where $0 < \rho_0 < 1$ and $0 < f_0 < 0.5$. Inserting Eq. (6.89) into (6.87) and (6.88) results in

$$P(z) = 1-(a_1+a_2)z^{-1}-(a_1+a_2)z^{-2}+z^{-3}$$
$$Q(z) = 1-(a_1-a_2)z^{-1}+(a_1-a_2)z^{-2}-z^{-3} \tag{6.90}$$

From Eq. (6.90) we see that $z=-1$ is a root of $P(z)$ and $z=1$ a root of $Q(z)$, which can be divided out and results in

$$P(z) = (1+z^{-1})(1-2\beta_1 z^{-1}+z^{-2})$$
$$Q(z) = (1-z^{-1})(1-2\beta_2 z^{-1}+z^{-2}) \tag{6.91}$$

where β_1 and β_2 are given by

$$\beta_1 = \frac{a_1+a_2+1}{2} = \rho_0\cos(2\pi f_0)+\frac{1-\rho_0^2}{2}$$
$$\beta_2 = \frac{a_1-a_2-1}{2} = \rho_0\cos(2\pi f_0)-\frac{1-\rho_0^2}{2} \tag{6.92}$$

It can be shown that $|\beta_1|<1$ and $|\beta_2|<1$ for all possible values of f_0 and ρ_0. With this property, the roots of $P(z)$ and $Q(z)$ in Eq. (6.91) are complex and given by $\beta_1 \pm j\sqrt{1-\beta_1^2}$ and $\beta_2 \pm j\sqrt{1-\beta_2^2}$, respectively. Because they lie in the unit circle, they can be uniquely represented by their angles

$$\cos(2\pi f_1) = \rho_0\cos(2\pi f_0)+\frac{1-\rho_0^2}{2}$$
$$\cos(2\pi f_2) = \rho_0\cos(2\pi f_0)-\frac{1-\rho_0^2}{2} \tag{6.93}$$

where f_1 and f_2 are the *line spectral frequencies* of $A(z)$. Since $|\rho_0|<1$, $\cos(2\pi f_2) < \cos(2\pi f_0)$, and thus $f_2 > f_0$. It's also the case that $\cos(2\pi f_1) > \cos(2\pi f_0)$ and thus $f_1 < f_0$. Furthermore, as $\rho_0 \to 1$, we see from Eq. (6.93) that $f_1 \to f_0$ and $f_2 \to f_0$. We conclude that, given a pole at f_0, the two line spectral frequencies bracket it, i.e., $f_1 < f_0 < f_2$, and that they are closer together as the pole of the second-order resonator gets closer to the unit circle.

We have proven that for a second-order predictor, the roots of $P(z)$ and $Q(z)$ lie in the unit circle, that ± 1 are roots, and that, once sorted, the roots of $P(z)$ and $Q(z)$ alternate. Although we do not prove it here, it can be shown that these conclusions hold for other predictor orders, and, therefore, the p predictor coefficients can be transformed into p line spectral frequencies. We also know that $z=1$ is always a root of $Q(z)$, whereas $z=-1$ is a root of $P(z)$ for even p and a root of $Q(z)$ for odd p.

To compute the LSF for $p>2$, we replace $z=\cos(\omega)$ and compute the roots of $P(\omega)$ and $Q(\omega)$ by any available root finding method. A popular technique, given that

Amazon/VB Assets
Exhibit 1011
Page 330

there are p roots which are real in ω and bounded between 0 and 0.5, is to bracket them by observing changes in sign of both functions in a dense grid. To compute the predictor coefficients from the LSF coefficients we can factor $P(z)$ and $Q(z)$ as a product of second-order filters as in Eq. (6.91), and then $A(z)=\big(P(z)+Q(z)\big)/2$.

In practice, LSF are useful because of *sensitivity* (a quantization of one coefficient generally results in a spectral change only around that frequency) and *efficiency* (LSF result in low spectral distortion). This doesn't occur with other representations. As long as the LSF coefficients are ordered, the resulting LPC filter is stable, though the proof is beyond the scope of this book. LSF coefficients are used extensively in Chapter 7.

6.3.5.2. Reflection Coefficients

For the autocorrelation method, the predictor coefficients may be obtained from the reflection coefficients by the following recursion:

$$\begin{aligned} a_i^i &= k_i \qquad\qquad i=1,\cdots,p \\ a_j^i &= a_j^{i-1}-k_i a_{i-j}^{i-1} \qquad 1\le j<i \end{aligned} \tag{6.94}$$

where $a_i=a_i^p$. Similarly, the reflection coefficients may be obtained from the prediction coefficients using a backward recursion of the form

$$\begin{aligned} k_i &= a_i^i \qquad\qquad i=p,\cdots,1 \\ a_j^{i-1} &= \frac{a_j^i+a_i^i a_{i-j}^i}{1-k_i^2} \qquad 1\le j<i \end{aligned} \tag{6.95}$$

where we initialize $a_i^p=a_i$.

Reflection coefficients are useful when implementing LPC filters whose values are interpolated over time, because, unlike the predictor coefficients, they are guaranteed to be stable at all times as long as the anchors satisfy Eq. (6.84).

6.3.5.3. Log-Area Ratios

The *log-area ratio* coefficients are defined as

$$g_i=\ln\left(\frac{1-k_i}{1+k_i}\right) \tag{6.96}$$

with the inverse being given by

$$k_i=\frac{1-e^{g_i}}{1+e^{g_i}} \tag{6.97}$$

Amazon/VB Assets
Exhibit 1011
Page 331

The log-area ratio coefficients are equal to the natural logarithm of the ratio of the areas of adjacent sections of a lossless tube equivalent of the vocal tract having the same transfer function. Since for stable predictor filters $-1 < k_i < 1$, we have from Eq. (6.96) that $-\infty < g_i < \infty$. For speech signals, it is not uncommon to have some reflection coefficients close to 1, and quantization of those values can cause a large change in the predictor's transfer function. On the other hand, the log-area ratio coefficients have relatively flat spectral sensitivity (i.e., a small change in their values causes a small change in the transfer function) and thus are useful in coding.

6.3.5.4. Roots of the Polynomial

An alternative to the predictor coefficients results from computing the complex roots of the predictor polynomial:

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k} = \prod_{k=1}^{p} (1 - z_k z^{-1}) \tag{6.98}$$

These roots can be represented as

$$z_k = e^{(-\pi b_k + j2\pi f_k)/F_s} \tag{6.99}$$

where b_k, f_k, and F_s represent the bandwidth, center frequency, and sampling frequency, respectively. Since a_k are real, all complex roots occur in conjugate pairs so that if (b_k, f_k) is a root, so is $(b_k, -f_k)$. The bandwidths b_k are always positive, because the roots are inside the unit circle ($|z_k| < 1$) for a stable predictor. Real roots $z_k = e^{-\pi b_k / F_s}$ can also occur. While algorithms exist to compute the complex roots of a polynomial, in practice there are sometimes numerical difficulties in doing so.

If the roots are available, it is straightforward to compute the predictor coefficients by using Eq. (6.98). Since the roots of the predictor polynomial represent resonance frequencies and bandwidths, they are used in the formant synthesizers of Chapter 16.

6.4. CEPSTRAL PROCESSING

A *homomorphic* transformation $\hat{x}[n] = D(x[n])$ is a transformation that converts a convolution

$$x[n] = e[n] * h[n] \tag{6.100}$$

into a sum

$$\hat{x}[n] = \hat{e}[n] + \hat{h}[n] \tag{6.101}$$

Amazon/VB Assets
Exhibit 1011
Page 332

In this section we introduce the *cepstrum* as one homomorphic transformation [32] that allows us to separate the source from the filter. We show that we can find a value N such that the cepstrum of the filter $\hat{h}[n] \approx 0$ for $n \geq N$, and that the cepstrum of the excitation $\hat{e}[n] \approx 0$ for $n < N$. With this assumption, we can approximately recover both $e[n]$ and $h[n]$ from $\hat{x}[n]$ by homomorphic filtering. In Figure 6.24, we show how to recover $h[n]$ with a homomorphic filter:

$$l[n] = \begin{cases} 1 & |n| < N \\ 0 & |n| \geq N \end{cases} \tag{6.102}$$

where D is the cepstrum operator.

The excitation signal can be similarly recovered with a homomorphic filter given by

$$l[n] = \begin{cases} 1 & |n| \geq N \\ 0 & |n| < N \end{cases} \tag{6.103}$$

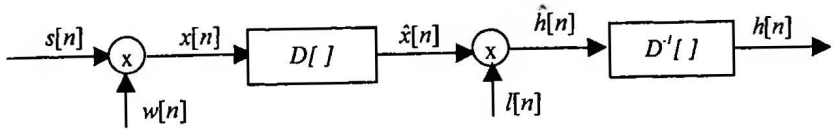


Figure 6.24 Homomorphic filtering to recover the filter's response from a periodic signal. We have used the homomorphic filter of Eq. (6.102).

6.4.1. The Real and Complex Cepstrum

The *real cepstrum* of a digital signal $x[n]$ is defined as

$$c[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln |X(e^{j\omega})| \, e^{j\omega n} d\omega \tag{6.104}$$

and the *complex cepstrum* of $x[n]$ is defined as

$$\hat{x}[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln X(e^{j\omega}) \, e^{j\omega n} d\omega \tag{6.105}$$

where the complex logarithm is used:

$$\hat{X}(e^{j\omega}) = \ln X(e^{j\omega}) = \ln |X(e^{j\omega})| + j\theta(\omega) \tag{6.106}$$

Amazon/VB Assets
Exhibit 1011
Page 333

and the phase $\theta(\omega)$ is given by

$$\theta(\omega) = \arg\left[X(e^{j\omega})\right] \tag{6.107}$$

You can see from Eqs. (6.104) and (6.105) that both the real and the complex cepstrum satisfy Eq. (6.101) and thus they are homomorphic transformations.

If the signal $x[n]$ is real, both the real cepstrum $c[n]$ and the complex cepstrum $\hat{x}[n]$ are also real signals. Therefore the term complex cepstrum doesn't mean that it is a complex signal but rather that the complex logarithm is taken.

It can easily be shown that $c[n]$ is the even part of $\hat{x}[n]$:

$$c[n] = \frac{\hat{x}[n] + \hat{x}[-n]}{2} \tag{6.108}$$

From here on, when we refer to cepstrum without qualifiers, we are referring to the real cepstrum, since it is the most widely used in speech technology.

The cepstrum was invented by Bogert et al. [6], and its term was coined by reversing the first syllable of the word spectrum, given that it is obtained by taking the inverse Fourier transform of the log-spectrum. Similarly, they defined the term *quefrency* to represent the independent variable n in $c[n]$. The quefrency has dimension of time.

6.4.2. Cepstrum of Pole-Zero Filters

A very general type of filters are those with rational transfer functions

$$H(z) = \frac{Az^r \prod_{k=1}^{M_i}(1-a_k z^{-1}) \prod_{k=1}^{M_o}(1-u_k z)}{\prod_{k=1}^{N_i}(1-b_k z^{-1}) \prod_{k=1}^{N_o}(1-v_k z)} \tag{6.109}$$

with the magnitudes of a_k, b_k, u_k, and v_k all less than 1. Therefore, $(1-a_k z^{-1})$ and $(1-b_k z^{-1})$ represent the zeros and poles inside the unit circle, whereas $(1-u_k z)$ and $(1-v_k z)$ represent the zeros and poles outside the unit circle, and z^r is a shift from the time origin. Thus, the complex logarithm is

$$\begin{aligned}\hat{H}(z) &= \ln[A] + \ln[z^r] + \sum_{k=1}^{M_i} \ln(1-a_k z^{-1}) \\ &- \sum_{k=1}^{N_i} \ln(1-b_k z^{-1}) + \sum_{k=1}^{M_o} \ln(1-u_k z) - \sum_{k=1}^{N_o} \ln(1-v_k z)\end{aligned} \tag{6.110}$$

Amazon/VB Assets
Exhibit 1011
Page 334

where the term $\ln[z^r]$ contributes to the imaginary part of the complex cepstrum only with a term $j\omega r$. Since it just carries information about the time origin, it's typically ignored. We use the Taylor series expansion

$$\ln(1-x) = -\sum_{n=1}^{\infty} \frac{x^n}{n} \tag{6.111}$$

in Eq. (6.110) and take inverse z-transforms to obtain

$$\hat{h}[n] = \begin{cases} \log[A] & n = 0 \\ \sum_{k=1}^{N_i} \frac{b_k^n}{n} - \sum_{k=1}^{M_i} \frac{a_k^n}{n} & n > 0 \\ \sum_{k=1}^{M_o} \frac{u_k^n}{n} - \sum_{k=1}^{N_o} \frac{v_k^n}{n} & n < 0 \end{cases} \tag{6.112}$$

If the filter's impulse response doesn't have zeros or poles outside the unit circle, the so-called *minimum phase* signals, then $\hat{h}[n] = 0$ for $n < 0$. *Maximum phase* signals are those with $\hat{h}[n] = 0$ for $n > 0$. If a signal is minimum phase, its complex cepstrum can be uniquely determined from its real cepstrum:

$$\hat{h}[n] = \begin{cases} 0 & n < 0 \\ c[n] & n = 0 \\ 2c[n] & n > 0 \end{cases} \tag{6.113}$$

It is easy to see from Eq. (6.112) that both the real and complex cepstrum are decaying sequences, which is the reason why, typically, a finite number of coefficients are sufficient to approximate it, and, therefore, people refer to the truncated cepstrum signal as a *cepstrum vector.*

6.4.2.1. LPC-Cepstrum

The case when the rational transfer function in Eq. (6.109) has been obtained with an LPC analysis is particularly interesting, since LPC analysis is such a widely used method. While Eq. (6.112) applies here, too, it is useful to find a recursion which doesn't require us to compute the roots of the predictor polynomial. Given the LPC filter

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{6.114}$$

Amazon/VB Assets
Exhibit 1011
Page 335

we take the logarithm

$$\hat{H}(z) = \ln G - \ln\left(1 - \sum_{l=1}^{p} a_l z^{-l}\right) = \sum_{k=-\infty}^{\infty} \hat{h}[k] z^{-k} \tag{6.115}$$

and the derivative of both sides with respect to z

$$\frac{-\sum_{n=1}^{p} n a_n z^{-n-1}}{1 - \sum_{l=1}^{p} a_l z^{-l}} = -\sum_{k=-\infty}^{\infty} k\hat{h}[k] z^{-k-1} \tag{6.116}$$

Multiplying both sides by $-z\left(1 - \sum_{l=1}^{p} a_l z^{-l}\right)$, we obtain

$$\sum_{n=1}^{p} n a_n z^{-n} = \sum_{n=-\infty}^{\infty} n\hat{h}[n] z^{-n} - \sum_{l=1}^{p} \sum_{k=-\infty}^{\infty} k\hat{h}[k] a_l z^{-k-l} \tag{6.117}$$

which, after replacing $l = n - k$, and equating terms in z^{-n}, results in

$$\begin{aligned} n a_n &= n\hat{h}[n] - \sum_{k=1}^{n-1} k\hat{h}[k] a_{n-k} \quad 0 < n \le p \\ 0 &= n\hat{h}[n] - \sum_{k=n-p}^{n-1} k\hat{h}[k] a_{n-k} \quad n > p \end{aligned} \tag{6.118}$$

so that the complex cepstrum can be obtained from the LPC coefficients by the following recursion:

$$\hat{h}[n] = \begin{cases} 0 & n < 0 \\ \ln G & n = 0 \\ a_n + \sum_{k=1}^{n-1} \left(\frac{k}{n}\right) \hat{h}[k] a_{n-k} & 0 < n \le p \\ \sum_{k=n-p}^{n-1} \left(\frac{k}{n}\right) \hat{h}[k] a_{n-k} & n > p \end{cases} \tag{6.119}$$

where the value for $n = 0$ can be obtained from Eqs. (6.115) and (6.111). We note that, while there are a finite number of LPC coefficients, the number of cepstrum coefficients is infinite. Speech recognition researchers have shown empirically that a finite number is sufficient: 12–20 depending on the sampling rate and whether or not frequency warping is done. In Chapter 8 we discuss the use of the cepstrum in speech recognition.

Amazon/VB Assets
Exhibit 1011
Page 336

This recursion should not be used in the reverse mode to compute the LPC coefficients from *any* set of cepstrum coefficients, because the recursion in Eq. (6.119) assumes an all-pole model with all poles inside the unit circle, and that might not be the case for an arbitrary cepstrum sequence, so that the recursion might yield a set of unstable LPC coefficients. In some experiments it has been shown that quantized LPC-cepstrum can yield unstable LPC coefficients over 5% of the time.

6.4.3. Cepstrum of Periodic Signals

It is important to see what the cepstrum of periodic signals looks like. To do so, let's consider the following signal:

$$x[n] = \sum_{k=0}^{M-1} \alpha_k \delta[n - kN] \tag{6.120}$$

which can be viewed as an impulse train of period N multiplied by an analysis window, so that only M impulses remain. Its z-transform is

$$X(z) = \sum_{k=0}^{M-1} \alpha_k z^{-kN} \tag{6.121}$$

which is a polynomial in z^{-N} rather than z^{-1}. Therefore, $X(z)$ can be expressed as a product of factors of the form $(1 - a_k z^{-Nk})$ and $(1 - u_k z^{Nk})$. Following the derivation in Section 6.4.2, it is clear that its complex cepstrum is nonzero only at integer multiples of N:

$$\hat{x}[n] = \sum_{k=-\infty}^{\infty} \beta_k \delta[n - kN] \tag{6.122}$$

A particularly interesting case is when $\alpha_k = \alpha^k$ with $0 < \alpha < 1$, so that Eq. (6.121) can be expressed as

$$X(z) = 1 + \alpha z^{-N} + \cdots + (\alpha z^{-N})^{M-1} = \frac{1 - (\alpha z^{-N})^M}{1 - \alpha z^{-N}} \tag{6.123}$$

so that taking the logarithm of Eq. (6.123) and expanding it in Taylor series using Eq. (6.111) results in

$$\hat{X}(z) = \ln X(z) = \sum_{r=1}^{\infty} \frac{\alpha^r}{r} z^{-rN} - \sum_{l=1}^{\infty} \frac{\alpha^{lM}}{l} z^{-lMN} = \sum_{n=1}^{\infty} \hat{x}[n] z^{-n} \tag{6.124}$$

Amazon/VB Assets
Exhibit 1011
Page 337

which lets us compute the complex cepstrum as

$$\hat{x}[n] = \sum_{r=1}^{\infty} \frac{\alpha^r}{r} \delta[n - rN] - \sum_{l=1}^{\infty} \frac{\alpha^{lM}}{l} \delta[n - lMN] \tag{6.125}$$

An infinite impulse train can be obtained by making $\alpha \to 1$ and $M \to \infty$ in Eq. (6.125).

$$\hat{x}[n] = \sum_{r=1}^{\infty} \frac{\delta[n - rN]}{r} \tag{6.126}$$

We see from Eq. (6.126) that the cepstrum of an impulse train goes to 0 as n increases. This justifies our assumption of homomorphic filtering.

6.4.4. Cepstrum of Speech Signals

We can compute the cepstrum of a speech segment by windowing the signal with a window of length N. In practice, the cepstrum is not computed through Eq. (6.112), since root-finding algorithms are slow and offer numerical imprecision for the large values of N used. Instead, we can compute the cepstrum directly through its definition of Eq. (6.105), using the DFT as follows:

$$X_a[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N}, \quad 0 \le k < N \tag{6.127}$$

$$\hat{X}_a[k] = \ln X_a[k], \quad 0 \le k < N \tag{6.128}$$

$$\hat{x}_a[n] = \frac{1}{N} \sum_{n=0}^{N-1} \hat{X}_a[k] e^{-j2\pi nk/N}, \quad 0 \le n < N \tag{6.129}$$

The subscript a means that the new complex cepstrum $\hat{x}_a[n]$ is an aliased version of $\hat{x}[n]$ given by

$$\hat{x}_a[n] = \sum_{r=-\infty}^{\infty} \hat{x}[n + rN] \tag{6.130}$$

which can be derived by using the sampling theorem of Chapter 5, by reversing the concepts of time and frequency.

This aliasing introduces errors in the estimation that can be reduced by choosing a large value for N.

Amazon/VB Assets
Exhibit 1011
Page 338

Computation of the complex cepstrum requires computing the complex logarithm and, in turn, the phase. However, given the principal value of the phase $\theta_p[k]$, there are infinite possible values for $\theta[k]$:

$$\theta[k] = \theta_p[k] + 2\pi n_k \tag{6.131}$$

From Chapter 5 we know that if $x[n]$ is real, $\arg\left[X(e^{j\omega})\right]$ is an odd function and also continuous. Thus we can do *phase unwrapping* by choosing n_k to guarantee that $\theta[k]$ is a smooth function, i.e., by forcing the difference between adjacent values to be small:

$$|\theta[k] - \theta[k-1]| < \pi \tag{6.132}$$

A linear phase term r as in Eq. (6.110), would contribute to the phase difference in Eq. (6.132) with $2\pi r/N$, which may result in errors in the phase unwrapping if $\theta[k]$ is changing sufficiently rapidly. In addition, there could be large changes in the phase difference if $X_a[k]$ is noisy. To guarantee that we can track small phase differences, a value of N several times larger than the window size is required: i.e., the input signal has to be zero-padded prior to the FFT computation. Finally, the delay r in Eq. (6.109), can be obtained by forcing the phase to be an odd function, so that:

$$\theta[N/2] = \pi r \tag{6.133}$$

For unvoiced speech, the unwrapped phase is random, and therefore only the real cepstrum has meaning. In practical situations, even voiced speech has some frequencies at which noise dominates (typically very low and high frequencies), which results in phase $\theta[k]$ that changes drastically from frame to frame. Because of this, the complex cepstrum in Eq. (6.105) is rarely used for real speech signals. Instead, the real cepstrum is used much more often:

$$C_a[k] = \ln|X_a[k]|, \quad 0 \le k < N \tag{6.134}$$

$$c_a[n] = \frac{1}{N}\sum_{n=0}^{N-1} C_a[k] e^{-j2\pi nk/N}, \quad 0 \le n < N \tag{6.135}$$

Similarly, it can be shown that for the new real cepstrum $c_a[n]$ is an aliased version of $c[n]$ given by

$$c_a[n] = \sum_{r=-\infty}^{\infty} c[n + rN] \tag{6.136}$$

which again has aliasing that can be reduced by choosing a large value for N.

Amazon/VB Assets
Exhibit 1011
Page 339

6.4.5. Source-Filter Separation via the Cepstrum

We have seen that, if the filter is a rational transfer function, and the source is an impulse train, the homomorphic filtering of Figure 6.24 can approximately separate them. Because of problems in estimating the phase in speech signals (see Section 6.4.4), we generally compute the real cepstrum using Eqs. (6.127), (6.134), and (6.135), and then compute the complex cepstrum under the assumption of a minimum phase signal according to Eq. (6.113). The result of separating source and filter using this cepstral deconvolution is shown in Figure 6.25 for voiced speech and Figure 6.26 for unvoiced speech.

The real cepstrum of white noise $x[n]$ with an expected magnitude spectrum $|X(e^{j\omega})|=1$ is 0. If colored noise is present, the cepstrum of the observed colored noise $\hat{y}[n]$ is identical to the cepstrum of the coloring filter $\hat{h}[n]$, except for a gain factor. The above is correct if we take an infinite number of noise samples, but in practice, this cannot be done and a limited number have to be used, so that this is only an approximation, though it is often used in speech processing algorithms.

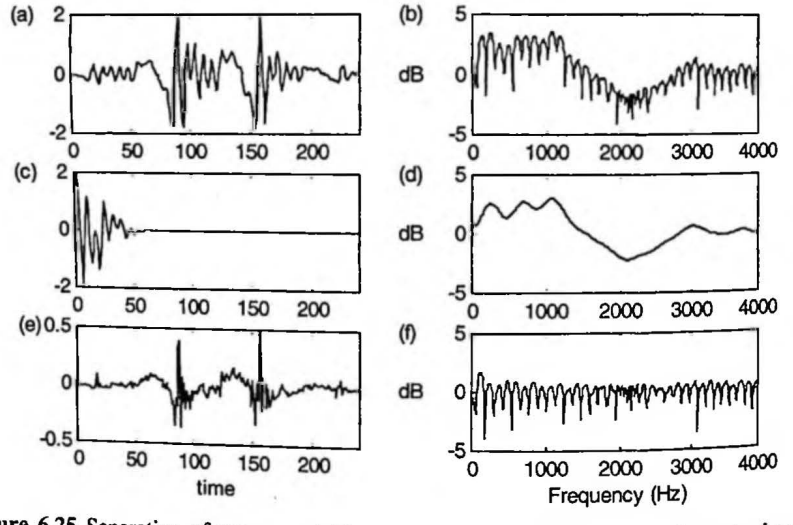


Figure 6.25 Separation of source and filter using homomorphic filtering for voiced speech with the scheme of Figure 6.24 with $N = 20$ in the homomorphic filter of Eq. (6.102) with the real cepstrum: (a) windowed signal, (b) log-spectrum, (c) filter's impulse response, (d) smoothed log-spectrum, (e) windowed excitation signal, (f) log-spectrum of high-part of cepstrum. Note that the windowed excitation is not a windowed impulse train because of the minimum phase assumption.

Amazon/VB Assets
Exhibit 1011
Page 340

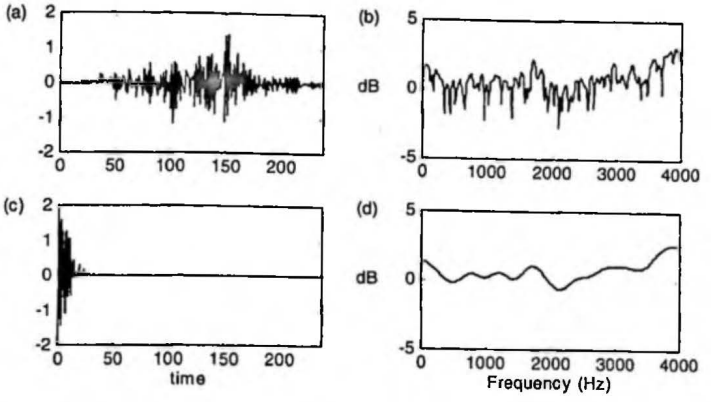


Figure 6.26 Separation of source and filter using homomorphic filtering for unvoiced speech with the scheme of Figure 6.24 with $N = 20$ in the homomorphic filter of Eq. (6.102) with the real cepstrum: (a) windowed signal, (b) log-spectrum, (c) filter's impulse response, (d) smoothed log-spectrum.

6.5. PERCEPTUALLY MOTIVATED REPRESENTATIONS

In this section we describe some aspects of human perception, and methods motivated by the behavior of the human auditory system: bilinearly transformed cepstrum, Mel-Frequency Cepstrum Coefficients (MFCC), and Perceptual Linear Prediction (PLP). These methods have been successfully used in speech recognition.

6.5.1. The Bilinear Transform

The transformation

$$s = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}} \tag{6.137}$$

for $0 < \alpha < 1$ belongs to the class of *bilinear* transforms. It is a mapping in the complex plane that maps the unit circle onto itself. The frequency transformation is obtained by making the substitution $z = e^{j\omega}$ and $s = e^{j\Omega}$:

$$\Omega = \omega + 2\arctan\left[\frac{\alpha \sin(\omega)}{1 - \alpha \cos(\omega)}\right] \tag{6.138}$$

Amazon/VB Assets
Exhibit 1011
Page 341

This transformation is very similar to the Bark and mel scale for an appropriate choice of the parameter α (see Chapter 2). Oppenheim [31] showed that the advantage of this transformation is that it can be used to transform a time sequence in the linear frequency into another time sequence in the warped frequency, as shown in Figure 6.27. This bilinear transform has been successfully applied to cepstral and autocorrelation coefficients.

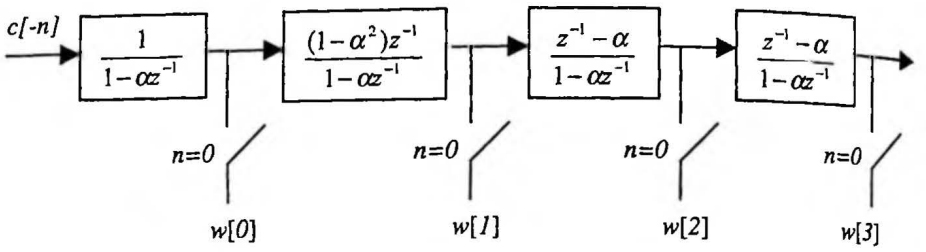


Figure 6.27 Implementation of the frequency-warped cepstral coefficients as a function of the linear-frequency cepstrum coefficients. Both sets of coefficients are causal. The input is the time-reversed cepstrum sequence, and the output can be obtained by sampling the outputs of the filters at time $n = 0$. The filters used for $w[m]$ $m > 2$ are the same. Note that, for a finite-length cepstrum, an infinite-length warped cepstrum results.

For a finite number of cepstral coefficients the bilinear transform in Figure 6.27 results in an infinite number of warped cepstral coefficients. Since truncation is usually done in practice, the bilinear transform is equivalent to a matrix multiplication, where the matrix is a function of the warping parameter α. Shikano [43] showed these warped cepstral coefficients were beneficial for speech recognition.

6.5.2. Mel-Frequency Cepstrum

The *Mel-Frequency Cepstrum Coefficients* (MFCC) is a representation defined as the real cepstrum of a windowed short-time signal derived from the FFT of that signal. The difference from the real cepstrum is that a nonlinear frequency scale is used, which approximates the behavior of the auditory system. Davis and Mermelstein [8] showed the MFCC representation to be beneficial for speech recognition.

Given the DFT of the input signal

$$X_a[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N}, \quad 0 \le k < N \tag{6.139}$$

we define a filterbank with M filters ($m = 1, 2, \cdots, M$), where filter m is triangular filter given by:

Amazon/VB Assets
Exhibit 1011
Page 342

$$H_m[k] = \begin{cases} 0 & k < f[m-1] \\ \dfrac{2(k - f[m-1])}{(f[m+1] - f[m-1])(f[m] - f[m-1])} & f[m-1] \le k \le f[m] \\ \dfrac{2(f[m+1] - k)}{(f[m+1] - f[m-1])(f[m+1] - f[m])} & f[m] \le k \le f[m+1] \\ 0 & k > f[m+1] \end{cases} \tag{6.140}$$

Such filters compute the average spectrum around each center frequency with increasing bandwidths, and they are displayed in Figure 6.28.

Alternatively, the filters can be chosen as

$$H'_m[k] = \begin{cases} 0 & k < f[m-1] \\ \dfrac{(k - f[m-1])}{(f[m] - f[m-1])} & f[m-1] \le k \le f[m] \\ \dfrac{(f[m+1] - k)}{(f[m+1] - f[m])} & f[m] \le k \le f[m+1] \\ 0 & k > f[m+1] \end{cases} \tag{6.141}$$

which satisfies $\sum_{m=1}^{M} H'_m[k] = 1$. The mel-cepstrum computed with $H_m[k]$ or $H'_m[k]$ will differ by a constant vector for all inputs, so the choice becomes unimportant when used in a speech recognition system that has been trained with the same filters.

Let's define f_l and f_h to be the lowest and highest frequencies of the filterbank in Hz, F_s the sampling frequency in Hz, M the number of filters, and N the size of the FFT. The boundary points $f[m]$ are uniformly spaced in the mel-scale:

$$f[m] = \left(\frac{N}{F_s}\right) B^{-1}\left(B(f_l) + m\frac{B(f_h) - B(f_l)}{M+1}\right) \tag{6.142}$$

where the mel-scale B is given by Eq. (2.6), and B^{-1} is its inverse

$$B^{-1}(b) = 700\left(\exp(b/1125) - 1\right) \tag{6.143}$$

We then compute the log-energy at the output of each filter as

$$S[m] = \ln\left[\sum_{k=0}^{N-1} |X_a[k]|^2 H_m[k]\right], \qquad 0 < m \le M \tag{6.144}$$

Amazon/VB Assets
Exhibit 1011
Page 343

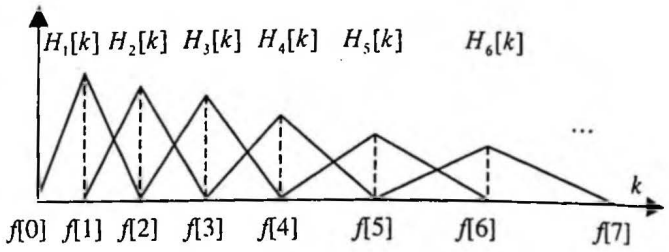


Figure 6.28 Triangular filters used in the computation of the mel-cepstrum using Eq. (6.140).

The mel-frequency cepstrum is then the discrete cosine transform of the M filter outputs:

$$c[n] = \sum_{m=0}^{M-1} S[m] \cos\left(\pi n (m - 1/2)/M\right) \qquad 0 \le n < M \tag{6.145}$$

where M varies for different implementations from 24 to 40. For speech recognition, typically only the first 13 cepstrum coefficients are used. It is important to note that the MFCC representation is no longer a homomorphic transformation. It would be if the order of summation and logarithms in Eq. (6.144) were reversed:

$$S[m] = \sum_{k=0}^{N-1} \ln\left(\left|X_a[k]\right|^2 H_m[k]\right) \qquad 0 < m \le M \tag{6.146}$$

In practice, however, the MFCC representation is approximately homomorphic for filters that have a smooth transfer function. The advantage of the MFCC representation using (6.144) instead of (6.146) is that the filter energies are more robust to noise and spectral estimation errors. This algorithm has been used extensively as a feature vector for speech recognition systems.

While the definition of cepstrum in Section 6.4.1 uses an inverse DFT, since $S[m]$ is even, a DCT-II can be used instead (see Chapter 5).

6.5.3. Perceptual Linear Prediction (PLP)

Perceptual Linear Prediction (PLP) [16] uses the standard Durbin recursion of Section 6.3.2.1.2 to compute LPC coefficients, and typically the LPC coefficients are transformed to LPC-cepstrum using the recursion in Section 6.4.2.1. But unlike standard linear prediction, the autocorrelation coefficients are not computed in the time domain through Eq. (6.55).

The autocorrelation $R_x[n]$ is the inverse Fourier transform of the power spectrum $|X(\omega)|^2$ of the signal. We cannot compute the continuous-frequency Fourier transform eas-

Amazon/VB Assets
Exhibit 1011
Page 344

ily, but we can take an FFT to compute $X[k]$, so that the autocorrelation can be obtained as the inverse Fourier transform of $|X[k]|^2$. Since the discrete Fourier transform is not performing linear convolution but circular convolution, we need to make sure that the FFT size is larger than twice the window length (see Section 5.3.4) for this to hold. This alternate way of computing autocorrelation coefficients, entailing two FFTs and N multiplies and adds, should yield identical results. Since normally only a small number p of autocorrelation coefficients are needed, this is generally not a cost-effective way to do it, unless the first FFT has to be computed for other reasons.

Perceptual linear prediction uses the above method, but replaces $|X[k]|^2$ by a perceptually motivated power spectrum. The most important aspect is the non-linear frequency scaling, which can be achieved through a set of filterbanks similar to those described in Section 6.5.2, so that this critical-band power spectrum can be sampled in approximately 1-Bark intervals. Another difference is that, instead of taking the logarithm on the filterbank energy outputs, a different non-linearity compression is used, often the cubic root. It is reported [16] that the use of this different non-linearity is beneficial for speech recognizers in noisy conditions.

6.6. FORMANT FREQUENCIES

Formant frequencies are the resonances in the vocal tract and, as we saw in Chapter 2, they convey the differences between different sounds. Expert spectrogram readers are able to recognize speech by looking at a spectrogram, particularly at the formants. It has been argued that they are very useful features for speech recognition, but they haven't been widely used because of the difficulty in estimating them.

One way of obtaining formant candidates at a frame level is to compute the roots of a p^{th}-order LPC polynomial [3, 26]. There are standard algorithms to compute the complex roots of a polynomial with real coefficients [36], though convergence is not guaranteed. Each complex root z_i can be represented as

$$z_i = \exp(-\pi b_i + j2\pi f_i) \tag{6.147}$$

where f_i and b_i are the formant frequency and bandwidth, respectively, of the i^{th} root. Real roots are discarded and complex roots are sorted by increasing f, discarding negative values. The remaining pairs (f_i, b_i) are the formant candidates. Traditional formant trackers discard roots whose bandwidths are higher than a threshold [46], say 200 Hz.

Closed-phase analysis of voiced speech [5] uses only the regions for which the glottis is closed and thus there is no excitation. When the glottis is open, there is a coupling of the vocal tract with the lungs and the resonance bandwidths are somewhat larger. Determination of the closed-phase regions directly from the speech signal is difficult, so often an *electroglottograph* (EGG) signal is used [23]. EGG signals, obtained by placing electrodes at the speaker's throat, are very accurate in determining the times when the glottis is closed. Using samples in the closed-phase covariance analysis can yield accurate results [46]. For female

Amazon/VB Assets
Exhibit 1011
Page 345

speech, the closed-phase is short, and sometimes non-existent, so such analysis can be a challenge. EGG signals are useful also for pitch tracking and are described in more detail in Chapter 16.

Another common method consists of finding the peaks on a smoothed spectrum, such as that obtained through an LPC analysis [26, 40]. The advantage of this method is that you can always compute the peaks and it is more computationally efficient than extracting the complex roots of a polynomial. On the other hand, this procedure generally doesn't estimate the formant's bandwidth. The first three formants are typically estimated this way for formant synthesis (see Chapter 16), since they are the ones that allow sound classification, whereas the higher formants are more speaker dependent.

Sometimes, the signal goes through some *conditioning*, which includes sampling rate conversion to remove frequencies outside the range we are interested in. For example, if we are interested only in the first three formants, we can safely downsample the input signal to 8 kHz, since we know all three formants should be below 4 kHz. This downsampling reduces computation and the chances of the algorithm to find formant values outside the expected range (otherwise peaks or roots could be chosen above 4 kHz which we know do not correspond to any of the first three formants). Pre-emphasis filtering is also often used to whiten the signal.

Because of the thresholds imposed above, it is possible that the formants are not continuous. For example, when the vocal tract's spectral envelope is changing rapidly, bandwidths obtained through the above methods are overestimates of the true bandwidths, and they may exceed the threshold and thus be rejected. It is also possible for the peak-picking algorithm to classify a harmonic as a formant during some regions where it is much stronger than the other harmonics. Due to the thresholds used, a given frame could have no formants, only one formant (either first, second, or third), two, three, or more. Formant alignment from one frame to another has often been done using heuristics to prevent such discontinuities.

6.6.1. Statistical Formant Tracking

It is desirable to have an approach that does not use any thresholds on formant candidates and uses a probabilistic model to do the tracking instead of heuristics [1]. The formant candidates can be obtained from roots of the LPC polynomial, peaks in the smoothed spectrum, or even from a dense sample of possible points. If the first n formants are desired, and we have $(p/2)$ formant candidates, a maximum of r n-tuples are considered, where r is given by

$$r = \binom{p/2}{n} \tag{6.148}$$

A Viterbi search (see Chapter 8) is then carried out to find the most likely path of formant n-tuples given a model with some a priori knowledge of formants. The prior distribution for formant targets is used to determine which formant candidate to use of all possible choices for the given phoneme (i.e., we know that F1 for an AE should be around 800 Hz).

Amazon/VB Assets
Exhibit 1011
Page 346

Formant continuity is imposed through the prior distribution of the formant slopes. This algorithm produces n formants for every frame, including silence.

Since we are interested in obtaining the first three formants ($n = 3$) and F3 is known to be lower than 4 kHz, it is advantageous to downsample the signal to 8 kHz in order to avoid obtaining formant candidates above 4 kHz and to let us use a lower-order analysis which offers fewer numerical problems when computing the roots. With $p = 14$, it results in a maximum of $r = 35$ triplets for the case of no real roots.

Let $\mathbf{X}$ be a sequence of T feature vectors $\mathbf{x}_t$ of dimension n:

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)' \tag{6.149}$$

where the prime denotes transpose.

We estimate the formants with the knowledge of what sound occurs at that particular time, for example by using a speech recognizer that segments the waveform into different phonemes (see Chapter 9) or states q_t within a phoneme. In this case we assume that the output distribution of each state i is modeled by one Gaussian density function with a mean μ_i and covariance matrix Σ_i. We can define up to N states, with λ being the set of all means and covariance matrices for all:

$$\lambda = (\mu_1, \Sigma_1, \mu_2, \Sigma_2, \cdots, \mu_N, \Sigma_N) \tag{6.150}$$

Therefore, the log-likelihood for $\mathbf{X}$ is given by

$$\ln p(\mathbf{X} \mid \hat{\mathbf{q}}, \lambda) = -\frac{TM}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=1}^{T}\ln|\Sigma_{q_t}| - \frac{1}{2}\sum_{t=1}^{T}(\mathbf{x}_t - \mu_{q_t})'\Sigma_{q_t}^{-1}(\mathbf{x}_t - \mu_{q_t}) \tag{6.151}$$

Maximizing $\mathbf{X}$ in Eq. (6.151) leads to the trivial solution $\hat{\mathbf{X}} = (\mu_{q_1}, \mu_{q_2}, \ldots, \mu_{q_T})'$, a piecewise function whose value is that of the best n-tuple candidate. This function has discontinuities at state boundaries and thus is not likely to represent well the physical phenomena of speech.

This problem arises because the slopes at state boundaries do not match the slopes of natural speech. To avoid these discontinuities, we would like to match not only the target formants at each state, but also the formant slopes at each state. To do that, we augment the feature vector $\mathbf{x}_t$ at frame t with the delta vector $\mathbf{x}_t - \mathbf{x}_{t-1}$. Thus, we increase the parameter space of λ with the corresponding means δ_t and covariance matrices Γ_t of these delta parameters, and assume statistical independence among them. The corresponding new log-likelihood has the form

$$\begin{aligned}\ln p(\mathbf{X} \mid \hat{\mathbf{q}}, \lambda) = K &- \frac{1}{2}\sum_{t=1}^{T}\ln|\Sigma_{q_t}| - \frac{1}{2}\sum_{t=2}^{T}\ln|\Gamma_{q_t}| \\ &- \frac{1}{2}\sum_{t=1}^{T}(\mathbf{x}_t - \mu_{q_t})'\Sigma_{q_t}^{-1}(\mathbf{x}_t - \mu_{q_t}) - \frac{1}{2}\sum_{t=2}^{T}(\mathbf{x}_t - \mathbf{x}_{t-1} - \delta_{q_t})'\Gamma_{q_t}^{-1}(\mathbf{x}_t - \mathbf{x}_{t-1} - \delta_{q_t})\end{aligned} \tag{6.152}$$

Amazon/VB Assets
Exhibit 1011
Page 347

Maximization of Eq. (6.152) with respect to $\mathbf{x}_t$ requires solving several sets of linear equations. If Γ_t and Σ_t are diagonal covariance matrices, it results in a set of linear equations for each of the M dimensions

$$\mathbf{BX} = \mathbf{c} \tag{6.153}$$

where $\mathbf{B}$ is a tridiagonal matrix (all values are zero except for those in the main diagonal and its two adjacent diagonals), which leads to a very efficient solution [36]. For example, the values of $\mathbf{B}$ and $\mathbf{c}$ for $T = 3$ are given by

$$\mathbf{B} = \begin{pmatrix} \frac{1}{\sigma_{q_1}^2} + \frac{1}{\gamma_{q_2}^2} & -\frac{1}{\gamma_{q_2}^2} & 0 \\ -\frac{1}{\gamma_{q_2}^2} & \frac{1}{\sigma_{q_2}^2} + \frac{1}{\gamma_{q_2}^2} + \frac{1}{\gamma_{q_3}^2} & -\frac{1}{\gamma_{q_3}^2} \\ 0 & -\frac{1}{\gamma_{q_3}^2} & \frac{1}{\sigma_{q_3}^2} + \frac{1}{\gamma_{q_3}^2} \end{pmatrix} \tag{6.154}$$

$$\mathbf{c} = \begin{pmatrix} \frac{\mu_{q_1}}{\sigma_{q_1}^2} - \frac{\delta_{q_2}}{\gamma_{q_2}^2} & \frac{\mu_{q_2}}{\sigma_{q_2}^2} + \frac{\delta_{q_2}}{\gamma_{q_2}^2} - \frac{\delta_{q_3}}{\gamma_{q_3}^2} & \frac{\mu_{q_3}}{\sigma_{q_3}^2} + \frac{\delta_{q_3}}{\gamma_{q_3}^2} \end{pmatrix}' \tag{6.155}$$

where just one dimension is represented, and the process is repeated for all dimensions with a computational complexity of $O(TM)$.

The maximum likelihood sequence $\hat{\mathbf{x}}_t$ is close to the targets μ_i while keeping the slopes close to δ_i for a given state i, thus estimating a continuous function. Because of the delta coefficients, the solution depends on all the parameters of all states and not just the current state. This procedure can be performed for the formants as well as the bandwidths.

The parameters μ_i, Σ_i, δ_i, and Γ_i can be re-estimated using the EM algorithm described in Chapter 8. In [1] it is reported that two or three iterations are sufficient for speaker-dependent data.

The formant track obtained through this method can be rough, and it may be desired to smooth it. Smoothing without knowledge about the speech signal would result in either blurring the sharp transitions that occur in natural speech, or maintaining ragged formant tracks where the underlying physical phenomena vary slowly with time. Ideally we would like a larger adjustment to the raw formant when the error in the estimate is large relative to the variance of the corresponding state within a phoneme. This can be done by modeling the formant measurement error as a Gaussian distribution. Figure 6.29 shows an utterance from a male speaker with the smoothed formant tracks, and Figure 6.30 compares the raw and smoothed formants. When no real formant is visible from the spectrogram, the algorithm tends to assign a large bandwidth (not shown in the figure).

Amazon/VB Assets
Exhibit 1011
Page 348

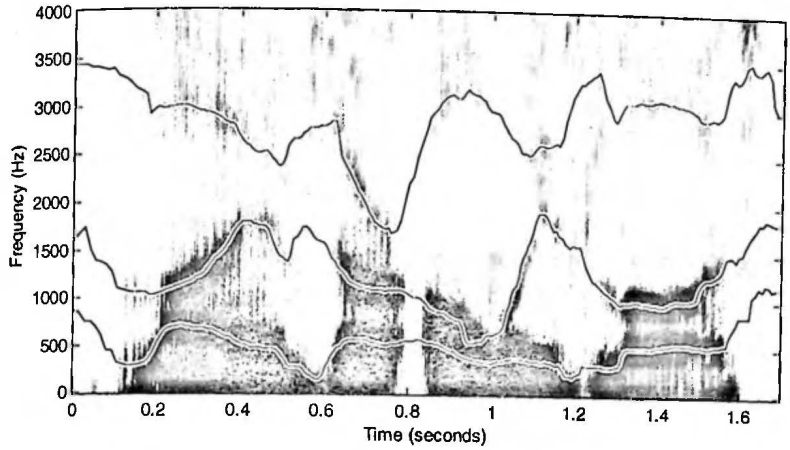


Figure 6.29 Spectrogram and three smoothed formants.

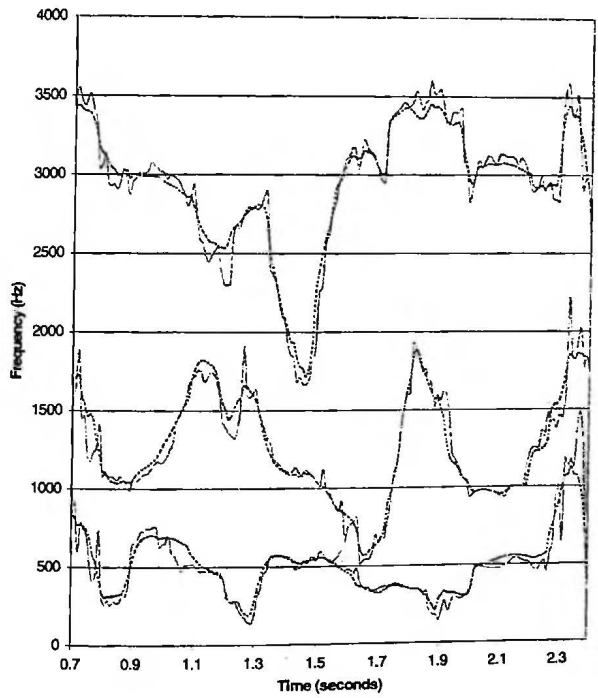


Figure 6.30 Raw formants (ragged gray line) and smoothed formants (dashed line).

Amazon/VB Assets
Exhibit 1011
Page 349

6.7. THE ROLE OF PITCH

Pitch determination is very important for many speech processing algorithms. The concatenative speech synthesis methods of Chapter 16 require pitch tracking on the desired speech segments if prosody modification is to be done. Chinese speech recognition systems use pitch tracking for tone recognition, which is important in disambiguating the myriad of homophones. Pitch is also crucial for prosodic variation in text-to-speech systems (see Chapter 15) and spoken language systems (see Chapter 17). While in the previous sections we have dealt with features representing the filter, pitch represents the source of the model illustrated in Figure 6.1.

Pitch determination algorithms also use short-term analysis techniques, which means that for every frame $\mathbf{x}_m$ we get a score $f(T \mid \mathbf{x}_m)$ that is a function of the candidate pitch periods T. These algorithms determine the optimal pitch by maximizing

$$T_m = \arg\max_T f(T \mid \mathbf{x}_m) \tag{6.156}$$

We describe several such functions computed through the autocorrelation method and the normalized cross-correlation method, as well as the signal conditioning that is often performed. Other approaches based on cepstrum [28] have also been used successfully. A good summary of techniques used for pitch tracking is provided by [17, 45].

Pitch determination using Eq. (6.156) is error prone, and a smoothing stage is often done. This smoothing, described in Section 6.7.4, takes into consideration that the pitch does not change quickly over time.

6.7.1. Autocorrelation Method

A commonly used method to estimate pitch is based on detecting the highest value of the autocorrelation function in the region of interest. This region must exclude $m = 0$, as that is the absolute maximum of the autocorrelation function [37]. As discussed in Chapter 5, the statistical autocorrelation of a sinusoidal random process

$$\mathbf{x}[n] = \cos(\omega_0 n + \varphi) \tag{6.157}$$

is given by

$$R[m] = E\{\mathbf{x}^*[n]\mathbf{x}[n+m]\} = \frac{1}{2}\cos(\omega_0 m) \tag{6.158}$$

which has maxima for $m = lT_0$, the pitch period and its harmonics, so that we can find the pitch period by computing the highest value of the autocorrelation. Similarly, it can be shown that any WSS periodic process $\mathbf{x}[n]$ with period T_0 also has an autocorrelation $R[m]$ which exhibits its maxima at $m = lT_0$.

Amazon/VB Assets
Exhibit 1011
Page 350

In practice, we need to obtain an estimate $\hat{R}[m]$ from knowledge of only N samples. If we use a window $w[n]$ of length N on $\mathbf{x}[n]$ and assume it to be real, the empirical autocorrelation function is given by

$$\hat{R}[m] = \frac{1}{N} \sum_{n=0}^{N-1-|m|} w[n]\mathbf{x}[n]w[n+|m|]\mathbf{x}[n+|m|] \tag{6.159}$$

whose expected value can be shown to be

$$E\left\{\hat{R}[m]\right\} = R[m]\left(w[m] * w[-m]\right) \tag{6.160}$$

where

$$w[m] * w[-m] = \sum_{n=0}^{N-|m|-1} w[n]w[n+|m|] \tag{6.161}$$

which, for the case of a rectangular window of length N, is given by

$$w[m] * w[-m] = \begin{cases} 1 - \dfrac{|m|}{N} & |m| < N \\ 0 & |m| \geq N \end{cases} \tag{6.162}$$

which means that $\hat{R}[m]$ is a biased estimator of $R[m]$. So, if we compute the peaks based on Eq. (6.159), the estimate of the pitch will also be biased. Although the variance of the estimate is difficult to compute, it is easy to see that as m approaches N, fewer and fewer samples of $x[n]$ are involved in the calculation, and thus the variance of the estimate is expected to increase. If we multiply Eq. (6.159) by $N/(N-m)$, the estimate will be unbiased but the variance will be larger.

Using the empirical autocorrelation in Eq. (6.159) for the random process in Eq. (6.157) results in an expected value of

$$E\left\{\hat{R}[m]\right\} = \left(1 - \frac{|m|}{N}\right)\frac{\cos(\omega_0 m)}{2}, \quad |m| < N \tag{6.163}$$

whose maximum coincides with the pitch period for $m > m_0$.

Since pitch periods can be as low as 40 Hz (for a very low-pitched male voice) or as high as 600 Hz (for a very high-pitched female or child's voice), the search for the maximum is conducted within a region. This F0 detection algorithm is illustrated in Figure 6.31 where the lag with highest autocorrelation is plotted for every frame. In order to see periodicity present in the autocorrelation, we need to use a window that contains at least two pitch periods, which, if we want to detect a 40 Hz pitch, implies 50 ms (see Figure 6.32). For window lengths so long, the assumption of stationarity starts to fail, because a pitch period at the beginning of the window can be significantly different than at the end of the window.

Amazon/VB Assets
Exhibit 1011
Page 351

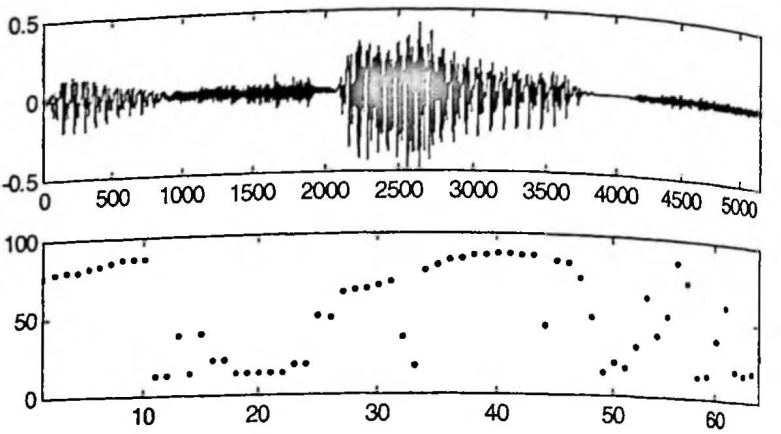


Figure 6.31 Waveform and unsmoothed pitch track with the autocorrelation method. A frame shift of 10 ms, a Hamming window of 30 ms, and a sampling rate of 8 kHz were used. Notice that two frames in the voiced region have an incorrect pitch. The pitch values in the unvoiced regions are essentially random.

One possible solution to this problem is to estimate the autocorrelation function with different window lengths for different lags m.

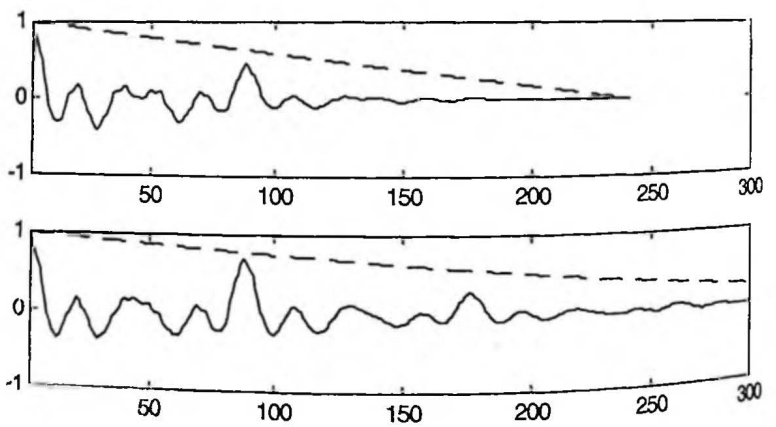


Figure 6.32 Autocorrelation function for frame 40 in Figure 6.31. The maximum occurs at 89 samples. A sampling frequency of 8 kHz and window shift of 10 ms are used. The top figure is using a window length of 30 ms, whereas the bottom one is using 50 ms. Notice the quasi-periodicity in the autocorrelation function.

Amazon/VB Assets
Exhibit 1011
Page 352

The candidate pitch periods in Eq. (6.156) can be simply $T_m = m$; i.e., the pitch period is any integer number of samples. For low values of T_m, the frequency resolution is lower than for high values. To maintain a relatively constant frequency resolution, we do not have to search all the pitch periods for large T_m. Alternatively, if the sampling frequency is not high, we may need to use fractional pitch periods (often done in the speech coding algorithms of Chapter 7).

The autocorrelation function can be efficiently computed by taking a signal, windowing it, and taking an FFT and then the square of the magnitude.

6.7.2. Normalized Cross-Correlation Method

A method that is free from these border problems and has been gaining in popularity is based on the *normalized cross-correlation* [2]

$$\alpha_t(T) = \cos(\theta) = \frac{< \mathbf{x}_t, \mathbf{x}_{t-T} >}{|\mathbf{x}_t||\mathbf{x}_{t-T}|} \tag{6.164}$$

where $\mathbf{x}_t = \{x[t-N/2], x[t-N/2+1], \cdots, x[t+N/2-1]\}$ is a vector of N samples centered at time t, and $< \mathbf{x}_t, \mathbf{x}_{t-T} >$ is the inner product between the two vectors defined as

$$< \mathbf{x}_n, \mathbf{y}_l > \sum_{m=-N/2}^{N/2-1} x[n+m]y[l+m] \tag{6.165}$$

so that, using Eq. (6.165), the normalized cross-correlation can be expressed as

$$\alpha_t(T) = \frac{\sum_{n=-N/2}^{N/2-1} x[t+n]x[t+n-T]}{\sqrt{\sum_{n=-N/2}^{N/2-1} x^2[t+n] \sum_{m=-N/2}^{N/2-1} x^2[t+m+T]}} \tag{6.166}$$

where we see that the numerator in Eq. (6.166) is very similar to the autocorrelation in Section 6.7.1, but where N terms are used in the summation for all values of T.

The maximum of the normalized cross-correlation method is shown in Figure 6.33 (b). Unlike the autocorrelation method, the estimate of the normalized cross-correlation is not biased by the term $(1-m/N)$. For perfectly periodic signals, this results in identical values of the normalized cross-correlation function for kT. This can result in pitch halving, where $2T$ can be chosen as the pitch period, which happens in Figure 6.33 (b) at the beginning of the utterance. Using a decaying bias $(1-m/M)$ with $M \gg N$, can be useful in reducing pitch halving, as we see in Figure 6.33 (c).

Amazon/VB Assets
Exhibit 1011
Page 353

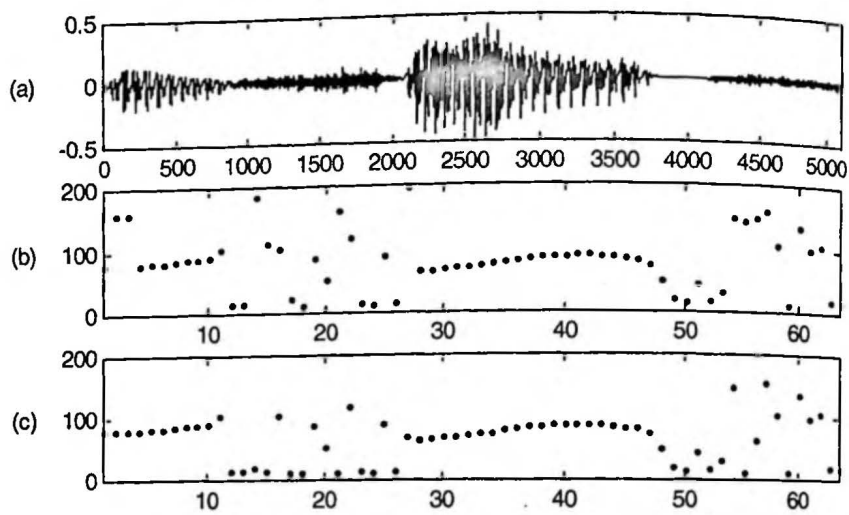


Figure 6.33 (a) Waveform and (b, c) unsmoothed pitch tracks with the normalized cross-correlation method. A frame shift of 10 ms, window length of 10 ms, and sampling rate of 8 kHz were used. (b) is the standard normalized cross-correlation method, whereas (c) has a decaying term. If we compare it to the autocorrelation method of Figure 6.31, the middle voiced region is correctly identified in both (b) and (c), but two frames at the beginning of (b) that have pitch halving are eliminated with the decaying term. Again, the pitch values in the unvoiced regions are essentially random.

Because the number of samples involved in the calculation is constant, this estimate is unbiased and has lower variance than that of the autocorrelation. Unlike the autocorrelation method, the window length could be lower than the pitch period, so that the assumption of stationarity is more accurate and it has more time resolution. While pitch trackers based on the normalized cross-correlation typically perform better than those based on the autocorrelation, they also require more computation, since all the autocorrelation lags can be efficiently computed through 2 FFTs and N multiplies and adds (see Section 5.3.4).

Let's gain some insight about the normalized cross-correlation. If $x[n]$ is periodic with period T, then we can predict it from a vector T samples in the past as:

$$\mathbf{x}_t = \rho \mathbf{x}_{t-T} + \mathbf{e}_t \tag{6.167}$$

where ρ is the prediction gain. The normalized cross-correlation measures the angle between the two vectors, as can be seen in Figure 6.34, and since it is a cosine, it has the property that $-1 \le \alpha_n(P) \le 1$.

Amazon/VB Assets
Exhibit 1011
Page 354

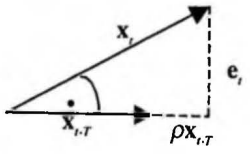


Figure 6.34 The prediction of $\mathbf{x}_t$ with $\mathbf{x}_{t-T}$ results in an error $\mathbf{e}_t$.

If we choose the value of the prediction gain ρ so as to minimize the prediction error

$$|\mathbf{e}_t|^2 = |\mathbf{x}_t|^2 - |\mathbf{x}_t|^2 \cos^2(\theta) = |\mathbf{x}_t|^2 - |\mathbf{x}_t|^2 \alpha_t^2(T) \tag{6.168}$$

and assume $\mathbf{e}_t$ is a zero-mean Gaussian random vector with a standard deviation $\sigma|\mathbf{x}_t|$, then

$$\ln f(\mathbf{x}_t \mid T) = K + \frac{\alpha_t^2(T)}{2\sigma^2} \tag{6.169}$$

so that the maximum likelihood estimate corresponds to finding the value T with highest normalized cross-correlation. Using Eq. (6.166), it is possible that $\alpha_t(T) < 0$. In this case, there is negative correlation between $\mathbf{x}_t$ and $\mathbf{x}_{t-T}$, and it is unlikely that T is a good choice for pitch. Thus, we need to force $\rho > 0$, so that Eq. (6.169) is converted into

$$\ln f(\mathbf{x}_t \mid T) = K + \frac{\left(\max(0, \alpha_t(T))\right)^2}{2\sigma^2} \tag{6.170}$$

The normalized cross-correlation of Eq. (6.164) predicts the current frame with a frame that occurs T samples before. Voiced speech may exhibit low correlation with a previous frame at a spectral discontinuity, such as those appearing at stops. To account for this, an enhancement can be done to consider not only the *backward* normalized cross-correlation, but also the *forward* normalized cross-correlation, by looking at a frame that occurs T samples ahead of the current frame, and taking the highest of both.

$$\ln f(\mathbf{x}_t \mid T) = K + \frac{\left(\max(0, \alpha_t(T), \alpha_t(-T))\right)^2}{2\sigma^2} \tag{6.171}$$

6.7.3. Signal Conditioning

Noise in the signal tends to make pitch estimation less accurate. To reduce this effect, signal conditioning or pre-processing has been proposed prior to pitch estimation [44]. Typically

Amazon/VB Assets
Exhibit 1011
Page 355

this involves bandpass filtering to remove frequencies above 1 or 2 kHz, and below 100 Hz or so. High frequencies do not have much voicing information and have significant noise energy, whereas low frequencies can have 50/60 Hz interference from power lines or non-linearities from some A/D subsystems that can also mislead a pitch estimation algorithm.

In addition to the noise in the very low frequencies and aspiration at high bands, the stationarity assumption is not as valid at high frequencies. Even a slowly changing pitch, say, nominal 100 Hz increasing 5 Hz in 10 ms, results in a fast-changing harmonic: the 30th harmonic at 3000 Hz changes 150 Hz in 10 ms. The corresponding short-time spectrum no longer shows peaks at those frequencies.

Because of this, it is advantageous to filter out such frequencies prior to the computation of the autocorrelation or normalized cross-correlation. If an FFT is used to compute the autocorrelation, this filter is easily done by setting to 0 the undesired frequency bins.

6.7.4. Pitch Tracking

Pitch tracking using the above methods typically fails in several cases:

- *Sub-harmonic errors*. If a signal is periodic with period T, it is also periodic with period $2T$, $3T$, etc. Thus, we expect the scores also to be high for the multiples of T, which can mislead the algorithm. Because the signal is never perfectly stationary, those multiples, or sub-harmonics, tend to have slightly lower scores than the fundamental. If the pitch is identified as $2T$, pitch halving is said to occur.
- *Harmonic errors*. If harmonic M dominates the signal's total energy, the score at pitch period T/M will be large. This can happen if the harmonic falls in a formant frequency that boosts its amplitude considerably compared to that of the other harmonics. If the pitch is identified as $T/2$, pitch doubling is said to occur.
- *Noisy conditions*. When the SNR is low, pitch estimates are quite unreliable for most methods.
- *Vocal fry*. While pitch is generally continuous, for some speakers it can suddenly change and even halve, particularly at the end of an unstressed voiced region. The pitch here is really not well defined and imposing smoothness constraints can hurt the system.
- *F0 jumps* up or down by an octave occasionally.
- *Breathy-voiced speech* is difficult to distinguish from periodic background noise.
- *Narrow-band filtering* of unvoiced excitations by certain vocal tract configurations can lead to signals that appear periodic.

Amazon/VB Assets
Exhibit 1011
Page 356

For these reasons, pitch trackers do not determine the pitch value at frame m based exclusively on the signal at that frame. For a frame where there are several pitch candidates with similar scores, the fact that pitch does not change abruptly with time is beneficial in disambiguation, because the following frame possibly has a clearer pitch candidate, which can help.

To integrate the normalized cross-correlation into a probabilistic framework, you can combine tracking with the use of a priori information [10]. Let's define $\mathbf{X}=\{\mathbf{x}_0,\mathbf{x}_1,\ldots,\mathbf{x}_{M-1}\}$ as a sequence of input vectors for M consecutive frames centered at equally spaced time instants, say 10 ms. Furthermore, if we assume that the $\mathbf{x}_i$ are independent of each other, the joint distribution takes on the form:

$$f(\mathbf{X}\,|\,\mathbf{T})=\prod_{i=0}^{M-1} f(\mathbf{x}_i\,|\,T_i) \tag{6.172}$$

where $\mathbf{T}=\{T_0,T_1,\ldots,T_{M-1}\}$ is the pitch track for the input. The *maximum a posteriori* (MAP) estimate of the pitch track is:

$$\mathbf{T}_{MAP}=\max_{\mathbf{T}} f(\mathbf{T}\,|\,\mathbf{X})=\max_{\mathbf{T}}\frac{f(\mathbf{T})f(\mathbf{X}\,|\,\mathbf{T})}{f(\mathbf{X})}=\max_{\mathbf{T}} f(\mathbf{T})f(\mathbf{X}\,|\,\mathbf{T}) \tag{6.173}$$

according to Bayes' rule, with the term $f(\mathbf{X}\,|\,\mathbf{T})$ being given by Eq. (6.172) and $f(\mathbf{x}_i\,|\,T_i)$ by Eq. (6.169), for example.

The function $f(\mathbf{T})$ constitutes the a priori statistics for the pitch and can help disambiguate the pitch, by avoiding pitch doubling or halving given knowledge of the speaker's average pitch, and by avoiding rapid transitions given a model of how pitch changes over time. One possible approximation is given by assuming that the a priori probability of the pitch period at frame i depends only on the pitch period for the previous frame:

$$f(\mathbf{T})=f(T_0,T_1,\ldots,T_{M-1})=f(T_{M-1}\,|\,T_{M-2})f(T_{M-2}\,|\,T_{M-3})\cdots f(T_1\,|\,T_0)f(T_0) \tag{6.174}$$

One possible choice for $f(T_t\,|\,T_{t-1})$ is to decompose it into a component that depends on T_t and another that depends on the difference (T_t-T_{t-1}). If we approximate both as Gaussian densities, we obtain

$$\ln f(T_t\,|\,T_{t-1})=K'-\frac{(T_t-\mu)^2}{2\beta^2}-\frac{(T_t-T_{t-1}-\delta)^2}{2\gamma^2} \tag{6.175}$$

so that when Eqs. (6.170) and (6.175) are combined, the log-probability of transitioning to T_i at time t from pitch T_j at time $t-1$ is given by

Amazon/VB Assets
Exhibit 1011
Page 357

$$S_t(T_i,T_j)=\frac{\left(\max(0,\alpha_t(T_i))\right)^2}{2\sigma^2}-\frac{(T_i-\mu)^2}{2\beta^2}-\frac{\left(T_i-T_j-\delta\right)^2}{2\gamma^2} \tag{6.176}$$

so that the log-likelihood in Eq. (6.173) can be expressed as

$$\ln f(\mathbf{T})f(\mathbf{X}\,|\,\mathbf{T})=\left(\max(0.\alpha_0(T_0))\right)^2+\max_{i_t}\sum_{t=1}^{M-1}S_t(T_i,T_{i_{t-1}}) \tag{6.177}$$

which can be maximized through dynamic programming. For a region where pitch is not supposed to change, $\delta=0$, the term $(T_i-T_j)^2$ in Eq. (6.176) acts as a penalty that keeps the pitch track from jumping around. A mixture of Gaussians can be used instead to model different rates of pitch change, as in the case of Mandarin Chinese with four tones characterized by different slopes. The term $(T_i-\mu)^2$ attempts to get the pitch close to its expected value to avoid pitch doubling or halving, with the average μ being different for male and female speakers. Pruning can be done during the search without loss of accuracy (see Chapter 12).

Pitch trackers also have to determine whether a region of speech is voiced or unvoiced. A good approach is to build a statistical classifier with techniques described in Chapter 8 based on energy and the normalized cross-correlation described above. Such classifiers, i.e., an HMM, penalize jumps between voiced and unvoiced frames to avoid voiced regions having isolated unvoiced frames inside and vice versa. A threshold can be used on the a posteriori probability to distinguish voiced from unvoiced frames.

6.8. HISTORICAL PERSPECTIVE AND FURTHER READING

In 1978, Lawrence R. Rabiner and Ronald W. Schafer [38] wrote a book summarizing the work to date on digital processing of speech, which remains a good source for the reader interested in further reading in the field. The book by Deller, Hansen, and Proakis [9] includes more recent work and is also an excellent reference. O'Shaughnessy [33] also has a thorough description of the subject. Malvar [25] covers filterbanks and lapped transforms extensively.

The extensive wartime interest in sound spectrography led Koenig and his colleagues at Bell Laboratories [22] in 1946 to the invaluable development of a tool that has been used for speech analysis since then: the spectrogram. Potter et al. [35] showed the usefulness of the analog spectrogram in analyzing speech. The spectrogram facilitated research in the field and led Peterson and Barney [34] to publish in 1952 a detailed study of formant values of different vowels. The development of computers and the FFT led Oppenheim, in 1970 [30], to develop digital spectrograms, which imitated the analog counterparts.

The MIT Acoustics Lab started work in speech in 1948 with Leo R. Beranek, who in 1954 published the seminal book *Acoustics*, where he studied sound propagation in tubes. In

Amazon/VB Assets
Exhibit 1011
Page 358

1950, Kenneth N. Stevens joined the lab and started work on speech perception. Gunnar Fant visited the lab at that time and as a result started a strong speech production effort at KTH in Sweden.

The 1960s marked the birth of digital speech processing. Two books, Gunnar Fant's *Acoustical Theory of Speech Production* [13] in 1960 and James Flanagan's *Speech Analysis: Synthesis and Perception* [14] in 1965, had a great impact and sparked interest in the field. The advent of the digital computer prompted Kelly and Gertsman to create in 1961 the first digital speech synthesizer [21]. Short-time Fourier analysis, cepstrum, LPC analysis, and pitch and formant tracking were the fruit of that decade.

Short-time frequency analysis was first proposed for analog signals by Fano [11] in 1950 and later by Schroeder and Atal [42].

The mathematical foundation behind linear predictive coding dates to the auto-regressive models of George Udny Yule (1927) and Gilbert Walker (1931), which led to the well-known Yule-Walker equations. These equations resulted in a Toeplitz matrix, named after Otto Toeplitz (1881-1940) who studied it extensively. N. Levinson suggested in 1947 an efficient algorithm to invert such a matrix, which J. Durbin refined in 1960 and is now known as the Levinson-Durbin recursion. The well-known LPC analysis consisted of the application of the above results to speech signals, as developed by Bishnu Atal [4], J. Burg [7], Fumitada Itakura and S. Saito [19] in 1968, and Markel [27] and John Makhoul [24] in 1973.

The cepstrum was first proposed in 1964 by Bogert, Healy, and John Tukey [6] and further studied by Alan V. Oppenheim [29] in 1965. The popular mel-frequency cepstrum was proposed by Davis and Mermelstein [8] in 1980, combining the advantages of cepstrum with knowledge of the non-linear perception of frequency by the human auditory system that had been studied by E. Zwicker [47] in 1961.

Formant tracking was first investigated by Ken Stevens and James Flanagan in the late 1950s, with the foundations for most modern techniques being developed by Schafer and Rabiner [40], Itakura [20], and Markel [26]. Pitch tracking through digital processing was first studied by B. Gold [15] in 1962 and then improved by A. M. Noll [28], M. Schroeder [41], and M. Sondhi [44] in the late 1960s.

REFERENCES

[1] Acero, A., "Formant Analysis and Synthesis Using Hidden Markov Models," *Eurospeech*, 1999, Budapest pp. 1047-1050.

[2] Atal, B.S., *Automatic Speaker Recognition Based on Pitch Contours*, PhD Thesis, 1968, Polytechnic Institute of Brooklyn.

[3] Atal, B.S. and L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," *Journal of the Acoustical Society of America*, 1971, **50**, pp. 637-655.

[4] Atal, B.S. and M.R. Schroeder, "Predictive Coding of Speech Signals," *Report of the 6th Int. Congress on Acoustics*, 1968, Tokyo, Japan.

Amazon/VB Assets
Exhibit 1011
Page 359

[5] Berouti, M.G., D.G. Childers, and A. Paige, "Glottal Area versus Glottal Volume Velocity," *Int. Conf. on Acoustics, Speech and Signal Processing*, 1977, Hartford, Conn, pp. 33-36.

[6] Bogert, B., M. Healy, and J. Tukey, "The Quefrency Alanysis of Time Series for Echoes," *Proc. Symp. on Time Series Analysis*, 1963, New York, J. Wiley, pp. 209-243.

[7] Burg, J., "Maximum Entropy Spectral Analysis," *Proc. of the 37th Meeting of the Society of Exploration Geophysicists*, 1967.

[8] Davis, S. and P. Mermelstein, "Comparison of Parametric Representations for Monosyllable Word Recognition in Continuously Spoken Sentences," *IEEE Trans. on Acoustics, Speech and Signal Processing*, 1980, **28**(4), pp. 357-366.

[9] Deller, J.R., J.H.L. Hansen, and J.G. Proakis, *Discrete-Time Processing of Speech Signals*, 2000, IEEE Press.

[10] Droppo, J. and A. Acero, "Maximum a Posteriori Pitch Tracking," *Int. Conf. on Spoken Language Processing*, 1998, Sydney, Australia, pp. 943-946.

[11] Fano, R.M., "Short-time Autocorrelation Functions and Power Spectra," *Journal of the Acoustical Society of America*, 1950, **22**(Sep), pp. 546-550.

[12] Fant, G., "On the Predictability of Formant Levels and Spectrum Envelopes from Formant Frequencies" in *For Roman Jakobson*, M. Halle, ed., 1956, The Hague, NL, Mouton & Co., pp. 109-120.

[13] Fant, G., *Acoustic Theory of Speech Production*, 1970, The Hague, NL, Mouton.

[14] Flanagan, J., *Speech Analysis Synthesis and Perception*, 1972, New York, Springer-Verlag.

[15] Gold, B., "Computer Program for Pitch Extraction," *Journal of the Acoustical Society of America*, 1962, **34**(7), pp. 916-921.

[16] Hermansky, H., "Perceptual Linear Predictive (PLP) Analysis of Speech," *Journal of the Acoustical Society of America*, 1990, **87**(4), pp. 1738-1752.

[17] Hess, W., *Pitch Determination of Speech Signals*, 1983, New York, Springer-Verlag.

[18] Itakura, F., "Line Spectrum Representation of Linear Predictive Coefficients," *Journal of the Acoustical Society of America*, 1975, **57**(4), pp. 535.

[19] Itakura, F. and S. Saito, "Analysis Synthesis Telephony Based on the Maximum Likelihood Method," *Proc. 6th Int. Congress on Acoustics*, 1968, Tokyo, Japan.

[20] Itakura, F. and S. Saito, "A Statistical Method for Estimation of Speech Spectral Density and Formant Frequencies," *Elec. and Comm. in Japan*, 1970, **53-A**(1), pp. 36-43.

[21] Kelly, J.L. and L.J. Gerstman, "An Artificial Talker Driven From Phonetic Input," *Journal of Acoustical Society of America*, 1961, **33**, pp. 835.

[22] Koenig, R., H.K. Dunn, and L.Y. Lacy, "The Sound Spectrograph," *Journal of the Acoustical Society of America*, 1946, **18**, pp. 19-49.

[23] Krishnamurthy, A.K. and D.G. Childers, "Two Channel Speech Analysis," *IEEE Trans. on Acoustics, Speech and Signal Processing*, 1986, **34**, pp. 730-743.

Amazon/VB Assets
Exhibit 1011
Page 360

[24] Makhoul, J., "Spectral Analysis of Speech by Linear Prediction," *IEEE Trans. on Acoustics, Speech and Signal Processing*, 1973, **21**(3), pp. 140-148.

[25] Malvar, H., *Signal Processing with Lapped Transforms*, 1992, Artech House.

[26] Markel, J.D., "Digital Inverse Filtering—A New Tool for Formant Trajectory Estimation," *IEEE Trans. on Audio and Electroacoustics*, 1972, **AU-20**(June), pp. 129-137.

[27] Markel, J.D. and A.H. Gray, "On Autocorrelation Equations as Applied to Speech Analysis," *IEEE Trans. on Audio and Electroacoustics*, 1973, **AU-21**(April), pp. 69-79.

[28] Noll, A.M., "Cepstrum Pitch Determination," *Journal of the Acoustical Society of America*, 1967, **41**, pp. 293-309.

[29] Oppenheim, A.V., *Superposition in a Class of Nonlinear Systems*, 1965, Research Lab. of Electronics, MIT, Cambridge, Massachusetts.

[30] Oppenheim, A.V., "Speech Spectrograms Using the Fast Fourier Transform," *IEEE Spectrum*, 1970, **7**(Aug), pp. 57-62.

[31] Oppenheim, A.V. and D.H. Johnson, "Discrete Representation of Signals," *The Proc. of the IEEE*, 1972, **60**(June), pp. 681-691.

[32] Oppenheim, A.V., R.W. Schafer, and T.G. Stockham, "Nonlinear Filtering of Multiplied and Convolved Signals," *Proc. of the IEEE*, 1968, **56**, pp. 1264-1291.

[33] O'Shaughnessy, D., *Speech Communication—Human and Machine*, 1987, Addison-Wesley.

[34] Peterson, G.E. and H.L. Barney, "Control Methods Used in a Study of the Vowels," *Journal of the Acoustical Society of America*, 1952, **24**(2), pp. 175-184.

[35] Potter, R.K., G.A. Kopp, and H.C. Green, *Visible Speech*, 1947, New York, D. Van Nostrand Co. Republished by Dover Publications, Inc., 1966.

[36] Press, W.H., *et al.*, *Numerical Recipes in C*, 1988, New York, NY, Cambridge University Press.

[37] Rabiner, L.R., "On the Use of Autocorrelation Analysis for Pitch Detection," *IEEE Trans. on Acoustics, Speech and Signal Processing*, 1977, **25**, pp. 24-33.

[38] Rabiner, L.R. and R.W. Schafer, *Digital Processing of Speech Signals*, 1978, Englewood Cliffs, NJ, Prentice-Hall.

[39] Rosenberg, A.E., "Effect of Glottal Pulse Shape on the Quality of Natural Vowels," *Journal of the Acoustical Society of America*, 1971, **49**, pp. 583-590.

[40] Schafer, R.W. and L.R. Rabiner, "System for Automatic Formant Analysis of Voiced Speech," *Journal of the Acoustical Society of America*, 1970, **47**, pp. 634-678.

[41] Schroeder, M., "Period Histogram and Product Spectrum: New Methods for Fundamental Frequency Measurement," *Journal of the Acoustical Society of America*, 1968, **43**(4), pp. 829-834.

[42] Schroeder, M.R. and B.S. Atal, "Generalized Short-Time Power Spectra and Autocorrelation," *Journal of the Acoustical Society of America*, 1962, **34**(Nov), pp. 1679-1683.

Amazon/VB Assets
Exhibit 1011
Page 361

[43] Shikano, K., K.-F. Lee, and R. Reddy, "Speaker Adaptation through Vector Quantization," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1986, Tokyo, Japan, pp. 2643-2646.

[44] Sondhi, M.M., "New Methods for Pitch Extraction," *IEEE Trans. on Audio and Electroacoustics*, 1968, **16**(June), pp. 262-268.

[45] Talkin, D., "A Robust Algorithm for Pitch Tracking" in *Speech Coding and Synthesis*, W.B. Kleijn and K.K. Paliwal, eds., 1995, Amsterdam, Elsevier, pp. 485-518, .

[46] Yegnanarayana, B. and R.N.J. Veldhuis, "Extraction of Vocal-Tract System Characteristics from Speech Signals," *IEEE Trans. on Speech and Audio Processing*, 1998, **6**(July), pp. 313-327.

[47] Zwicker, E., "Subdivision of the Audible Frequency Range into Critical Bands," *Journal of the Acoustical Society of America*, 1961, **33**(Feb), p. 248.

Amazon/VB Assets
Exhibit 1011
Page 362

CHAPTER 7

Speech Coding

Transmission of speech using data networks requires the speech signal to be digitally encoded. Voice over IP has become very popular because of the Internet, where bandwidth limitations make it necessary to compress the speech signal. Digital storage of audio signals, which can result in higher quality and smaller size than the analog counterpart, is commonplace in compact discs, digital video discs, and MP3 files. Many spoken language systems also use coded speech for efficient communication. For these reasons we devote a chapter to speech and audio coding techniques.

Rather than exhaustively cover all the existing speech and audio coding algorithms, we uncover their underlying technology and enumerate some of the most popular standards. The coding technology discussed in this chapter has a strong link to both speech recognition and speech synthesis. For example, the speech synthesis algorithms described in Chapter 16 use many techniques described here.

Amazon/VB Assets
Exhibit 1011
Page 363

7.1. SPEECH CODERS ATTRIBUTES

How do we compare different speech or audio coders? We can refer to a number of factors, such as signal bandwidth, bit rate, quality of reconstructed speech, noise robustness, computational complexity, delay, channel-error sensitivity, and standards.

Speech signals can be bandlimited to 10 kHz without significantly affecting the hearer's perception. The telephone network limits the bandwidth of speech signals to between 300 and 3400 Hz, which gives *telephone speech* a lower quality. Telephone speech is typically sampled at 8 kHz. The term *wideband speech* is used for a bandwidth of 50–7000 Hz and a sampling rate of 16 kHz. Finally, *audio coding* is used in dealing with high-fidelity audio signals, in which case the signal is sampled at 44.1 kHz.

Reduction in bit rate is the primary purpose of speech coding. The previous bit stream can be compressed to a lower rate by removing redundancy in the signal, resulting in savings in storage and transmission bandwidth. If only redundancy is removed, the original signal can be recovered exactly (*lossless* compression). In *lossy* compression, the signal cannot be recovered exactly, though hopefully it will sound similar to the original.

Depending on system and design constraints, fixed-rate or variable-rate speech coders can be used. Variable-rate coders are used for non-real time applications, such as voice storage (silence can be coded with fewer bits than fricatives, which in turn use fewer bits than vowels), or for packet voice transmissions, such as CDMA cellular for better channel utilization. Transmission of coded speech through a noisy channel may require devoting more bits to channel coding and fewer to source coding. For most real-time communication systems, a maximum bit rate is specified.

The quality of the reconstructed speech signal is a fundamental attribute of a speech coder. Bit rate and quality are intimately related: the lower the bit rate, the lower the quality. While the bit rate is inherently a number, it is difficult to quantify the quality. The most widely used measure of quality is the *Mean Opinion Score* (MOS) [25], which is the result of averaging opinion scores for a set of between 20 and 60 untrained subjects. Each listener characterizes each set of utterances with a score on a scale from 1 (unacceptable quality) to 5 (excellent quality), as shown in Table 7.1. An MOS of 4.0 or higher defines *good* or *toll* quality, where the reconstructed speech signal is generally indistinguishable from the original signal. An MOS between 3.5 and 4.0 defines *communication* quality, which is sufficient for telephone communications. We show in Section 7.2.1 that if each sample is quantized with 16 bits, the resulting signal has *toll* quality (essentially indistinguishable from the unquantized signal). See Chapter 16 for more details on perceptual quality measurements.

Table 7.1 Mean Opinion Score (MOS) is a numeric value computed as an average for a number of subjects, where each number maps to the above subjective quality.

Excellent	Good	Fair	Poor	Bad
5	4	3	2	1

Amazon/VB Assets
Exhibit 1011
Page 364

Another measure of quality is the *signal-to-noise ratio* (SNR), defined as the ratio between the signal's energy and the noise's energy in terms of dB:

$$SNR = \frac{\sigma_x^2}{\sigma_e^2} = \frac{E\{x^2[n]\}}{E\{e^2[n]\}} \tag{7.1}$$

The MOS rating of a codec on noise-free speech is often higher than its MOS rating for noisy speech. This is generally caused by specific assumptions in the speech coder that tend to be violated when a significant amount of noise is present in the signal. This phenomenon is more accentuated for lower-bit-rate coders that need to make more assumptions.

The computational complexity and memory requirements of a speech coder determine the cost and power consumption of the hardware on which it is implemented. In most cases, real-time operation is required at least for the decoder. Speech coders can be implemented in inexpensive *Digital Signal Processors* (DSP) that form part of many consumer devices, such as answering machines and DVD players, for which storage tends to be relatively more expensive than processing power. DSPs are also used in cellular phones because bit rates are limited.

All speech coders have some delay, which, if excessive, can affect the dynamics of a two-way communication. For instance, delays over 150 ms can be unacceptable for highly interactive conversations. Coder delay is the sum of different types of delay. The first is the *algorithmic delay* arising because speech coders usually operate on a block of samples, called a *frame,* which needs to be accumulated before processing can begin. Often the speech coder requires some additional *look-ahead* beyond the frame to be encoded. The *computational delay* is the time that the speech coder takes to process the frame. For real-time operation, the computational delay has to be smaller than the algorithmic delay. A block of bits is generally assembled by the encoder prior to transmission, possibly to add error-correction properties to the bit stream, which cause *multiplexing delay.* Finally, there is the *transmission delay,* due to the time it takes for the frame to traverse the channel. The decoder will incur a *decoder delay* to reconstruct the signal. In practice, the total delay of many speech coders is at least three frames.

If the coded speech needs to be transmitted over a channel, we need to consider possible channel errors, and our speech decoder should be insensitive to at least some of them. There are two types of errors: random errors and burst errors, and they could be handled differently. One possibility to increase the robustness against such errors is to use channel coding techniques, such as those proposed in Chapter 3. Joint source and channel coding allows us to find the right combination of bits to devote to speech coding with the right amount devoted to channel coding, adjusting this ratio adaptively depending on the channel. Since channel coding will only reduce the number of errors, and not eliminate them, graceful degradation of speech quality under channel errors is typically a design factor for speech coders. When the channel is the Internet, complete frames may be missing because they have not arrived in time. Therefore, we need techniques that degrade gracefully with missing frames.

Amazon/VB Assets
Exhibit 1011
Page 365

7.2. SCALAR WAVEFORM CODERS

In this section we describe several waveform coding techniques, such as linear PCM, μ-law, and A-law PCM, APCM, DPCM, DM, and ADPCM, that quantize each sample using scalar quantization. These techniques attempt to approximate the waveform, and, if a large enough bit rate is available, will get arbitrarily close to it.

7.2.1. Linear Pulse Code Modulation (PCM)

Analog-to-digital converters perform both sampling and quantization simultaneously. To better understand how this process affects the signal it's better to study them separately. We analyzed the effects of sampling in Chapter 5, so now we analyze the effects of quantization, which encodes each sample with a fixed number of bits. With B bits, it is possible to represent 2^B separate quantization levels. The output of the quantizer $\hat{x}[n]$ is given by

$$\hat{x}[n] = Q\{x[n]\} \tag{7.2}$$

Linear *Pulse Code Modulation* (PCM) is based on the assumption that the input discrete signal $x[n]$ is bounded

$$|x[n]| \leq X_{max} \tag{7.3}$$

and that we use *uniform quantization* with quantization step size Δ which is constant for all levels x_l

$$x_l - x_{l-1} = \Delta \tag{7.4}$$

The input/output characteristics are shown by Figure 7.1 for the case of a 3-bit uniform quantizer. The so-called *mid-riser* quantizer has the same number of positive and negative levels, whereas the *mid-tread* quantizer has one more negative than positive levels. The code $c[n]$ is expressed in two's complement representation, which for Figure 7.1 varies between –4 and +3. For the mid-riser quantizer the output $\hat{x}[n]$ can be obtained from the code $c[n]$ through

$$\hat{x}[n] = sign(c[n])\frac{\Delta}{2} + c[n]\Delta \tag{7.5}$$

and for the mid-tread quantizer

$$\hat{x}[n] = c[n]\Delta \tag{7.6}$$

which is often used in computer systems that use two's complement representation.

There are two independent parameters for a uniform quantizer: the number of levels $N = 2^B$, and the step size Δ. Assuming Eq. (7.3), we have the relationship

$$2X_{max} = \Delta 2^B \tag{7.7}$$

Amazon/VB Assets
Exhibit 1011
Page 366

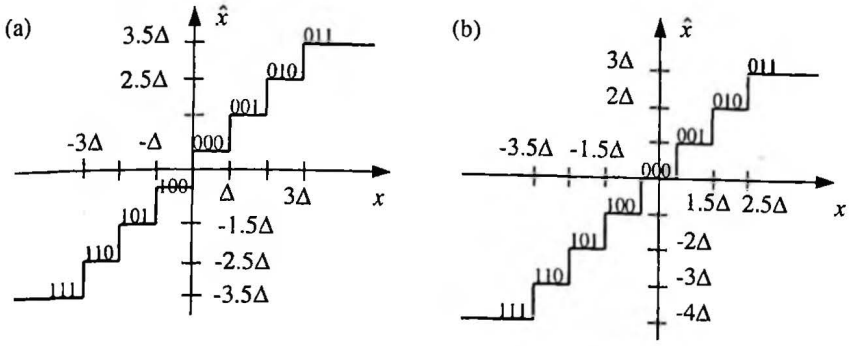


Figure 7.1 Three-bit uniform quantization characteristics: (a) mid-riser, (b) mid-tread.

In quantization, it is useful to express the relationship between the unquantized sample $x[n]$ and the quantized sample $\hat{x}[n]$ as

$$\hat{x}[n] = x[n] + e[n] \tag{7.8}$$

with $e[n]$ being the quantization noise. If we choose Δ and B to satisfy Eq. (7.7), then

$$-\frac{\Delta}{2} \le e[n] \le \frac{\Delta}{2} \tag{7.9}$$

While there is obviously a deterministic relationship between $e[n]$ and $x[n]$, it is convenient to assume a probabilistic model for the quantization noise:

1. $e[n]$ is white: $E\{e[n]e[n+m]\} = \sigma_e^2 \delta[m]$
2. $e[n]$ and $x[n]$ are uncorrelated: $E\{x[n]e[n+m]\} = 0$
3. $e[n]$ is uniformly distributed in the interval $(-\Delta/2, \Delta/2)$

These assumptions are unrealistic for some signals, except in the case of speech signals, which rapidly fluctuate between different quantization levels. The assumptions are reasonable if the step size Δ is small enough, or alternatively the number of levels is large enough (say, more than 2^6).

The variance of such uniform distribution (see Chapter 3) is

$$\sigma_e^2 = \frac{\Delta^2}{12} = \frac{X_{\max}^2}{3 \times 2^{2B}} \tag{7.10}$$

after using Eq. (7.7). The SNR is given by

$$SNR(dB) = 10 \log_{10}\left(\frac{\sigma_x^2}{\sigma_e^2}\right) = (20 \log_{10} 2)B + 10 \log_{10} 3 - 20 \log_{10}\left(\frac{X_{\max}}{\sigma_x}\right) \tag{7.11}$$

which implies that each bit contributes to 6 dB of *SNR*, since $20 \log_{10} 2 \cong 6$.

Amazon/VB Assets
Exhibit 1011
Page 367

Speech samples can be approximately described as following a *Laplacian* distribution [40]

$$p(x) = \frac{1}{\sqrt{2}\sigma_x} e^{-\frac{\sqrt{2}|x|}{\sigma_x}} \tag{7.12}$$

and the probability of x falling outside the range $(-4\sigma_x, 4\sigma_x)$ is 0.35%. Thus, using $X_{max} = 4\sigma_x$, $B = 7$ bits in Eq. (7.11) results in an *SNR* of 35 dB, which would be acceptable in a communications system. Unfortunately, signal energy can vary over 40 dB, due to variability from speaker to speaker as well as variability in transmission channels. Thus, in practice, it is generally accepted that 11 bits are needed to achieve an SNR of 35 dB while keeping the clipping to a minimum.

Digital audio stored in computers (Windows WAV, Apple AIF, Sun AU, and SND formats among others) use 16-bit linear PCM as their main format. The *Compact Disc-Digital Audio* (CD-DA or simply CD) also uses 16-bit linear PCM. Invented in the late 1960s by James T. Russell, it was launched commercially in 1982 and has become one of the most successful examples of consumer electronics technology: there were about 700 million audio CD players in 1997. A CD can store up to 74 minutes of music, so the total amount of digital data that must be stored on a CD is 44,100 samples/(channel*second) * 2 bytes/sample * 2 channels * 60 seconds/minute * 74 minutes = 783,216,000 bytes. This 747 MB are stored in a disk only 12 centimeters in diameter and 1.2 mm thick. CD-ROMs can record only 650 MB of computer data because they use the remaining bits for error correction.

7.2.2. μ-law and A-law PCM

Human perception is affected by SNR, because adding noise to a signal is not as noticeable if the signal energy is large enough. Ideally, we want SNR to be constant for all quantization levels, which requires the step size to be proportional to the signal value. This can be done by using a logarithmic *compander*[1]

$$y[n] = \ln|x[n]| \tag{7.13}$$

followed by a uniform quantizer on $y[n]$ so that

$$\hat{y}[n] = y[n] + \varepsilon[n] \tag{7.14}$$

and, thus,

$$\hat{x}[n] = \exp\{\hat{y}[n]\}\text{sign}\{x[n]\} = x[n]\exp\{\varepsilon[n]\} \tag{7.15}$$

[1] A compander is a nonlinear function that compands one part of the x-axis.

Amazon/VB Assets
Exhibit 1011
Page 368

after using Eqs. (7.13) and (7.14). If $\varepsilon[n]$ is small, then Eq. (7.15) can be expressed as

$$\hat{x}[n] \cong x[n](1+\varepsilon[n]) = x[n] + x[n]\varepsilon[n] \tag{7.16}$$

and, thus, the $SNR = 1/\sigma_\varepsilon^2$ is constant for all levels. This type of quantization is not practical, because an infinite number of quantization steps would be required. An approximation is the so-called μ-law [51]:

$$y[n] = X_{\max} \frac{\log\left[1+\mu\dfrac{|x[n]|}{X_{\max}}\right]}{\log[1+\mu]} \operatorname{sign}\{x[n]\} \tag{7.17}$$

which is approximately logarithmic for large values of $x[n]$ and approximately linear for small values of $x[n]$. A related compander called A-law is also used

$$y[n] = X_{\max} \frac{1+\log\left[\dfrac{A|x[n]|}{X_{\max}}\right]}{1+\log A} \operatorname{sign}\{x[n]\} \tag{7.18}$$

which has greater resolution than μ-law for small sample values, but a range equivalent to 12 bits. In practice, they both offer similar quality. The μ-law curve can be seen in Figure 7.2.

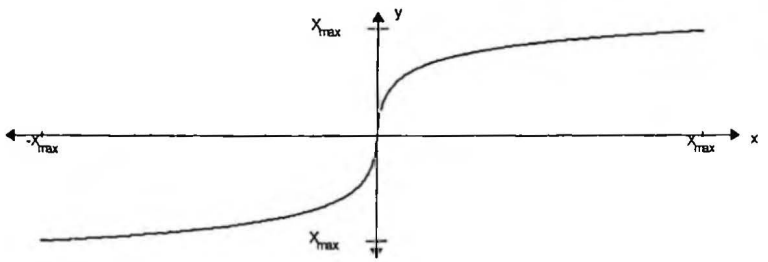


Figure 7.2 Nonlinearity used in the μ-law compression.

In 1972 the ITU-T[2] recommendation G.711 standardized telephone speech coding at 64 kbps for digital transmission of speech through telephone networks. It uses 8 bits per sample and an 8-kHz sampling rate with either μ-law or *A*-law. In North America and Japan, μ-law with $\mu = 255$ is used, whereas, in the rest of the world, *A*-law with $A = 87.56$ is used. Both compression characteristics are very similar and result in an approximate SNR of 35 dB. Without the logarithmic compressor, a uniform quantizer requires approximately 12 bits

[2] The International Telecommunication Union (ITU) is a part of the United Nations Economic, Scientific and Cultural Organization (UNESCO). ITU-T is the organization within ITU responsible for setting global telecommunication standards. Within ITU-T, Study Group 15 (SG15) is responsible for formulating speech coding standards. Prior to 1993, telecommunication standards were set by the *Comité Consultatif International Téléphonique et Télégraphique* (CCITT), which was reorganized into the ITU-T that year.

Amazon/VB Assets
Exhibit 1011
Page 369

per sample to achieve the same level of quality. All the speech coders for telephone speech described in this chapter use G.711 as a baseline reference, whose quality is considered *toll*, and an MOS of about 4.3. G.711 is used by most digital central office switches, so that when you make a telephone call using your plain old telephone service (POTS), your call is encoded with G.711.

7.2.3. Adaptive PCM

When quantizing speech signals we confront a dilemma. On the one hand, we want the quantization step size to be large enough to accommodate the maximum peak-to-peak range of the signal and avoid clipping. On the other hand, we need to make the step size small to minimize the quantization noise. One possible solution is to adapt the step size to the level of the input signal.

The basic idea behind *Adaptive PCM* (APCM) is to let the step size $\Delta[n]$ be proportional to the standard deviation of the signal $\sigma[n]$:

$$\Delta[n] = \Delta_0 \sigma[n] \tag{7.19}$$

An equivalent method is to use a fixed quantizer but have a time-varying gain $G[n]$, which is inversely proportional to the signal's standard deviation

$$G[n] = G_0 / \sigma[n] \tag{7.20}$$

Estimation of the signal's variance, or short-time energy, is typically done by low-pass filtering $x^2[n]$. With a first-order IIR filter, the variance $\sigma^2[n]$ is computed as

$$\sigma^2[n] = \alpha\sigma^2[n-1] + (1-\alpha)x^2[n-1] \tag{7.21}$$

with α controlling the time constant of the filter $T = -1/(F_s \ln\alpha)$, F_s the sampling rate, and $0 < \alpha < 1$. In practice, α is chosen so that the time constant ranges between 1 ms ($\alpha = 0.88$ at 8 kHz) and 10 ms ($\alpha = 0.987$ at 8 kHz).

Alternatively, $\sigma^2[n]$ can be estimated from the past M samples:

$$\sigma^2[n] = \frac{1}{M} \sum_{m=n-M}^{n-1} x^2[m] \tag{7.22}$$

In practice, it is advantageous to set limits on the range of values of $\Delta[n]$ and $G[n]$:

$$\Delta_{min} \le \Delta[n] \le \Delta_{max} \tag{7.23}$$

$$G_{min} \le G[n] \le G_{max} \tag{7.24}$$

with the ratios $\Delta_{max} / \Delta_{min}$ and G_{max} / G_{min} determining the dynamic range of the system. If our objective is to obtain a relatively constant SNR over a range of 40 dB, these ratios can be 100.

Amazon/VB Assets
Exhibit 1011
Page 370

Feedforward adaptation schemes require us to transmit, in addition to the quantized signal, either the step size $\Delta[n]$ or the gain $G[n]$. Because these values evolve slowly with time, they can be sampled and quantized at a low rate. The overall rate will be the sum of the bit rate required to transmit the quantized signal plus the bit rate required to transmit either the gain or the step size.

Another class of adaptive quantizers use *feedback adaptation* to avoid having to send information about the step size or gain. In this case, the step size and gain are estimated from the quantizer output, so that they can be recreated at the decoder without any extra information. The corresponding short-time energy can then be estimated through a first-order IIR filter as in Eq. (7.21) or a rectangular window as in Eq. (7.22), but replacing $x^2[n]$ by $\hat{x}^2[n]$.

Another option is to adapt the step size

$$\Delta[n] = P\Delta[n-1] \tag{7.25}$$

where $P > 1$ if the previous codeword corresponds to the largest positive or negative quantizer level, and $P < 1$ if the previous codeword corresponds to the smallest positive or negative quantizer level. A similar process can be done for the gain.

APCM exhibits an improvement between 4–8 dB over μ-law PCM for the same bit rate.

7.2.4. Differential Quantization

Speech coding is about finding redundancy in the signal and removing it. We know that there is considerable correlation between adjacent samples, because on the average the signal doesn't change rapidly from sample to sample. A simple way of capturing this is to quantize the difference $d[n]$ between the current sample $x[n]$ and its predicted value $\tilde{x}[n]$

$$d[n] = x[n] - \tilde{x}[n] \tag{7.26}$$

with its quantized value represented as

$$\hat{d}[n] = Q\{d[n]\} = d[n] + e[n] \tag{7.27}$$

where $e[n]$ is the quantization error. Then, the quantized signal is the sum of the predicted signal $\tilde{x}[n]$ and the quantized difference $\hat{d}[n]$

$$\hat{x}[n] = \tilde{x}[n] + \hat{d}[n] = x[n] + e[n] \tag{7.28}$$

If the prediction is good, Eq. (7.28) tells us that the quantization error will be small. Statistically, we need the variance of $e[n]$ to be lower than that of $x[n]$ for differential coding to provide any gain. Systems of this type are generically called *Differential Pulse Code Modulation* (DPCM) [11] and can be seen in Figure 7.3.

Amazon/VB Assets
Exhibit 1011
Page 371

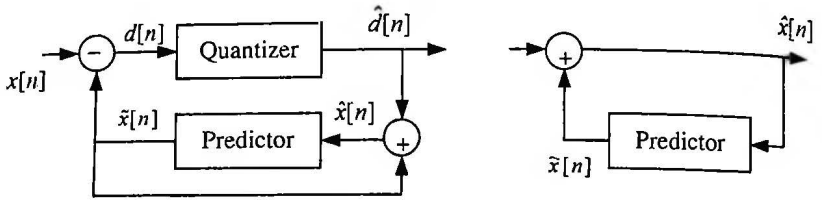


Figure 7.3 Block diagram of a DPCM encoder and decoder with feedback prediction.

Delta Modulation (DM) [47] is a 1-bit DPCM, which predicts the current sample to be the same as the past sample:

$$\tilde{x}[n] = x[n-1] \tag{7.29}$$

so that we transmit whether the current sample is above or below the previous sample.

$$d[n] = \begin{cases} \Delta & x[n] > x[n-1] \\ -\Delta & x[n] \le x[n-1] \end{cases} \tag{7.30}$$

with Δ being the step size. If Δ is too small, the reconstructed signal will not increase as fast as the original signal, a condition known as *slope overload distortion*. When the slope is small, the step size Δ also determines the peak error; this is known as *granular noise*. Both quantization errors can be seen in Figure 7.4. The choice of Δ that minimizes the mean squared error will be a tradeoff between slope overload and granular noise.

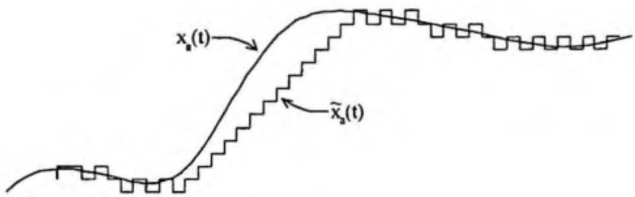


Figure 7.4 An example of slope overload distortion and granular noise in a DM encoder.

If the signal is oversampled by a factor N, and the step size is reduced by the same amount (i.e., Δ/N), the slope overload will be the same, but the granular noise will decrease by a factor N. While the coder is indeed very simple, sampling rates of over 200 kbps are needed for SNRs comparable to PCM, so DM is rarely used as a speech coder.

However, delta modulation is useful in the design of analog-digital converters, in a variant called sigma-delta modulation [44] shown in Figure 7.5. First the signal is lowpass filtered with a simple analog filter, and then it is oversampled. Whenever the predicted signal $\tilde{x}[n]$ is below the original signal $x[n]$, the difference $d[n]$ is positive. This difference $d[n]$

Amazon/VB Assets
Exhibit 1011
Page 372

is averaged over time with a digital integrator whose output is $e[n]$. If this situation persists, the accumulated error $e[n]$ will exceed a positive value A, which causes a 1 to be encoded into the stream $q[n]$. A digital-analog converter is used in the loop which increments by one the value of the predicted signal $\tilde{x}[n]$. The system acts in the opposite way if the predicted signal $\tilde{x}[n]$ is above the original signal $x[n]$ for an extended period of time. Since the signal is oversampled, it changes very slowly from one sample to the next, and this quantization can be accurate. The advantages of this technique as an analog-digital converter are that inexpensive analog filters can be used and only a simple 1-bit A/D is needed. The signal can next be low-passed filtered with a more accurate digital filter and then downsampled.

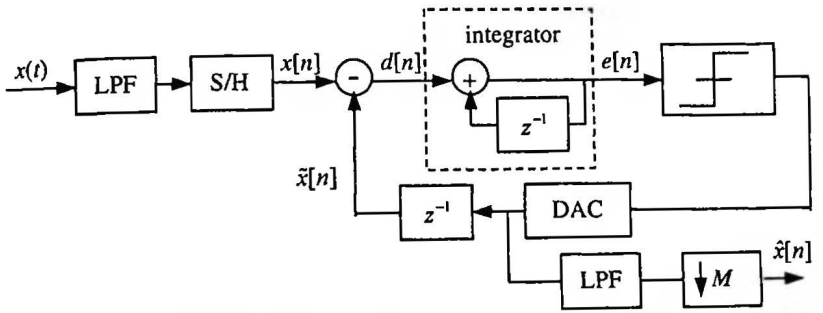


Figure 7.5 A sigma-delta modulator used in an oversampling analog-digital converter.

Adaptive Delta Modulation (ADM) combines ideas from adaptive quantization and delta modulation with the so-called *Continuously Variable Slope Delta Modulation* (CVSDM) [22] having a step size that increases

$$\Delta[n] = \begin{cases} \alpha\Delta[n-1] + k_1 & \text{if } e[n], e[n-1] \text{ and } e[n-2] \text{ have same sign} \\ \alpha\Delta[n-1] + k_2 & \text{otherwise} \end{cases} \tag{7.31}$$

with $0 < \alpha < 1$ and $0 < k_2 << k_1$. The step size increases if the last three errors have the same sign and decreases otherwise.

Improved DPCM is achieved through linear prediction in which $\tilde{x}[n]$ is a linear combination of past quantized values $\hat{x}[n]$

$$\tilde{x}[n] = \sum_{k=1}^{p} a_k \hat{x}[n-k] \tag{7.32}$$

DPCM systems with fixed prediction coefficients can provide from 4 to 11 dB improvement over direct linear PCM, for prediction orders up to $p = 4$, at the expense of increased computational complexity. Larger improvements can be obtained by adapting the

Amazon/VB Assets
Exhibit 1011
Page 373

prediction coefficients. The coefficients can be transmitted in a feedforward fashion or not transmitted if the feedback scheme is selected.

ADPCM [6] combines differential quantization with adaptive step-size quantization. ITU-T Recommendation G.726 uses ADPCM at bit rates of 40, 32, 24, and 16 kbps, with 5, 4, 3, and 2 bits per sample, respectively. It employs an adaptive feedback quantizer and an adaptive feedback pole-zero predictor. Speech at bit rates of 40 and 32 kbps offer toll quality, while the other rates don't. G.727 is called embedded ADPCM because the 2-bit quantizer is embedded into the 3-bit quantizer, which is embedded into the 4-bit quantizer, and into the 5-bit quantizer. This makes it possible for the same codec to use a lower bit rate, with a graceful degradation in quality, if channel capacity is temporarily limited. Earlier standards G.721 [7, 13] (created in 1984) and G.723 have been subsumed by G.726 and G.727. G.727 has a MOS of 4.1 for 32 kbps and is used in submarine cables. The Windows WAV format also supports a variant of ADPCM. These standards are shown in Table 7.2.

Table 7.2 Common scalar waveform standards used.

Standard	Bit Rate (kbits/sec)	MOS	Algorithm	Sampling Rate (kHz)
Stereo CD Audio	1411	5.0	16-bit linear PCM	44.1
WAV, AIFF, SND	Variable	-	16/8-bit linear PCM	8, 11.025, 16, 22.05, 44.1, 48
G.711	64	4.3	μ-law/A-law PCM	8
G.727	40, 32, 24, 16	4.2 (32k)	ADPCM	8
G.722	64, 56, 48		Subband ADPCM	16

Wideband speech (50–7000 Hz) increases intelligibility of fricatives and overall perceived quality. In addition, it provides more subject presence and adds a feeling of transparent communication. ITU-T Recommendation G.722 encodes wideband speech with bit rates of 48, 56, and 64 kbps. Speech is divided into two subbands with QMF filters (see Chapter 5). The upper band is encoded using a 16-kbps ADPCM similar to the G.727 standard. The lower band is encoded using a 48-kbps ADPCM with the 4- and 5-bit quantizers embedded in the 6-bit quantizer. The quality of this system scores almost 1 MOS higher than that of telephone speech.

7.3. SCALAR FREQUENCY DOMAIN CODERS

Frequency domain is advantageous because:

1. The samples of a speech signal have a great deal of correlation among them, whereas frequency domain components are approximately uncorrelated and

Amazon/VB Assets
Exhibit 1011
Page 374

2. The perceptual effects of masking described in Chapter 2 can be more easily implemented in the frequency domain. These effects are more pronounced for high-bandwidth signals, so frequency-domain coding has been mostly used for CD-quality signals and not for 8-kHz speech signals.

7.3.1. Benefits of Masking

As discussed in Chapter 2, masking is a phenomenon by which human listeners cannot perceive a sound if it is below a certain level. The consequence is that we don't need to encode such sound. We now illustrate how this masked threshold is computed for MPEG[3]-1 layer 1. Given an input signal $s[n]$ quantized with b bits, we obtain the normalized signal $x[n]$ as

$$x[n] = \frac{s[n]}{N2^{b-1}} \tag{7.33}$$

where $N = 512$ is the length of the DFT. Then, using a Hanning window,

$$w[n] = 0.5 - 0.5\cos(2\pi n / N) \tag{7.34}$$

we obtain the log-power spectrum as

$$P[k] = P_0 + 10\log_{10}\left|\sum_{n=0}^{N-1} w[n]x[n]e^{-j2\pi nk/N}\right|^2 \tag{7.35}$$

where P_0 is the playback SPL, which, in the absence of any volume information, is defined as 90 dB.

Tonal components are identified in Eq. (7.35) as local maxima, which exceed neighboring components within a certain Bark distance by at least 7 dB. Specifically, bin k is tonal if and only if

$$P[k] > P[k \pm 1] \tag{7.36}$$

and

$$P[k] > P[k \pm l] + 7dB \tag{7.37}$$

where $1 < l \le \Delta_k$, and Δ_k is given by

$$\Delta_k = \begin{cases} 2 & 2 < k < 63 \quad (170\,\text{Hz} - 5.5\,\text{kHz}) \\ 3 & 63 \le k < 127 \quad (5.5\,\text{kHz}, 11\,\text{kHz}) \\ 6 & 127 \le k \le 256 \quad (11\,\text{kHz}, 22\,\text{kHz}) \end{cases} \tag{7.38}$$

[3] MPEG (Moving Picture Experts Group) is the nickname given to a family of International Standards for coding audiovisual information.

Amazon/VB Assets
Exhibit 1011
Page 375

so that the power of that tonal masker is computed as the sum of the power in that bin and its left and right adjacent bins:

$$P_{TM}[k] = 10\log_{10}\left(\sum_{j=-1}^{1} 10^{0.1P[k+j]}\right) \tag{7.39}$$

The noise maskers are computed as the sum of power spectrum of the remaining frequency bins $\bar{k}$ in a critical band not within a neighborhood Δ_k of the tonal maskers:

$$P_{NM}[\bar{k}] = 10\log_{10}\left(\sum_{j} 10^{0.1P[j]}\right) \tag{7.40}$$

where j spans a critical band.

To compute the overall masked threshold we need to sum all masking thresholds contributed by each frequency bin i, which is approximately equal to the maximum (see Chapter 2):

$$T[k] = \max\left(T_h[k], \max_i\left(T_i[k]\right)\right) \tag{7.41}$$

In Chapter 2 we saw that whereas temporal postmasking can last from 50 to 300 ms, temporal premasking tends to last about 5 ms. This is also important because when a frequency transform is quantized, the blocking effects of transform's coders can introduce noise above the temporal premasking level that can be audible, since 1024 points corresponds to 23 ms at a 44-kHz sampling rate. To remove this pre-echo distortion, audible in the presence of castanets and other abrupt transient signals, subband filtering has been proposed, whose time constants are well below the 5-ms premasking time constant.

7.3.2. Transform Coders

We now use the *Adaptive Spectral Entropy Coding* (ASPEC) of High Quality Music Signals algorithm, which is the basis for the MPEG1 Layer 1 audio coding standard [24], to illustrate how transform coders work. The DFT coefficients are grouped into 128 subbands, and 128 scalar quantizers are used to transmit all the DFT coefficients. It has been empirically found that a difference of less than 1 dB between the original amplitude and the quantized value cannot be perceived. Each subband j has a quantizer having k_j levels and step size of T_j as

$$k_j = 1 + 2\times \mathrm{rnd}\left(P_j / T_j\right) \tag{7.42}$$

where T_j is the quantized JND threshold, P_j is the quantized magnitude of the largest real or imaginary component of the j^{th} subband, and rnd( ) is the nearest integer rounding function. Entropy coding (see Chapter 3) is used to encode the coefficients of that subband. Both

Amazon/VB Assets
Exhibit 1011
Page 376

T_j and P_j are quantized on a dB scale using 8-bit uniform quantizers with a 170-dB dynamic range, thus with a step size of 0.66 dB. Then they are transmitted as side information.

There are two main methods of obtaining a frequency-domain representation:

1. Through subband filtering via a filterbank (see Chapter 5). When a filterbank is used, the bandwidth of each band is chosen to increase with frequency following a perceptual scale, such as the Bark scale. As shown in Chapter 5, such filterbanks yield perfect reconstruction in the absence of quantization.
2. Through frequency-domain transforms. Instead of using a DFT, higher efficiency can be obtained by the use of an MDCT (see Chapter 5).

The exact implementation of the MPEG1 Layer I standard is much more complicated and beyond the scope of this book, though it follows the main ideas described here; the same is true for the popular MPEG1 Layer III, also known as MP3. Implementation details can be found in [42].

7.3.3. Consumer Audio

Dolby Digital, MPEG, DTS, and the Perceptual Audio Coder (PAC) [28] are all audio coders based on frequency-domain coding. Except for MPEG-1, which supports only stereo signals, the rest support multichannel.

Dolby Digital is multichannel digital audio, using lossy AC-3 [54] coding technology from original PCM with a sample rate of 48 kHz at up to 24 bits. The bit rate varies from 64 to 448 kbps, with 384 being the normal rate for 5.1 channels and 192 the normal rate for stereo (with or without surround encoding). Most Dolby Digital decoders support up to 640 kbps. Dolby Digital is the format used for audio tracks on almost all Digital Video/Versatile Discs (DVD). A DVD-5 with only one surround stereo audio stream (at 192 kbps) can hold over 55 hours of audio. A DVD-18 can hold over 200 hours.

MPEG was established in 1988 as part of the joint ISO (International Standardization Organization) / IEC (International Electrotechnical Commission) Technical Committee on Information Technology. MPEG-1 was approved in 1992 and MPEG-2 in 1994. Layers I to III define several specifications that provide better quality at the expense of added complexity. MPEG-1 audio is limited to 384 kbps. MPEG1 Layer III audio [23], also known as MP3, is very popular on the Internet, and many compact players exist.

MPEG-2 audio, one of the audio formats used in DVD, is multichannel digital audio, using lossy compression from 16-bit linear PCM at 48 kHz. Tests have shown that for nearly all types of speech and music, at a data rate of 192 kbps and over, on a stereo channel, scarcely any difference between original and coded versions was observable (ranking of coded item > 4.5), with the original signal needing 1.4 Mbps on a CD (reduction by a factor of 7). One advantage of the MPEG audio technique is that future findings regarding psychoacoustic effects can be incorporated later, so it can be expected that today's quality level

Amazon/VB Assets
Exhibit 1011
Page 377

using 192 kbps will be achievable at lower data rates in the future. A variable bit rate of 32 to 912 kbps is supported for DVDs.

DTS (Digital Theater Systems) Digital Surround is another multi-channel (5.1) digital audio format, using lossy compression derived from 20-bit linear PCM at 48 kHz. The compressed data rate varies from 64 to 1536 kbps, with typical rates of 768 and 1536 kbps.

7.3.4. Digital Audio Broadcasting (DAB)

Digital Audio Broadcasting (DAB) is a means of providing current AM and FM listeners with a new service that offers: sound quality comparable to that of compact discs, increased service availability (especially for reception in moving vehicles), flexible coverage scenarios, and high spectrum efficiency.

Different approaches have been considered for providing listeners with such a service. Currently, the most advanced system is one commonly referred to as Eureka 147 DAB, which has been under development in Europe under the Eureka Project EU147 since 1988. Other approaches include various American in-band systems (IBOC, IBAC, IBRC, FMDigital, and FMeX) still in development, as well as various other systems promising satellite delivery, such as WorldSpace and CD Radio, still in development as well. One satellite-delivery system called MediaStar (formerly Archimedes) proposes to use the Eureka 147 DAB signal structure, such that a single receiver could access both terrestrial and satellite broadcasts.

DAB has been under development since 1981 at the Institut für Rundfunktechnik (IRT) and since 1987 as part of a European research project (Eureka 147). The Eureka 147 DAB specification was standardized by the European Telecommunications Standards Institute (ETSI) in February 1995 as document ETS 300 401, with a draft second edition issued in June 1996. In December 1994, the International Telecommunication Union—Radiocommunication (ITU-R) recommended that this technology, referred to as Digital System A, be used for implementing DAB services.

The Eureka 147 DAB signal consists of multiple carriers within a 1.536-MHz channel bandwidth. Four possible modes of operation define the channel coding configuration, specifying the total number of carriers, the carrier spacing, and also the guard interval duration. Each channel provides a raw data rate of 2304 kbps; after error protection, a useful data rate of anywhere between approximately 600 kbps up to 1800 kbps is available to the service provider, depending on the user-specified multiplex configuration. This useful data rate can be divided into an infinite number of possible configurations of audio and data programs. All audio programs are individually compressed using MUSICAM (MPEG-1 Layer II).

For each useful bit, 1 1/3 ... 4 bits are transmitted. This extensive redundancy makes it possible to reconstruct the transmitted bit sequence in the receiver, even if part of it is disrupted during transmission (FEC—forward error correction). In the receiver, error concealment can be carried out at the audio reproduction stage, so that residual transmission errors which could not be corrected do not always cause disruptive noise.

Amazon/VB Assets
Exhibit 1011
Page 378

7.4. CODE EXCITED LINEAR PREDICTION (CELP)

The use of linear predictors removes redundancy in the signal, so that coding of the residual signal can be done with simpler quantizers. We first introduce the LPC vocoder and then introduce coding of the residual signal with a very popular technique called CELP.

7.4.1. LPC Vocoder

A typical model for speech production is shown in Figure 7.6, which has a source, or excitation, driving a linear time-varying filter. For voiced speech, the excitation is an impulse train spaced P samples apart. For unvoiced speech, the source is white random noise. The filter $h_m[n]$ for frame m changes at regular intervals, say every 10 ms. If this filter is represented with linear predictive coding, it is called an *LPC vocoder* [3].

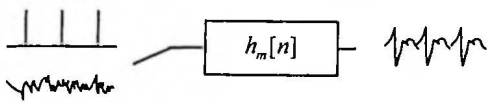


Figure 7.6 Block diagram of an LPC vocoder.

In addition to transmitting the gain and LPC coefficients, the encoder has to determine whether the frame is voiced or unvoiced, as well as the pitch period P for voiced frames.

The LPC vocoder produces reasonable quality for unvoiced frames, but often results in somewhat mechanical sound for voiced sounds, and a buzzy quality for voiced fricatives. More importantly, the LPC vocoder is quite sensitive to voicing and pitch errors, so that an accurate pitch tracker is needed for reasonable quality. The LPC vocoder also performs poorly in the presence of background noise. Nonetheless, it can be highly intelligible. The Federal Standard 1015 [55], proposed for secure communications, is based on a 2.4-kbps LPC vocoder.

It's also possible to use linear predictive coding techniques together with Huffman coding [45] to achieve lossless compression of up to 50%.

7.4.2. Analysis by Synthesis

Code Excited Linear Prediction (CELP) [5] is an umbrella term for a family of techniques that quantize the LPC residual using VQ, thus the term *code excited*, using analysis by synthesis. In addition CELP uses the fact that the residual of voiced speech has periodicity and can be used to predict the residual of the current frame. In CELP coding the LPC coefficients are quantized and transmitted (feedforward prediction), as well as the codeword index. The prediction using LPC coefficients is called *short-term prediction*. The prediction of the residual based on pitch is called *long-term prediction*. To compute the quantized coefficients we use an *analysis-by-synthesis* technique, which consists of choosing the combina-

Amazon/VB Assets
Exhibit 1011
Page 379

tion of parameters whose reconstructed signal is closest to the analysis signal. In practice, not all coefficients of a CELP coder are estimated in an analysis-by-synthesis manner.

We first estimate the p^{th}-order LPC coefficients from the samples $x[n]$ for frame t using the autocorrelation method, for example. We then quantize the LPC coefficients to $(a_1, a_2, \cdots, a_p)$ with the techniques described in Section 7.4.5. The residual signal $e[n]$ is obtained by inverse filtering $x[n]$ with the quantized LPC filter

$$e[n] = x[n] - \sum_{i=1}^{p} a_i x[n-i] \tag{7.43}$$

Given the transfer function of the LPC filter

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} = \sum_{i=0}^{\infty} h_i z^{-i} \tag{7.44}$$

we can obtain the first M coefficients of the impulse response $h[n]$ of the LPC filter by driving it with an impulse as

$$h[n] = \begin{cases} 1 & n = 0 \\ \sum_{i=1}^{n} a_i h[n-i] & 0 < n < p \\ \sum_{i=1}^{p} a_i h[n-i] & p \le n < M \end{cases} \tag{7.45}$$

so that if we quantize a frame of M samples of the residual $\mathbf{e} = (e[0], e[1], \cdots e[M-1])^T$ to $\mathbf{e}_i = (e_i[0], e_i[1], \cdots e_i[M-1])^T$, we can compute the reconstructed signal $\hat{x}_i[n]$ as

$$\hat{x}_i[n] = \sum_{m=0}^{n} h[m] e_i[n-m] + \sum_{m=n+1}^{\infty} h[m] e[n-m] \tag{7.46}$$

where the second term in the sum depends on the residual for previous frames, which we already have. Let's define signal $r_0[n]$ as the second term of Eq (7.46):

$$r_0[n] = \sum_{m=n+1}^{\infty} h[m] e[n-m] \tag{7.47}$$

which is the output of the LPC filter when there is no excitation for frame t. The important thing to note is that $r_0[n]$ does not depend on $e_i[n]$.

It is convenient to express Eqs. (7.46) and (7.47) in matrix form as

$$\hat{\mathbf{x}}_i = \mathbf{H}\mathbf{e}_i + \mathbf{r}_0 \tag{7.48}$$

where matrix $\mathbf{H}$ corresponds to the LPC filtering operation with its memory set to 0:

Amazon/VB Assets
Exhibit 1011
Page 380

$$\mathbf{H} = \begin{bmatrix} h_0 & 0 & \cdots & 0 & 0 \\ h_1 & h_0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ h_{M-1} & h_{M-2} & \cdots & h_0 & 0 \\ h_M & h_{M-1} & \cdots & h_1 & h_0 \end{bmatrix} \tag{7.49}$$

Given the large dynamic range of the residual signal, we use gain-shape quantization, where we quantize the gain and the gain-normalized residual separately:

$$\mathbf{e}_i = \lambda \mathbf{c}_i \tag{7.50}$$

where λ is the gain and $\mathbf{c}_i$ is the codebook entry i. This codebook is known as the *fixed codebook* because its vectors do not change from frame to frame. Usually the size of the codebook is selected as 2^N so that full use is made of all N bits. Codebook sizes typically vary from 128 to 1024. Combining Eq. (7.48) with Eq. (7.50), we obtain

$$\hat{\mathbf{x}}_i = \lambda \mathbf{H}\mathbf{c}_i + \mathbf{r}_0 \tag{7.51}$$

The error between the original signal $\mathbf{x}$ and the reconstructed signal $\hat{\mathbf{x}}_i$ is

$$\boldsymbol{\varepsilon}_i = \mathbf{x} - \hat{\mathbf{x}}_i \tag{7.52}$$

The optimal gain λ and codeword index i are the ones that minimize the squared error between the original signal and the reconstructed[4] signal:

$$E(i,\lambda) = \left|\mathbf{x} - \hat{\mathbf{x}}_i\right|^2 = \left|\mathbf{x} - \lambda\mathbf{H}\mathbf{c}_i - \mathbf{r}_0\right|^2 = \left|\mathbf{x} - \mathbf{r}_0\right|^2 + \lambda^2 \mathbf{c}_i^T \mathbf{H}^T \mathbf{H}\mathbf{c}_i - 2\lambda \mathbf{c}_i^T \mathbf{H}^T (\mathbf{x} - \mathbf{r}_0) \tag{7.53}$$

where the term $\left|\mathbf{x} - \mathbf{r}_0\right|^2$ does not depend on λ or i and can be neglected in the minimization. For a given $\mathbf{c}_i$, the gain λ_i that minimizes Eq. (7.53) is given by

$$\lambda_i = \frac{\mathbf{c}_i^T \mathbf{H}^T (\mathbf{x} - \mathbf{r}_0)}{\mathbf{c}_i^T \mathbf{H}^T \mathbf{H}\mathbf{c}_i} \tag{7.54}$$

Inserting Eq. (7.54) into (7.53) lets us compute the index j as the one that minimizes

$$j = \arg\min_i \left\{ -\frac{\left(\mathbf{c}_i^T \mathbf{H}^T (\mathbf{x} - \mathbf{r}_0)\right)^2}{\mathbf{c}_i^T \mathbf{H}^T \mathbf{H}\mathbf{c}_i} \right\} \tag{7.55}$$

So we first obtain the codeword index j according to Eq. (7.55) and then the gain λ_j according to Eq. (7.54), which is scalarly quantized to $\hat{\lambda}_j$. Both codeword index j and $\hat{\lambda}_j$ are transmitted. In the algorithm described here, we first chose the quantized LPC coeffi-

[4] A beginner's mistake is to find the codebook index that minimizes the squared error of the residual. This does not minimize the difference between the original signal and the reconstructed signal.

Amazon/VB Assets
Exhibit 1011
Page 381

cients $(a_1, a_2, \cdots, a_p)$ independently of the gains and codeword index, and then we chose the codeword index independently of the quantized gain $\hat{\lambda}_j$. This procedure is called *open-loop* estimation, because some parameters are obtained independently of the others. This is shown in Figure 7.7. *Closed-loop* estimation [49] means that all possible combinations of quantized parameters are explored. Closed-loop is more computationally expensive but yields lower squared error.

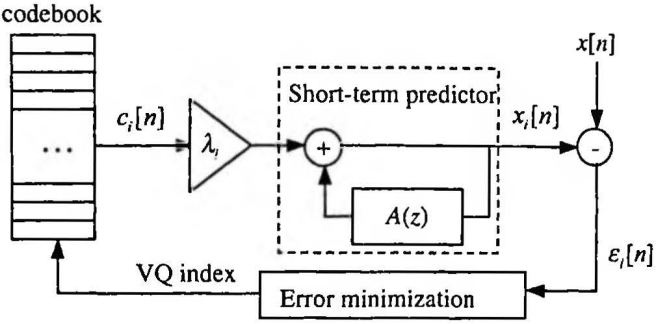


Figure 7.7 Analysis-by-synthesis principle used in a basic CELP.

7.4.3. Pitch Prediction: Adaptive Codebook

The fact that speech is highly periodic during voiced segments can also be used to reduce redundancy in the signal. This can be done by predicting the residual signal $e[n]$ at the current vector with samples from the past residual signal shifted a pitch period t:

$$e[n] = \lambda_i^a e[n-t] + \lambda_i^f c_i^f[n] = \lambda_i^a c_i^a[n] + \lambda_i^f c_i^f[n] \tag{7.56}$$

Using the matrix framework we described before, Eq. (7.56) can be expressed as

$$\mathbf{e}_{ti} = \lambda_i^a \mathbf{c}_t^a + \lambda_i^f \mathbf{c}_i^f \tag{7.57}$$

where we have made use of an *adaptive codebook* [31], where $\mathbf{c}_t^a$ is the adaptive codebook entry j with corresponding gain λ^a, and $\mathbf{c}_i^f$ is the fixed or stochastic codebook entry i with corresponding gain λ^f. The adaptive codebook entries are segments of the recently synthesized excitation signal

$$\mathbf{c}_t^a = (e[-t], e[1-t], \cdots, e[M-1-t])^T \tag{7.58}$$

where t is the delay which specifies the start of the adaptive codebook entry t. The range of t is often between 20 and 147, since this can be encoded with 7 bits. This corresponds to a range in pitch frequency between 54 and 400 Hz for a sampling rate of 8 kHz.

Amazon/VB Assets
Exhibit 1011
Page 382

The contribution of the adaptive codebook is much larger than that of the stochastic codebook for voiced sounds. So we generally search for the adaptive codebook first, using Eq. (7.58) and a modified version of Eqs. (7.55) and (7.54), replacing i by t. Closed-loop search of both t and gain here often yields a much larger error reduction.

7.4.4. Perceptual Weighting and Postfiltering

The objective of speech coding is to reduce the bit rate while maintaining a perceived level of quality; thus, minimization of the error is not necessarily the best criterion. A perceptual weighting filter tries to shape the noise so that it gets masked by the speech signal (see Chapter 2). This generally means that most of the quantization noise energy is located in spectral regions where the speech signal has most of its energy. A common technique [4] consists in approximating this perceptual weighting with a linear filter

$$W(z) = \frac{A(z/\beta)}{A(z/\gamma)} \tag{7.59}$$

where $A(z)$ is the predictor polynomial

$$A(z) = 1 - \sum_{i=1}^{p} a_i z^{-i} \tag{7.60}$$

Choosing γ and β so that and $0 < \gamma < \beta \leq 1$, implies that the roots of $A(z/\beta)$ and $A(z/\gamma)$ will move closer to the origin of the unit circle than the roots of $A(z)$, thus resulting in a frequency response with wider resonances. This perceptual filter therefore deemphasizes the contribution of the quantization error near the formants. A common choice of parameters is $\beta = 1.0$ and $\gamma = 0.8$, since it simplifies the implementation. This filter can easily be included in the matrix **H**, and a CELP coder incorporating the perceptual weighting is shown in Figure 7.8.

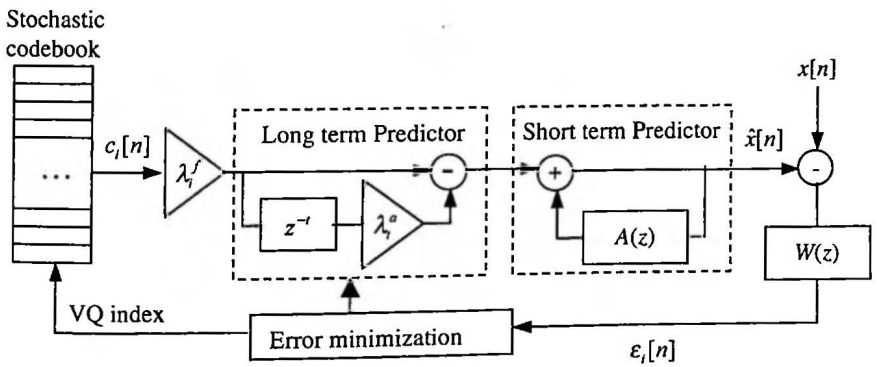


Figure 7.8 Diagram of a CELP coder. Both long-term and short-term predictors are used, together with a perceptual weighting.

Amazon/VB Assets
Exhibit 1011
Page 383

Despite the perceptual weighting filter, the reconstructed signal still contains audible noise. This filter reduces the noise in those frequency regions that are perceptually irrelevant without degrading the speech signal. The postfilter generally consists of a short-term postfilter to emphasize the formant structure and a long-term postfilter to enhance the periodicity of the signal [10]. One possible implementation follows Eq. (7.59) with values of $\beta = 0.5$ and $\gamma = 0.75$.

7.4.5. Parameter Quantization

To achieve a low bit rate, all the coefficients need to be quantized. Because of its coding efficiency, vector quantization is the compression technique of choice to quantize the predictor coefficients. The LPC coefficients cannot be quantized directly, because small errors produced in the quantization process may result in large changes in the spectrum and possibly unstable filters. Thus, equivalent representations that guarantee stability are used, such as reflection coefficients, log-area ratios, and the line spectral frequencies (LSF) described in Chapter 6. LSF are used most often, because it has been found empirically that they behave well when they are quantized and interpolated [2]. For 8 kHz, 10 predictor coefficients are often used, which makes using a single codebook impractical because of the large dimension of the vector. Split-VQ [43] is a common choice, where the vectors are divided into several subvectors, and each is vector quantized. Matrix quantization can also be used to exploit the correlation of these subvectors across consecutive time frames. *Transparent quality*, defined as average spectral distortion below 1 dB with no frames above 4 dB, can be achieved with fewer than 25 bits per frame.

A frame typically contains around 20 to 30 milliseconds, which at 8 kHz represents 160–240 samples. Because of the large vector dimension, it is impractical to quantize a whole frame with a single codebook. To reduce the dimensionality, the frame is divided into four or more nonoverlapping sub-frames. The LSF coefficients for each subframe are linearly interpolated between the two neighboring frames.

The typical range of the pitch prediction for an 8-kHz sampling rate goes from 54 to 400 Hz, from 20 to 147 samples, and from 2.5 ms to 18.375 ms, which can be encoded with 7 bits. An additional bit is often used to encode fractional delays for the lower pitch periods. These fractional delays can be implemented through upsampling as described in Chapter 5. The subframe gain of the adaptive codebook can be effectively encoded with 3 or 4 bits. Alternatively, the gains of all sub-frames within a frame can be encoded through VQ, resulting in more efficient compression.

The fixed codebook can be trained from data using the techniques described in Chapter 4. This will offer the lowest distortion for the training set but doesn't guarantee low distortion for mismatched test signals. Also, it requires additional storage, and full search increases computation substantially.

Since subframes should be approximately white, the codebook can be populated from samples of a white process. A way of reducing computation is to let those noise samples be only +1, 0, or –1, because only additions are required. Codebooks of a specific type, known as *algebraic codebooks* [1], offer even more computational savings because they contain

Amazon/VB Assets
Exhibit 1011
Page 384

many 0s. Locations for the 4 pulses per subframe under the G.729 standard are shown in Table 7.3.

Full search can efficiently be done with this codebook structure. Algebraic codebooks can provide almost as low distortion as trained codebooks can, with low computational complexity.

Table 7.3 Algebraic codebooks for the G.729 standard. Each of the four codebooks has one pulse in one possible location indicated by 3 bits for the first three codebooks and 4 bits for the last codebook. The sign is indicated by an additional bit. A total of 17 bits are needed to encode a 40-sample subframe.

Amplitude	Positions
±1	0, 5, 10, 15, 20, 25, 30, 35
±1	1, 6, 11, 16, 21, 26, 31, 36
±1	2, 7, 12, 17, 22, 27, 32, 37
±1	3, 8, 13, 18, 23, 28, 33, 38
	4, 9, 14, 19, 24, 29, 34, 39

7.4.6. CELP Standards

There are many standards for speech coding based on CELP, offering various points in the bit-rate/quality plane, mostly depending on when they were created and how refined the technology was at that time.

Voice over Internet Protocol (Voice over IP) consists of transmission of voice through data networks such as the Internet. H.323 is an umbrella standard which references many other ITU-T recommendations. H.323 provides the system and component descriptions, call model descriptions, and call signaling procedures. For audio coding, G.711 is mandatory, while G.722, G.728, G.723.1, and G.729 are optional. G.728 is a low-delay CELP coder that offers toll quality at 16 kbps [9], using a feedback 50^{th}-order predictor, but no pitch prediction. G.729 [46] offers toll quality at 8 kbps, with a delay of 10 ms. G.723.1, developed by DSP Group, including Audiocodes Ltd., France Telecom, and the University of Sherbrooke, has slightly lower quality at 5.3 and 6.3 kbps, but with a delay of 30 ms. These standards are shown in Table 7.4.

Table 7.4 Several CELP standards used in the H.323 specification used for teleconferencing and voice streaming through the Internet.

Standard	Bit Rate (kbps)	MOS	Algorithm	H.323	Comments
G.728	16	4.0	No pitch prediction	Optional	Low-delay
G.729	8	3.9	ACELP	Optional	
G.723.1	5.3, 6.3	3.9	ACELP for 5.3k	Optional	

Amazon/VB Assets
Exhibit 1011
Page 385

In 1982, the Conference of European Posts and Telegraphs (CEPT) formed a study group called the Groupe Spécial Mobile (GSM) to study and develop a pan-European public land mobile system. In 1989, GSM responsibility was transferred to the European Telecommunication Standards Institute (ETSI), and the phase I GSM specifications were published in 1990. Commercial service was started in mid 1991, and by 1993 there were 36 GSM networks in 22 countries, with 25 additional countries considering or having already selected GSM. This is not only a European standard; South Africa, Australia, and many Middle and Far East countries have chosen GSM. The acronym GSM now stands for Global System for Mobile telecommunications. The GSM group studied several voice coding algorithms on the basis of subjective speech quality and complexity (which is related to cost, processing delay, and power consumption once implemented) before arriving at the choice of a Regular Pulse Excited–Linear Predictive Coder (RPE-LPC) with a Long Term Predictor loop [56]. Neither the original full-rate at 13 kbps [56] nor the half-rate at 5.6 kbps [19] achieves toll quality, though the enhanced full-rate (EFR) standard based on ACELP [26] has toll quality at the same rates.

The *Telecommunication Industry Association* (TIA) and the *Electronic Industries Alliance* (EIA) are organizations accredited by the *American National Standards Institute* (ANSI) to develop voluntary industry standards for a wide variety of telecommunication products. TR-45 is the working group within TIA devoted to mobile and personal communication systems. Time Division Multiple Access (TDMA) is a digital wireless technology that divides a narrow radio channel into framed time slots (typically 3 or 8) and allocates a slot to each user. The TDMA Interim Standard 54, or TIA/EIA/IS54, was released in early 1991 by both TIA and EIA. It is available in North America at both the 800-MHz and 1900-MHz bands. IS54 [18] at 7.95 kbps is used in North America's TDMA (Time Division Multiple Access) digital telephony and has quality similar to the original full-rate GSM. TDMA IS-136 is an update released in 1994.

Table 7.5 CELP standards used in cellular telephony.

Standard	Bit Rate (kbps)	MOS	Algorithm	Cellular	Comments
Full-rate GSM	13	3.6	VSELP RTE-LTP	GSM	
EFR GSM	12.2	4.5	ACELP	GSM	
IS-641	7.4	4.1	ACELP	PCS1900	
IS-54	7.95	3.9	VSELP	TDMA	
IS-96a	max 8.5	3.9	QCELP	CDMA	Variable-rate

Code Division Multiple Access (CDMA) is a form of *spread spectrum*, a family of digital communication techniques that have been used in military applications for many years. The core principle is the use of noiselike carrier waves, and, as the name implies,

Amazon/VB Assets
Exhibit 1011
Page 386

bandwidths much wider than that required for simple point-to-point communication at the same data rate. Originally there were two motivations: either to resist enemy efforts to jam the communications (anti-jam, or AJ) or to hide the fact that communication was even taking place, sometimes called low probability of intercept (LPI). The service started in 1996 in the United States, and by the end of 1999 there were 50 million subscribers worldwide. IS-96 QCELP [14], used in North America's CDMA, offers variable-rate coding at 8.5, 4, 2, and 0.8 kbps. The lower bit rate is transmitted when the coder detects background noise. TIA/EIA/IS-127-2 is a standard for an enhanced variable-rate codec, whereas TIA/EIA/IS-733-1 is a standard for high-rate. Standards for CDMA, TDMA, and GSM are shown in Table 7.5.

Third generation (3G) is the generic term used for the next generation of mobile communications systems. 3G systems will provide enhanced services to those—such as voice, text, and data—predominantly available today. The Universal Mobile Telecommunications System (UMTS) is a part of ITU's International Mobile Telecommunications (IMT)-2000 vision of a global family of third-generation mobile communications systems. It has been assigned to the frequency bands 1885–2025 and 2110–2200 MHz. The first networks are planned to launch in Japan in 2001, with European countries following in early 2002. A major part of 3G is General Packet Radio Service (GPRS), under which carriers charge by the packet rather than by the minute. The speech coding standard for CDMA2000, the umbrella name for the third-generation standard in the United States, gained approval for its first phase in 2000. An adaptive multi-rate wideband speech codec has also been proposed for the GSM's 3G [16], which has five modes of operation from 24 kbps down to 9.1 kbps.

While most of the work described above uses a sampling rate of 8 kHz, there has been growing interest in using CELP techniques for high bandwidth and particularly in a scalable way so that a basic layer contains the lower frequency and the higher layer either is a full-band codec [33] or uses a parametric model [37].

7.5. LOW-BIT RATE SPEECH CODERS

In this section we describe a number of low-bit-rate speech coding techniques including the mixed-excitation LPC vocoder, harmonic coding, and waveform interpolation. These coding techniques are also used extensively in speech synthesis.

Waveform-approximating coders are designed to minimize the difference between the original signal and the coded signal. Therefore, they produce a reconstructed signal whose SNR goes to infinity as the bit rate increases, and they also behave well when the input signal is noisy or music. In this category we have the scalar waveform coders of Section 7.2, the frequency-domain coders of Section 7.3, and the CELP coders of Section 7.4.

Low-bit-rate coders, on the other hand, do not attempt to minimize the difference between the original signal and the quantized signal. Since these coders are designed to operate at low bit rates, their SNR does not generally approach infinity even if a large bit rate is used. The objective is to compress the original signal with another one that is perceptually equivalent. Because of the reliance on an inaccurate model, these low-bit-rate coders often

Amazon/VB Assets
Exhibit 1011
Page 387

distort the speech signal even if the parameters are not quantized. In this case, the distortion can consist of more than quantization noise. Furthermore, these coders are more sensitive to the presence of noise in the signal, and they do not perform as well on music.

In Figure 7.9 we compare the MOS of waveform approximating coders and low-bit-rate coders as a function of the bit rate. CELP uses a model of speech to obtain as much prediction as possible, yet allows for the model not to be exact, and thus is a waveform-approximating coder. CELP is a robust coder that works reasonably well when the assumption of only a clean speech signal breaks either because of additive noise or because there is music in the background. Researchers are working on the challenging problem of creating more scalable coders that offer best performance at all bit rates.

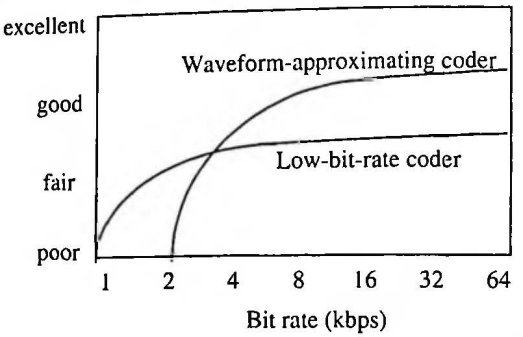


Figure 7.9 Typical subjective performance of waveform-approximating and low-bit-rate coders as a function of the bit rate. Note that waveform-approximating coders are a better choice for bit rates higher than about 3 kbps, whereas parametric coders are a better choice for lower bit rates. The exact cutoff point depends on the specific algorithms compared.

7.5.1. Mixed-Excitation LPC Vocoder

The main weakness of the LPC vocoder is the binary decision between voiced and unvoiced speech, which results in errors especially for noisy speech and voiced fricatives. By having a separate voicing decision for each of a number of frequency bands, the performance can be enhanced significantly [38]. The new proposed U.S. Federal Standard at 2.4 kbps is a Mixed Excitation Linear Prediction (MELP) LPC vocoder [39], which has a MOS of about 3.3. This exceeds the quality of the older 4800-bps Federal Standard 1016 [8] based on CELP. The bit rate of the proposed standard can be reduced while maintaining the same quality by jointly quantizing several frames together [57]. A hybrid codec that uses MELP in strongly voiced regions and CELP in weakly voiced and unvoiced regions [53] has shown to yield lower bit rates. MELP can also be combined with the waveform interpolation technique of Section 7.5.3 [50].

Amazon/VB Assets
Exhibit 1011
Page 388

7.5.2. Harmonic Coding

Sinusoidal coding decomposes the speech signal [35] or the LP residual signal [48] into a sum of sinusoids. The case where these sinusoids are harmonically related is of special interest for speech synthesis (see Chapter 16), so we will concentrate on it in this section, even though a similar treatment can be followed for the case where the sinusoids are not harmonically related. In fact, a combination of harmonically related and nonharmonically related sinusoids can also be used [17]. We show in Section 7.5.2.2 that we don't need to transmit the phase of the sinusoids, only the magnitude.

As shown in Chapter 5, a periodic signal $\tilde{s}[n]$ with period T_0 can be expressed as a sum of T_0 harmonic sinusoids

$$\tilde{s}[n] = \sum_{l=0}^{T_0-1} A_l \cos(nl\omega_0 + \phi_l) \tag{7.61}$$

whose frequencies are multiples of the fundamental frequency $\omega_0 = 2\pi / T_0$, and where A_l and ϕ_l are the sinusoid amplitudes and phases, respectively. If the pitch period T_0 has fractional samples, the sum in Eq. (7.61) includes only the integer part of T_0 in the summation. Since a real signal $s[n]$ will not be perfectly periodic in general, we have a modeling error

$$e[n] = s[n] - \tilde{s}[n] \tag{7.62}$$

We can use short-term analysis to estimate these parameters from the input signal $s[n]$ at frame k, in the neighborhood of $t = kN$, where N is the frame shift:

$$s_k[n] = s[n]w_k[n] = s[n]w[kN - n] \tag{7.63}$$

if we make the assumption that the sinusoid parameters for frame k (ω_0^k, A_l^k and ϕ_l^k) are constant within the frame.

At resynthesis time, there will be discontinuities at unit boundaries, due to the block processing, unless we specifically smooth the parameters over time. One way of doing this is with overlap-add method between frames $(k-1)$ and k:

$$\hat{s}[n] = w[n]\tilde{s}_{k-1}[n] + w[n-N]\tilde{s}_k[n-N], \quad 0 \le n < N \tag{7.64}$$

where the window $w[n]$ must be such that

$$w[n] + w[n-N] = 1, \quad 0 \le n < N \tag{7.65}$$

to achieve perfect reconstruction. This is the case for the common Hamming and Hanning windows.

This harmonic model [35] is similar to the classic filterbank, though rather than the whole spectrum we transmit only the fundamental frequency ω_0 and the amplitudes A_l and phases ϕ_l of the harmonics. This reduced representation doesn't result in loss of quality for a frame shift N that corresponds to 12 ms or less. For unvoiced speech, using a default pitch of 100 Hz results in acceptable quality.

Amazon/VB Assets
Exhibit 1011
Page 389

7.5.2.1. Parameter Estimation

For simplicity in the calculations, let's define $\tilde{s}[n]$ as a sum of complex exponentials

$$\tilde{s}[n] = \sum_{l=0}^{T_0-1} A_l \exp\{j(nl\omega_0 + \phi_l)\} \tag{7.66}$$

and perform short-time Fourier transform with a window $w[n]$

$$\tilde{S}_W(\omega) = \sum_{l=0}^{T_0-1} A_l e^{j\phi_l} W(\omega - l\omega_0) \tag{7.67}$$

where $W(\omega)$ is the Fourier transform of the window function. The goal is to estimate the sinusoid parameters as those that minimize the squared error:

$$E = |S(\omega) - \tilde{S}_W(\omega)|^2 \tag{7.68}$$

If the main lobes of the analysis window do not overlap, we can estimate the phases ϕ_l as

$$\phi_l = \arg S(l\omega_0) \tag{7.69}$$

and the amplitudes A_l as

$$A_l = \frac{|S(l\omega_0)|}{W(0)} \tag{7.70}$$

For example, the Fourier transform of a $(2N + 1)$ point rectangular window centered around the origin is given by

$$W(\omega) = \frac{\sin((2N+1)\omega/2)}{\sin(\omega/2)} \tag{7.71}$$

whose main lobes will not overlap in Eq. (7.67) if $2T_0 < 2N+1$: i.e., the window contains at least two pitch periods. The implicit assumption in the estimates of Eqs. (7.69) and (7.70) is that there is no spectral leakage, but a rectangular window does have significant spectral leakage, so a different window is often used in practice. For windows such as Hanning or Hamming, which reduce the leakage significantly, it has been found experimentally that these estimates are acceptable if the window contains at least two and a half pitch periods.

Typically, the window is centered around 0 (nonzero in the interval $-N \le n \le N$) to avoid numerical errors in estimating the phases.

Another implicit assumption in Eqs. (7.69) and (7.70) is that we know the fundamental frequency ω_0 ahead of time. Since, in practice, this is not the case, we can estimate it as the one which minimizes Eq. (7.68). This pitch-estimation method can generate pitch doubling or tripling when a harmonic falls within a formant that accounts for the majority of the signal's energy.

Amazon/VB Assets
Exhibit 1011
Page 390

Voiced/unvoiced decisions can be computed from the ratio between the energy of the signal and that of the reconstruction error

$$SNR = \frac{\sum_{n=-N}^{N} |s[n]|^2}{\sum_{n=-N}^{N} |s[n] - \tilde{s}[n]|^2} \tag{7.72}$$

where it has been empirically found that frames with SNR higher than 13 dB are generally voiced and lower than 4 dB unvoiced. In between, the signal is considered to contain a mixed excitation. Since speech is not perfectly stationary within the analysis frame, even noise-free periodic signals will yield finite SNR.

For unvoiced speech, a good assumption is to default to a pitch of 100 Hz. The use of fewer sinusoids leads to perceptual artifacts.

Improved quality can be achieved by using an analysis-by-synthesis framework [17, 34] since the closed-loop estimation is more robust to pitch-estimation and voicing decision errors.

7.5.2.2. Phase Modeling

An impulse train $e[n]$, a periodic excitation, can be expressed as a sum of complex exponentials

$$e[n] = T_0 \sum_{k=-\infty}^{\infty} \delta[n - n_0 - kT_0] = \sum_{l=0}^{T_0-1} e^{j(n-n_0)\omega_0 l} \tag{7.73}$$

which, if passed through a filter $H(\omega) = A(\omega)\exp\Phi(\omega)$, will generate

$$s[n] = \sum_{l=0}^{T_0-1} A(l\omega_0)\exp\{j[(n-n_0)\omega_0 l + \Phi(l\omega_0)]\} \tag{7.74}$$

Comparing Eq. (7.66) with (7.74), the phases of our sinusoidal model are given by

$$\phi_l = -n_0\omega_0 l + \Phi(l\omega_0) \tag{7.75}$$

Since the sinusoidal model has too many parameters to lead to low-rate coding, a common technique is to not encode the phases. In Chapter 6 we show that if a system is considered minimum phase, the phases can be uniquely recovered from knowledge of the magnitude spectrum.

The magnitude spectrum is known at the pitch harmonics, and the remaining values can be filled in by interpolation: e.g., linear or cubic splines [36]. This interpolated magnitude spectrum can be approximated through the real cepstrum:

$$|\tilde{A}(\omega)| = c_0 + 2\sum_{k=1}^{K} c_k \cos(k\omega) \tag{7.76}$$

Amazon/VB Assets
Exhibit 1011
Page 391

and the phase, assuming a minimum phase system, is given by

$$\tilde{\Phi}(\omega) = -2\sum_{k=1}^{K} c_k \sin(k\omega) \tag{7.77}$$

The phase $\phi_0(t)$ of the first harmonic between frames $(k-1)$ and k can be obtained from the instantaneous frequency $\omega_0(t)$

$$\phi_0(t) = \phi_0((k-1)N) + \int_{(k-1)N}^{t} \omega_0(t)dt \tag{7.78}$$

if we assume the frequency $\omega_0(t)$ in that region to vary linearly between frames $(k-1)$ and k:

$$\omega_0(t) = \omega_0^{k-1} + \frac{\omega_0^k - \omega_0^{k-1}}{N} t \tag{7.79}$$

and insert Eq. (7.79) into (7.78), evaluating at $t = kN$, to obtain

$$\phi_0^k = \phi_0(kN) = \phi_0((k-1)N) + (\omega_0^{k-1} + \omega_0^k)(N/2) \tag{7.80}$$

the phase of the sinusoid at ω_0 as a function of the fundamental frequencies at frames $(k-1)$, k and the phase at frame $(k-1)$:

$$\phi_l^k = \Phi^k(l\omega_0) + l\phi_0^k \tag{7.81}$$

The phases computed by Eqs. (7.80) and (7.81) are a good approximation in practice for perfectly voiced sounds. For unvoiced sounds, random phases are needed, or else the reconstructed speech sounds buzzy. Voiced fricatives and many voiced sounds have an aspiration component, so that a mixed excitation is needed to represent them. In these cases, the source is split into different frequency bands and each band is classified as either voiced or unvoiced. Sinusoids in voiced bands use the phases described above, whereas sinusoids in unvoiced bands have random phases.

7.5.2.3. Parameter Quantization

To quantize the sinusoid amplitudes, we can use an LPC fitting and then quantize the line spectral frequencies. Also we can do a cepstral fit and quantize the cepstral coefficients. To be more effective, a mel scale should be used.

While these approaches help in reducing the number of parameters and in quantizing those parameters, they are not the most effective way of quantizing the sinusoid amplitudes. A technique called *Variable-Dimension Vector Quantization* (VDVQ) [12] has been devised to address this. Each codebook vector $\mathbf{c}_i$ has a fixed dimension N determined by the length of the FFT used. The vector of sinusoid amplitudes $\mathbf{A}$ has a dimension l that depends on the number of harmonics and thus the pitch of the current frame. To compute the distance between $\mathbf{A}$ and $\mathbf{c}_i$, the codebook vectors are resampled to a size l and the distance is computed between two vectors of dimension l. Euclidean distance of the log-amplitudes is often used.

Amazon/VB Assets
Exhibit 1011
Page 392

In this method, only the distance at the harmonics is evaluated instead of the distance at the points in the envelope that are not actually present in the signal. Also, this technique does not suffer from inaccuracies of the model used, such as the inability of linear predictive coding to model nasals.

7.5.3. Waveform Interpolation

The main idea behind waveform interpolation (WI) [29] is that the pitch pulse changes slowly over time for voiced speech. During voiced segments, the speech signal is nearly periodic. WI coders can operate as low as 2.4 kbps.

Starting at an arbitrary time instant, it is easy to identify a first pitch cycle $x_1[n]$, a second $x_2[n]$, a third $x_3[n]$, and so on. We then express our signal $x[n]$ as a function of these pitch cycle waveforms $x_m[n]$

$$x[n] = \sum_{m=-\infty}^{\infty} x_m[n - t_m] \tag{7.82}$$

where $P_m = t_m - t_{m-1}$ is the pitch period at time t_m in samples, and the pitch cycle is a windowed version of the input

$$x_m[n] = w_m[n]x[n] \tag{7.83}$$

for example, with a rectangular window. To transmit the signal in a lossless fashion we need to transmit all pitch waveforms $x_m[n]$.

If the signal is perfectly periodic, we need to transmit only one pitch waveform $x_m[n]$ and the pitch period P. In practice, voiced signals are not perfectly periodic, so that we need to transmit more than just one pitch waveform. On the other hand, voiced speech is nearly periodic, and consecutive pitch waveforms are very similar. Thus, we probably do not need to transmit all, and we could send every other pitch waveform, for example.

It is convenient to define a two-dimensional surface $u[n,l]$ (shown in Figure 7.10) such that the pitch waveform $x_m[n]$ can be obtained as

$$x_m[n] = u[n, t_m] \tag{7.84}$$

so that $u[n,l]$ is defined for $l = t_m$, with the remaining points having been computed through interpolation. A frequency representation of the pitch cycle can also be used instead of the time pitch cycle.

This surface can then be sampled at regular time intervals $l = sT$. It has been shown empirically that transmitting the pitch waveform $x_i[n]$ about 40 times per second (a 25-ms interval is equivalent to $T = 200$ samples for an $F_s = 8000$ Hz sampling rate) is sufficient for voiced speech. The so-called *slowly evolving waveform* (SEW) $\tilde{u}[n,l]$ can be generated by low-pass filtering $u[n,l]$ along the l-axis:

$$x_i[n] = \tilde{u}[n, sT] = \frac{\sum_m h[sT - t_m]u[n, t_m]}{\sum_m h[sT - t_m]} \tag{7.85}$$

Amazon/VB Assets
Exhibit 1011
Page 393

where $h[n]$ is a low-pass filter and $x_s[n]$ is a sampled version of $\tilde{u}[n,l]$.

The decoder has to reconstruct each pitch waveform $x_m[n]$ from the SEW $x_s[n]$ by interpolation between adjacent pitch waveforms, and thus the name *waveform interpolation (WI) coding*:

$$\tilde{x}_m[n] = \tilde{u}[n,t_m] = \frac{\sum_s h[t_m - sT]x_s[n]}{\sum_s h[t_m - sT]} \tag{7.86}$$

If the sampling period is larger than the local pitch period ($T > P_m$), perfect reconstruction will not be possible, and there will be some error in the approximation

$$x_m[n] = \tilde{x}_m[n] + \hat{x}_m[n] \tag{7.87}$$

or alternatively in the two-dimensional representation

$$u[n,l] = \tilde{u}[n,l] + \hat{u}[n,l] \tag{7.88}$$

where $\hat{x}_m[n]$ and $\hat{u}[n,l]$ represent the *rapidly evolving waveforms* (REW).

Since this technique can also be applied to unvoiced speech, where the concept of pitch waveform doesn't make sense, the more general term *characteristic waveform* is used instead. For unvoiced speech, an arbitrary *period* of around 100 Hz can be used.

For voiced speech, we expect the rapidly varying waveform $\hat{u}[n,l]$ in Eq. (7.88) to have much less energy than the slowly evolving waveform $\tilde{u}[n,l]$. For unvoiced speech the converse is true: $\hat{u}[n,l]$ has more energy than $\tilde{u}[n,l]$. For voiced fricatives, both components may be comparable and thus we want to transmit both.

In Eqs. (7.85) and (7.86) we need to average characteristic waveforms that have, in general, different lengths. To handle this, all characteristic waveforms are typically normalized in length prior to the averaging operation. This length normalization is done by padding with zeros $x_m[n]$ to a certain length M, or truncating $x_m[n]$ if $P_m > M$. Another possible normalization is done via linear resampling. This decomposition is shown in Figure 7.10.

Another representation uses the Fourier transform of $x_m[n]$. This case is related to the harmonic model of Section 7.5.2. In the harmonic model, a relatively long window is needed to average the several pitch waveforms within the window, whereas this waveform interpolation method has higher time resolution. In constructing the characteristic waveforms we have implicitly used a rectangular window of length one pitch period, but other windows can be used, such as a Hanning window that covers two pitch periods. This frequency-domain representation offers advantages in coding both the SEW and the REW, because properties of the human auditory system can help reduce the bit rate. This decomposition is often done on the LPC residual signal.

In particular, the REW $\hat{u}[n,l]$ has the characteristics for noise, and as such only a rough description of its power spectral density is needed. At the decoder, random noise is generated with the transmitted power spectrum. The spectrum of $\hat{u}[n,l]$ can be vector quantized to as few as eight shapes with little or no degradation.

Amazon/VB Assets
Exhibit 1011
Page 394

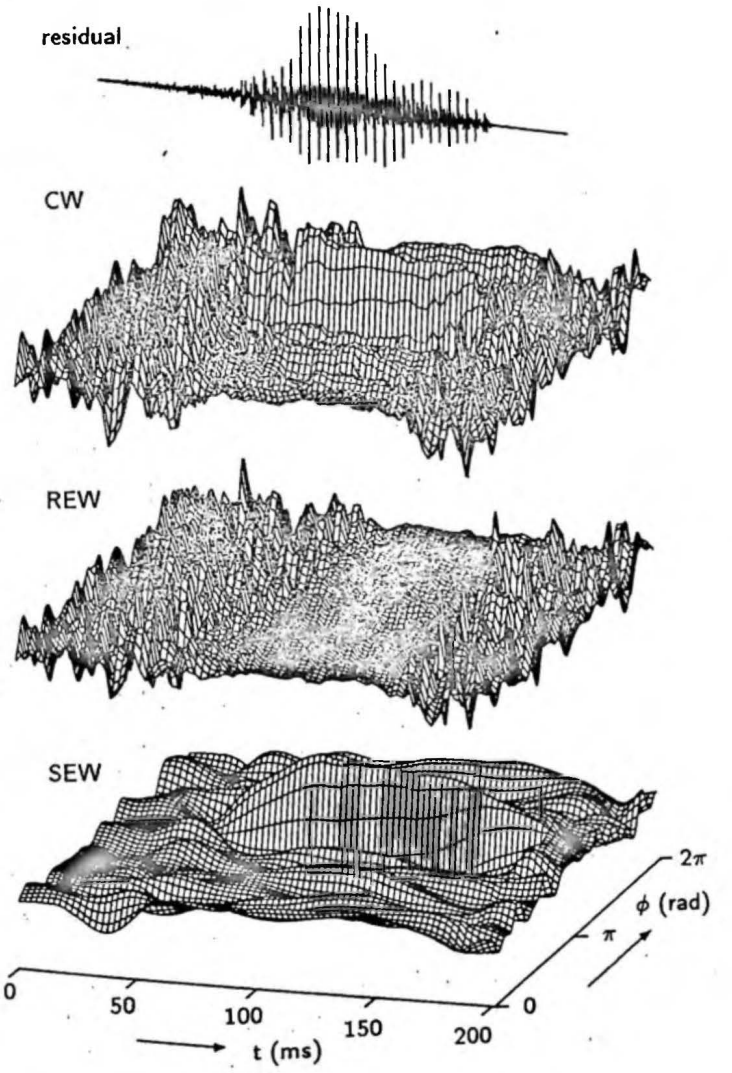


Figure 7.10 LP residual signal and its associated characteristic waveform (CW) $u(t,\phi)$. In the ϕ axis we have a normalized pitch pulse at every given time t. Decomposition of the surface into a slowly evolving waveform (SEW) and a rapidly evolving waveform (REW). (After Kleijn and Haagen [30], reprinted by permission of IEEE).

Amazon/VB Assets
Exhibit 1011
Page 395

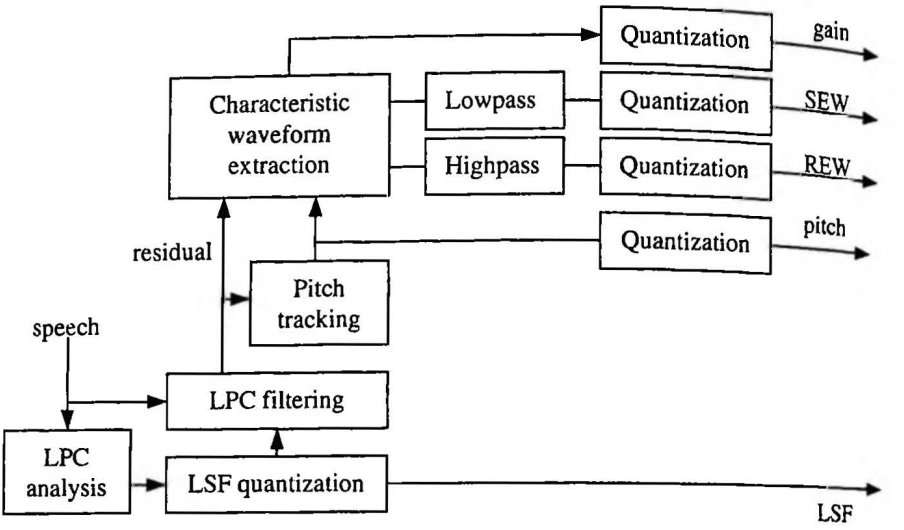


Figure 7.11 Block diagram of the WI encoder.

The SEW $\tilde{u}[n,l]$ is more important perceptually, and for high quality the whole shape needs to be transmitted. Higher accuracy is desired at lower frequencies so that a perceptual frequency scale (mel or Bark) is often used. Since the magnitude of $\tilde{u}[n,l]$ is perceptually more important than the phase, for low bit rates the phase of the SEW is not transmitted. The magnitude spectrum can be quantized with the VDVQ described in Section 7.5.2.3.

To obtain the characteristic waveforms, the pitch needs to be computed. We can find the pitch period such that the energy of the REW is minimized. To do this we use the approaches described in Chapter 6. Figure 7.11 shows a block diagram of the encoder and Figure 7.12 of the decoder.

Parameter estimation using an analysis-by-synthesis framework [21] can yield better results than the open-loop estimation described above.

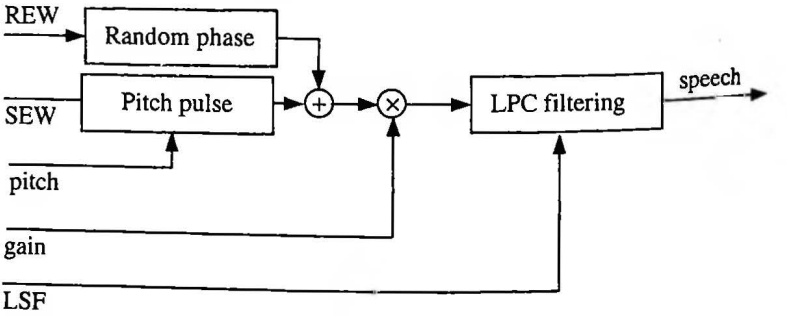


Figure 7.12 Block diagram of the WI decoder.

Amazon/VB Assets
Exhibit 1011
Page 396

7.6. HISTORICAL PERSPECTIVE AND FURTHER READING

This chapter is only an introduction to speech and audio coding technologies. The reader is referred to [27, 32, 41, 52] for coverage in greater depth. A good source of the history of speech coding can be found in [20].

In 1939, Homer Dudley of AT&T Bell Labs first proposed the channel vocoder [15], the first analysis-by-synthesis system. This vocoder analyzed slowly varying parameters for both the excitation and the spectral envelope. Dudley thought of the advantages of bandwidth compression and information encryption long before the advent of digital communications.

PCM was first conceived in 1937 by Alex Reeves at the Paris Laboratories of AT&T, and it started to be deployed in the United States Public Switched Telephone Network in 1962. The digital compact disc, invented in the late 1960s by James T. Russell and introduced commercially in 1984, also uses PCM as coding standard. The use of μ-law encoding was proposed by Smith [51] in 1957, but it wasn't standardized for telephone networks (G.711) until 1972. In 1952, Schouten et al. [47] proposed delta modulation and Cutler [11] invented differential PCM. ADPCM was developed by Barnwell [6] in 1974.

Speech coding underwent a fundamental change with the development of linear predictive coding in the early 1970s. Atal [3] proposed the LPC vocoder in 1971, and then CELP [5] in 1984. The majority of coding standards for speech signals today use a variation on CELP.

Sinusoidal coding [35] and waveform interpolation [29] were developed in 1986 and 1991, respectively, for low-bit-rate telephone speech. Transform coders such as MP3 [23], MPEG II, and Perceptual Audio Coder (PAC) [28] have been used primarily in audio coding for high-fidelity applications.

Recently, researchers have been improving the technology for cellular communications by trading off source coding and channel coding. For poor channels more bits are allocated to channel coding and fewer to source coding to reduce dropped calls. Scalable coders that have different layers with increased level of precision, or bandwidth, are also of great interest.

REFERENCES

[1] Adoul, J.P., *et al.*, "Fast CELP Coding Based on Algebraic Codes," *Int. Conf. on Acoustics, Speech and Signal Processing*, 1987, Dallas, TX, pp. 1957-1960.

[2] Atal, B.S., R.V. Cox, and P. Kroon, "Spectral Quantization and Interpolation for CELP Coders," *Int. Conf. on Acoustics, Speech and Signal Processing*, 1989, Glasgow, pp. 69-72.

[3] Atal, B.S. and L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," *Journal of the Acoustical Society of America*, 1971, **50**, pp. 637-655.

[4] Atal, B.S. and M.R. Schroeder, "Predictive Coding of Speech Signals and Subjective Error Criteria," *IEEE Trans. on Acoustics, Speech and Signal Processing*, 1979, **ASSP-27**(3), pp. 247-254.

Amazon/VB Assets
Exhibit 1011
Page 397

[5] Atal, B.S. and M.R. Schroeder, "Stochastic Coding of Speech at Very Low Bit Rates," *Proc. Int. Conf. on Comm.*, 1984, Amsterdam, pp. 1610-1613.

[6] Barnwell, T.P., *et al.*, *Adaptive Differential PCM Speech Transmission*, 1974, Rome Air Development Center.

[7] Benvenuto, N., G. Bertocci, and W.R. Daumer, "The 32-kbps ADPCM Coding Standard," *AT&T Technical Journal*, 1986, **65**, pp. 12-22.

[8] Campbell, J.P., T.E. Tremain, and V.C. Welch, "The DoD 4.8 kbps Standard (Proposed Federal Standard 1016)" in *Advances in Speech Coding*, B. Atal, V. Cuperman, and A. Gersho, eds. 1991, Kluwer Academic Publishers, pp. 121-133.

[9] Chen, J.H., *et al.*, "A Low-Delay CELP Coder for the CCITT 16 kbps Speech Coding Standard," *IEEE Journal on Selected Areas Communcations*, 1992, **10**(5), pp. 830-849.

[10] Chen, J.H. and A. Gersho, "Adaptive Postfiltering for Quality Enhancement of Coded Speech," *IEEE Trans. on Speech and Audio Processing*, 1995, **3**(1), pp. 59-71.

[11] Cutler, C.C., *Differential Quantization for Communication Signals*, 1952, US Patent 2,605,361.

[12] Das, A. and A. Gersho, "Variable Dimension Vector Quantization," *IEEE Signal Processing Letters*, 1996, **3**(7), pp. 200-202.

[13] Daumer, W.R., *et al.*, "Overview of the 32kbps ADPCM Algorithm," *Proc. IEEE Global Telecomm*, 1984, pp. 774-777.

[14] DeJaco, P.J.A., W. Gardner, and C. Lee, "QCELP: The North American CDMA Digital Cellular Variable Speech Coding Standard," *Proc. Workshop on Speech Coding for Telecommunications*, 1993, Sainte Adele, Quebec, pp. 5-6.

[15] Dudley, H., "The Vocoder," *Bell Labs Record*, 1939, **17**, pp. 122-126.

[16] Erdmann, C., *et al.*, "An Adaptive Rate Wideband Speech Codec with Adaptive Gain Re-Quantization," *IEEE Workshop on Speech Coding*, 2000, Delavan, Wisconsin.

[17] Etemoglu, C.O., V. Cuperman, and A. Gersho, "Speech Coding with an Analysis-by-Synthesis Sinusoidal Model," *Int. Conf. on Acoustics, Speech and Signal Processing*, 2000, Istanbul, Turkey, pp. 1371-1374.

[18] Gerson, I.A. and M.A. Jasiuk, "Vector Sum Excited Linear Prediction (VSELP)" in *Advances in Speech Coding*, B.S. Atal, V. Cuperman, and A. Gersho, eds. 1991, Boston, MA, Kluwer Academic Publishers, pp. 69-79.

[19] Gerson, I.A. and M.A. Jasiuk., "Techniques for Improving the Performance of CELP-type Speech Coders," *IEEE Journal Selected Areas Communications*, 1991, **10**(5), pp. 858-865.

[20] Gold, B. and N. Morgan, *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*, 2000, New York, John Wiley.

[21] Gottesman, O. and A. Gersho, "High Quality Enhanced Waveform Interpolative Coding at 2.8 kbps," *Int. Conf. on Acoustics, Speech and Signal Processing*, 2000, Istanbul, Turkey, pp. 1363-1366.

[22] Greefkes, J.A., "A Digitally Companded Delta Modulation Modem for Speech Transmission," *Proc. Int. Conf. on Communications*, 1970, pp. 7.33-7.48.

[23] ISO, *Coding of Moving Pictures and Associated Audio—Audio Part*, 1993. Int. Standards Organization.

[24] ISO/IEC, *Information Technology—Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbps, Part 3: Audio (MPEG-1)*, 1992, Int. Standards Organization.

Amazon/VB Assets
Exhibit 1011
Page 398

[25] ITU-T, *Methods for Subjective Determination of Transmission Quality*, 1996, Int. Telecommunication Unit.

[26] Jarvinen, K., *et al.*, "GSM Enhanced Full Rate Speech Codec," *Int. Conf. on Acoustics, Speech and Signal Processing*, 1997, Munich, Germany, pp. 771-774.

[27] Jayant, N.S. and P. Noll, *Digital Coding of Waveforms*, 1984. Upper Saddle River, NJ, Prentice Hall.

[28] Johnston, J.D., *et al.*, "ATT Perceptual Audio Coding (PAC)" in *Audio Engineering Society (AES) Collected Papers on Digital Audio Bit Rate Reduction*, N. Gilchrist and C. Grewin, eds. 1996, pp. 73-82.

[29] Kleijn, W.B., "Continuous Representations in Linear Predictive Coding," *Int. Conf. on Acoustics, Speech and Signal Processing*, 1991, Toronto, Canada, pp. 201-204.

[30] Kleijn, W.B. and J. Haagen, "Transformation and Decomposition of the Speech Signal for Coding," *IEEE Signal Processing Letters*, 1994, **1**, pp. 136-138.

[31] Kleijn, W.B., D.J. Krasinski, and R.H. Ketchum, "An Efficient Stochastically Excited Linear Predictive Coding Algorithm for High Quality Low Bit Rate Transmission of Speech," *Speech Communication*, 1988, **7**, pp. 305-316.

[32] Kleijn, W.B. and K.K. Paliwal, *Speech Coding and Synthesis*, 1995, Amsterdam, Netherlands, Elsevier.

[33] Koishida, K., V. Cuperman, and A. Gersho, "A 16-KBIT/S Bandwidth Scalable Audio Coder Based on the G.729 Standard," *Int. Conf. on Acoustics, Speech and Signal Processing*, 2000, Istanbul, Turkey, pp. 1149-1152.

[34] Li, C. and V. Cuperman, "Analysis-by-Synthesis Multimode Harmonic Speech Coding at 4 kbps," *Int. Conf. on Acoustics, Speech and Signal Processing*, 2000, Istanbul, Turkey, pp. 1367-1370.

[35] McAulay, R.J. and T.F. Quateri, "Speech Analysis/Synthesis Based on a Sinusoidal Representation," *IEEE Trans. on Acoustics, Speech and Signal Processing*, 1986, **34**, pp. 744-754.

[36] McAulay, R.J. and T.F. Quateri, "Sinusoidal Coding" in *Speech Coding and Synthesis*, W.B. Kleijn and K.K. Paliwal, eds. 1995, Elsevier, pp. 121-174.

[37] McCree, A., "A 14 kbps Wideband Speech Coder with a Parametric Highband Model," *Int. Conf. on Acoustics, Speech and Signal Processing*, 2000, Istanbul, Turkey, pp. 1153-1156.

[38] McCree, A.V. and T.P. Barnwell, "Improving the Performance of a Mixed-Excitation LPC Vocoder in Acoustic Noise," *Int. Conf. on Acoustics, Speech and Signal Processing*, 1992, San Francisco, pp. II-137-138.

[39] McCree, A.V., *et al.*, "A 2.4 kbit/s MELP Coder Candidate for the New U.S. Federal Standard," *Int. Conf. on Acoustics, Speech and Signal Processing*, 1996, Atlanta, GA, pp. 200-203.

[40] Paez, M.D. and T.H. Glisson, "Minimum Squared-Error Quantization in Speech," *IEEE Trans. on Comm*, 1972, **20**, pp. 225-230.

[41] Painter, T. and A. Spanias, "A Review of Algorithms for Perceptual Coding of Digital Audio Signals," *Proc. Int. Conf. on DSP*, 1997, pp. 179-205.

[42] Painter, T. and A. Spanias, "Perceptual Coding of Digital Audio," *Proc. of IEEE*, 2000(April), pp. 451-513.

[43] Paliwal, K.K. and B. Atal, "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame," *IEEE Trans. on Speech and Audio Processing*, 1993, **1**(1), pp. 3-14.

[44] Prevez, M.A., H.V. Sorensen, and J.V.D. Spiegel, "An Overview of Sigma-Delta Converters," *IEEE Signal Processing Magazine*, 1996, **13**(1), pp. 61-84.

Amazon/VB Assets
Exhibit 1011
Page 399

[45] Robinson, T., *Simple Lossless and Near-Lossless Waveform Compression*, 1994, Cambridge University Engineering Department.

[46] Salami, R., C. Laflamme, and B. Bessette, "Description of ITU-T Recommendation G.729 Annex A: Reduced Complexity 8 kbps CS-ACELP Codec," *Int. Conf. on Acoustics, Speech and Signal Processing*, 1997, Munich, Germany, pp. 775-778.

[47] Schouten, J.S., F.E. DeJager, and J.A. Greefkes, *Delta Modulation, a New Modulation System for Telecommunications*, 1952, Phillips, pp. 237-245.

[48] Shlomot, E., V. Cuperman, and A. Gersho, "Combined Harmonic and Waveform Coding of Speech at Low Bit Rates," *Int. Conf. on Acoustics, Speech and Signal Processing*, 1998, Seattle, WA, pp. 585-588.

[49] Singhal, S. and B.S. Atal, "Improving Performance of Multi-Pulse LPC Coders at Low Bit Rates," *Int. Conf. on Acoustics, Speech and Signal Processing*, 1984, San Diego, pp. 1.3.1-1.3.4.

[50] Skoglund, J., R. Cox, and J. Collura, "A Combined WI and MELP Coder at 5.2KBPS," *Int. Conf. on Acoustics, Speech and Signal Processing*, 2000, Istanbul, Turkey, pp. 1387-1390.

[51] Smith, B., "Instantaneous Companding of Quantized Signals," *Bell Systems Technical Journal*, 1957, **36**(3), pp. 653-709.

[52] Spanias, A.S., "Speech Coding: A Tutorial Review," *Proc. of the IEEE*, 1994, **82**(10), pp. 1441-1582.

[53] Stachurski, J. and A. McCree, "A 4 kbps Hybrid MELP/CELP Coder with Alignment Phase Encoding and Zero Phase Equalization," *Int. Conf. on Acoustics, Speech and Signal Processing*, 2000, Istanbul, Turkey, pp. 1379-1382.

[54] Todd, C., "AC-3: Flexible Perceptual Coding for Audio Transmission and Storage," *Audio Engineering Society 96th Convention*, 1994.

[55] Tremain, T.E., *The Government Standard Linear Predictive Coding Algorithm*, in *Speech Technology Magazine*, 1982, pp. 40-49.

[56] Vary, P., *et al.*, "A Regular-Pulse Excited Linear Predictive Code," *Speech Communication*, 1988, **7**(2), pp. 209-215.

[57] Wang, T., *et al.*, "A 1200 BPS Speech Coder Based on MELP," *Int. Conf. on Acoustics, Speech and Signal Processing*, 2000, Istanbul, Turkey, pp. 1375-1378.

Amazon/VB Assets
Exhibit 1011
Page 400

PART III

SPEECH RECOGNITION

CHAPTER 8

Hidden Markov Models

The hidden Markov model (HMM) is a very powerful statistical method of characterizing the observed data samples of a discrete-time series. Not only can it provide an efficient way to build parsimonious parametric models, but can also incorporate the dynamic programming principle in its core for a unified pattern segmentation and pattern classification of time-varying data sequences. The data samples in the time series can be discretely or continuously distributed; they can be scalars or vectors. The underlying assumption of the HMM is that the data samples can be well characterized as a parametric random process, and the parameters of the stochastic process can be estimated in a precise and well-defined framework. The basic HMM theory was published in a series of classic papers by Baum and his colleagues [4]. The HMM has become one of the most powerful statistical methods for modeling speech signals. Its principles have been successfully used in automatic speech recognition, formant and pitch tracking, speech enhancement, speech synthesis, statistical language modeling, part-of-speech tagging, spoken language understanding, and machine translation [3, 4, 8, 10, 12, 18, 23, 37].

Amazon/VB Assets
Exhibit 1011
Page 403

8.1. THE MARKOV CHAIN

A *Markov chain* models a class of random processes that incorporates a minimum amount of memory without being completely memoryless. In this subsection we focus on the discrete-time Markov chain only.

Let $\mathbf{X} = X_1, X_2, \ldots X_n$ be a sequence of random variables from a finite discrete alphabet $O = \{o_1, o_2, \ldots, o_M\}$. Based on the Bayes' rule, we have

$$P(X_1, X_2, \ldots, X_n) = P(X_1)\prod_{i=2}^{n} P(X_i \mid X_1^{i-1}) \tag{8.1}$$

where $X_1^{i-1} = X_1, X_2, \ldots, X_{i-1}$. The random variables X are said to form a first-order Markov chain, provided that

$$P(X_i | X_1^{i-1}) = P(X_i | X_{i-1}) \tag{8.2}$$

As a consequence, for the first-order Markov chain, Eq. (8.1) becomes

$$P(X_1, X_2, \ldots, X_n) = P(X_1)\prod_{i=2}^{n} P(X_i \mid X_{i-1}) \tag{8.3}$$

Equation (8.2) is also known as the *Markov assumption*. This assumption uses very little memory to model dynamic data sequences: the probability of the random variable at a given time depends only on the value at the preceding time. The Markov chain can be used to model time-invariant (stationary) events if we discard the time index i,

$$P(X_i = s | X_{i-1} = s') = P(s|s') \tag{8.4}$$

If we associate X_i to a state, the Markov chain can be represented by a finite state process with transition between states specified by the probability function $P(s|s')$. Using this finite state representation, the Markov assumption [Eq. (8.2)] is translated to the following: the probability that the Markov chain will be in a particular state at a given time depends only on the state of the Markov chain at the previous time.

Consider a Markov chain with N distinct states labeled by $\{1, \ldots, N\}$, with the state at time t in the Markov chain denoted as s_t; the parameters of a Markov chain can be described as follows:

$$a_{ij} = P(s_t = j \mid s_{t-1} = i) \quad 1 \le i, j \le N \tag{8.5}$$

$$\pi_i = P(s_1 = i) \quad 1 \le i \le N \tag{8.6}$$

where a_{ij} is the transition probability from state i to state j; and π_i is the initial probability that the Markov chain will start in state i. Both transition and initial probabilities are bound to the constraints:

Amazon/VB Assets
Exhibit 1011
Page 404

$$\sum_{j=1}^{N} a_{ij} = 1; \quad 1 \le i \le N$$
$$\sum_{j=1}^{N} \pi_j = 1 \tag{8.7}$$

The Markov chain described above is also called the observable Markov model because the output of the process is the set of states at each time instance t, where each state corresponds to an observable event X_t. In other words, there is one-to-one correspondence between the observable event sequence $\mathbf{X}$ and the Markov chain state sequence $\mathbf{S} = s_1, s_2, \ldots s_n$. Consider a simple three-state Markov chain for the Dow Jones Industrial average as shown in Figure 8.1. At the end of each day, the Dow Jones Industrial average may correspond to one of the following states:

state 1 – *up* (in comparison to the index of previous day)
state 2 – *down* (in comparison to the index of previous day)
state 3 – *unchanged* (in comparison to the index of previous day)

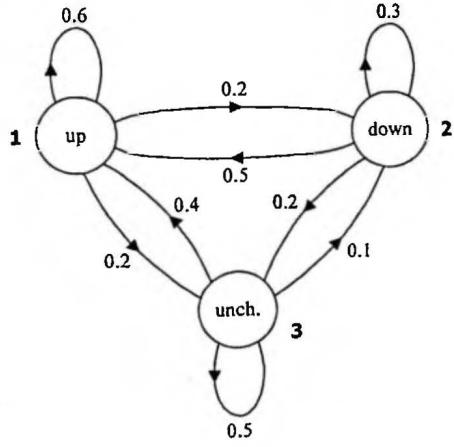


Figure 8.1 A Markov chain for the Dow Jones Industrial average. Three states represent *up*, *down*, and *unchanged*, respectively.

The parameter for this Dow Jones Markov chain may include a state-transition probability matrix

$$A = \{a_{ij}\} = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.5 & 0.3 & 0.2 \\ 0.4 & 0.1 & 0.5 \end{bmatrix}$$

and an initial state probability matrix

Amazon/VB Assets
Exhibit 1011
Page 405

$$\pi = (\pi_i)^t = \begin{pmatrix} 0.5 \\ 0.2 \\ 0.3 \end{pmatrix}$$

Suppose you would like to find out the probability for five consecutive *up* days. Since the observed sequence of *up-up-up-up-up* corresponds to the state sequence of (1, 1, 1, 1, 1), the probability will be

$$P(5 \text{ consecutive } up \text{ days}) = P(1,1,1,1,1)$$
$$= \pi_1 a_{11} a_{11} a_{11} a_{11} = 0.5 \times (0.6)^4 = 0.0648$$

8.2. DEFINITION OF THE HIDDEN MARKOV MODEL

In the Markov chain, each state corresponds to a deterministically observable event, i.e., the output of such sources in any given state is not random. A natural extension to the Markov chain introduces a non-deterministic process that generates output observation symbols in any given state. Thus, the observation is a probabilistic function of the state. This new model is known as a *hidden Markov model*, which can be viewed as a double-embedded stochastic process with an underlying stochastic process (the state sequence) not directly observable. This underlying process can only be probabilistically associated with another observable stochastic process producing the sequence of features we can observe.

A hidden Markov model is basically a Markov chain where the output observation is a random variable X generated according to a output probabilistic function associated with each state. Figure 8.2 shows a revised hidden Markov model for the Dow Jones Industrial average. You see that each state now can generate all three output observations: *up*, *down*, and *unchanged*, according to its output pdf. This means that there is no longer a one-to-one correspondence between the observation sequence and the state sequence, so you cannot unanimously determine the state sequence for a given observation sequence, i.e., the state sequence is not observable and therefore hidden. This is why the world *hidden* is placed in front of *Markov models*. Although the state of an HMM is hidden, it often contains salient information about the data we are modeling. For example, the first state in Figure 8.2 indicates a *bull* market, and the second state indicates a *bear* market as specified by the output probability in each state. Formally speaking, a hidden Markov model is defined by:

- $O = \{o_1, o_2, \ldots, o_M\}$—An output observation alphabet.[1] The observation symbols correspond to the physical output of the system being modeled. In the case of the Dow Jones Industrial average HMM, the output observation alphabet is the collection of three categories— $O = \{up, down, unchanged\}$.

[1] Although we use the discrete output observation here, we can extend it to the continuous case with a continuous pdf. You can also use vector quantization to map a continuous vector variable into a discrete alphabet set.

Amazon/VB Assets
Exhibit 1011
Page 406