pitch in Time-Variant Quasiperiodic Acoustic Signals 182
10.1.1 Basic Definitions 182
10.1.2 Why is the Problem Difficult? 184
10.1.3 Categorizing the Methods 185
10.2 Short-Term Analysis PDAs 185
10.2.1 Correlation and Distance Function 185
10.2.2 Cepstrum and Other Double-Transform Methods 187
10.2.3 Frequency-Domain Methods: Harmonic Analysis 188
10.2.4 Active Modeling 190
10.2.5 Least Squares and Other Statistical Methods 191
10.2.6 Concluding Remarks 192
10.3 Selected Time-Domain Methods 192
10.3.1 Temporal Structure Investigation 192
10.3.2 Fundamental Harmonic Processing 193
10.3.3 Temporal Structure Simplification 193
10.3.4 Cascaded Solutions 195
10.4 A Short Look into Voicing Determination 195
10.4.1 Simultaneous Pitch and Voicing Determination 196
10.4.2 Pattern-Recognition VDAs 197
10.5 Evaluation and Postprocessing 197
10.5.1 Developing Reference PDAs with Instrumental Help 197
10.5.2 Error Analysis 198
10.5.3 Evaluation of PDAs and VDAs– Some Results 200
10.5.4 Postprocessing and Pitch Tracking 201
10.6 Applications in Speech and Music 201
10.7 Some New Challenges and Developments 203
10.7.1 Detecting the Instant of Glottal Closure 203
10.7.2 Multiple Pitch Determination 204
10.7.3 Instantaneousness Versus Reliability 206
10.8 Concluding Remarks 207
References 208

11 **Formant Estimation and Tracking**
D. O'Shaughnessy 213
11.1 Historical 213
11.2 Vocal Tract Resonances 215
11.3 Speech Production 216
11.4 Acoustics of the Vocal Tract 218
11.4.1 Two-Tube Models for Vowels 218

11.4.2 Three-Tube Models for Nasals and Fricatives 219
11.4.3 Obstruents 220
11.4.4 Coarticulation 220
11.5 Short-Time Speech Analysis 221
11.5.1 Vowels 221
11.5.2 Nasals 221
11.5.3 Fricatives and Stops 222
11.6 Formant Estimation 223
11.6.1 Continuity Constraints 223
11.6.2 Use of Phase Shift 224
11.6.3 Smoothing 225
11.7 Summary 226
References 226

12 **The STFT, Sinusoidal Models, and Speech Modification**
M. M. Goodwin 229
12.1 The Short-Time Fourier Transform 230
12.1.1 The STFT as a Sliding-Window Transform 230
12.1.2 The STFT as a Modulated Filter Bank 233
12.1.3 Original Formulation of the STFT 234
12.1.4 The Time Reference of the STFT 234
12.1.5 The STFT as a Heterodyne Filter Bank 235
12.1.6 Reconstruction Methods and Signal Models 235
12.1.7 Examples 236
12.1.8 Limitations of the STFT 240
12.2 Sinusoidal Models 242
12.2.1 Parametric Extension of the STFT 242
12.2.2 The Sinusoidal Signal Model 244
12.2.3 Sinusoidal Analysis and Synthesis 244
12.2.4 Signal Modeling by Matching Pursuit 245
12.2.5 Sinusoidal Matching Pursuits 246
12.2.6 Sinusoidal Analysis 247
12.2.7 Sinusoidal Synthesis 250
12.3 Speech Modification 253
12.3.1 Comparing the STFT and Sinusoidal Models 253
12.3.2 Linear Filtering 253
12.3.3 Enhancement 254
12.3.4 Time-Scale Modification 254
12.3.5 Pitch Modification 255
12.3.6 Cross-Synthesis and Other Modifications 255
12.3.7 Audio Coding with Decode-Side Modification 256
References 256

13 **Adaptive Blind Multichannel Identification**
Y. Huang, J. Benesty, J. Chen 259
13.1 Overview 259
13.2 Signal Model and Problem Formulation 260

13.3 Identifiability and Principle 261
13.4 Constrained Time-Domain Multichannel LMS and Newton Algorithms 262
13.4.1 Unit-Norm Constrained Multichannel LMS Algorithm 262
13.4.2 Unit-Norm Constrained Multichannel Newton Algorithm 265
13.5 Unconstrained Multichannel LMS Algorithm with Optimal Step-Size Control 266
13.6 Frequency-Domain Blind Multichannel Identification Algorithms 268
13.6.1 Frequency-Domain Multichannel LMS Algorithm 268
13.6.2 Frequency-Domain Normalized Multichannel LMS Algorithm 273
13.7 Adaptive Multichannel Exponentiated Gradient Algorithm 276
13.8 Summary 279
References 279

Part C Speech Coding

14 Principles of Speech Coding

W. B. Kleijn 283
14.1 The Objective of Speech Coding 283
14.2 Speech Coder Attributes 284
14.2.1 Rate 284
14.2.2 Quality 285
14.2.3 Robustness to Channel Imperfections 285
14.2.4 Delay 286
14.2.5 Computational and Memory Requirements 286
14.3 A Universal Coder for Speech 286
14.3.1 Speech Segment as Random Vector 286
14.3.2 Encoding Random Speech Vectors 287
14.3.3 A Model of Quantization 288
14.3.4 Coding Speech with a Model Family 289
14.4 Coding with Autoregressive Models 293
14.4.1 Spectral-Domain Index of Resolvability 293
14.4.2 A Criterion for Model Selection 294
14.4.3 Bit Allocation for the Model 295
14.4.4 Remarks on Practical Coding 296
14.5 Distortion Measures and Coding Architecture 296
14.5.1 Squared Error 297
14.5.2 Masking Models and Squared Error 298
14.5.3 Auditory Models and Squared Error 299
14.5.4 Distortion Measure and Coding Architecture 301
14.6 Summary 302
References 303

15 **Voice over IP: Speech Transmission over Packet Networks**
J. Skoglund, E. Kozica, J. Linden, R. Hagen, W. B. Kleijn 307
15.1 Voice Communication 307
15.1.1 Limitations of PSTN 307
15.1.2 The Promise of VoIP 308
15.2 Properties of the Network 308
15.2.1 Network Protocols 308
15.2.2 Network Characteristics 309
15.2.3 Typical Network Characteristics 312
15.2.4 Quality-of-Service Techniques 313
15.3 Outline of a VoIP System 313
15.3.1 Echo Cancelation 314
15.3.2 Speech Codec 315
15.3.3 Jitter Buffer 315
15.3.4 Packet Loss Recovery 316
15.3.5 Joint Design of Jitter Buffer and Packet Loss Concealment 316
15.3.6 Auxiliary Speech Processing Components 316
15.3.7 Measuring the Quality of a VoIP System 317
15.4 Robust Encoding 317
15.4.1 Forward Error Correction 317
15.4.2 Multiple Description Coding 320
15.5 Packet Loss Concealment 326
15.5.1 Nonparametric Concealment 326
15.5.2 Parametric Concealment 327
15.6 Conclusion 327
References 328

16 **Low-Bit-Rate Speech Coding**
A. V. McCree 331
16.1 Speech Coding 331
16.2 Fundamentals: Parametric Modeling of Speech Signals 332
16.2.1 Speech Production 332
16.2.2 Human Speech Perception 334
16.2.3 Vocoders 335
16.3 Flexible Parametric Models 337
16.3.1 Mixed Excitation Linear Prediction (MELP) 337
16.3.2 Sinusoidal Coding 341
16.3.3 Waveform Interpolation 342
16.3.4 Comparison and Contrast of Modeling Approaches 343
16.4 Efficient Quantization of Model Parameters 344
16.4.1 Vector Quantization 344
16.4.2 Exploiting Temporal Properties 344
16.4.3 LPC Filter Quantization 345
16.5 Low-Rate Speech Coding Standards 345
16.5.1 MIL-STD 3005 345
16.5.2 The NATO STANAG 4591 346
16.5.3 Satellite Communications 346

16.5.4 ITU 4 kb/s Standardization 347
16.6 Summary 347
References 347

17 **Analysis-by-Synthesis Speech Coding**
J.-H. Chen, J. Thyssen 351
17.1 Overview 352
17.2 Basic Concepts of Analysis-by-Synthesis Coding 353
17.2.1 Definition of Analysis-by-Synthesis 353
17.2.2 From Conventional Predictive Waveform Coding to a Speech Synthesis Model 353
17.2.3 Basic Principle of Analysis by Synthesis 354
17.2.4 Generic Analysis-by-Synthesis Encoder Structure 355
17.2.5 Reasons for the Coding Efficiency of Analysis by Synthesis 357
17.3 Overview of Prominent Analysis-by-Synthesis Speech Coders 357
17.4 Multipulse Linear Predictive Coding (MPLPC) 360
17.5 Regular-Pulse Excitation with Long-Term Prediction (RPE-LTP) 362
17.6 The Original Code Excited Linear Prediction (CELP) Coder 363
17.7 US Federal Standard FS1016 CELP 367
17.8 Vector Sum Excited Linear Prediction (VSELP) 368
17.9 Low-Delay CELP (LD-CELP) 370
17.10 Pitch Synchronous Innovation CELP (PSI-CELP) 371
17.11 Algebraic CELP (ACELP) 371
17.11.1 ACELP Background 372
17.11.2 ACELP Efficient Search Methods 373
17.11.3 ACELP in Standards 377
17.12 Conjugate Structure CELP (CS-CELP) and CS-ACELP 377
17.13 Relaxed CELP (RCELP) – Generalized Analysis by Synthesis 378
17.13.1 Generalized Analysis by Synthesis Applied to the Pitch Parameters 378
17.13.2 RCELP in Standards 381
17.14 eX-CELP 381
17.14.1 eX-CELP in Standards 382
17.15 iLBC 382
17.16 TSNFC 383
17.16.1 Excitation VQ in TSNFC 385
17.16.2 TSNFC in Standards 386
17.17 Embedded CELP 386
17.18 Summary of Analysis-by-Synthesis Speech Coders 388
17.19 Conclusion 390
References 390

18 **Perceptual Audio Coding of Speech Signals**
J. Herre, M. Lutzky 393
18.1 History of Audio Coding 393
18.2 Fundamentals of Perceptual Audio Coding 394
18.2.1 General Background 394

18.2.2 Coder Structure 394
18.2.3 Perceptual Audio Coding Versus Speech Coding 395
18.3 Some Successful Standardized Audio Coders 396
18.3.1 MPEG-1 396
18.3.2 MPEG-2 396
18.3.3 MPEG-2 Advanced Audio Coding 397
18.3.4 MPEG-4 Advanced Audio Coding 397
18.3.5 Progress in Coding Performance 397
18.4 Perceptual Audio Coding for Real-Time Communication 398
18.4.1 Delay Sources in Perceptual Audio Coding 398
18.4.2 MPEG-4 Low-Delay AAC 399
18.4.3 ITU-T G.722.1-C 401
18.4.4 Ultra-Low-Delay Perceptual Audio Coding 402
18.5 Hybrid/Crossover Coders 403
18.5.1 MPEG-4 Scalable Speech/Audio Coding 404
18.5.2 ITU-T G.729.1 405
18.5.3 AMR-WB+ 406
18.5.4 ARDOR 408
18.6 Summary 409
References 409

Part D Text-to-Speech Synthesis

19 **Basic Principles of Speech Synthesis**
J. Schroeter 413
19.1 The Basic Components of a TTS System 413
19.1.1 TTS Frontend 413
19.1.2 TTS Backend 415
19.2 Speech Representations and Signal Processing for Concatenative Synthesis 421
19.2.1 Time-Domain Pitch Synchronous Overlap Add (TD-PSOLA) 422
19.2.2 LPC-Based Synthesis 423
19.2.3 Sinusoidal Synthesis 423
19.3 Speech Signal Transformation Principles 423
19.3.1 Prosody Transformation Principles 424
19.3.2 Principle Methods for Changing Speaker Characteristics and Speaking Style 424
19.4 Speech Synthesis Evaluation 425
19.5 Conclusions 426
References 426

20 **Rule-Based Speech Synthesis**
R. Carlson, B. Granström 429
20.1 Background 429
20.2 Terminal Analog 429
20.2.1 Formant Synthesizers 429

Detailed Cont.

20.2.2 Higher-Level Parameters 430
20.2.3 Voice Source Models 431
20.3 Controlling the Synthesizer 432
20.3.1 Rule Compilers for Speech Synthesis 432
20.3.2 Data-Driven Parametric Synthesis 433
20.4 Special Applications of Rule-Based Parametric Synthesis 434
20.5 Concluding Remarks 434
References 434

21 **Corpus-Based Speech Synthesis**
T. Dutoit 437
21.1 Basics 437
21.2 Concatenative Synthesis with a Fixed Inventory 438
21.2.1 Diphone-Based Synthesis 438
21.2.2 Modifying Prosody 440
21.2.3 Smoothing Joints 444
21.2.4 Up from Diphones 445
21.3 Unit-Selection-Based Synthesis 447
21.3.1 Selecting Units 447
21.3.2 Target Cost 448
21.3.3 Concatenation Cost 448
21.3.4 Speech Corpus 449
21.3.5 Computational Cost 450
21.4 Statistical Parametric Synthesis 450
21.4.1 HMM-Based Synthesis Framework 451
21.4.2 The State of the Art and Perspectives 453
21.5 Conclusion 453
References 453

22 **Linguistic Processing for Speech Synthesis**
R. Sproat 457
22.1 Why Linguistic Processing is Hard 457
22.2 Fundamentals: Writing Systems and the Graphical Representation of Language 457
22.3 Problems to be Solved and Methods to Solve Them 458
22.3.1 Text Preprocessing 458
22.3.2 Morphological Analysis and Word Pronunciation 461
22.3.3 Syntactic Analysis, Accenting, and Phrasing 462
22.3.4 Sense Disambiguation: Dealing with Ambiguity in Written Language 464
22.4 Architectures for Multilingual Linguistic Processing 465
22.5 Document-Level Processing 465
22.6 Future Prospects 466
References 467

23 **Prosodic Processing**
J. van Santen, T. Mishra, E. Klabbers 471
23.1 Overview 471
23.1.1 What Is Prosody? 471
23.1.2 Prosody in Human–Human Communication 472
23.2 Historical Overview 475
23.2.1 Rule-Based Approaches in Formant Synthesis 475
23.2.2 Statistical Approaches in Diphone Synthesis 475
23.2.3 Using as-is Prosody in Unit Selection Synthesis 475
23.3 Fundamental Challenges 476
23.3.1 Challenge to Unit Selection: Combinatorics of Language ... 476
23.3.2 Challenge to Target Prosody-Based Approaches: Multitude of Interrelated Acoustic Prosodic Features 476
23.4 A Survey of Current Approaches 477
23.4.1 Timing 477
23.4.2 Intonation 479
23.5 Future Approaches 484
23.5.1 Hybrid Approaches 484
23.6 Conclusions 485
References 485

24 **Voice Transformation**
Y. Stylianou 489
24.1 Background 489
24.2 Source–Filter Theory and Harmonic Models 490
24.2.1 Harmonic Model 490
24.2.2 Analysis Based on the Harmonic Model 491
24.2.3 Synthesis Based on the Harmonic Model 491
24.3 Definitions 492
24.3.1 Source Modifications 492
24.3.2 Filter Modifications 493
24.3.3 Combining Source and Filter Modifications 494
24.4 Source Modifications 494
24.4.1 Time-Scale Modification 495
24.4.2 Pitch Modification 496
24.4.3 Joint Pitch and Time-Scale Modification 496
24.4.4 Energy Modification 497
24.4.5 Generating the Source Modified Speech Signal 497
24.5 Filter Modifications 498
24.5.1 The Gaussian Mixture Model 499
24.6 Conversion Functions 499
24.7 Voice Conversion 500
24.8 Quality Issues in Voice Transformations 501
24.9 Summary 502
References 502

25 **Expressive/Affective Speech Synthesis**
N. Campbell 505
25.1 Overview 505
25.2 Characteristics of Affective Speech 506
25.2.1 Intentions and Emotions 506
25.2.2 Message and Filters 507
25.2.3 Coding and Expression 507
25.3 The Communicative Functionality of Speech 508
25.3.1 Multiple Layers of Prosodic Information 509
25.3.2 Text Data versus Speech Synthesis 509
25.4 Approaches to Synthesizing Expressive Speech 510
25.4.1 Emotion in Expressive Speech Synthesis 510
25.5 Modeling Human Speech 512
25.5.1 Discourse-Act Labeling 513
25.5.2 Expressive Speech and Emotion 513
25.5.3 Concatenative Synthesis Using Expressive Speech Samples 515
25.6 Conclusion 515
References 515

Part E Speech Recognition

26 **Historical Perspective of the Field of ASR/NLU**
L. Rabiner, B.-H. Juang 521
26.1 ASR Methodologies 521
26.1.1 Issues in Speech Recognition 522
26.2 Important Milestones in Speech Recognition History 523
26.3 Generation 1 – The Early History of Speech Recognition 524
26.4 Generation 2 – The First Working Systems for Speech Recognition .. 524
26.5 Generation 3 – The Pattern Recognition Approach to Speech Recognition 525
26.5.1 The ARPA SUR Project 527
26.5.2 Research Outside of the ARPA Community 529
26.6 Generation 4 – The Era of the Statistical Model 530
26.6.1 DARPA Programs in Generation 4 532
26.7 Generation 5 – The Future 534
26.8 Summary 534
References 535

27 **HMMs and Related Speech Recognition Technologies**
S. Young 539
27.1 Basic Framework 539
27.2 Architecture of an HMM-Based Recognizer 540
27.2.1 Feature Extraction 540
27.2.2 HMM Acoustic Models 541
27.2.3 *N*-Gram Language Models 543
27.2.4 Decoding and Lattice Generation 544

27.3 HMM-Based Acoustic Modeling 547
27.3.1 Discriminative Training 547
27.3.2 Covariance Modeling 549
27.4 Normalization 550
27.4.1 Mean and Variance Normalization 550
27.4.2 Gaussianization 550
27.4.3 Vocal-Tract-Length Normalization 551
27.5 Adaptation 551
27.5.1 Maximum A Posteriori (MAP) Adaptation 552
27.5.2 ML-Based Linear Transforms 552
27.5.3 Adaptive Training 553
27.6 Multipass Recognition Architectures 554
27.7 Conclusions 554
References 555

28 **Speech Recognition with Weighted Finite-State Transducers**
M. Mohri, F. Pereira, M. Riley 559
28.1 Definitions 559
28.2 Overview 560
28.2.1 Weighted Acceptors 560
28.2.2 Weighted Transducers 561
28.2.3 Composition 562
28.2.4 Determinization 563
28.2.5 Minimization 565
28.2.6 Speech Recognition Transducers 566
28.3 Algorithms 567
28.3.1 Preliminaries 567
28.3.2 Composition 568
28.3.3 Determinization 570
28.3.4 Weight Pushing 572
28.3.5 Minimization 573
28.4 Applications to Speech Recognition 574
28.4.1 Speech Recognition Transducers 574
28.4.2 Transducer Standardization 577
28.5 Conclusion 582
References 582

29 **A Machine Learning Framework for Spoken-Dialog Classification**
C. Cortes, P. Haffner, M. Mohri 585
29.1 Motivation 585
29.2 Introduction to Kernel Methods 586
29.3 Rational Kernels 587
29.4 Algorithms 589
29.5 Experiments 591
29.6 Theoretical Results for Rational Kernels 593
29.7 Conclusion 594
References 595

30 **Towards Superhuman Speech Recognition**
M. Picheny, D. Nahamoo 597
30.1 Current Status 597
30.2 A Multidomain Conversational Test Set 598
30.3 Listening Experiments 599
30.3.1 Baseline Listening Tests 599
30.3.2 Listening Tests to Determine Knowledge Source Contributions 600
30.4 Recognition Experiments 601
30.4.1 Preliminary Recognition Results 602
30.4.2 Results on the Multidomain Test Set 602
30.4.3 System Redesign 606
30.4.4 Coda 606
30.5 Speculation 607
30.5.1 Proposed Human Listening Experiments 607
30.5.2 Promising Incremental Approaches 608
30.5.3 Promising Disruptive Approaches 612
References 614

31 **Natural Language Understanding**
S. Roukos 617
31.1 Overview of NLU Applications 618
31.1.1 Context Dependence 619
31.1.2 Semantic Representation 619
31.2 Natural Language Parsing 620
31.2.1 Decision Tree Parsers 621
31.3 Practical Implementation 623
31.3.1 Classing 623
31.3.2 Labeling 623
31.4 Speech Mining 623
31.4.1 Word Tagging 624
31.5 Conclusion 625
References 626

32 **Transcription and Distillation of Spontaneous Speech**
S. Furui, T. Kawahara 627
32.1 Background 627
32.2 Overview of Research Activities on Spontaneous Speech 628
32.2.1 Classification of Spontaneous Speech 628
32.2.2 Major Projects and Corpora of Spontaneous Speech 629
32.2.3 Issues in Design of Spontaneous Speech Corpora 630
32.2.4 Corpus of Spontaneous Japanese (CSJ) 631
32.3 Analysis for Spontaneous Speech Recognition 632
32.3.1 Observation in Spectral Analysis 632
32.3.2 Analysis of Speaking Rate 634
32.3.3 Analysis of Factors Affecting ASR Accuracy 634

32.4 Approaches to Spontaneous Speech Recognition 635
32.4.1 Effect of Corpus Size 635
32.4.2 Acoustic Modeling 636
32.4.3 Models Considering Speaking Rate 637
32.4.4 Pronunciation Variation Modeling 637
32.4.5 Language Model 638
32.4.6 Adaptation of Acoustic Model 638
32.4.7 Adaptation of Language Model 639
32.5 Metadata and Structure Extraction of Spontaneous Speech 640
32.5.1 Sentence Boundary Detection 640
32.5.2 Disfluency Detection 642
32.5.3 Detection of Topic and Discourse Boundaries 643
32.6 Speech Summarization 644
32.6.1 Categories of Speech Summarization 644
32.6.2 Key Sentence Extraction 645
32.6.3 Summary Generation 646
32.7 Conclusions 647
References 647

33 **Environmental Robustness**
J. Droppo, A. Acero 653
33.1 Noise Robust Speech Recognition 653
33.1.1 Standard Noise-Robust ASR Tasks 654
33.1.2 The Acoustic Mismatch Problem 655
33.1.3 Reducing Acoustic Mismatch 655
33.2 Model Retraining and Adaptation 656
33.2.1 Retraining on Corrupted Speech 656
33.2.2 Single-Utterance Retraining 657
33.2.3 Model Adaptation 657
33.3 Feature Transformation and Normalization 657
33.3.1 Feature Moment Normalization 658
33.3.2 Voice Activity Detection 662
33.3.3 Cepstral Time Smoothing 662
33.3.4 SPLICE – Normalization Learned from Stereo Data 663
33.4 A Model of the Environment 664
33.5 Structured Model Adaptation 667
33.5.1 Analysis of Noisy Speech Features 667
33.5.2 Log-Normal Parallel Model Combination 667
33.5.3 Vector Taylor-Series Model Adaptation 668
33.5.4 Comparison of VTS and Log-Normal PMC 670
33.5.5 Strategies for Highly Nonstationary Noises 670
33.6 Structured Feature Enhancement 671
33.6.1 Spectral Subtraction 671
33.6.2 Vector Taylor-Series Speech Enhancement 673
33.7 Unifying Model and Feature Techniques 675
33.7.1 Noise Adaptive Training 676
33.7.2 Uncertainty Decoding and Missing Feature Techniques 676

33.8 Conclusion 677
References 677

34 **The Business of Speech Technologies**
J. Wilpon, M. E. Gilbert, J. Cohen 681
34.1 Introduction 682
34.1.1 Economic Value of Network-Based Speech Services 682
34.1.2 Economic Value of Device-Based Speech Applications 683
34.1.3 Technology Overview 683
34.2 Network-Based Speech Services 686
34.2.1 The Industry 686
34.2.2 The Service Paradigm and Historical View of Service Deployments 688
34.2.3 Paradigm Shift from Directed-Dialog- to Open-Dialog-Based Services 690
34.2.4 Technical Challenges that Lay Ahead for Network-Based Services 691
34.3 Device-Based Speech Applications 692
34.3.1 The Industry 692
34.3.2 The Device-Based Speech Application Marketplace 692
34.3.3 Technical Challenges that Enabled Mass Deployment 693
34.3.4 History of Device-Based ASR 694
34.3.5 Modern Use of ASR 695
34.3.6 Government Applications of Speech Recognition 695
34.4 Vision/Predications of Future Services – Fueling the Trends 697
34.4.1 Multimodal-Based Speech Services 697
34.4.2 Increased Automation of Service Development Process 699
34.4.3 Complex Problem Solving 699
34.4.4 Speech Mining 700
34.4.5 Mobile Devices 701
34.5 Conclusion 701
References 702

35 **Spoken Dialogue Systems**
V. Zue, S. Seneff 705
35.1 Technology Components and System Development 707
35.1.1 System Architecture 707
35.1.2 Spoken Input Processing 708
35.1.3 Spoken Output Processing 710
35.1.4 Dialogue Management 711
35.2 Development Issues 712
35.2.1 Data Collection 712
35.2.2 Evaluation 713
35.3 Historical Perspectives 714
35.3.1 Large-Scale Government Programs 714
35.3.2 Some Example Systems 715

35.4 New Directions 715
35.4.1 User Simulation 715
35.4.2 Machine Learning and Dialogue Management 716
35.4.3 Portability 717
35.4.4 Multimodal, Multidomain, and Multilingual Application Development 717
35.5 Concluding Remarks 718
References 718

Part F Speaker Recognition

36 Overview of Speaker Recognition

A. E. Rosenberg, F. Bimbot, S. Parthasarathy 725
36.1 Speaker Recognition 725
36.1.1 Personal Identity Characteristics 725
36.1.2 Speaker Recognition Definitions 726
36.1.3 Bases for Speaker Recognition 726
36.1.4 Extracting Speaker Characteristics from the Speech Signal 727
36.1.5 Applications 728
36.2 Measuring Speaker Features 729
36.2.1 Acoustic Measurements 729
36.2.2 Linguistic Measurements 730
36.3 Constructing Speaker Models 731
36.3.1 Nonparametric Approaches 731
36.3.2 Parametric Approaches 732
36.4 Adaptation 735
36.5 Decision and Performance 735
36.5.1 Decision Rules 735
36.5.2 Threshold Setting and Score Normalization 736
36.5.3 Errors and DET Curves 736
36.6 Selected Applications for Automatic Speaker Recognition 737
36.6.1 Indexing Multispeaker Data 737
36.6.2 Forensics 737
36.6.3 Customization: SCANmail 738
36.7 Summary 739
References 739

37 Text-Dependent Speaker Recognition

M. Hébert 743
37.1 Brief Overview 743
37.1.1 Features 744
37.1.2 Acoustic Modeling 744
37.1.3 Likelihood Ratio Score 745
37.1.4 Speaker Model Training 746
37.1.5 Score Normalization and Fusion 746
37.1.6 Speaker Model Adaptation 747

37.2 Text-Dependent Challenges 747
37.2.1 Technological Challenges 747
37.2.2 Commercial Deployment Challenges 748
37.3 Selected Results 750
37.3.1 Feature Extraction 750
37.3.2 Accuracy Dependence on Lexicon 751
37.3.3 Background Model Design 752
37.3.4 T-Norm in the Context of Text-Dependent Speaker Recognition 753
37.3.5 Adaptation of Speaker Models 753
37.3.6 Protection Against Recordings 757
37.3.7 Automatic Impostor Trials Generation 759
37.4 Concluding Remarks 760
References 760

38 **Text-Independent Speaker Recognition**
D. A. Reynolds, W. M. Campbell 763
38.1 Introduction 763
38.2 Likelihood Ratio Detector 764
38.3 Features 766
38.3.1 Spectral Features 766
38.3.2 High-Level Features 766
38.4 Classifiers 767
38.4.1 Adapted Gaussian Mixture Models 767
38.4.2 Support Vector Machines 771
38.4.3 High-Level Feature Classifiers 774
38.4.4 System Fusion 775
38.5 Performance Assessment 776
38.5.1 Task and Corpus 776
38.5.2 Systems 777
38.5.3 Results 777
38.5.4 Computational Considerations 778
38.6 Summary 778
References 779

Part G Language Recognition

39 **Principles of Spoken Language Recognition**
C.-H. Lee 785
39.1 Spoken Language 785
39.2 Language Recognition Principles 786
39.3 Phone Recognition Followed by Language Modeling (PRLM) 788
39.4 Vector-Space Characterization (VSC) 789
39.5 Spoken Language Verification 790
39.6 Discriminative Classifier Design 791

39.7 Summary 793
References 793

40 **Spoken Language Characterization**
M. P. Harper, M. Maxwell 797
40.1 Language versus Dialect 798
40.2 Spoken Language Collections 800
40.3 Spoken Language Characteristics 800
40.4 Human Language Identification 804
40.5 Text as a Source of Information on Spoken Languages 806
40.6 Summary 807
References 807

41 **Automatic Language Recognition Via Spectral and Token Based Approaches**
D. A. Reynolds, W. M. Campbell, W. Shen, E. Singer 811
41.1 Automatic Language Recognition 811
41.2 Spectral Based Methods 812
41.2.1 Shifted Delta Cepstral Features 812
41.2.2 Classifiers 813
41.3 Token-Based Methods 815
41.3.1 Tokens 815
41.3.2 Classifiers 816
41.4 System Fusion 818
41.4.1 Methods 819
41.4.2 Output Scores 820
41.5 Performance Assessment 820
41.5.1 Task and Corpus 820
41.5.2 Systems 821
41.5.3 Results 822
41.5.4 Computational Considerations 822
41.6 Summary 823
References 823

42 **Vector-Based Spoken Language Classification**
H. Li, B. Ma, C.-H. Lee 825
42.1 Vector Space Characterization 826
42.2 Unit Selection and Modeling 827
42.2.1 Augmented Phoneme Inventory (API) 828
42.2.2 Acoustic Segment Model (ASM) 828
42.2.3 Comparison of Unit Selection 829
42.3 Front-End: Voice Tokenization and Spoken Document Vectorization 830
42.4 Back-End: Vector-Based Classifier Design 831
42.4.1 Ensemble Classifier Design 832
42.4.2 Ensemble Decision Strategy 833
42.4.3 Generalized VSC-Based Classification 833

42.5 Language Classification Experiments and Discussion 835
42.5.1 Experimental Setup 835
42.5.2 Language Identification 836
42.5.3 Language Verification 837
42.5.4 Overall Performance Comparison 838
42.6 Summary 838
References 839

Part H Speech Enhancement

43 Fundamentals of Noise Reduction
J. Chen, J. Benesty, Y. Huang, E. J. Diethorn 843
43.1 Noise 843
43.2 Signal Model and Problem Formulation 845
43.3 Evaluation of Noise Reduction 846
43.3.1 Signal-to-Noise Ratio 846
43.3.2 Noise-Reduction Factor and Gain Function 846
43.3.3 Speech-Distortion Index and Attenuation Frequency Distortion 847
43.4 Noise Reduction via Filtering Techniques 847
43.4.1 Time-Domain Wiener Filter 847
43.4.2 A Suboptimal Filter 851
43.4.3 Subspace Method 852
43.4.4 Frequency-Domain Wiener Filter 855
43.4.5 Short-Time Parametric Wiener Filter 857
43.5 Noise Reduction via Spectral Restoration 857
43.5.1 MMSE Spectral Estimator 857
43.5.2 MMSE Spectral Amplitude and Phase Estimators 859
43.5.3 Maximum A Posteriori (MAP) Spectral Estimator 861
43.5.4 Maximum-Likelihood Spectral Amplitude Estimator 861
43.5.5 Maximum-Likelihood Spectral Power Estimator 862
43.5.6 MAP Spectral Amplitude Estimator 863
43.6 Speech-Model-Based Noise Reduction 863
43.6.1 Harmonic-Model-Based Noise Reduction 864
43.6.2 Linear-Prediction-Based Noise Reduction 864
43.6.3 Hidden-Markov-Model-Based Noise Reduction 866
43.7 Summary 868
References 869

44 Spectral Enhancement Methods
I. Cohen, S. Gannot 873
44.1 Spectral Enhancement 874
44.2 Problem Formulation 875
44.3 Statistical Models 876
44.4 Signal Estimation 879
44.4.1 MMSE Spectral Estimation 879
44.4.2 MMSE Log-Spectral Amplitude Estimation 881

44.5 Signal Presence Probability Estimation 881
44.6 A Priori SNR Estimation 882
44.6.1 Decision-Directed Estimation 882
44.6.2 Causal Recursive Estimation 883
44.6.3 Relation Between Causal Recursive Estimation and Decision-Directed Estimation 886
44.6.4 Noncausal Recursive Estimation 887
44.7 Noise Spectrum Estimation 888
44.7.1 Time-Varying Recursive Averaging 888
44.7.2 Minima-Controlled Estimation 889
44.8 Summary of a Spectral Enhancement Algorithm 891
44.9 Selection of Spectral Enhancement Algorithms 896
44.9.1 Choice of a Statistical Model and Fidelity Criterion 896
44.9.2 Choice of an A Priori SNR Estimator 897
44.9.3 Choice of a Noise Estimator 898
44.10 Conclusions 898
References 899

45 **Adaptive Echo Cancelation for Voice Signals**
M. M. Sondhi 903
45.1 Network Echoes 904
45.1.1 Network Echo Canceler 905
45.1.2 Adaptation Algorithms 906
45.2 Single-Channel Acoustic Echo Cancelation 915
45.2.1 The Subband Canceler 916
45.2.2 RLS for Subband Echo Cancelers 917
45.2.3 The Delayless Subband Structure 917
45.2.4 Frequency-Domain Adaptation 918
45.2.5 The Two-Echo-Path Model 919
45.2.6 Variable-Step Algorithm for Acoustic Echo Cancelers 920
45.2.7 Cancelers for Nonlinear Echo Paths 920
45.3 Multichannel Acoustic Echo Cancelation 921
45.3.1 Nonuniqueness of the Misalignment Vector 923
45.3.2 Solutions for the Nonuniqueness Problem 924
45.4 Summary 925
References 926

46 **Dereverberation**
Y. Huang, J. Benesty, J. Chen 929
46.1 Background and Overview 929
46.1.1 Why Speech Dereverberation? 929
46.1.2 Room Acoustics and Reverberation Evaluation 930
46.1.3 Classification of Speech Dereverberation Methods 930
46.2 Signal Model and Problem Formulation 931
46.3 Source Model-Based Speech Dereverberation 932
46.3.1 Speech Models 932
46.3.2 LP Residual Enhancement Methods 933

Amazon EX1010 Page 1149

46.3.3 Harmonic Filtering 934
46.3.4 Speech Dereverberation Using Probabilistic Models 935
46.4 Separation of Speech and Reverberation via Homomorphic Transformation 936
46.4.1 Cepstral Liftering 936
46.4.2 Cepstral Mean Subtraction and High-Pass Filtering of Cepstral Frame Coefficients 936
46.5 Channel Inversion and Equalization 937
46.5.1 Single-Channel Systems 937
46.5.2 Multichannel Systems 938
46.6 Summary 941
References 942

47 **Adaptive Beamforming and Postfiltering**
S. Gannot, I. Cohen 945
47.1 Problem Formulation 947
47.2 Adaptive Beamforming 948
47.2.1 Frequency-Domain Frost Algorithm 948
47.2.2 Frequency-Domain Generalized Sidelobe Canceller 950
47.2.3 Time-Domain Generalized Sidelobe Canceller 952
47.3 Fixed Beamformer and Blocking Matrix 953
47.3.1 Using Acoustical Transfer Functions 953
47.3.2 Using Delay-Only Filters 954
47.3.3 Using Relative Transfer Functions 954
47.4 Identification of the Acoustical Transfer Function 955
47.4.1 Signal Subspace Method 955
47.4.2 Time Difference of Arrival 956
47.4.3 Relative Transfer Function Estimation 956
47.5 Robustness and Distortion Weighting 960
47.6 Multichannel Postfiltering 962
47.6.1 MMSE Postfiltering 963
47.6.2 Log-Spectral Amplitude Postfiltering 964
47.7 Performance Analysis 967
47.7.1 The Power Spectral Density of the Beamformer Output 967
47.7.2 Signal Distortion 968
47.7.3 Stationary Noise Reduction 968
47.8 Experimental Results 972
47.9 Summary 972
47.A Appendix: Derivation of the Expected Noise Reduction for a Coherent Noise Field 973
47.B Appendix: Equivalence Between Maximum SNR and LCMV Beamformers 974
References 975

Amazon EX1010 Page 1150

48 **Feedback Control in Hearing Aids**
A. Spriet, S. Doclo, M. Moonen, J. Wouters 979
48.1 Problem Statement 980
48.1.1 Acoustic Feedback in Hearing Aids 980
48.1.2 Feedforward Suppression Versus Feedback Cancellation ... 981
48.1.3 Performance of a Feedback Canceller 982
48.2 Standard Adaptive Feedback Canceller 982
48.2.1 Adaptation of the CAF 982
48.2.2 Bias of the CAF 984
48.2.3 Reducing the Bias of the CAF 985
48.3 Feedback Cancellation Based on Prior Knowledge of the Acoustic Feedback Path 986
48.3.1 Constrained Adaptation (C-CAF) 986
48.3.2 Bandlimited Adaptation (BL-CAF) 988
48.4 Feedback Cancellation Based on Closed-Loop System Identification 990
48.4.1 Closed-Loop System Setup 990
48.4.2 Direct Method 991
48.4.3 Desired Signal Model 992
48.4.4 Indirect and Joint Input–Output Method 994
48.5 Comparison 995
48.5.1 Steady-State Performance 995
48.5.2 Tracking Performance 995
48.5.3 Measurement of the Actual Maximum Stable Gain 995
48.6 Conclusions 997
References 997

49 **Active Noise Control**
S. M. Kuo, D. R. Morgan 1001
49.1 Broadband Feedforward Active Noise Control 1002
49.1.1 Filtered-X LMS Algorithm 1003
49.1.2 Analysis of the FXLMS Algorithm 1004
49.1.3 Leaky FXLMS Algorithm 1004
49.1.4 Feedback Effects and Solutions 1005
49.2 Narrowband Feedforward Active Noise Control 1006
49.2.1 Introduction 1006
49.2.2 Waveform Synthesis Method 1007
49.2.3 Adaptive Notch Filters 1008
49.2.4 Multiple-Frequency ANC 1009
49.2.5 Active Noise Equalization 1010
49.3 Feedback Active Noise Control 1010
49.4 Multichannel ANC 1011
49.4.1 Principles 1011
49.4.2 Multichannel FXLMS Algorithms 1012
49.4.3 Frequency-Domain Convergence Analysis 1013
49.4.4 Multichannel IIR Algorithm 1014
49.4.5 Multichannel Adaptive Feedback ANC Systems 1015

49.5 Summary 1015
References 1015

Part I Multichannel Speech Processing

50 Microphone Arrays
G. W. Elko, J. Meyer 1021
50.1 Microphone Array Beamforming 1021
50.1.1 Delay-and-Sum Beamforming 1023
50.1.2 Filter-and-Sum Beamforming 1028
50.1.3 Arrays with Directional Elements 1028
50.2 Constant-Beamwidth Microphone Array System 1029
50.3 Constrained Optimization of the Directional Gain 1030
50.4 Differential Microphone Arrays 1031
50.5 Eigenbeamforming Arrays 1034
50.5.1 Spherical Array 1034
50.5.2 Eigenbeamformer 1035
50.5.3 Modal Beamformer 1037
50.6 Adaptive Array Systems 1037
50.6.1 Constrained Broadband Arrays 1038
50.7 Conclusions 1040
References 1040

51 Time Delay Estimation and Source Localization
Y. Huang, J. Benesty, J. Chen 1043
51.1 Technology Taxonomy 1043
51.2 Time Delay Estimation 1044
51.2.1 Problem Formulation and Signal Models 1044
51.2.2 The Family of the Generalized Cross-Correlation Methods . 1045
51.2.3 Adaptive Eigenvalue Decomposition Algorithm 1047
51.2.4 Adaptive Blind Multichannel Identification Based Methods 1048
51.2.5 Multichannel Spatial Prediction and Interpolation Methods 1049
51.2.6 Multichannel Cross-Correlation Coefficient Algorithm 1050
51.2.7 Minimum-Entropy Method 1052
51.3 Source Localization 1054
51.3.1 Problem Formulation 1054
51.3.2 Measurement Model and Cramèr–Rao Lower Bound 1054
51.3.3 Maximum-Likelihood Estimator 1055
51.3.4 Least-Squares Estimators 1056
51.3.5 Least-Squares Error Criteria 1056
51.3.6 Spherical Intersection (SX) Estimator 1057
51.3.7 Spherical Interpolation (SI) Estimator 1057
51.3.8 Linear-Correction Least-Squares Estimator 1058
51.4 Summary 1061
References 1062

52 **Convolutive Blind Source Separation Methods**
M. S. Pedersen, J. Larsen, U. Kjems, L. C. Parra 1065
52.1 The Mixing Model 1066
52.1.1 Special Cases 1067
52.1.2 Convolutive Model in the Frequency Domain 1067
52.1.3 Block-Based Model 1068
52.2 The Separation Model 1068
52.2.1 Feedforward Structure 1068
52.2.2 Relation Between Source and Separated Signals 1068
52.2.3 Feedback Structure 1069
52.2.4 Example: The TITO System 1070
52.3 Identification 1071
52.4 Separation Principle 1071
52.4.1 Higher-Order Statistics 1072
52.4.2 Second-Order Statistics 1073
52.4.3 Sparseness in the Time/Frequency Domain 1075
52.4.4 Priors from Auditory Scene Analysis and Psychoacoustics 1076
52.5 Time Versus Frequency Domain 1076
52.5.1 Frequency Permutations 1077
52.5.2 Time–Frequency Algorithms 1077
52.5.3 Circularity Problem 1078
52.5.4 Subband Filtering 1078
52.6 The Permutation Ambiguity 1078
52.6.1 Consistency of the Filter Coefficients 1078
52.6.2 Consistency of the Spectrum of the Recovered Signals 1081
52.6.3 Global Permutations 1083
52.7 Results 1084
52.8 Conclusion 1084
References 1084

53 **Sound Field Reproduction**
R. Rabenstein, S. Spors 1095
53.1 Sound Field Synthesis 1095
53.2 Mathematical Representation of Sound Fields 1096
53.2.1 Coordinate Systems 1096
53.2.2 Wave Equation 1096
53.2.3 Plane Waves 1097
53.2.4 General Wave Fields in Two Dimensions 1098
53.2.5 Spherical Waves 1099
53.2.6 The Kirchhoff–Helmholtz Integral 1099
53.3 Stereophony 1100
53.3.1 Sine Law 1100
53.3.2 Tangent Law 1101
53.3.3 Application of Amplitude Panning 1102
53.3.4 The Sweet Spot 1102
53.4 Vector-Based Amplitude Panning 1103
53.4.1 Two-Dimensional Vector-Based Amplitude Panning 1103

53.4.2 Three-Dimensional Vector-Based Amplitude Panning 1104
53.4.3 Perception of Vector-Based Amplitude Panning 1104
53.5 Ambisonics 1104
53.5.1 Two-Dimensional Ambisonics 1105
53.5.2 Three-Dimensional Ambisonics 1108
53.5.3 Extensions of Ambisonics 1109
53.6 Wave Field Synthesis 1109
53.6.1 Description of Acoustical Scenes by the Kirchhoff–Helmholtz Integral 1109
53.6.2 Monopole and Dipole Sources 1110
53.6.3 Reduction to Two Spatial Dimensions 1111
53.6.4 Spatial Sampling 1112
53.6.5 Determination of the Loudspeaker Driving Signals 1112
References 1113

Acknowledgements 1115
About the Authors 1117
Detailed Contents 1133
Subject Index 1161

Subject Index

A

abbreviation expansion 460
ABX method 85
accenting 462
acceptor equivalence 563
ACELP 371
acoustic
– alphabet 786
– feature 474
– feedback 980, 1005
– holography 1109
– mismatch 655
– model 529
– model adaptation 604
– modeling 603, 744, 787
– processor 529
– SIMO system 260
– wave propagation 1021
acoustic echo cancelation (AEC) 314, 315, 915
acoustic segment model (ASM) 788, 828
– advantage 829
– training process 829
acoustic word (AW) 786, 788
acoustical environment 653
acoustical transfer-function (ATF) 947
acoustic–phonetic approach 521
active noise cancellation 150
active noise control (ANC) 1001
active noise equalization 1010
adaptation 551, 735
adaptation speed 986
adaptive
– algorithm 906
– array system 1037
– beamforming 948
– blind SIMO identification 260
– codebook 364, 365
– differential pulse code modulation (ADPCM) 352
– feedback cancellation 980
– feedback canceller 982
– filter 110
– noise canceller (ANC) 947, 951, 970
– notch filter 1007, 1009
– predictive coding (APC) 357
– spectral enhancement 338
additive noise 654
advanced front-end (AFE) 672
affect 505
affective speech 506
affine projection 914
affine projection algorithm (APA) 914
agent (A) 706
airflow 10
airline travel information system (ATIS) 532
algebraic code excited linear prediction (ACELP) 407
all-pole transfer function 121
ambisonics 1104
AMI project 630
amplitude modulation 1081
– spectrogram 78
amplitude panning 1100
analog-to-digital (A/D) 283
– converter (ADC) 980
analysis
– by synthesis 296, 353
– by synthesis speech coding 351
– by synthesis vocoder 175
– filterbank 394
– of variance (ANOVA) 89
– synthesis coding 172
– window 875, 892
ANC 960, 961
anti-resonance 217
antisymmetric polynomial 129
aperiodic pulse 338
AR 157
ARDOR (adaptive rate-distortion optimized sound coder) 408
ARISE (automatic railway information systems for Europe) 714
ARMA 157
ARMA filtering 662
ARPA (Advanced Research Projects Agency) 527
ARPA SUR 527
array gain 1031
array processing 946
array response 1024
ART (advanced recognition technology) 695
articulation index (AI) 61, 72
articulatory feature 802
articulatory mechanism 22
articulatory organs 14
artificial neural network (ANN) 530, 790
ASR methodologies 521
ASWD 460
ATF 960, 964, 968–970
ATIS 619
attenuation frequency distortion 847
AT&T (American Telephone and Telegraph) 683, 685–693, 698, 699
audio coding 393
auditory
– event 83
– masking 27
– model 300
– periphery 300
– scene analysis 1076
– spectrogram 77
– system 150
augmented transition network (ATN) 712
AURORA 654
autocorrelation coefficient sequence (ACS) 876
autocorrelation function 169
autocorrelation PDA 186
automatic
– continuous speech recognition 539
– gain control (AGC) 316
– gain normalization (AGN) 658
– language recognition 811
– repeat request (ARQ) 316
– speech attribute transcription (ASAT) 786
– speech recognition (ASR) 3, 80, 149, 560, 627, 653, 708, 786, 787, 817
– speech recognizer 214, 457
– speech transcription 628
automation rate 617
autoregressive (AR) 142, 153, 490, 1068
autoregressive model 293, 985
autoregressive moving-average (ARMA) 142, 143
auxiliary phone symbol 567, 575

average magnitude difference function 186
a posteriori error 113, 276
a posteriori SNR 846–856, 879
– instantaneous narrow-band 859
– instantaneous subband 860, 861
a priori error 113, 276
a priori SNR 846–856, 879–897
– narrow-band 858

B

babble noise 893
back-off weight 574
backward
– integration 930
– linear prediction 123
– masking 66
– prediction error signal 123
– prediction-error power 123
– predictor 108, 123
– predictor matrix 131
Balian–Low theorem 242
bandlimited adaptation 988
bandpass liftering 936
bandwidth extension 397
Bark scale 395, 727
Bark spectral distortion (BSD) 92
Bark spectrogram 77
base phones 541
basilar membrane (BM) 30
– nonlinearity 36
– signal processing 27
Baum–Welch 547
Baum–Welch algorithm 531
Bayes' Rule 540
Bayesian belief network (BBN) 683, 687, 690, 696, 716
Bayesian estimators 135, 151
Bayesian formulation 529
Bayesian information criterion (BIC) 603, 737
beam search 545
beamformer 946
beamforming 938, 946, 1021, 1080
beamwidth 1025
beats 47
behavioral features 725
Bezout theorem 940
bias compensation 889, 891
bias of the CAF 984
bigram 546, 548
binaural hearing 67, 72
binaural intelligibility level difference (BILD) 69
binaural noise suppression 68
binaural technique 1096
biometrics 696, 697, 725
bit errors 285
bitstream encoding 395
Blackman–Harris window 232, 250
BL-CAF 988
blind identifiability 1047
blind identification 1071
blind multichannel identification 259
blind SIMO identifiability 261
blind source separation (BSS) 962, 1065, 1068
block least-squares 913
block Toeplitz 117, 131, 1068
blocking matrix (BM) 947, 951, 953
bluetooth 311
BM 954, 961, 968
BNF grammar 620
Bolt, Beranek, and Newman (BBN) 527, 696
Boolean semiring 568
BoosTexter 592
broadband feedforward ANC 1002
broadcast news (BN) 533, 642
burst 310
BV16 384
BV32 384

C

cache model 639
CAF (continuous adaptation feedback) 982, 997
call home (CH) 533
call routing 618
caller identification system (CIS) 738
cancellative semiring 568
car interior noise 893
cardinal vowel 218
Carnegie Mellon University (CMU) 527
Cartesian coordinates 157, 1096
categorical estimation (CE) 846
causal 170
causal recursive SNR estimation 883, 897
C-CAF 986
center clipper 906
center frequency 224
cepstral
– distance (CD) 90
– features 766
– histogram normalization (CHN) 659
– liftering 936
– mean and variance normalization (CMVN) 659
– mean normalization 658
– mean subtraction (CMS) 730, 766, 937
– time-smoothing 662
– variance normalization (CVN) 659
– weighting 936
cepstrum 162
– analysis 936
– PDA 188
chain 579
channel coding 285
channel diversity 261
channel inversion and equalization 937
channel mismatch 744, 748, 751, 752
channel splitting 322
channel usage 748
channel vocoder 242
CHIL project 630
chirp z-transform (CZT) 225
Cholesky factorization 124
chunking problem 641
circulant matrix 118
circularity problem 1078
claim of identity 748, 749
clarity index 930
class-based language model 544
classification and regression tree (CART) 475, 477, 790
classifier 765–767, 813
classifier fusion 819
classing 623
client (C) 706
clock drift 311
closed loop 353
closed semiring 568
closed-loop
– system 981
– system identification 990
– transfer function 991
closed-set 726
closed-set speaker identification 728
cluster adaptive training (CAT) 553
clustering 530
coarticulation 23, 220
cochlea 27, 31
cochlear amplifier (CA) 44
cochlear map function 33, 34

cochlear modeling 28
cocktail-party effect 921
cocktail-party phenomenon 67, 72
cocktail-party problem 1065
code excited linear prediction (CELP) 332
codebook 287, 363
codec 284, 398, 750
code-excited linear prediction (CELP) 352, 363, 490
coder structure 394
coding 395, 507
coherence function 92, 119, 924
coloration 930
comfort noise generation 316, 317
command-and-control routing 529
common zeros 261, 1047
commutative semiring 567
co-modulation masking release (CMR) 67
comparison category rating (CCR) 86
comparison mean opinion score (CMOS) 86
complex cepstrum 163
complex logarithm 163
complex orthography 806
component-normalization constraint 262
composition algorithm 568, 589
computational auditory scene analysis (CASA) 1076
concept-to-speech generation 711
condition number 108, 129
conditional random fields (CRF) 641
conditional variance 884
confidence estimation 765
confusion network 546
congestion 312
conjugate structure CELP (CS-CELP) 352, 377
connected speech 527
consonant 219, 802
constrained adaptation 986
constrained lexicon 747
constrained MLLR (CMLLR) 552
constrained optimization 879
constrained-entropy coding 289, 290
constrained-resolution coding 289, 291
contact center 681–702
context dependence 619, 623
context-dependency transducer 566, 576
context-oriented clustering (COC) 446
continuity constraint 223
continuous adaptation 982
– feedback (CAF) 997
– feedback (CAF) cancellation 979
continuous speech recognition (CSR) 539
conversational quality 84, 86
conversational telephone speech (CTS) 642
convolutive mixtures 1065
convolutive model 1066
co-occurrence statistics 786
co-prime 1047
co-prime channel impulse responses 261
corollary to the principle of orthogonality 107
corpus of spontaneous Japanese (CSJ) 628, 629, 631, 647
corpus-based speech synthesis 437
correlating transforms 324
correlation matrix 107, 123
correlation vector 123
counting JNDs 50
Cramèr–Rao lower bound (CRLB) 862, 1054, 1055
critical band 65
critical sampling 240
cross relation (CR) 260, 261
cross-correlation vector 107
cross-relation method 261
cross-synthesis 255
CS-ACELP 377
cumulant 1072
cumulative distribution function (CDF) 660
customer Care 690
customer-care application 585
cutoff time delay 930
cycle 568
cyclo-stationarity 1075

D

DARPA 532, 690–696
data collection 712
data-driven parallel model combination (DPMC) 668
data-driven parametric synthesis 433
decision tree 622, 623
– parser 621
decision-directed SNR estimation 882, 886, 897
DECOMP 461
decorrelation 961
degradation mean opinion score (DMOS) 85
delay 286, 309, 398
delay-and-sum beamformer 938, 946, 962
delay-and-sum beamforming 1023
delayed sources 1067
delta cepstrum 177
denoising 254
dependency structure 642, 646
derivation order 621, 622
desired signal 981
desired signal model 992
DET 777
detection cost function (DCF) 736
detection error tradeoff (DET) 736, 777, 822
detection threshold 47
deterministic algorithm 110, 111
deterministic automaton 563
deterministic finite automaton (DFA) 565
deterministic transducer 570
determinization algorithm 563, 570, 577
diagnostic acceptability measure (DAM) 87
diagnostic rhyme test (DRT) 87
diagnostic speech quality 87
dialect 798
dialog
– design 748
– management 690, 691, 698–700
– manager 619
– system 617
dialogue
– management 711
– modeling 711
– system 529
dictation 628, 683, 685, 693, 695, 698
difference of arrival (TDOA) 1044
differential entropy 289
differential microphone array 1031
diffused noise field 970
digital compact cassette, DCC 396
digital signal processing (DSP) 1002
digital-to-analog converter (DAC) 980

diglossia 806
dimensionality reduction 826
– error-correcting-output-coding 839
– feature reduction 834
– linear discriminant analysis 834
– principal component analysis (PCA) 834
– singular value decomposition (SVD) 834
diphone 438
direct method 991
directed dialog 689, 690
directed-dialogue transaction 705
direct-inverse equalizer 938
direction of arrival (DOA) estimation 1044
directivity factor 1030
directivity index (DI) 1030
directivity pattern 1026, 1080
directory services 684–690
discourse marker 642–645
discourse modeling 709
discrete Fourier transform (DFT) 230, 250, 253, 298, 340, 664, 845, 933, 1067
discrete-time Fourier transform (DTFT) 162, 236
discriminative training (DT) 547, 684, 791
disfluency 638–642, 708
distance function 186
distinctive feature 76
distortion 287
distortion measure 285, 301, 875, 896
doctor/specialists paradigm 612
document structure detection 413
dot product 587
double-talk 906
– detector 909
downsampling 233
Dragon 685, 696, 698
duct-acoustic ANC 1005
duration 74
dynamic Bayesian network (DBN) 637
dynamic frequency warping (DFW) 498
dynamic masking 27
dynamic network decoder 547
dynamic programming 525, 526, 544, 684
dynamic time warping (DTW) 424, 494, 525, 745

E

early reverberation 930
early to late energy ratio (ELER) 930
echo 309, 903
– cancelation 314, 904
– canceler 903
– cancellation 946
– return loss (ERL) 905
– return loss enhancement (ERLE) 906
– suppressor 905
eigenbeam 1036
eigenbeamformer 1035
eigendecomposition 111
eigenfrequency 491
eigenvoice modeling 735
electroglottograph (EGG) 441
electroglottography (EGG) 13
electromyographic (EMG) 20
EM (expectation-maximization) algorithm 768, 935
embedded market 683, 692–694
E-model 94
emotion 506, 510, 513
energy modification 497
enhanced variable rate coder (EVRC) 359
enhancement 254
entropy 787, 1052
epsilon cycle 568
equal error rate (EER) 736, 752, 777, 822, 827
equalization and cancelation (EC) 69
equivalent rectangular bandwidth (ERB) 65
– scale 395
ergodic 1053
error concealment 326
error signal 106
estimate smoothing 225
estimate stage (E-step) 150
estimate-maximize (EM) 149–154
Ethnologue 798
Euclidean distance 277
Euclidean norm 155
evaluation
– accuracy 425
– intelligibility 425
– naturalness 425
excess mean-square error 112
excitation signal 121
exclusive-or (XOR) 318
expectation maximization (EM) 499, 541, 768, 813
experienced listener 88
EXPN 460
expression 507
expressive speech 510, 513
– synthesis 505
extended Baum–Welch 548
eXtended CELP (eX-CELP) 381
extended Kalman filter 136, 148
extended wide-band adaptive multirate coder (AMR-WB+) 406
extensible mark-up language (XML) 630
extension model 622

F

F16 cockpit noise 893
factoring algorithm 579
false accept (FA) 749
– rate 749
false reject (FR) 749
– rate 749
fast affine projection (FAP) 915
fast Fourier transform (FFT) 230, 246, 253, 268, 728, 812, 918, 1067
fast recursive least squares (FRLS) 914
fast transversal filter (FTF) 914
FBF (fixed beamformer) 952–954, 959–961, 964, 969, 970, 972
feature 765, 766
– extraction 540, 750
– extraction algorithm 744, 750, 751
– mapping 750
– moment normalization 658
– normalization 657
– set 76
Fechner's postulate 51
feedback
– ANC 1011
– ANC system 1015
– cancellation 982
– canceller 982
– control 980
– effects 1005
– path 980
– signal 981
– structure 1069
feedforward
– structure 1068
– suppression 981
fidelity criterion 896

filler 642
filler rate 635
filter 507
– and-sum beamformer 938, 946
– and-sum beamforming 1028
– bank analysis 727
– bank summation (FBS) 235
– modification 493, 498
filtered-X LMS (FXLMS) 1003
finite
– state
network (FSN) 527
finite impulse response (FIR) 105, 847, 952, 1001, 1027, 1068
finite-state
– machine (FSM) 531
– machines 685, 698, 700
– network (FSN) 522, 531
– transducer (FST) 717
first-in first-out (FIFO) 316
first-order differential array 1032
first-pass transducer 580
Fisher information matrix 1055
Fitzpatrick wide-coverage deterministic syntactic parser (FIDDITCH) 463
fixed beamformer (FBF) 947, 951
fixed rate coder 284
forensic applications 725
forensics 737
formal semantic language 619
formant 19, 213
– bandwidth 223
– correction 255
– estimation 213, 223
– feature 75
– merging 224
– synthesizer 429
– tracker 214
forward
– backward algorithm 531, 589
– compatibility 749
– linear prediction 122
– masking 27, 28
– path 980
– prediction error 122, 1050
– prediction-error power 123
– predictor 108, 122
– predictor matrix 130
forward backward linear predictive coding (FB-LPC) 359
forward error correction (FEC) 317–319
Fourier matrix 269
Fourier transform (FT) 214
fractional pitch 370
frame erasure concealment (FEC) 387
frame-count-dependent thresholding (FCDT) 756
frequency domain 1067
frequency selective switch (FSS) 404
frequency-domain
– adaptation 918
– block error signal 269
– convergence analysis 1013
– FXLMS 1013
– Newton algorithm 274
– PDA 189
– unconstrained multichannel LMS algorithm 270
– unit-norm constrained multichannel LMS algorithm 270
frication 74
fricative 222
fricative sound 308
Frobenius norm 108, 129
Frost algorithm 948, 1039
FS1016 367
Fujisaki intonation model 481
fully excited 261
functional contour (FC) 482
fundamental
– frequency 12, 182, 730
– frequency determination 181
– harmonic processing 193
fuzzy vector quantization (FVQ) 498
FXLMS algorithm 1003, 1005, 1008, 1012, 1013

G

Gabor
– feature 78
– transform 229
gamma model 877, 880, 884, 896
Gaussian
– distribution 135, 150–152
– mixture model (GMM) 93, 425, 499, 733, 745, 767, 786, 788, 812, 813, 834, 835, 837, 839
– model 877, 879, 884, 896
– window 229
Gaussianization 550
Gauss–Markov estimator 959
Geigel algorithm 909
generalized analysis-by-synthesis 378
generalized cross-correlation (GCC) 154, 1045, 1061
generalized eigenvalue decomposition (GEVD) 955
generalized likelihood ratio (GLR) 737
generalized linear discriminant sequence (GLDS) 773, 814, 821
generalized probabilistic descent (GPD) 793
generalized sidelobe canceler (GSC) 939, 950, 952, 956, 1039
GLDS kernel 773
glottal closure instant (GCI) 203, 495
glottal pulse 332
glottis 9
glottography 13
GLOVE synthesizer 430
GMM classifier 813
government applications 682, 685, 687, 690, 694–697, 701, 702
grammar 528
grapheme-to-phoneme conversion 414
graphical representation of language 457
grating lobes 1025
Green's function 1110
group delay spectrum 178
groupe spéciale mobile (GSM) 359
GSC (generalized sidelobe canceller) 954–972

H

Hadamard product 570
half rate standard coder (PDC-HR) 371
Hamming window 232
Hanning window 232
harmonic 217
– analysis 188
– filtering 934
– model 490
– plus-noise model (HNM) 423, 864
– structure 188
H-criterion 549
head-related transfer functions (HRTF) 1076
hearing
– aid 980
– impairment 70, 72, 74
– threshold 49
Helmholtz
– equation 1034
– integral equation 1099

heterodyne filter bank 235
heteroscedastic LDA (HLDA) 550
hidden Markov model (HMM) 3, 94, 151, 421, 521, 528, 530, 539, 559, 607, 628, 684, 685, 734, 745, 767, 787, 815, 828, 834, 863, 932, 1073
hidden Markov process (HMP) 874
high-efficiency AAC (HE-AAC) 397
higher-order Ambisonics 1107
higher-order statistics (HOS) 150, 259, 936, 1071, 1072
high-level features 727, 766
high-pass filter (HPF) 406
Hilbert envelope 933
HMM specification 579
HN model 443
H-norm 746, 765
holophony 1109
homograph disambiguation 414
homomorphic
– formant vocoder 174
– system 162
– transformation 936
– vocoder 173
honestly significant difference (HSD) 89
How May I Help You (HMIHY) 690
human
– human communication 472
– recognition performance 599
– speech 512
– speech perception 334
– speech recognition (HSR) 80
– vocal tract 727
human language
– identification capability 804
– recognition 785
– technology (HLT) 705, 707
hybrid 904
hybrid coder 403
hypopharyngeal cavity 20
hypothesis testing 790

I

IBM (International Business Machines) 683–698
iCampus project 629
idempotent 1058
identification 763
idiolect 727
IIR algorithm 1006, 1014
iLBC 382
ill-conditioned system 1048
image method 157, 953
imposter trials generation 759
improved minima-controlled recursive averaging (IMCRA) 873, 891, 898
improved PNLMS (IPNLMS) 911
impulse response
– maximum phase 174
– minimum phase 173
– zero phase 173
independent component analysis (ICA) 1075
index of resolvability 291
indirect method 994
infinite impulse response (IIR) 105, 1005, 1068
infomax 1073
information
– extraction 618
– index (II) 92
– retrieval (IR) 786, 787, 826
– state update (ISU) 712
inner ear 27, 28
inner hair cell (IHC) 30
instability 981
instant of glottal closure 183
instantaneous frequency 190
instantaneous fundamental frequency 490
instantaneous mixing model 1067
instantaneous phase 491
Institute for Defense Analyses (IDA) 530
integral speech quality 85
intelligibility 438, 845
intelligibility level difference (ILD) 69
intensity 47
– increment 47
– JND 27, 46, 48
– level 47
– modification 493
intention 506
interacting multiple model (IMM) 151
interactive voice response (IVR) 686, 705
internal noise 48
internal representation 62, 77
International Phonetic Alphabet (IPA) 801
International Telecommunication Union (ITU) 85, 426
Internet Engineering Task Force (IETF) 359
Internet protocol (IP) 308, 697
interpolation error signal 128
interpolator 128
intersection algorithm 570
intonation 415, 479
– modeling 475
intrusive objective quality measure 90
inverse discrete Fourier transform (IDFT) 231, 856
inverse discrete-time Fourier transform (IDTFT) 162, 1046
inverse frequency selective switch (IFSS) 404
inverse MDCT (IMDCT) 404
IP phone 313
IPNLMS algorithm 114, 115
IPO approach 481
irrelevancy 284
I-smoothing 549
ISO language codes 798
isophones 62
Itakura distance 122, 846
Itakura–Saito (IS) distance 90, 122, 846
Itakuro–Saito criterion 295
ITU-T (International Telecommunication Union – Telecommunication Standardization Sector) 309

J

J_ToBI 631
Jacobian matrix 1055
jaw 14
jitter 11, 310
– buffer 315
JNAS corpus 634
JND model 51
joint input–output method 994
just noticeable difference (JND) 27, 65, 299

K

Kalman
– filter 135, 136, 138–153, 845, 865
– gain 139, 140, 149, 156
– predictor 136, 144, 145
– smoother 136, 145
Karhunen–Loève transform (KLT) 874
kernel
– dot product 587
– expected counts 589, 594

– gappy 590
– Mercer's condition 587
– methods 586, 587
– n-gram 591
– positive definite symmetric 593
– rational 587
– sequence learning 588
– trick 587
– variance of counts 593
key sentence indexing 645
keyword spotting 530
KIM model 482
Kirchhoff–Helmholtz integral 1099
Klatt model 430, 477
knowledge sources for spoken language identification 800
Kullback–Leibler divergence 277
kurtosis 1072
– maximization 933

L

label model 622
labeling 623
Lagrange multiplier 949, 1052, 1058
language 798
– characterization 797
– detection 827
– identification (LID) 786, 797, 827, 835
– model (LM) 529, 540, 543, 574, 638, 836
– model scaling 540
– modeling 604, 787
– perplexity 528
– recognition evaluation (LRE) 787
– variability 799
– verification 827, 835
– versus dialect 798
language cues 825
– acoustic letters 826, 831, 839
– acoustic words 826, 831
– lexical constraints 831
– spoken document 826
– vocabulary size 826
Laplace distribution 1053
Laplacian model 877, 880, 885, 896
large number of rare events (LNRE) 449
large-margin classification 586, 587, 592
large-vocabulary continuous speech recognition (LVCSR) 628, 803
laryngeal
– cavity 21
– endoscopy 13
– excitation 183
laryngogram 198
laryngograph 198
larynx 9
late reverberation 930
latency 309
latent semantic analysis (LSA) 639, 645
latent semantic indexing (LSI) 787
lattice predictor 126
lattice-based MMI 548
lazy evaluation 562
leaky FXLMS algorithm 1013
leaky LMS 961
learning curve 112
least-mean-square (LMS) 907, 947, 1001
least-mean-square (LMS) algorithm 112
least-mean-square (LMS) convergence 907
least-squares (LS) 938
least-squares estimator (LSE) 1056
least-squares PDA 191
lemmatization 462
Levenstein distance 759
Levinson–Durbin algorithm 110, 124
Levinson–Wiggins–Robinson algorithm 132
lexical content 743, 751, 757
lexical information 803
lexical mismatch 744, 749, 752
LID technology 786
liftering 168
lightly supervised training 630
likelihood 764
– ratio 733, 745
– ratio detector 764
– ratio score 745
– ratio test (LRT) 764, 790
line spectral frequency (LSF) 296, 345, 449, 493
line spectrum pair representation 129
linear
– alignment model 481
– classifier 622, 624, 625
– discriminant analysis (LDA) 685, 819
– discriminant function (LDF) 789, 835
– filtering 253
– interpolation 128
– multivariate regression (LMR) 498
linear prediction (LP) 121, 190, 293, 423, 490, 845, 863
– analysis 223
– coding (LPC) 417, 845
– coefficient (LPC) 3, 149, 151
– residual 186
linear predictive cepstral coefficient (LPCC) 751
linear predictive coding 169, 336, 360, 525, 684, 727
linear shift-invariant system 105
linear time-invariant causal (LTIC) 32
linearly constrained minimum variance (LCMV) 939, 960
– beamformer 948, 949, 974
linearly constrained minimum variance (LCMV) 946–963
Linguistic Data Consortium (LDC) 776
linguistic decoder 529
linguistic processing 457
lips 17
listening effort scale 87
listening quality 84–86
listening region 1109
LMS (least mean square) 961
log mel-frequency filterbank (LMFB) 665
log semiring 564, 568
(log) likelihood ratio (LLR) 736
log-area-ratio (LAR) 90
log-likelihood (LL) 90
log-likelihood ratio 765
log-spectral
– amplitude (LSA) 966
– amplitude estimation 881
– distance 846
– distortion (LSD) 295, 894
Lombard effect 88
long-term prediction (LTP) 400
long-term predictor 362, 993
long-term synthesis filter 356
loudness 49, 62
– JND 27, 49
– recruitment 28, 30
– SNR 50, 54
– temporal integration 48
low sampling rates (LSR) 396
low-bit-rate speech coding 331
low-delay CELP (LD-CELP) 359, 370
low-level speaker features 727
low-time liftering 936

LP residual enhancement methods 933
LP residual signal 932
LPC (linear predictive coding) 521
LPC filter quantization 345
LRE 820
LS (least squares) 938
LS equalizer 938
LSEQ 460
lung volume 9

M

MA (moving average) 157
machine learning 585, 716
machine recognition performance 599
magnetic resonance imaging (MRI) 12, 24
magnitude subtraction rule 672
main lobe 1025
MALACH (multilingual access to large spoken archives) 629
MAP adaptation 552
marker tracking system 24
Markov decision process (MDP) 716
Markov model toolkit (HTK) 821
masked threshold 47
masked threshold intensity 48
masking 49, 298
– curve 299
– effect 394
matched condition training 656
matched filtering 939
matching pursuit 242, 245–247, 249–252
– conjugate-subspace 246, 247, 249
– order recursive 246
– orthogonal 246
– subspace 246
matrix inversion lemma 949, 963, 974
matrix quantization 344
maximal figure-of-merit (MFoM) 792, 833, 835
maximization stage (M-step) 150
maximum a posteriori (MAP) 769, 788, 845, 896
maximum entropy 624
maximum likelihood (ML) 149, 154, 521, 547, 862, 1055, 1073
maximum matching 459
maximum mutual information (MMI) 548, 684
maximum phase 165
maximum signal-to-noise ratio (MSNR) 946
maximum SNR beamformer 949, 974
maximum stable gain 982
maximum-entropy method 641
maximum-likelihood
– decoding 529
– estimate 294
– feature-space regression (FMLLR) 603
– linear regression (MLLR) 552, 603, 638, 685
– model 290
– solution 529
MCRA 965
mean absolute error (MAE) 116
mean opinion score (MOS) 85, 285, 317, 378, 846
mean square error (MSE) 106, 122, 123, 135, 137, 138, 141, 223, 276, 394, 490, 844, 847, 946, 983
measuring normalizing blocks (MNB) 93
mel filter 766
mel frequency cepstral coefficient (MFCC) 493
mel-filter cepstral coefficient 751
mel-frequency cepstral coefficient (MFCC) 179, 214, 449, 540, 602, 666, 729
mel-frequency filterbank 665
Mercer condition 587, 771
message 507
microphone array 946, 1021
million operations per second (MOPS) 380
MIL-STD 3005 345
MIMO system 106
MIMO Wiener filter 116
MIMO Wiener–Hopf equations 117
minima-controlled recursive averaging (MCRA) 889, 960
minimal automaton 573
minimal pairs intelligibility (MPI) 87
minimax 150
minimization 565, 573, 578
minimum Bayes risk (MBR) 647
minimum classification error (MCE) 548, 734, 835
minimum classification error training (MCE) 684
minimum decision cost function minDCF 777
minimum error classification (MCE) 792
minimum error training 533
minimum mean square error (MMSE) 107, 123, 845, 879, 938, 946, 947
minimum phase 126, 165, 170
minimum phone error (MPE) 548
minimum significant difference (MSD) 89
minimum statistics (MS) 889
minimum variance (MV) 1044
minimum verification error (MVE) 792
minimum-phase mixing 1074
minimum-phase system 937
MINT (multichannel inverse theorem) 940
– equalizer 940
MIPS (million instructions per second) 376
misalignment 982
– vector 111
MISO NLMS algorithm 117
MISO system 105
MISO Wiener–Hopf equations 117
MIT 687, 690, 696
mixed excitation 337
mixed excitation linear prediction (MELP) 337
mixed-initiative 706
mixing model 1066
mixture components 287
mixture splitting 543
MMSE 879, 938
– equalizer 938
MMSE of the log-spectral amplitude (MMSE-LSA) 879
MMSE of the spectral amplitude (MMSE-SA) 879
MMSE spectral estimation 879
MMSE-LSA 881, 897
MMSE-STSA 879
mobile devices 682, 692, 693, 697–699, 701
mode matching 1105
model adaptation 656, 657
model retraining 656
model-based noise reduction 863
– harmonic model 864
– HMM model 866
– linear prediction model 864
modern standard Arabic (MSA) 806
modified Bessel function 1053
modified discrete cosine transform (MDCT) 256, 394, 398

modified frequency-domain normalized multichannel LMS algorithm 276
modified rhyme test (MRT) 87
modulated filter bank 233, 235, 242
modulated noise reference unit (MNRU) 88
modulation spectrogram 78
monophone 542
morphological analyzer 461
morphology 803
moving-average (MA) 142, 143
MP3 396
MPEG (moving pictures expert group) 396
MPEG-1 396
MPEG-2 396
MPEG-2 advanced audio coding (MPEG-2 AAC) 397
MPEG-4 397
MPEG-4 low-delay advanced audio coding (AAC-LD) 399
MPEG-4 scalable coding 404
MPLPC (multipuls linear predictive coding) 360
multiband excited (MBE) 341
multiband resynthesis OLA (MBROLA) 444
multicategory (MC) 791
multichannel
– acoustic echo cancelation 921
– backward prediction error vector 131
– cross-correlation coefficient (MCCC) 1051
– EG± algorithm 279
– forward prediction error vector 130
– linear prediction 130
– LMS (MCLMS) 1048
– Newton (MCN) 262
– Wiener–Hopf equations 131
multidelay filter (MDF) 919
multidimensional metric subjective test 87
multidomain conversational test set 598
multilayer perceptron (MLP) 531
multimodal 682, 683, 697–699, 708, 717
– services 682, 683, 692, 697–699
– user interface (MUI) 523
multiple description coding (MDC) 307, 317, 320–324
multiple pitch determination 204
multiple signal classification (MUSIC) 1044
multiple-frequency ANC 1009
multiple-input multiple-output (MIMO) 104
multipulse linear predictive coding (MPLPC) 352, 361
multistage VQ (MSVQ) 344
multistimulus test with hidden reference and anchor (MUSHRA) 86
multistyle training 656
muscular hydrostat 14
musical noise 672
mutual fund 618

N

naïve listener 88
narrowband feedforward ANC 1006
narrowband noise 1013
narrowband spectrogram 217
nasal 221
– cavity 18
National Aeronautics and Space Administration (NASA) 136
National Institute of Science and Technology 811
National Institute of Standards and Technology (NIST) 726
NATO STANAG 4591 346
natural language
– generation (NLG) 710
– processing (NLP) 640
– understanding (NLU) 522, 617, 618, 681, 682, 690, 691, 698, 700, 709
natural modes 111
naturalness 308, 438, 474
N-best interface 709
near miss to Stevens' law 54
nearest-neighbor modeling 732
network echo 903, 904
– cancelation 314
– canceler 905
network-based services 682, 686–689, 691, 701
Neumann series 1060
neural
– masking 27, 45
– network (NN) 93, 150, 477, 521, 745
Neyman–Pearson lemma 790
NIST (National Institute of Standards and Technology) 533, 743, 764, 776, 820, 825
– evaluation tests 533
– language recognition evaluation 825, 827
– LRE 787
NLMS 115, 983
nodal grammar 522
noise
– adaptive training (NAT) 676
– cancelation 149, 150
– estimation 888, 898
– feedback coding (NFC) 383
– reduction 844
– reduction factor 846, 852, 854, 856
– reduction gain 846
noncausal recursive SNR estimation 887, 897
noncontinuous adaptation 982
nondeterministic finite automaton (NFA) 565
nonintrusive quality assessment 93
nonlinear cochlea 27, 35
nonlinear processor 906
nonlinear transformation 925
nonparametric approaches 731
non-stationarity 1074
nonwhiteness 1075
normalized frequency-domain multichannel LMS algorithm 274
normalized least-mean-square (NLMS) 104, 403
normalized LMS 113, 115, 907
normalized misalignment 111
normalized MMSE 107
normalized step-size parameter 113
Nortel 687
North American Business (NAB) 532, 567
North Atlantic Treaty Organization (NATO) 346
notch filter 981
NTT (Nippon Telegraph and Telephone) 687–689
NUANCE 687–695
nuisance attribute projection 774
null space 118
number expression expansion 461
NUU (non-uniform units) 449

O

object (O) 804
objective 83
objective quality measure 90
obstruent 220, 222
OM-LSA 967

open dialog 690–692
open loop 353, 356, 379
open quotient (OQ) 11
open systems interconnection reference (OSI) 285
open-loop system 991
open-set 726
open-set identification 728
optimum step size 266
order-recursive matching pursuit 246
orthogonal matching pursuit 246
orthogonality principle 136
OSI model 308
otoacoustic emission (OAE) 28, 36
outer hair cell (OHC) 27, 30, 41
outer hair cell transduction 41
out-of-vocabulary (OOV) 532, 604
– rate 635, 636
– words 534
overall perception 85
overcomplete 234
overlap-add (OLA) 231, 232, 235, 240, 243, 245–247, 252–254, 326, 444, 893
overlap-save 253, 268
oversampling 231, 234, 250
oversubtraction rule 672

P

packet headers (IP, UDP, RTP) 315
packet loss 310
– concealment (PLC) 310, 316, 326
– recovery 316
packet networks 285
palatography 23
panning law 1100
parallel distributed processing (PDP) 531
parallel model combination (PMC) 667
parallel phone recognition followed by language model (PPRLM) 789, 802, 817, 818
parallel processing method 522
parametric approaches 732
parametric speech coder 332
parametric stereo (PS) 397
PARCOR (partial correlation coefficient) 126, 151
parsing 620
partial correlation coefficients (PARCOR) 126, 151
partial tracking 243, 245
partial-rank algorithm (PRA) 915
particle Kalman filter 151
partitioned-block frequency-domain (PBFD) 983
part-of-speech (POS) 463, 631
path 568
pattern
– clustering 525
– recognition 176
– recognition approach 521
– recognition VDA 197
PBFD (partioned-block frequency domain) 983
PDF (probability density function) 932
peakiness 338
peak-picking 223
PEMAFC 991
perceived quality 297
perceptual
– audio coder 393
– audio coding 393
– domain 297
– evaluation of speech quality (PESQ) 92, 317, 894
– linear prediction (PLP) 93, 540, 602
– model 394
– quality assessment for digital audio (PEAQ) 92
– speech quality measure (PSQM) 92
perfect reconstruction 231–234, 240
performance bounds 321
permutation ambiguity 1078
permutation problem 1077
personal digital cellular (PDC) 359, 371
personal identifying characteristics 725
phantom source 1100
phase
– mismatch 444
– shift 224
– unwrapping 164
– vocoder 242
phone 74, 540
– lattice 817
– recognition 815
– recognition followed by language modeling (PRLM) 788
– recognizer (PR) 815
– recognizer training 816
– reduction 633
phoneme 74, 801
phoneme recognition followed by language modeling (PRLM) 787, 815
– classifier 815
phoneme tracking 526
phonetic 76
– class-based verification (PCBV) 746
– components 458
– decision tree 542
phonology 801
phonotactics 802
photoglottography (PGG) 13
phrasing 463
physiological characteristics 725
Pierrehumbert theory 479
piriform fossa 20
pitch 62, 64, 182, 473, 490, 728, 730
– accent language 462
– contour smoothing 201
– detection 171
– determination 181
algorithms 185
– determination algorithm (PDA) 181
– lag 367
– marker 183, 192
– mismatch 444
– modification 255, 492, 496
formant-corrected 255
– period 332
– predictor 361
– synchronous analysis 494
– synchronous innovation (PSI) 371
– synchronous innovation CELP (PSI-CELP) 359, 371
– tracking 201
plane wave 1022, 1097
– solution 1097
PNLMS algorithm 114, 115
polar coordinates 157, 1096
portability 717
POS tagger 463
positive definite symmetric (PDS) 594
postfilter 150, 962, 963, 968, 972
power spectral density (PSD) 293, 855, 949, 960
power spectral subtraction rule 672
PPRLM classifier 818
precedence effect 68, 930
prediction error 991
prediction error method (PEM) 991
predictor 121
principle of orthogonality 106, 123

probabilistic context-free grammar (PCFG) 621
probability density function (PDF) 499, 875, 1052
probability ratio test 790
probability semiring 568
probe signal 986
product-rule fusion 819
projection matrix 950, 1057, 1059
pronunciation
– dictionary 542
– lexicon 566, 575
– variation 541, 638
proportionate normalized LMS (PNLMS) 276, 910
prosodic feature 642, 643
prosodic information 803
prosodic processing 471
prosody 439, 471
protection against recordings 749, 757, 758
prototype window 243
PSD (power spectral density) 963, 965, 968
psychoacoustics 62, 394
public switched telephone network (PSTN) 307
pulse dispersion filter 339

Q

QMF synthesis filterbank 406
QR decomposition 947
quadratic function 107
quadrature mirror filterbank (QMF) 386, 405
quality 285
quality of service (QoS) 90, 313
quantization 287, 395
– cell 288
quarter-wavelength resonator 218
quefrency 161, 188
question answering (QA) 644

R

random walk 151, 155
range difference 1054
rank 118
rapidly evolving waveform (REW) 343
RASTA (relative spectral processing) 937
– filtering 662, 766
rate 284
rational kernels 586, 587
raw phone accuracy (RPA) 548
RCELP (relaxed CELP) 378
reading machine 509
real-time transport protocol (RTP) 309
receiver operating characteristic (ROC) 736, 822
recruitment 53
recursive 136
– averaging 888
– least squares (RLS) 914
redundancy 284, 289
Reed–Solomon code (RS) 318
reference conditions 88
reflection coefficient 125
regression tree 553
regularization 108
regular-pulse excitation with long-term prediction (RPE-LTP) 359, 362
regulated transducer 568
relative entropy 277
relative spectral processing (RASTA) method 937
relaxed CELP (RCELP) 359
repair 642, 643
repair interval model (RIM) 642
repair rate 635
re-prompt rate (RR) 749
resonant tectorial membrane (RTM) 43
resource management (RM) 532
respiratory system 8
reticular lamina (RL) 30
reverberation 903
– time 157, 930
reweighting 565
rich transcription (RT) 628, 640, 643
RNN intonation model 480
robust algorithm 912
robust encoding 317
robust processing 526
robust risk minimization 624
robust speech recognition 653
robustness 960
room impulse response (RIR) 947, 953, 954
root mean square (RMS) 363
– error (RMSE) 95
ROUGE 647
RSVP protocol 313
RTF 954, 968
rule compiler 432
rule-based speech synthesis 429

S

SALT 698
sample autocorrelation coefficient 876
saturated Poisson internal noise (SPIN) 54
sausage 600
scalar quantization 322
scale ambiguity 262
SCANmail 738
Schur complement 108
score fusion 746
score normalization 746, 765
SCTE (Society of Cable Telecommunications Engineers) 359
secant method 1060
second-order Ambisonics 1107
second-order statistics (SOS) 259, 1047, 1071, 1073
second-pass transducer 581
segment model 636
segmental SNR 90, 893
segmentation 220
segmentation and labeling 525
selectable mode vocoder (SMV) 359
semantic 522
– annotation 621
– language 619
– radicals 458
– representation 619, 620
semiring 588, 589
semi-tied covariance 549
sensation level (SL) 37
sense disambiguation 464
sensitivity function 991
sensitivity matrix 300
sentence
– compaction 644, 646
– extraction 644
– unit 640
SFC model 482
shifted delta cepstral (SDC) 821
– coefficient 812
shimmer 11
shortest-distance algorithm 572, 574, 589, 593
short-term autocorrelation function 186
short-term spectral analysis 727
short-term synthesis filter 356
short-term transformation 185

short-time Fourier transform (STFT) 215, 221, 229, 231–242, 245, 251, 253–256, 875, 892, 948
short-time spectrum 729
short-time speech analysis 221
signal bias removal (SBR) algorithm 937
signal decorrelation operation 985
signal distortion 968
signal presence probability 876, 881, 889, 891
signal-to-noise ratio (SNR) 90, 150, 394, 844, 846, 874, 951
sign-data algorithm 116
sign-error algorithm 116
sign-sign algorithm 116
SIMO system 105, 1045
simultaneous dynamic-masking 37
simultaneous masking 299
sine law of stereophony 1101
single metric subjective test 85
single-input multiple-output (SIMO) 104, 260
single-input single-output (SISO) 103, 261
single-trial loudness 49
singular value decomposition (SVD) 639, 645
sinusoidal
– analysis 244
– coding 341
– model 242–247, 249–256, 490
– synthesis 244
– transform coder (STC) 341
Siren 14 401
SISO system 104
sliding-window transform 230
slowly evolving waveform (SEW) 343
smoothed coherence transform (SCOT) 1046
SNR (signal-to-noise ratio) 960, 970
software library
– AT&T FSM library 592
– GRM library 592
– LLAMA 592
sone scale 64
sonorant 215
sound field 1096
– reproduction 1095
– synthesis 1096
sound pressure level (SPL) 62
source localization 1043, 1054
source localization algorithms
– linear-correction least-squares estimator 1058
– maximum-likelihood estimator 1055
– plane intersection (PI) algorithm 1059
– quadratic-correction least-squares estimator 1060
– spherical interpolation estimator 1057
– spherical intersection estimator 1057
source localization least-squares errors
– hyperbolic least-squares error 1056
– spherical least-squares error 1057
source model-based speech dereverberation 932
source modification 492, 494
source-filter speech synthesis 355
SPAM model 603
sparseness 910, 1071, 1075
spatial
– aperture 1022
– encoding 1107
– filtering 1022
– Fourier transform 1022
– location 62
spatial aliasing 1025
spatial sampling requirement 1025
speaker
– adaptive training 553
– adaptive training (SAT) 636
– detection 726
– diarization 630
– identification 696, 697, 726
– localization 135, 154
– model adaptation 747, 753
– model aging 748
– model behavioral changes 748
– model synthesis 751, 755, 771
– model training 746
– recognition 725, 726, 763
– segmentation 726
– tracking 726
– verification 697, 726
speaker mode 522
– speaker adaptive 522
– speaker-independent 522
– speaker-trained 522
speakerphone 904
speaking environment 522
speaking mode 522
– connected words 522
– continuous speech 522
– isolated words 522
speaking rate 634, 635, 637
speaking situation 522
spectral
– continuity 224
– distortion (SD) 90
– enhancement 874
– enhancement algorithm 891
– envelope mismatch 444
– features 766
– flattening 187
– floor 880
– pitch cues 65
– subtraction 150, 671, 844, 874
– tilt 223
– zero 219
spectral restoration 845, 857
– MAP spectral amplitude estimator 863
– MAP spectral estimator 861
– maximum-likelihood power estimator 861
– maximum-likelihood spectral power estimator 862
– MMSE amplitude estimator 859
– MMSE phase estimator 859
– MMSE spectral estimator 857
spectrogram 215, 221
spectrum 127
speech
– aids 214
– apparatus 7
– archive 628, 644
– articulation 14
– codec 315
– coder attributes 284
– coding 351, 393
– corpus 449
– dereverberation 929
– distortion regularized generalized sidelobe canceller (SDR-GSC) 962
– distortion-weighted multichannel Wiener filter (SDW-MWF) 962
– document 628, 644
– enhancement 136, 149, 151, 844, 874
– in noisy environments (SPINE) 655
– intelligibility (SI) 70, 73, 930
– intelligibility index (SII) 61, 70, 72
– material 71, 72
– mining 617, 618, 623, 701
– pathology 74

– perception 61
– presence probability 958
– production 7, 216, 332
– production model 165, 932
– quality 83
– quality assessment 83
– reception threshold (SRT) 70
– recognition 214, 559, 597, 681–696, 698–701
– recognition transducer 566, 574
– recognition unit 522
– representations 421
– signal processing 421
– signal processing, PSRELP 422
– signal processing, TD-PSOLA 422
– summarization 644
– synthesis 429, 471, 710
– synthesizer 505
– transmission index (STI) 61, 72
– units 419
– units, demisyllable 419
– units, diphone 419
– user interface (SUI) 523
speech model
– harmonic model 863
– hidden Markov model 863
– linear prediction model 863
speech-distortion index 847, 850, 852
speech-to-noise ratio (SNR) 70
speech-to-speech summarization 644
speech-to-speech translation (S2S) 618
speech-to-text summarization 644
Speechworks 687
speed of sound 1054
speed quotient (SQ) 11
spherical array 1034
spherical harmonics 1108
spherical wave 1099
SPLICE (stereo piecewise linear compensation for environment) 663
spoken dialogue system 705–707
spoken-dialog classification 585, 586, 591
spoken-language
– characteristics 800
– characterization 797
– classification 825
– corpora 800
– identification 786
– recognition 785
– verification 790
spontaneous speech recognition 627
SRE 776
stack decoder 547
standardized transducer 579
state clustering 542
statistical decoding algorithm 529
statistical language model (SLM) 640, 709
statistical model 876, 896
statistical parametric synthesis 453
statistical parser 621
statistical speech models 932
steepest-descent algorithm 110
STEM-ML model 483
step-size parameter 111
stereophony 1096, 1100
STFT 967
stochastic
– automaton 572, 574
– dependency context-free grammar (SD-CFG) 646
– gradient 907
– gradient algorithm 112
– language model (SLM) 684
stop 222
stop transition 222
stress language 462
string semiring 564
structured model adaptation 667
structured query language (SQL) 619
subband canceler 916
subglottal pressure 10
subglottal tract 18
subject (S) 804
subjective 83
– quality test 85
subsequential transducer 570
subspace
– method 845, 847, 852
– pursuit 245, 246
subspace-constrained precision and means (SPAM) 549
summarization accuracy 647
sums-of-products approach 478
SUNDIAL (speech understanding and dialog) 714
superhuman speech recognition 597, 601
super-resolution PDA 187
supervised adaptation 747, 754
support vector machine (SVM) 586, 587, 641, 684, 734, 745, 765, 771, 786, 788, 812, 814
support vector modeling 734
suprasegmental 727
SUR (speech understanding research) 527
surface form 637
surveillance applications 725
SVM classifier 814
SVM token classifier 818
sweet spot 1102
switchboard (SB) 533
switchboard corpus 629
syllable 802
symmetric polynomial 129
synchronized OLA (SOLA) 326
synchronized overlap add (SOLA) 498
syntax 522, 804
synthesis
– articulatory 415
– backend 413, 415
– concatenative 419
– formant 416
– frontend 413
– HMM-based 421
– LPC-based 423
– sinusoidal 423
– unit-selection 420
– unit-selection, join cost 420
– unit-selection, target cost 420
synthesis model 429
synthesis window 875, 893
system combinations 837
– PVT–LM 837
– PVT–VSC 837
– UVT–LM 837
– UVT–VSC 837
System Development Corporation (SDC) 527
system fusion 775, 818, 819
system identification 956
system-initiative 705
Systems Development Corporation (SDC) 696

T

tag 472
tagging model 622
tail effect 107
talker normalization 526, 529
tangent law of stereophony 1101
task perplexity 522
task syntax 522, 528
TC-STAR project 630
TDOA 155, 157
TD-PSOLA 440, 497
TDT (topic detection and tracking) 643

tectorial membrane (TM) 30
Telecommunications Industry Association (TIA) 359
telecommunications standardization sector of the International Telecommunications Union (ITU-T) 347
teleconferencing 915, 921
Tellme 688
templates 731
temporal cues 65
temporal envelope fluctuations 66
temporal masking 66, 299
temporal noise shaping (TNS) 400
temporomandibular joint (TMJ) 15
term-document matrix 786
text
– analysis 414
– analysis, component 472
– analysis, etymological 414
– analysis, linguistic–syntactic 414
– analysis, morphological 414
– analysis, phonetic 414
– analysis, prosodic 414
– categorization (TC) 786, 788
– dependent 726
– dependent speaker recognition 743
– markup 414
– normalization 413
– preprocessing 414, 458
– preprocessor 459
text categorization (TC) 826
text-independent 726
text-independent speaker recognition 763
text-to-speech (TTS) 214, 413, 437, 457, 472, 490, 644, 710, 757
text-to-speech synthesizer 699
TF-GSC 964, 965, 967, 972
tilt intonation model 480
timbre 62
time alignment method 526
time constant 111
time delay estimation (TDE) 1043, 1044
time delay estimation algorithm
– adaptive blind multichannel identification (ABMCI) based algorithm 1048
– adaptive eigenvalue decomposition (AED) algorithm 1047
– classical cross-correlation (CCC) algorithm 1046
– minimum entropy algorithm 1053
– multichannel cross-correlation coefficient (MCCC) algorithm 1051
– multichannel spatial interpolation algorithm 1050
– multichannel spatial prediction algorithm 1050
– phase transform (PHAT) algorithm 1047
– smoothed coherence transform (SCOT) algorithm 1046
time difference of arrival (TDOA) 154, 155, 956, 1044, 1061
time scaling 254
time warping 379
time-domain
– aliasing 231, 233, 241, 253
– aliasing cancelation (TDAC) 386, 400
– bandwidth extension (TDBWE) 386
– pitch-synchronous overlap add (TD-PSOLA) 422
– pitch-synchronous overlap-add (TD-PSOLA) 440
time–frequency
– domain 1075, 1076
– masking 1075
– tiling 231
time-localized spectrum 231
time-scale modification 254, 492, 495
timing 472, 477
– of individual phones 414
TITO System 1070
T-norm 746, 753, 765
ToBI (tone and break indices) 415, 479
Toeplitz 107, 123
token 766, 815
– passing 545
tokenization 459, 826
– acoustic units 827
– augmented phoneme inventory (API) 828
– international phonetic alphabet (IPA) 828
– parallel phone recognition (PPR) 828
– parallel voice tokenization (PVT) 829–831, 836, 837
– universal voice tokenization (UVT) 829, 830, 836, 837
tonality 65
tone language 462
tongue 15
topic detection 643
trajectory model 636
transducer 522
– composition 562
– equivalence 564
transfer-function generalized sidelobe canceller (TF-GSC) 947
transform coded excitation (TCX) 407
transform predictive coder (TPC) 386
transient beam-to-reference ratio (TBRR) 965
transinformation 74
– index (TI) 74
transmission channel 523
transmission control protocol (TCP) 285, 308
travel reservation 618
treebank 620, 621
trigger model 639
trim transducer 568
triphone 542
tropical semiring 564, 568
t-test 89
turbulent noise 222
twins property 571
two-echo-path model 919
two-input-two-output (TITO) 1070
two-stage noise feedback coding (TSNFC) 359
two-state noise feedback coding (TSNFC) 383
two-tone suppression (2TS) 28, 37, 39
tympanic membrane (TM) 32

U

ultra-low delay (ULD) 402
ultrasonography 24
uncertainty decoding 676
unconstrained multichannel LMS algorithm 267
undersampling 231
uniformly spaced linear array 1024
unit concatenative distortion (UCD) 420
unit segmental distortion (USD) 420
unit selection 447, 827
unit-norm constrained multichannel LMS algorithm 262
unit-norm constrained multichannel Newton algorithm 265
unit-norm constraint 262

universal background model (UBM) 746, 768, 769, 813
universal speech units 787
unscented Kalman filter (UKF) 150
unscented transform (UT) 150
unsupervised adaptation 638, 747, 754
unvoiced speech 150, 151, 332
unweighted RMS level 63
upsampling 233
upward spread of masking (USM) 28, 37
user (U) 706
user datagram protocol (UDP) 285, 309
user experience (UE) 686
user-initiative 706

V

variable rate coder 284
variable rate smoothing (VRS) 754
variable step-size unconstrained multichannel LMS algorithm 268
variable-rate multimode wide-band (VMR-WB) 359
variable-step-size NLMS algorithm 113
variance kernels 593
variance normalization 550
vector based amplitude panning (VBAP) 1103
vector quantization (VQ) 323, 344, 357, 363, 498, 521, 525
vector quantization modeling 732
vector sum excited linear prediction (VSELP) 359, 368
vector Taylor series 668, 673
vector-based classifier 826, 831
– artificial neural network (ANN) 835
– directed acyclic graph 833
– ensemble classifiers 832
– generalized SVM 832
– maximum wins 833
– one-versus-one 832, 834
– one-versus-rest 832, 834
– quadratic program (QP) 832
– support vector machine 831
vectorization 826, 830
– bag-of-sounds 831, 839
– composite document vector 830
– dimensionality 830
– document vector 830
vector-space
– characterization (VSC) 786, 825–827, 830, 831, 834–837
– modeling 786, 787, 789
velum 17
ventilation 8
verb (V) 804
verification 763
very large-scale integration (VLSI) 904
video conference 154
virtual source 1100
Viterbi algorithm 544
vocabulary size 522
vocal
– fold 10
– fold vibration 10, 183
vocal tract 18, 121, 213
– cavity 222
– constriction 222
– excitation 215, 415
– geometry 415
– impulse response 166
– length normalization (VTLN) 551, 603, 636, 685
– model 219
– resonance 215, 217
vocoder 242, 335
voder (voice coder) 413
voice
– activity detection 316, 317, 662
– activity detector (VAD) 883, 890, 898
– alteration 424
– conversion 425, 500
– dialing 529, 681, 682, 688, 690, 693, 695
– morphing 494
– onset time (VOT) 22
– over IP (VoIP) 307
– portal 682, 687
– production 7
– quality 20
– recognition call processing (VRCP) 523
– source model 431
– tokenization (VT) 830
– transformation 423, 489
– transformation, prosody 424
– XML 692
voiced sound 21
voiced speech 150, 151, 181, 332
voiceless
– excitation 181
– sound 21
voice-over-Internet protocol (VoIP) 359
voiceprint 737
VoiceTone 691
voicing 74
voicing determination 181, 195
voicing determination algorithm (VDA) 195
VoIP (voice over IP) 307, 308
Volterra expansion 920
Voronoi region 288
vowel 221, 802
vowel production 19
VTLN (vocal-tract-length normalization) 684
VXML 692

W

Wall Street Journal (WSJ) 654
wave equation 1097
wave field synthesis 1109
waveform
– coder 352, 353
– interpolation (WI) 342
– similarity OLA (WSOLA) 326
– similarity overlap add (WSOLA) 498
– synthesis method 1007
wavelet transform 933
wavevector 1022
weakly left-divisible semiring 568
Weber's law 51
Weber-Fechner law 51
Weber-fraction formula 54
weighted acceptor 560
weighted automaton 568, 585, 588
weighted distance measure 177
weighted finite-state acceptors (WFSA) 561
weighted finite-state transducer (WFST) 465, 561, 647
weighted least-squares (WLS) 136, 957
weighted MOPS (WMOPS) 381
weighted speech level 64
weighted squared-error criterion 297
weighted transducer 559, 561, 568, 588
weight-pushing algorithm 572
white Gaussian noise 893
white noise 123
whitening 123
wide dynamic-range multiband compression (WDRC) 28

wideband AMR speech coder (AMR-WB) 406
wideband spectrogram 217, 221
Wiener filter 103, 106, 107, 136, 149, 845, 879, 951, 962, 963, 969
– frequency-domain 855
– parametric 847, 857
– suboptimal 851
– time-domain 847
Wiener–Hopf equations 107, 122, 123
wireless (WiFi) 309
wireless LAN (WLAN) 309, 311
Wishart density 552
word
– branching factor 528
– error rate (WER) 629, 713
– insertion penalty 540
– lattice 546, 585, 586, 588, 589
– link record 545
– perplexity 635
– verification 528
word/phrase spotting 523
written versus spoken forms of a language 806

Y

Yule–Walker 149, 150, 153

Z

zero crossing (ZC) 525
zero-forcing equalizer 938
zero-input response (ZIR) 365, 385
zero-padding 1079
zero-state response (ZSR) 365, 385
zero-sum-free semiring 568
Zipf's law 449, 831
Z-norm 746, 765