One possible accent structure is indicated in boldface. Here *birthday party* functions as a complex nominal, with lexical stress on *birthday*. The word *party* should not receive stress at all, nor should its later stand-alone form *parties*. Accentuation of *improve* is optional: it is a full content word, yet somehow it also feels predictable from the context, so deaccentuation is possible. Some of the complex nominals, like *birthday party*, are fully fixed and can be entered into the lexicon as such. Others form small families of binary or *n*-ary phrases, which may be detected by local syntactic and lexical analysis. Ambiguous cases such as *moving van* or *hot dog*, which could be either nominals or adjective-noun phrases, may have to be resolved by user markup or text understanding processes.

Dwight Bolinger opined that *Accent is predictable—if you're a mind reader* [6], asserting that accentuation algorithms will never achieve perfect performance, because a writer's exact intentions cannot be inferred from text alone, and understanding is needed. However, work in [20] (similar to [3] but incorporating more sophisticated mechanics for name identification and memory of recent accented items), showed that reasonably straightforward procedures, if applied separately and combined intelligently, can yield adequate results on the accentuation task. This research has also determined that improvement occurs when the system learns that not all 'closed-class' categories are equally likely to be deaccented. For example, *closed accented* items include the negative article, negative modals, negative *do*, most nominal pronouns, most nominative and all reflexive pronouns, pre- and postqualifiers (e.g., quite), prequantifiers (e.g., *all*), postdeterminers (e.g., *next*), nominal adverbials (e.g., *here*), interjections, particles, most wh-words, plus some prepositions (e.g., *despite, unlike*).

One area of current and future development is the introduction of discourse analysis to synthesis of dialog. Discourse analysis algorithms attempt to delimit the time within which a given word/concept can be considered newly introduced, given, old, or reintroduced, and combined with analysis of segments within discourse and their boundary cues (turn-taking, digressions, interruptions, summarization, etc.) can supplement algorithms for accent assignment. This kind of work improves the naturalness of computer responses in human-computer dialog, as well as the accentuation in TTS renditions of pure text, when dialog must be performed (e.g., in reading a novel out loud) [44].

As noted above, user- or application-supplied annotations, based on intimate knowledge of the purpose and content of the speech event, can greatly enhance the quality by offloading the task of automatic accent prediction. The /EMPHASIS/ tag described in Section 15.7, with several levels of strength including *reduced accent* and *no accent*, is ideally suited for this purpose.

15.4.4. Tone

Tones can be understood as labels for perceptually salient levels or movements of F0 on syllables. Pitch levels and movements on accented and phrase-boundary syllables can exhibit a bewildering diversity, based on the speaker's characteristics, the nature of the speech event, and the utterance itself, as discussed above. For modeling purposes, it is useful to

Amazon/VB Assets
Exhibit 1011
Page 779

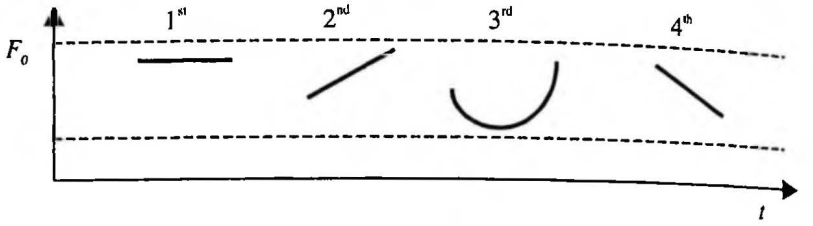


Figure 15.4 The four Chinese tones.

have an inventory of basic, abstract pitch types that could in principle serve as the base inventory for expression of linguistically significant contrasts. Chinese, a lexical tone language, is said to have an inventory of 4 lexical tones (5 if neutral tone is included), as shown in Figure 15.4. Different speakers can realize these tones differently according to their physiology, mood, utterance content, and the speech occasion. But the variance in the tones' shapes, and contrasts with one another, remain fairly predictable, within broad limits.

By analogy, linguists have proposed a relatively small set of tonal primitives for English, which can be used, in isolation or in combination, to specify the gross phonological typology of linguistically relevant contrasts found in theories of English intonational meaning [17, 28, 39]. A basic set of tonal contrasts has been codified for American English as part of the Tones and Break Indices (ToBI) system [4, 55]. These categories can be used for annotation of prosodic training data for machine learning, and also for internal modular control of F0 generation in a TTS system. The set specifies 2 abstract levels, H(igh) and L(ow), indicating a relatively higher or lower point in a speaker's range. The H/L primitive distinctions form the foundations for 2 types of entities: pitch accents, which signal prominence or culmination; and boundary tones, which signal unit completion, or delimitation. The boundary tones are further divided into phrase types and full boundary types, which would mark the ends of intonational phrases or whole utterances.

While useful as a link to syntax/semantics, the term *accent* as defined in Section 15.4.3 is a bit too abstract, even for symbolic prosody. What is required is a way of labeling linguistically significant types of pitch contrast on accented syllables. Such a system could serve as the basis for a theory of intonational meaning. The ToBI standard specifies six types of pitch accents (see Table 15.1) in American English, where the * indicates direct alignment with an accented syllable, two intermediate phrasal tones (see Table 15.2), and five boundary tones [4] (see Table 15.3).

In American English one sometimes hears a string of strictly descending pitch accent levels across a short phrase. When judiciously applied, this *downstep* effect can be pleasantly natural, as in the following sentence:

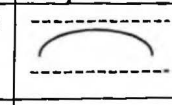
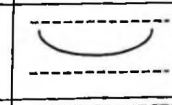
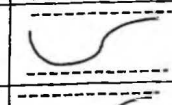
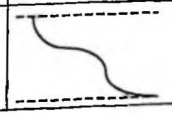
*"I saw a big-H**
*fat-!H**
pig-!H (L-L%)"*

Amazon/VB Assets
Exhibit 1011
Page 780

A basic rule used in the baseline F0 generation system of Section 15.6.2 consists in having all the pitch accents realized as H*, associated with the lexically stressed syllable of accented words. In general, ToBI representations of intonation should be sparse, specifying only what is linguistically significant. So, words lacking accent should not receive ToBI pitch accent annotations, and their pitch must be derived via interpolation over neighbors, or by some other default means. Low excursions can be linguistically significant also, in the crude sense that if you dip very low in your range on a given word, it may be perceived as prominent by listeners. L*+H and L+H* are both F0 rises on the accented syllable, but in the case of L*+H, the association of the starred tone (L*) with the accented syllable may push the realization of H off to the following syllable. !H* can be used for successively lowered high accents, such as might be found on *big red car*, or *tall, dark, and handsome*. A ToBI labeled utterance is shown in Figure 15.5.

A typical boundary tone is the *final lowering*, the marked tendency for the final syllable in all kinds of noninterrogative utterances to be realized on a pitch level close to the absolute bottom of a speaker's range. The final low (L-L%) may 'pull down' the height of some few accents to its left as well [41].

Table 15.1 ToBI pitch accent tones.

ToBI tone	Description	Graph
H*	*peak accent*—a tone target on an accented syllable which is in the upper part of the speaker's pitch range.	
L*	*low accent*—a tone target on an accented syllable which is in the lowest part of the speaker's pitch range	
L*+H	*scooped accent*—a low tone target on an accented syllable which is immediately followed by a relatively sharp rise to a peak in the upper part of the speaker's pitch range.	
L*+!H	*scooped downstep accent*—a low tone target on an accented syllable which is immediately followed by a relatively flat rise to a downstep peak.	
L+H*	*rising peak accent*—a high peak target on an accented syllable which is immediately preceded by a relatively sharp rise from a valley in the lowest part of the speaker's pitch range.	
!H*	*downstep high tone*—a clear step down onto an accented syllable from a high pitch which itself cannot be accounted for by an H phrasal tone ending the preceding phrase or by a preceding H pitch accent in the same phrase.	

Amazon/VB Assets
Exhibit 1011
Page 781

Table 15.2 ToBI intermediate phrasal tones.

ToBI tone	Description
L-	Phrase accent, which occurs at an intermediate phrase boundary (level 3 and above).
H-	Phrase accent, which occurs at an intermediate phrase boundary (level 3 and above).

Ultimately, abstract linguistic categories should correlate with, or provide labels for expressing, contrasts in meaning. While the ToBI pitch accent inventory is useful for generating a variety of English-like F0 effects, the distinction between perceptual contrast, functional contrast, and semantic contrast is particularly unclear in the case of prosody [41]. For example, whether or not the L*, an alternative method of signaling accentual prominence, functions in full linguistic contrast to H* is unclear.

In addition, we have mentioned that junctures are typically marked with perceptible pitch movements that are independent of accent. The ToBI specification also allows for combinations of the H and L primitives that signal phrase, clause, and utterance boundaries. These are called phrasal tones. The ToBI specification further points out that since intonation phrases are composed of one or more intermediate phrases plus a boundary tone, full intonation phrase boundaries have two final tones.

The symbolic ToBI transcription alone is not sufficient to generate a full F0 contour. The remaining components are discussed in Section 15.6.

Table 15.3 ToBI boundary tones.

ToBI tone	Description
L-L%	For a full intonation phrase with an L phrase accent ending its final intermediate phrase and an L% boundary tone falling to a point low in the speaker's range, as in the standard 'declarative' contour of American English.
L-H%	For a full intonation phrase with an L phrase accent closing the last intermediate phrase, followed by an H boundary tone, as in 'continuation rise.'
H-H%	For an intonation phrase with a final intermediate phrase ending in an H phrase accent and a subsequent H boundary tone. as in the canonical 'yes-no question' contour. Note that the H- phrase accent causes 'upstep' on the following boundary tone, so that the H% after H- rises to a very high value.
H- L%	For an intonation phrase in which the H phrase accent of the final intermediate phrase upsteps the L% to a value in the middle of the speaker's range, producing a final level plateau.
%H	High initial boundary tones; marks a phrase that begins relatively high in the speaker's pitch range when not explained by an initial H* or preceding H%.

Amazon/VB Assets
Exhibit 1011
Page 782

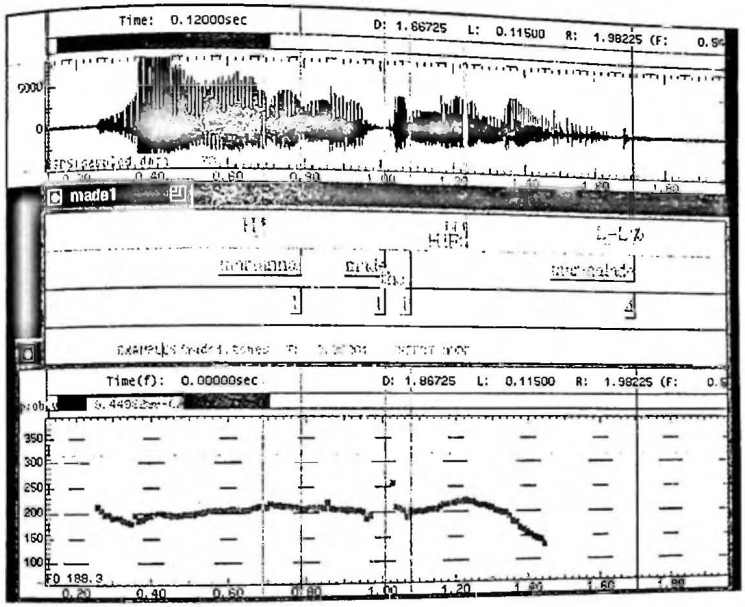


Figure 15.5 "*Marianna made the marmalade*", with an H* accent on *Marianna* and *marmalade*, and a final L-L% marking the characteristic sentence-final pitch drop. Note the use of 1 for the weak inter-word breaks, and 4 for the sentence-final break (after Beckman [4]).

15.4.5. Tune

Nyaah nuh nyaah you get
nyaah nyaah, can't me!

—Children's chant

Some pitch contours appear to be immediately recognizable and emotionally interpretable, independent of lexical content, such as the English *children's chant* above [40]. Can this idea of stylized *tunes*, perhaps decomposable into the *tones* we examined above, be applied to the intonation of ordinary speech? In fact, the ideal use of the ToBI pitch accent labels above would be as primitive elements in holistic prosodic contour descriptions, analogous to the role of phonemes in words. Ultimately, a dictionary of meaningful contours, described abstractly by ToBI tone symbols to allow for variable phonetic realization, would constitute a theory of intonational meaning for American English. Ideally, the meanings of such contours could perhaps be derived compositionally from the meanings of their constituent pitch accent and boundary tones, thus allowing us to dispense with the dictionary altogether. Contour stylization approaches describe contours holistically and index them for

Amazon/VB Assets
Exhibit 1011
Page 783

application on the basis of utterance type, usually based on a naïve syntactic typology, e.g., question, declarative, etc.

The holistic representation of contours can perhaps be defended, but the categorizing of types via syntactic description (usually triggered by punctuation) is questionable. Typically, use of punctuation as a rule trigger for pitch effects is making certain hidden assumptions about the relation between punctuation and syntax, and in turn between syntax and prosody. An obvious example is question intonation. If you find a question mark at the end of a sentence, are you justified in applying a final high rise (which might be denoted as H-H% in ToBI)? First, the intonation of yes-no questions in general differs from that of *wh*-questions. *Wh*-questions usually lack the extreme final upturn of F0 heard in some yes-no questions:

i. *Are you going?*
ii. *Where are you going?*

However, there are cases where an extreme final upturn is acceptable on ii. As [8] puts it, "It has been emphasized repeatedly ... that no intonation is an infallible clue to any sentence type: any intonation that can occur with a statement, a command, or an exclamation can also occur with a question."

Admittedly, there is a rough correspondence between syntactic types and speech acts,[2] as shown in Table 15.4. Nevertheless, the correspondence between syntactic types and acts is not deterministic, and prosody in spontaneous speech is definitely mediated via speech acts (the pragmatic context and use of an utterance) rather than syntactic types. Thus, it is difficult to obtain high-quality simulation of spontaneous speech based on linguistic descriptions that do not include speech acts and pragmatics. Likewise, even simulation of prose reading without due consideration of pragmatics and speech acts, and based solely on syntactic types, is difficult because prose that is read may:

- Include acted dialog
- Have limited occurrence of most types other than declarative, lessening variety in practice
- Include long or complex sentences, blunting 'stereotypical' effects based on utterance type
- Lack text cues as to syntactic type, or analysis grammar may be incomplete

Table 15.4 Relationship between syntactic types and speech acts.

Type	Speech Act	Example
interrogative	Questioning	Is it good?
declarative	Stating	It's good.
imperative	Commanding	Be good!
exclamatory	Exclaiming	How good it is!

[2] For a more in-depth coverage of speech acts, consult Chapter 17.

Amazon/VB Assets
Exhibit 1011
Page 784

Thus, description of an entire speech event, rather than inferences about text content, is again the ultimate guarantor of quality. This is why the future of automatic prosody lies with *concept-to-speech* systems (see Chapter 17) incorporating explicit pragmatic and semantic context specification to guide message rendition.

For commercial TTS systems that must infer structure from raw text, there are a few characteristic fragmentary pitch patterns that can be taken as tunes and applied to special segments of utterances. These include:

- Phone numbers—downstepping with pauses
- List intonation—downstepping with pauses (melons, pears, and eggplants)
- Tag and quotative tag intonation—low rise on tag (*Never!* he blurted. Come here, Jonathan.)

15.4.6. Prosodic Transcription Systems

ToBI, introduced above, can be used as a notation for transcription of prosodic training data and as a high-level specification for the symbolic phase of prosodic generation. Alternatives to ToBI also exist for these purposes, and some of them are amenable to automated prosody annotation of corpora. Some examples of this type of system are discussed in this section.

PROSPA was developed specially to meet the needs of discourse and conversation analysis, and it has also influenced the Prosody Group in the European ESPRIT 2589 SAM (Multilingual Speech Input/Output Assessment, Methodology and Standardization) project [50]. The system has annotations for general or global trends over long spans shown in Table 15.5, short, accent-lending pitch movements on particular vowels are transcribed in Table 15.6, and the pitch shape after the last accent in a () sequence, or *tail*, is indicated in Table 15.7.

Table 15.5 Annotations for general or global trends over long spans.

( )	extent of a sequence of cohesive accents
F	globally falling intonation
R	globally rising intonation
H	level intonation on high tone level
M	level intonation on middle tone level
L	level intonation on low tone level
H/F	falling intonation on a globally high tone level
...	sequence of weakly accented or unaccented syllables

Amazon/VB Assets
Exhibit 1011
Page 785

Table 15.6 Annotations for accent-lending pitch movements on particular vowels.

+	Upward pitch movement
-	Downward pitch movement
=	level pitch accent

Table 15.7 Annotations for pitch shape after the last accent in a () sequence, or *tail.*

	falling tails
/	rising tails
-	level tails
^	combinations of tails (rising-falling here)

INTSINT is a coding system of intonation described in [22]. It provides a formal encoding of the symbolic or phonologically significant events on a pitch curve. Each such target point of the stylized curve is coded by a symbol, either as an absolute tone, scaled globally with respect to the speakers pitch range, or as a relative tone, defined locally in conjunction with the neighboring target points. Absolute tones in INTSINT are defined according to the speaker's pitch range as shown in Table 15.8. Relative tones are notated in INTSINT with respect to the height of the preceding and following target points, as shown in Table 15.9.

Table 15.8 The definition of absolute tones in INTSINT.

T	top of the speaker's pitch range
M	initial, mid value
B	bottom of the speaker's pitch range

Table 15.9 The definition of relative tones in INTSINT.

H	target higher than both immediate neighbours
L	target lower than both immediate neighbours
S	target not different from preceding target
U	target in a rising sequence
D	target in a falling sequence

In a transcription, numerical values can be retained for all F0 target points. TILT [60] is one of the most interesting models of prosodic annotation. It can represent a curve in both its qualitative (ToBI-like) and quantitative (parametrized) aspects. Generally any 'interesting' movement (potential pitch accent or boundary tone) in a syllable can be described in terms of TILT events, and this allows annotation to be done quickly by humans or machines without specific attention to linguistic/functional considerations, which are paramount for ToBI labeling. The linguistic/functional correlations of TILT events can be linked by subsequent analysis of the pragmatic, semantic, and syntactic properties of utterances.

Amazon/VB Assets
Exhibit 1011
Page 786

The automatic parametrization of a pitch event on a syllable is in terms of:

- starting F0 value (Hz)
- duration
- amplitude of rise (A_{rise}, in Hz)
- amplitude of fall (A_{fall}, in Hz)
- starting point, time aligned with the signal and with the vowel onset

The tone shape, mathematically represented by its *tilt*, is a value computed directly from the F0 curve by the following formula:

$$tilt = \frac{|A_{rise}| - |A_{fall}|}{|A_{rise}| + |A_{fall}|} \tag{15.1}$$

A likely syllable for tilt analysis in the contour can be automatically detected based on high energy and relatively extreme F0 values or movements. Human annotators can select syllables for attention and label their qualities according to Table 15.10.

Table 15.10 Label scheme for syllables.

sil	Silence
c	Connection
a	Major pitch accent
fb	Falling boundary
rb	Rising boundary
afb	Accent+falling boundary
arb	Accent+rising boundary
m	Minor accent
mfb	Minor accent+falling boundary
mrb	Minor accent+rising boundary
l	Level accent
lrb	Level accent+rising boundary
lfb	Level accent+falling boundary

15.5. DURATION ASSIGNMENT

Pitch and duration are not entirely independent, and many of the higher-order semantic factors that determine pitch contours may also influence durational effects. The relation between duration and pitch events is a complex and subtle area, in which only initial

Amazon/VB Assets
Exhibit 1011
Page 787

exploration has been done [63]. Nonetheless, most systems often treat duration and pitch independently because of practical considerations [61].

Numerous factors, including semantics and pragmatic conditions, might ultimately influence phoneme durations. Some factors that are typically neglected include:

- The issue of speech rate relative to speaker intent, mood, and emotion.
- The use of duration and rhythm to possibly signal document structure above the level of phrase or sentence (e.g., paragraph).
- The lack of a consistent and coherent practical definition of the phone such that boundaries can be clearly located for measurement.

15.5.1. Rule-Based Methods

Klatt [1] identified a number of first-order perceptually significant effects that have largely been verified by subsequent research. These effects are summarized in Table 15.11.

Table 15.11. Perceptually significant effects for duration. After Klatt [1].

Lengthening of the final vowel and following consonants in prepausal syllables.
Shortening of all syllabic segments[3] in nonprepausal position.
Shortening of syllabic segments if not in a word final syllable.
Consonants in non-word-initial position are shortened.
Unstressed and secondary stressed phones are shortened.
Emphasized vowels are lengthened.
Vowels may be shortened or lengthened according to phonetic features of their context.
Consonants may be shortened in clusters.

The rule-based duration-modeling mechanism involves table lookup of minimum and inherent durations for every phone type. The minimum duration is rate dependent, so all phones could be globally scaled in their minimum durations for faster or slower rates. The inherent duration is the raw material for the rules above: it may be stretched or contracted by a prespecified percentage attached to each rule type above applied in sequence, then it is finally added back onto the minimum duration to yield a millisecond time for a given phone. The duration of a phone is expressed as

$$d = d_{min} + r(\bar{d} - d_{min}) \tag{15.2}$$

[3] Syllabic segments include vowels and syllabic consonants.

Amazon/VB Assets
Exhibit 1011
Page 788

where d_{min} is the minimum duration of the phoneme, $\bar{d}$ is the average duration of the phoneme, and the correction r is given by

$$r = \prod_{i=1}^{N} r_i \tag{15.3}$$

for the case of N rules being applied where each rule has a correction r_i. At the very end, a rule may apply that lengthens vowels when they are preceded by voiceless plosives (/p/, /t/, /k/). This is also the basis for the additive-multiplicative duration model [49] that has been widely used in the field.

15.5.2. CART-Based Durations

A number of generic machine-learning methods have been applied to the duration assignment problem, including CART and linear regression [43, 62]. The voice datasets generally rely on less than the full set of possible joint duration predictors implied in the rule list of Table 15.11. It has been shown that a model restricted to the following features and contexts can compare favorably, in listeners' perceptions, with durations from natural speech [43]:

- Phone identity
- Primary lexical stress (binary feature)
- Left phone context (1 phone)
- Right phone context (1 phone)

In addition, a single rule of vowel and post-vocalic consonant lengthening (rule 1 in Table 15.11) is applied in prepausal syllables. The restriction of phone context to immediate left and right neighbors results in a triphone duration model, congruent with the voice triphone model underlying the basic synthesis in the system [23]. In perceptual testing this simple triphone duration model yielded judgments nearly identical to those elicited by utterances with phone durations from natural speech [43]. From this result, you may conjecture that even the simplified list of *first-order* factors above may be excessive, and that only the handful of factors implicit in the triphones themselves, supplemented by a single-phrase final-syllable coda lengthening rule, is required. This would simplify data collection and analysis for system construction.

15.6. PITCH GENERATION

We now describe the issues involved in generating synthetic pitch contours. Pitch, or F0, is probably the most characteristic of all the prosody dimensions. As discussed in Section 15.8, the quality of a prosody module is dominated by the quality of its pitch-generation component.

Amazon/VB Assets
Exhibit 1011
Page 789

Since generating pitch contours is an incredibly complicated problem, pitch generation is often divided into two levels, with the first level computing the so-called symbolic prosody described in Section 15.4 and the second level generating pitch contours from this symbolic prosody. This division is somewhat arbitrary since, as we shall see below, a number of important prosodic phenomena do not fall cleanly on one side or the other but seem to involve aspects of both. Often it is useful to add several other attributes of the pitch contour prior to its generation, which are discussed in Section 15.6.1.

15.6.1. Attributes of Pitch Contours

A pitch contour is characterized not only by its symbolic prosody but also by several other attributes such as pitch range, gradient prominence, declination, and microprosody. Some of these attributes often cross into the realm of symbolic prosody. These attributes are also known in the field as phonetic prosody (termed as an analogy to phonology and phonemics).

15.6.1.1. Pitch Range

Pitch range refers to the high and low limits within which all the accent and boundary tones must be realized: a floor and ceiling, so to speak, which are typically specified in Hz. This may be considered in terms of stable, speaker-specific limits as well as in terms of an utterance or passage. For a TTS system, each voice typically has a characteristic pitch range representing some average of the pitch extremes over test utterances. This speaker-specific range can be set as an initial default for the voice or character. These limits may be changed by an application.

Another sense of pitch range is the actual exploitation of zones within the *hard* limits at any point in time for linguistic purposes, having to do with expression of the content or feeling of the message. Pitch-range variation that is correlated with emotion or other aspects of the speech event is sometimes called *paralinguistic*. This linguistic and paralinguistic use of pitch range includes aspects of both symbolic and phonetic prosody. Since it is quantitative, it certainly is a phonetic property of an utterance's F0 contour. Furthermore, it seems that most linguistic contrasts involving pitch accents, boundary tones, etc. can be realized in any pitch range. These settings can be estimated from natural speech (for research purposes) by calculating F0 mean and variance over an utterance or set of utterances, or by simply adopting the minimum and maximum measurements (perhaps the 5th and 95th percentile to minimize the effect of pitch tracker errors).

But, although pitch range is a phonetic property, it can be systematically manipulated to express states of mind and feeling in ways that other strictly phonetic properties, such as characteristic formant values, rarely are. Pitch range interacts with all the prosodic attributes you have examined above, and certain pitch-range settings may be characteristic of particular styles or utterance events. For example, it is noted [8] that: "*we cannot speak of an into-*

Amazon/VB Assets
Exhibit 1011
Page 790

nation of exclamation ... Exclamation draws impartially upon the full repertory of up-down patterns. What characterizes the class is not shape but range: exclamations reach for the extreme—usually higher but sometimes lower." In this sense, then, pitch range cannot be considered an arbitrary or physiological attribute—it is directly manipulated for communicative purposes.

In prosodic research, distinguishing emotive and iconic use of pitch (analogous to gesture) from strictly linguistic (logical, syntactic, and semantic expression, with arbitrary relation between signifier and signified) prosodic phenomena has been difficult. Pitch-range variation seems to straddle emotional, linguistic, and phonetic expression.

A linguistic pitch range may be narrowed or widened, and the zone of current pitch variation may be placed anywhere within a speaker's wider, physically determined range. So, for example, a male speaker might adopt a falsetto speaking style for some purpose, with his pitch range actually narrowed, but with all pitch variation realized in a high portion of his overall range, close to his physical limits.

Pitch range is a gradient property, without categorical bounds. It seems to trade off with other model components: accent, relative prominence, downstep, and declination. For example, if our model of prosody incorporates, say, an accent-strength component, but if we also recognize that pitch range can be manipulated for linguistic purposes, we may have difficulty determining, in analysis, whether a given accent is at partial strength in a wide range or at full strength in a reset, narrower range. This analytic uncertainty may be reflected in the quality of models based on the analysis.

A practical TTS system has to stay within, and make some attempt to maximize the exploitation of, the current system default or user-specified range. Thus, for general TTS purposes, the simplest approach is to use about 90% of the user-set or system default range for general prose reading, most of the time, and use the reserved 10% in special situations, such as the paragraph initial *resets*, exclamations, and emphasized words and phrases.

15.6.1.2. Gradient Prominence

Gradient prominence refers to the relative strength of a given accent position with respect to its neighbors and the current pitch-range setting. The simplest approach, where every accented syllable is realized as a H(igh) tone, at uniform strength, within an invariant range, can sound unnatural. At first glance, the prominence property of accents might appear to be a phonetic detail, in that it is quantitative, and certainly any single symbolic tonal transcription can be realized in a wide variety of relative per-accent prominence settings. However, the relative height of accents can fundamentally alter the information content of a spoken message by determining focus, contrast, and emphasis. You would hope that such linguistic content would be determined by the presence and absence, or perhaps the types (H, L, etc.), of the symbolic accents themselves. But an accented syllable at a low prominence might be perceived as unaccented in some contexts, and there is no guaranteed minimum degree of prominence for accent perception. Furthermore, as noted above, the realization of prominence of an accent is context-sensitive, depending on the current pitch-range setting.

Amazon/VB Assets
Exhibit 1011
Page 791

The key knowledge deficit here is a theory of the interpretation of prominence that would allow designers to make sensible decisions. It appears that relative prominence is related to the information status of accent-bearing words and is in that sense linguistic, yet there is no theory of prominence categories that would license any abstraction. For the present, many commercial TTS systems adopt a pseudorandom pattern of alternating stronger/weaker prominence, simply to avoid monotony. If a word is tagged for emphasis, or if its information status can otherwise be inferred, its prominence can be heightened within the local range.

In the absence of information on the relative semantic salience of accented words in the utterance, successive prominence levels are varied in some simple alternating pattern, to avoid monotony. Rather than limiting the system to a single peak F0 value per accented syllable, several points could be specified, which, when connected by interpolation and smoothing, could give varied effects within the syllable, such as rising, falling, and scooped accents.

15.6.1.3. Declination

Related to both pitch range and gradient prominence is the long-term downward trend of accent heights across a typical reading-style, semantically neutral, declarative sentence. This is called *declination*. Although this tendency, if overdone, can simply give the effect of a bored or uncomprehending reader, it is a favorite prosodic effect for TTS systems, because it is simple to implement and licenses some pitch change across a single sentence. If a system uses a 'top line' as a reference for calculating the height of every accent, the slope of that top line can simply be declined across the utterance. Otherwise, each accent's prominence can be realized as a certain percentage of the height of the preceding one. Declination can be reset at utterance boundaries, or within an utterance at the boundaries of certain linguistic structures, such as the beginning of quoted speech. Intrasentence phrase and clause types that typically narrow the pitch range, such as parentheticals and certain relative clauses, can be modeled by suspending the declination, or adopting a new declination line for the temporary narrowed range, then resuming the suspended longer-term trend as the utterance progresses. Needless to say, declination is not a prominent feature of spontaneous speech and in any case should not be overdone.

The minor effect of declination should not be confused with the tendency in all kinds of nonquestioning utterances to end with a very low pitch, close to the bottom of the speaker's range. In prosodic research this is called *final lowering* and is well-attested as a phenomenon that is independent of declination [29]. The ToBI notation used to specify final lowering is the complex boundary tone L-L%. In Figure 15.6 we show the declination line together with the other two *downers of intonation*: downstep and final lowering described in Section 15.4.4.

Amazon/VB Assets
Exhibit 1011
Page 792

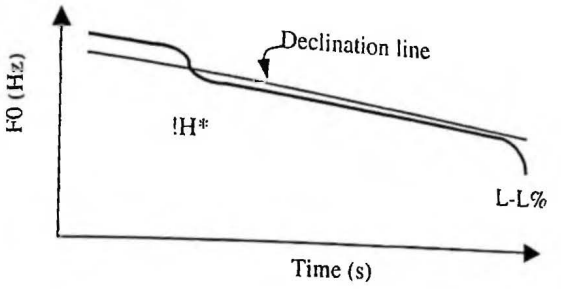


Figure 15.6 The three downers of intonation: the declination line, a downstep (!H*), and the final lowering (L-L%).

15.6.1.4. Phonetic F0—Microprosody

Microprosody refers to those aspects of the pitch contour that are unambiguously phonetic and that often involve some interaction with the speech carrier phones. These may be regarded as *second-order* effects, in the sense that rendering them well cannot compensate for incorrect accentuation or other mistakes at the symbolic level. Conversely, making no attempt to model these but putting a great deal of care into the semantic basis for determining accentuation, contrast, focus, emphasis, phrasing, etc. can result in a system of reasonable quality. Nevertheless, all else being equal, it is advisable to make some attempt to capture the local phonetic properties of natural pitch contours.

If the strength of accents is controlled semantically, by having equal degrees of focus on words of differing phonetic makeup, it has been observed that high vowels described in Chapter 2 carrying H* accents are uniformly higher in the phonetic pitch range than low vowels with the same kinds of accent. The distinction between high and low vowels correlates with the position of the tongue in articulation (high or low in the mouth). The highest English vowels by this metric are */iy/* (as in *bee*) and */uw/* (as in *too*), while the lowest vowel is */aa/* as in *father*. The predictability of F0 under these conditions may relate to the degree of tension placed on the laryngeal mechanisms by the raised tongue position in the high vowels as opposed to the low. In any case, this effect, while probably perceptually important in natural speech, is challenging for a synthesizer. The reason relates again to the issue of gradient prominence, discussed above. Apart from experimental prompts in the lab, there is currently no principled way to assign prominence for accent height realization based on utterance content in general TTS. It may therefore be difficult for a listener to correctly factor pitch accent height that is due to correctly (or incorrectly) assigned gradient prominence from height variation related to the lower-level phonetic effects of vowel height.

Amazon/VB Assets
Exhibit 1011
Page 793

Another phonetic effect is the level F0 in the early portion of a vowel that follows a voiced obstruent such as /b/, contrasted with the typical fall in F0 following a voiceless obstruent such as /p/. This phonetic conditioning effect, of both preceding and following consonants, can be observed most clearly when identical underlying accent types are assigned to the carrier vowel, and may persist as long as 50 ms or more into the vowel. The exact contribution of the pre-vocalic consonant, the post-vocalic consonant, and the underlying accent type are difficult to untangle, though [54] is a good survey of all research in this area and adds new experimental results. For commercial synthesizers, this is definitely a second-order effect and is probably more important for rule-based formant synthesizers (see Chapter 16), which need to use every possible cue to enforce distinctions among consonants in phoneme perception, than for strictly intonational synthesis. However, in order to achieve completely natural prosody in the future, this area will have to be addressed.

Last, and perhaps least, *jitter* is a variation of individual cycle lengths in pitch-period measurement, and *shimmer* is variation in energy values of the individual cycles. These are distinct concepts, though somewhat correlated. Obviously, this is an influence of glottal pulse shape and strength on the quality of vowels. Speech with jitter and shimmer over 15% sounds pathological, but complete regularity in the glottal pulse may sound unnatural. For a deeper understanding of how these could be controlled, see Chapter 16.

15.6.2. Baseline F0 Contour Generation

We now examine a simple system that generates F0 contours. Although each stage of an F0 contour algorithm ideally requires a complete natural language and semantic analysis system, in practice a number of rules are often used. The system described here illustrates most of the important features common to the pitch-generation systems of commercial synthesizers.

First, let's consider a natural speech sample and describe what initial information is needed to characterize it, and how an artificial pitch contour can be synthesized based on the input analysis. The chosen sample is the utterance *"Don't hit it to Joey!"*, an exclamation, from the ToBI Labeling Guidelines sample utterance set [4]. The natural waveform, aligned pitch contour, and abstract ToBI labels are shown in Figure 15.7. This utterance is about 1.63 seconds and it has three major ToBI pitch events:

H*	high pitch accent on *Don't*
L*+!H	low pitch accent with following downstepped high on *Joey*
L-L%	low utterance-final boundary tone at the very end of utterance

Amazon/VB Assets
Exhibit 1011
Page 794

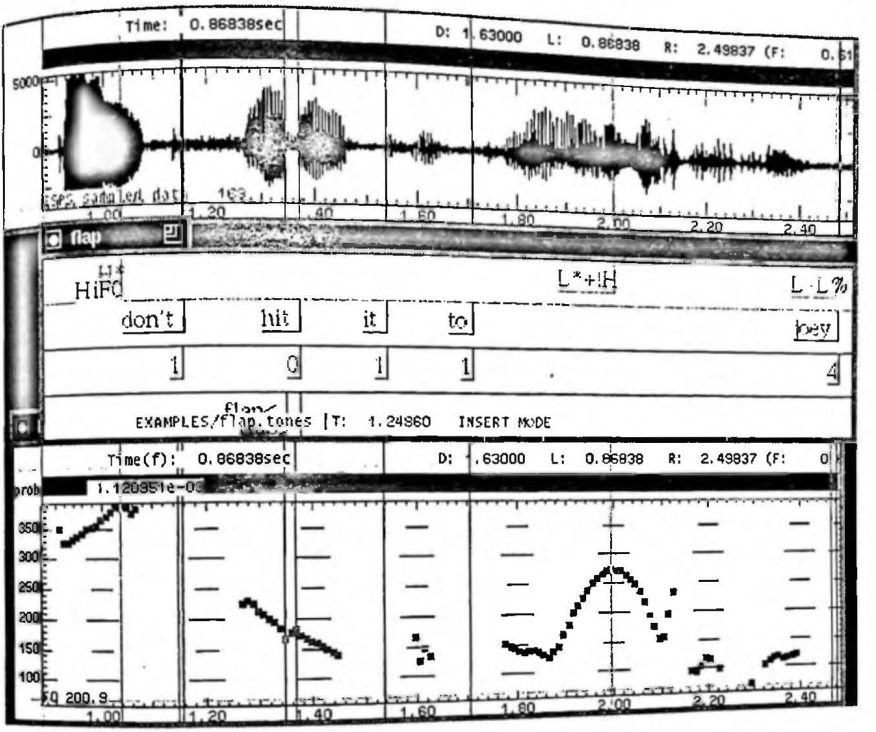


Figure 15.7 Time waveform, segmentation, ToBI marks, and pitch contour for the utterance *"Don't hit it to Joey!"* spoken by a female speaker (after Beckman [4]).

The input to the F0 contour generator includes:

- Word segmentation.
- Phone labels within words.
- Durations for phones, in milliseconds.
- Utterance type and/or punctuation information.
- Relative salience of words as determined by grammatical/semantic analysis.
- Current pitch-range settings for voice.

Amazon/VB Assets
Exhibit 1011
Page 795

15.6.2.1. Accent Determination

Although accent determination ideally requires a complete natural language and semantic analysis system (see Section 15.4.3), in practice a number of rules are often used. The first rule is: *Content word categories of noun, verb, adjective, and adverb are to be accented, while the function word categories (everything else, such as pronoun, preposition, conjunction, etc.) are to be left unaccented.* Rules can be used to tune this by specifying which POS is accented or not and in which context.

If we apply that simple metric to the natural sample of Figure 15.7, we see that it does not account for the accentuation of 'hit', which, as a verb, should have been accented. In a real system perhaps we would have accented it, and this might have resulted in the typical overaccented quality of synthetic prosody. For this sample discussion, let's adopt a simplified version of a rule found in some commercial synthesizers: *Monosyllabic common verbs are left unaccented.*

What about *"Don't"*? A simplistic view would state that the POS-based policy has done the right thing, after all *"Don't"* can be regarded as a verbal form. However, usually *do* is considered an auxiliary verb and is not accented. For now we adopt another rule that says: *In a negative imperative exclamation, determined by presence of a second-person negative auxiliary form and a terminal exclamation point, the negative term gets accented.* The adoption of these corollaries to the simple POS-based accentuation rule accounts for our accent placement in the present example, but of course it sometimes fails, as does any rigid policy. So our utterance would now appear (with words selected for accent in upper case) as *"DON'T hit it to JOEY!"*

15.6.2.2. Tone Determination

In the limit, tone determination (see Section 15.4.4) also requires a complete natural language and semantic analysis system, but in practice a number of rules are often used. Generally, in working systems, H* is used for all pitch accent tones, and this is actually very realistic, as H* is the most frequent tone in natural speech.

Sometimes complex tones of the type L*+!H are thrown in for a kind of pseudovariety in TTS. In our sample natural utterance this is the tone that is used, so here we assume that this is the accent type assigned.

We also need to mark punctuation-adjacent and utterance-final phonemes as rise, continue, or fall boundaries. In this case we mark it as L-L%.

15.6.2.3. Pitch Range

To determine the pitch range, we are going to make use of three lines as a frame within which all pitches are calculated. The top line and bottom line would presumably be derived from the current or default pitch-range settings as controlled by an application or user. Here

Amazon/VB Assets
Exhibit 1011
Page 796

we set them in accord with the limits of our natural sample. Note that while, for this example, the pitch contour is generated within an actual pitch range, it could also be done within an abstract range of, say, 1–100, which the voice-generation module could map to the current actual setting. So we set the top line at $T = 375$ Hz and the base line at $B = 100$ Hz.

It is more advantageous to work in a logarithmic scale, because it is more easily ported from males to females, and because this better represents human prosody. There are 24 semitones in an octave; thus a semitone corresponds to a ratio of $a = 2^{1/24}$. The pitch range can be expressed in semitones as

$$n = 24 \log_2 (T/B) \simeq 80 \log_{10} (T/B) \tag{15.4}$$

so that we can express frequencies in semitones as

$$f_0 = 80 \log_{10} F_0 \tag{15.5}$$

and its inverse

$$F_0 = 10^{f_0/80} \tag{15.6}$$

Using Eq. (15.5), the top line is $t = 206$ and the base line is $b = 160$. The reference line is a kind of midline for the range, used in the accent scaling calculations, and is set halfway between the bottom and top lines, i.e., $r = 183$, and using Eq. (15.6), $R = 194$ Hz.

15.6.2.4. Prominence Determination

The relative prominence of the words (see Section 15.6.1.2) allows the pitch module to scale the pitch within any given pitch range. Here we assume (arbitrarily) that $N = 5$ degrees of abstract relative prominence are sufficient. This means that, e.g., an H* pitch accent with prominence 5 will be at or near the very top of the current pitch-range setting, while an L* with the same prominence will be at or near the very bottom of the range. Smaller prominence numbers indicate less salience, placing their pitch events closer to the middle of the range.

Converting the abstract tone types plus prominence into pitch numbers is more art than science (but see Section 15.6.4 for a discussion of data-based methods for this process). Here we assume a simple linear relationship between the tone's type and relative prominence:

$$f_0[i] = r + (t - r) * p[i] / N \tag{15.7}$$

In the limit, prominence determination also requires a complete natural language and semantic analysis system, but in practice a number of heuristics are often used. One such heuristic is: *In a negative imperative exclamation, the negative term gets the most emphasis,* leading to a relative prominence assignment of 5 on 'don't.' Using Eqs. (15.7) and (15.6), the anchor equals the top range of 375 Hz.

Amazon/VB Assets
Exhibit 1011
Page 797

Then, since the L*+!H involves a *downstepped* term, it must by definition be lower than the preceding H* accent, so we arbitrarily assign it a relative prominence of '2'. The L*+!H is more complex, requiring calculation and placement of two separate anchor points. For simplicity we are using a single prominence value for complex tones like L*+!H, but we could also use a value-per-tone approach, at the cost of greater analytical complexity. Using Eq. (15.7), it corresponds to 192 semitones, and with Eq. (15.6), the value of !H is 251 Hz. For L*, we use a prominence of –2 (we use negative values for L tones), which, using Eq. (15.7), results in an anchor of 174, or alternatively 149 Hz.

The L-L% tone is a boundary tone, so it always goes on the final frame of a syllable-final (in this case, utterance-final) phone. The L-L% in most ToBI systems is not treated as a two-part complex tone but rather as a *super low* L% boundary, falling at the very bottom of the speaker's pitch range, i.e., prominence of 5, for a few frames. Thus the F0 value of these anchor point is 100 Hz.

We also need to set anchors for the initial point. The initial anchor is usually set at some arbitrary but high place within the speaker's range (perhaps a rule looking at utterance type can be used). A prominence of 4 can be used, yielding a value of 329 Hz.

Finally we need to determine where to place the anchors within the accented syllable. Often they are placed in the middle of the vowel. All the anchor points are shown in Figure 15.8.

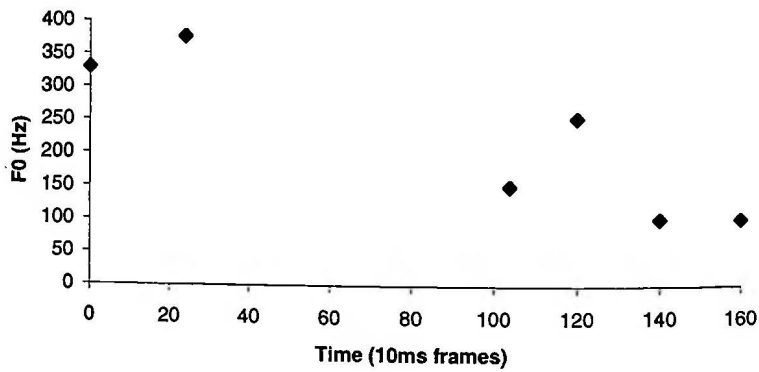


Figure 15.8 Anchor points of the F0 contour.

15.6.2.5. F0 Contour Interpolation

To obtain the full F0 contour we need some kind of interpolation. One way is to interpolate linearly and follow with a multipoint moving-average window over the resulting (angular) contour to smooth it out. Another possibility is a higher-order interpolation polynomial. In this case a cubic interpolation routine is called, which has the advantage of retaining the exact anchor points in a smoothed final contour (as opposed to moving average, which smears the anchor points). In general the choice of interpolation algorithm makes little per-

Amazon/VB Assets
Exhibit 1011
Page 798

ceptual difference, as long as no sharp 'corners' remain in the contour. In Figure 15.9 the contour was interpolated fully, without regard to voicing properties of underlying phones. In the graph, the sections corresponding to unvoiced phones have been replaced with zero, for ease of comparison to the sample in Figure 15.7. The interpolation can be done in the linear frequency, as in Figure 15.9, or in the log-frequency.

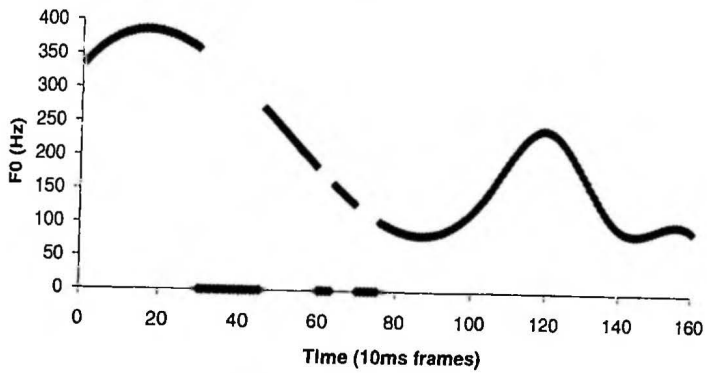


Figure 15.9 F0 contour of Figure 15.8 after cubic interpolation. Sections corresponding to unvoiced phones have been replaced with zero.

In order for the interpolation algorithm to operate properly we need to have phone durations so that the anchor points are appropriately spaced apart. In this baseline algorithm, we followed the algorithm described in Section 15.5.2.

15.6.2.6. Interface to Synthesis Module

Finally, most synthesizers cannot accept an arbitrary number of pitch controls on a given phoneme, nor it this necessary. We can downsample the pitch buffer to allocate a few characteristic points per phoneme record, and, if the synthesizer can interpolate pitch, it may be desirable to skip pitch controls for unvoiced phones altogether. The F0 targets can be placed at default locations (such as the left edge and middle of each phone), or the placements can be indicated by percent values on each target, depending on what the synthesizer supports. This has to be in agreement with the specific interface between the prosody module and the synthesis module as described in Section 15.2.

15.6.2.7. Evaluation and Possible Improvements

In comparing the output contour of Figure 15.9 to the natural one of Figure 15.7, how well have we done? As a first-order approximation, from visual inspection, it is somewhat similar

Amazon/VB Assets
Exhibit 1011
Page 799

to the original. Of course, we have used hand-coded information for the accent property, accent type, and prominence! However, these choices were reasonable, and could apply them as defaults to many other utterances. At a minimum, almost exactly the code given above would apply without change and give a decent contour for a whole 'family' of similar utterances, such as *"Don't hit the ball to Joey!"* or *"Never give the baseball to Henry!"* A higher-order discourse module would need to determine that *ball* and *baseball* are not accented, however, in order to use the given contour with the same rhetorical effect (presumably *ball* and *baseball* in these cases could be given/understood information).

Something very much like the system described here has been used in most commercially marketed synthesizers throughout the 1990s. This model seems overly simple, even crude, and presumably it could be substantially augmented, or completely replaced by something more sophisticated.

However, many weaknesses are apparent also. For one thing, the contour appears very smooth. The slight *jitter* of real contours can be easily simulated at a final stage of pitch buffer processing by modifying +/- 3 or 4 Hz to the final value of each frame. The degree to which such niceties actually affect listener perceptions depend entirely on the quality of the synthetic speech and the quality of the pitch-modification algorithms in the synthesizer.

The details of peak placement obviously differ between the natural and synthetic contours. This is partly due to the crude uniform durations used, but in practice synthesizers may incorporate large batteries of rules to decide exactly (for example) which frame of a phone the H* definition point should appear in—early, middle, late? Sometimes this decision is based on surrounding phonetic structure, word and syllable structure, and prosodic context. The degree to which this matters in perception depends partly on synthetic speech quality overall.

15.6.3. Parametric F0 Generation

To realize all the prosodic effects discussed above, some systems make almost direct use of a real speaker's measured data, via table lookup methods. Other systems use data indirectly, via parametrized algorithms with generic structure. The simplest systems use an invariant algorithm that has no particular connection to any single speaker's data, such as the algorithm described in the baseline F0 generation system of Section 15.6.2. Each of these approaches has advantages and disadvantages, and none of them has resulted in a system that fully mimics human prosodic performance to the satisfaction of all listeners. As in other areas of TTS, researchers have not converged on any single standard family of approaches. Once we venture beyond the simplest approaches, we find an interesting variety of systems, based on different assumptions, with differing characteristics. We now discuss a few of the more representative approaches.

Even models that make little or no attempt to analyze the internal components of an F0 contour must be indexed somehow. System designers should choose indexing or predictive factors that are derivable from text analysis, are general enough to cover most prosodic situations, and are powerful enough to specify high-quality prosody. In practice, most mod-

Amazon/VB Assets
Exhibit 1011
Page 800

els' predictive factors have a rough correspondence to, or are an elaboration of, the elements of the baseline algorithm of Section 15.6.2. A typical list might include the following:

- Word structure (stress, phones, syllabification)
- Word class and/or POS
- Punctuation and prosodic phrasing
- Local syntactic structure
- Clause and sentence *type* (declarative, question, exclamation, quote, etc.)
- Externally specified focus and emphasis
- Externally specified speech style, pragmatic style, emotional tone, and speech act goals

These factors jointly determine an output contour's characteristics, as listed below. Ideally, any or all of these may be externally and directly specified, or they may be inferred or implied within the F0 generation model itself:

- Pitch-range setting
- Gradient, relative prominence on each syllable
- Global declination trend, if any
- Local shape of F0 movement
- Timing of F0 events relative to phone (carrier) structure

The combinatorial complexity of these predictive factors, and the size of the resulting models, can be serious issues for practical systems that strive for high coverage of prosodic variability and high-quality output. The possibility of using *externally specified* symbolic markups gives the whole system a degree of modularity, in that prosodic annotation can be specified directly by an authoritative outside source or can be derived automatically by the symbolic prosody prediction process that precedes F0 contour generation.

Parametric models propose an underlying architecture of prosodic production or perception that constrains the set of possible outputs to conform to universal constants of the human speech mechanism. Naturally, these models need settings to distinguish different speakers, different styles, and the specifics of utterances. We describe superposition models and ToBI Realization models.

15.6.3.1. Superposition Models

An influential class of parametric models was initiated by the work [35] for Swedish, which proposed additive superposition of component contours to synthesize a complex final F0 track. In the version refined and elaborated in [14], the component contours, which may all have different strengths and decay characteristics, may correspond to longer-term trends, such as phrase or utterance declination, as well as shorter-time events, such as pitch accents

Amazon/VB Assets
Exhibit 1011
Page 801

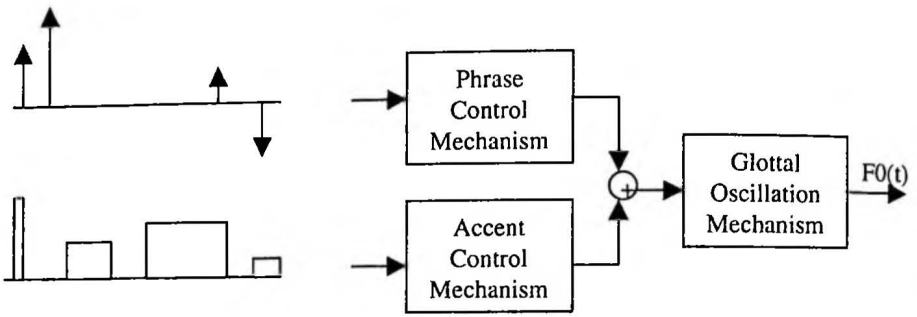


Figure 15.10 Fujisaki pitch model [15]. F0 is a superposition of phrase effects with accent effects. The phrase mechanism controls things like the declination of a declarative sentence or a question, whereas the accent mechanism accounts for accents in individual syllables.

on words. The component contours are modeled as the critically damped responses of second-order systems to impulse functions for the longer-term, slowly decaying phrasal trends, and step or rectangular functions of shorter-term accent events. The components so generated are added and ride a baseline that is speaker specific. The basic ingredients of the system, known as Fujisaki's model [15, 19], are shown in Figure 15.10. The resulting contour is shown in Figure 15.11. Obviously, similar effects can be generated with linear accent shapes as described in the simpler model above, with smoothing. However, there are some plausible claims for the articulatory correlates of the constraints imposed in the second-order damping and superposition effects of this model [33].

Superposition models of this type can, if supplied with accurate parameters in the form of time alignments and strengths of the impulses and steps, generate contours closely mimicking natural examples. In this respect, the remaining quality gap for general application is in the parametric knowledge driving the model, not in the model structure per se. These kinds of models have been particularly successful in replicating the relatively constrained Japanese reading-style. Whether these models can account straightforwardly for the immense variety of a large range of English speakers and text genre, or whether, on the contrary, the parameters proliferate and the settings become increasingly arbitrary, remains to be seen.

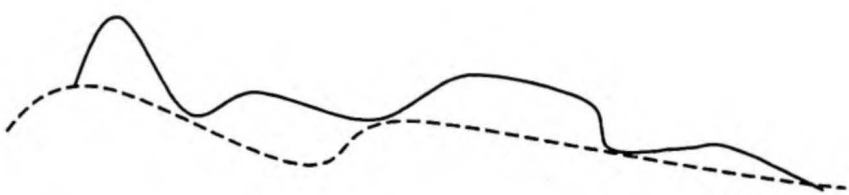


Figure 15.11 Composite contour obtained by low-pass filtering the impulses and boxes in the Fujisaki model of Figure 15.10.

Amazon/VB Assets
Exhibit 1011
Page 802

15.6.3.2. ToBI Realization Models

One simple parametric model, which in its inherent structure makes only modest claims for principled correspondence to perceptual or articulatory reality, is designed to support prosodic symbols such as the *Tones and Break Indices* (ToBI) system. This model, variants of which are developed in [2, 54], posits two or three control lines, by reference to which ToBI-like prosody symbols can be scaled. This provides for some independence between symbolic and phonetic prosodic subsystems. In the model shown in Figure 15.12, the top line is an upper limit of the pitch range. It can be slanted down to simulate declination. The bottom line represents the bottom of the speaker's range. Pitch accents and boundary tones (as in ToBI) are scaled from a reference line, which is often midway in the range in a logarithmic scale of the pitch range, as described in the baseline algorithm of 15.6.2. You can think of this scaling as operating within a percentage of the current range, rather than absolute values, so a generic method can be applied to any arbitrary pitch-range setting. The quantitative instantiation of accent height is done at the final stage. The accents and boundary tones consist of one or more points, which can be aligned with the carrier phones; then interpolation is applied between points, and smoothing is performed over the resulting contour.

In Figure 15.12, t, r, and b are the top, reference, and baseline pitch values, respectively. They are set from the defaults of the voice character and by user choice. The base b is considered a physiological constraint of voice. P is the prominence of the accent and N is the number of prominence steps. Declination can be modeled by slanting the top and/or reference lines down. The lowered position of the reference in Figure 15.12 reflects the observation that the realization of H(igh) and L(ow) ToBI abstract tones in a given pitch range is asymmetric, with a greater portion available for H, while L saturates more quickly. This is why placing the reference line midway between the top and base lines in a log-frequency scale automatically takes care of this phenomenon. After target points are located and scaled according to their gradient prominence specifications, the (hopefully sparse) targets can be interpolated and the resulting contour smoothed. If the contour is calculated in, say, 10-ms frames, two pitch targets sampled from the contour vector per phone usually suffice to reproduce the intended prosodic effects faithfully.

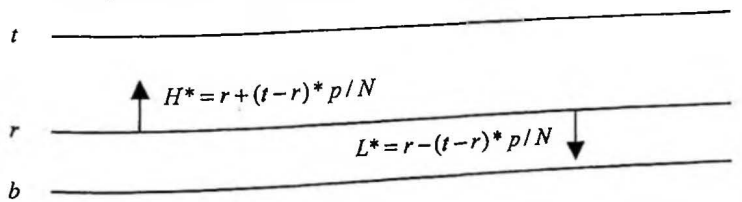


Figure 15.12 A typical model of tone scaling with an abstract pitch range.

Amazon/VB Assets
Exhibit 1011
Page 803

If a database of recorded utterances with phone labeling and F0 measurements has been reliably labeled with ToBI pitch annotations, it may be possible to automate the implementation of the ToBI-style parametrized model. This was attempted with some success in [5], where linear regression was used to predict syllable initial, vowel medial, and syllable final F0 based on simple, accurately measurable factors such as:

- ToBI accent type of target and neighbor syllables
- ToBI boundary pitch type of target and neighbor syllables
- Break index on target and neighbor syllables
- Lexical stress of target and neighbor syllables
- Number of syllables in phrase
- Target syllable position phrase
- Number and location of stressed syllable(s)
- Number and location of accented syllable(s)

Models of this sort do not incorporate an explicit mechanism (like the scaling direction from r in Figure 15.12) to distinguish H(igh) from L(ow) tone space, beyond what the data and its annotations imply.

The work in [47] consists of a ToBI realization model in which the 'smoothing' mechanism is built-in as a dynamical system whose parameters are also learnt from data. This work could be viewed as a stochastic realization of Fujisaki's superposition model without the phrase controls and where the accents are given by ToBI labels.

Both the ToBI realization models and the superposition models could, if supplied with sufficiently accurate measurements of an example contour, reproduce it fairly accurately. Both models require much detailed knowledge (global and local pitch range; location, type, and relative strength of accents and boundary tones; degree of declination; etc.) to function at human-level quality for a given utterance. If a system designer is in possession of a completely annotated, high-quality database of fully representative prosodic forms for his/her needs, the question of deployment of the database in a model can be made based on performance tradeoffs, maintenance issues, and other engineering considerations. If, on the other hand, no such database is available for the given application purpose, extremely high prosodic quality, including lively yet principled variation, should not be expected to result simply from choosing the 'mathematically correct' model type.

15.6.4. Corpus-Based F0 Generation

It is possible to have F0 parameters trained from a corpus of natural recordings. The simplest models are the direct models, where an exact match is required. Models that offer more generalization have a library of F0 contours that are indexed either from features from the parse tree or from ToBI labels. Finally, there are F0 generation models from a statistical

Amazon/VB Assets
Exhibit 1011
Page 804

network such as a neural network or an HMM. In all cases, once the model is set, the parameters are learned automatically from data.

15.6.4.1. Transplanted Prosody

The most direct approach of all is to store a single contour from a real speaker's utterance corresponding to every possible input utterance that one's TTS system will ever face. This seems to limit the ability to freely synthesize any input text. However, this approach can be viable under certain special conditions and limitations. These controls are so detailed that they are tedious to write manually. Fortunately, they can be generated automatically by speech recognition algorithms.

When these controls (*transplanted prosody*), taken from an authentic digitized utterance, are applied to synthetic voice units, the results can be very convincing. sometimes nearly as good as the original digitized samples [43]. A system with this capability can mix predefined utterances having natural-quality prosody, such as greetings, with flexible synthesis capabilities for system response, using a consistent synthetic voice. The transplanted prosody for the *frozen* phrases can be derived either from the original voice data donor used to create the synthetic voice model, or any other speaker, with global adjustment for pitch-range differences. Another use of the transplanted prosody capability is to compress a spoken message into ASCII (phone labels plus the prosodic controls) for playback, preserving much of the quality, if not the full individuality, of the original speaker's recording.

15.6.4.2. F0 Contours Indexed by Parsed Text

In a more generalized variant of the direct approach, once could imagine collecting and indexing a gigantic database of clauses, phrases, words, or syllables, and then annotating all units with their salient prosodic features. If the terms of annotation (word structure, POS, syntactic context, etc.) can be applied to new utterances at runtime, a prosodic description for the closest matching database unit can be recovered and applied to the input utterance [23]. The advantages here are that prosodic quality can be made arbitrarily high, by collecting enough exemplars to cover arbitrarily large quantities of input text, and that detailed analysis of the deeper properties of the prosodic phenomena can be sidestepped. The potential disadvantages are:

- Data-collection time is long (which affects the capability to create new voices).
- A large amount of runtime storage is required (presumably less important as technology progresses).
- Database annotation may have to be manual, or if automated, may be of poor quality.
- The model cannot be easily modified/extended, owing to lack of fundamental understanding.

Amazon/VB Assets
Exhibit 1011
Page 805

- Coverage can never be complete, therefore rulelike generalization, fuzzy match capability, or back-off, is needed.
- Consistency control for the prosodic attributes (to prevent unit boundary mismatches) can be difficult.

The first two disadvantages are self-explanatory. The difficulty of annotating the database, to form the basis of the indexing and retrieval scheme, depends on the type and depth of the indexing parameters chosen. Any such scheme requires annotations to identify the component phones of each unit (syllable, word, or phrase) and their durations. This can usually be obtained from speech recognition tools [23], which may be independently required to create a synthetic voice (see Chapter 16). Lexical or word stress attributes can be extracted from an online dictionary or NLP system, though, as we have seen above, lexical stress is neither a necessary nor a sufficient condition for predicting pitch accent placement.

If only a very high level of description is sought, based primarily on the pragmatics of utterance use and some syntactic typology, it may not be necessary to recover a detailed symbolic pitch analysis. An input text can be described in high-level pragmatic/semantic terms, and pitch from the nearest matching word or phrase from the database can be applied with the expectation that its contour is likely correct. For example, such a system might have multiple prosodic versions of a word that can be used in different pragmatic senses, such as *ok*, which could be a question, a statement, an exclamation, a signal of hesitation or uncertainty, etc. The correct version must be selected based on the runtime requirements of the application.

Direct prosody schemes of this type often preserve the original phone carrier material of each instance in order to assure optimal match between prosody and spectrum. However, with DSP techniques enabling arbitrary modifications of waveforms (see Chapter 16), this is not strictly necessary; the prosodic annotations could stand alone, with phone label annotation only. If more detailed prosodic control is required, such as being aware of the type of accent, its pitch range, prominence, and other features, the annotation task is much more difficult.

A straightforward and elegant formulation of the lookup-table direct model approach can be found in [30]. This system, created for Spanish but generally adaptable, is based on a large single-speaker recorded database of a variety of sentence types. The sentences are linguistically analyzed, and prosodic structure is hypothesized based on syllables, accent groups (groups of syllables with one lexical stress), breath groups (groups of accent-groups between pauses regardless of the duration of the pause), and sentences. Note that these structures are hypothesized based on the textual material alone, and the speaker will not always perform accordingly. Pitch (initial, mid, and final F0) and duration data for each spoken syllable is automatically measured and stored. At runtime, the input sentence is analyzed using the same set of structural attributes inferred from the text, and a vector of candidate syllables from the database is constructed with identical, or similar, structural context and attributes for each successive input syllable position.

The best path through the set of such candidates is selected by minimizing the F0 distance and disjuncture across the utterance. This is a clean and simple approach to jointly

Amazon/VB Assets
Exhibit 1011
Page 806

utilizing both shallow linguistic features and genuine phonetic data (duration and F0), with dynamic programming to smooth and integrate the output F0 contour. However, as with any direct modeling approach, it lacks significant generalization capabilities outside the textual material and speaking style specified during the data collection phase, so a number of separate models may have to be constructed.

The CHATR system of ATR (Japan) [9] takes a similar approach, in that optimal prosody is selected from prerecorded units, rather than synthesized from a more general model. The CHATR system consists of a large database of digitized speech, indexed by the speaker identity, the phoneme sequences of the words, and some pragmatic and semantic attributes. Selection of phonemes proceeds by choosing the minimal-cost path from among the similarly indexed database candidate units available for each phoneme or longer segment of speech to be synthesized. This system achieves high quality by allowing the carrier phones to bear only their original prosody—pitch modification of the contour is minimized or eliminated. Of course, the restriction of DSP modification implies a limitation of the generalizability of the database. This type of approach obtains the prosody implicitly from the database [31], and as such combines both the prosody and speech synthesis modules. This type of minimal-cost search is described in more detail in Chapter 16.

15.6.4.3. F0 Contours Indexed by ToBI

The architecture for a simple and straightforward direct model indexed by ToBI is diagrammed in Figure 15.13. This model combines the two often-conflicting goals: it is empirically (corpus) based, but it permits specification in terms of principled abstract prosodic categories. In this model, an utterance to be synthesized is annotated in terms of its linguistic features—perhaps POS, syntactic structure, word emphasis (based on information structure), etc. The utterance so characterized is matched against a corpus of actual utterances that are annotated with linguistic features and ToBI symbols, Corpus (a). A *fuzzy* matching capability based on edit distance or dynamic programming can be incorporated. If Corpus (a) is sufficiently large and varied, a number of possible ToBI renderings of either the entire utterance or selected parts of it may be recovered. At this level of abstraction, the ToBI labels would not encode relative prominence specifications (strength of pitch extrusions) or pitch range. The set of such abstractly described contours can then be fuzzy matched into Corpus (b), a set of ToBI annotated actual contours, and the best set of matches recovered.

Note that while it is possible that Corpus (a) is the exact same base material as Corpus (b), the model does not enforce an identity, and there may be reasons to desire such flexibility and modularity, depending on the degree and quality of data and annotation at each level. Once a number of likely actual contours have been identified, they can be passed to a voice-unit selection module. The module can select the combination of segmental strings (sometimes called 'long units,' since they may combine more than one phoneme) from the voice database whose original prosody is closest to one of the candidate contours, using root-mean-square-error, correlation, or other statistical tests. Those units are then concatenated (with their prosody unmodified) and sent to the application or played out.

Amazon/VB Assets
Exhibit 1011
Page 807

A model of this type has some of the disadvantages of direct models as listed above. It also assumes availability of large and varied databases of both prosodic contours and segmental (phone) long units for concatenation (see Chapter 16). It further requires that these databases be annotated, either by human labelers or automated systems. However, it has certain advantages as well:

- It allows for symbolic, phonological coding of prosody.
- It has high-quality natural contours.
- It has high-quality phonetic units, with unmodified pitch.
- Its modular architecture can work with user-supplied prosodic symbols.

It also allows the immediate, temporary use of data that is collected for deeper analysis, in the hope of eventual construction of smaller, parametrized models. The model of Figure 15.13 is a generalization of the prosody system described in [23].

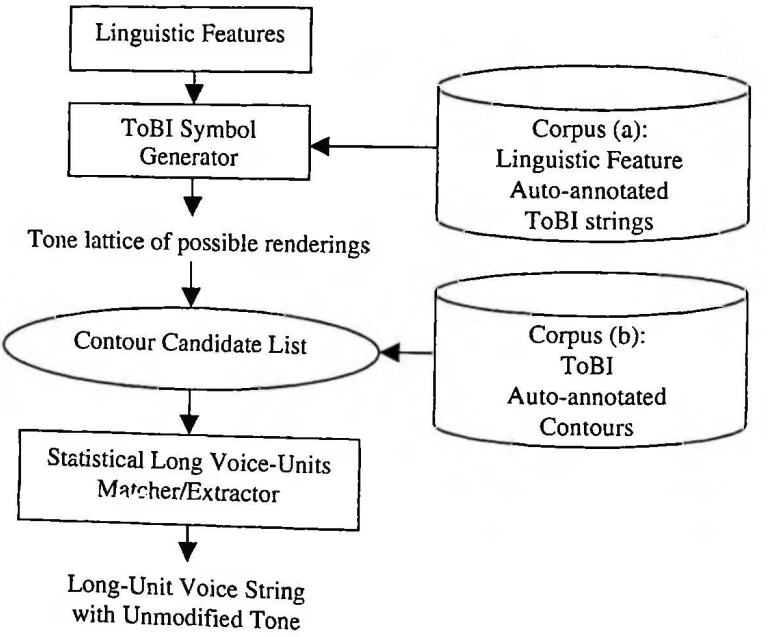


Figure 15.13 A corpus-based prosodic generation model.

Amazon/VB Assets
Exhibit 1011
Page 808

15.7. PROSODY MARKUP LANGUAGES

Chapter 14 discussed generalized document markup schemes for text analysis. Most TTS engines provide simple text tags and application programming interface controls that allow at least rudimentary hints to be passed along from an application to a TTS engine. We expect to see more sophisticated speech-specific annotation systems, which eventually incorporate current research on the use of semantically structured inputs to synthesizers, sometimes called concept-to-speech systems. A standard set of prosodic annotation tags would likely include tags for insertion of silence pause, emotion, pitch baseline and range, speed in words-per-minute, and volume. This would be in addition to general tags for specifying the language of origin if not predictable, character of the voice, and text normalization context such as address, date, email, etc.

For prosodic processing, text may be marked with tags that have scope, in the general fashion of XML. Some examples of the form and function of a few common TTS tags for prosodic processing, based loosely on the proposals of [65], are introduced below. Other tags can be added by intermediate subcomponents to indicate variables such as accents and tones. This extension allows for even finer research and prosody models.

- **Pause** or **Break** commands might accept either an absolute duration of silence in milliseconds, or, as in the W3C proposal, a mnemonic describing the relative salience of the pause (Large, Medium, Small, None), or a *prosodic punctuation* symbol from the set ',', '.', '?', '!', '...', etc., which not only indicates a pause insertion but also influences the typical pitch contour of the phone segments entering and leaving the pause area. For example, specifying ',' as the argument of a Pause command might determine the use of a continuation rise on the phones immediately preceding the pause, indicating incompletion or listing intonation.
- **Rate** controls the speed of output. The usual measurement is *words per minute*, which can be a bit vague, since words are of very different durations. However, this metric is familiar to many TTS users and works reasonably well in practice. For non-IndoEuropean languages, different metrics must be contemplated. Some power listeners who use a TTS system routinely can tolerate (in fact, demand) rates of over 300 words per minute, while 150 or fewer might be all that a novice listener could expect to reliably comprehend.
- **Baseline Pitch** specifies the desired average pitch: a level around which, or up from which, pitch is to fluctuate.
- **Pitch Range** specifies within what bounds around the baseline pitch level line the pitch is to fluctuate.
- **Pitch** commands can override the system's default prosody, giving an application or document author greater control. Generally, TTS engines require some freedom to express their typical pitch patterns within the broad limits specified by a Pitch markup.

Amazon/VB Assets
Exhibit 1011
Page 809

- **Emphasis** emphasizes or deemphasizes one or more words, signaling their relative importance in an utterance. Its scope could be indicated by XML style. Control over emphasis brings up a number of interesting considerations. For one thing, it may be desirable to have degrees of emphasis. The notion of gradient prominence—the apparent fact that there are no categorical constraints on levels of relative emphasis or accentuation—has been a perpetual thorn in the side for prosodic researchers. This means that in principle any positive real number could be used as an argument to this tag. In practice, most TTS engines would artificially constrain the range of emphasis to a smaller set of integers, or perhaps use semantic labels, such as *strong*, *moderate*, *weak*, *none* for degree of emphasis. Emphasis may be realized with multiple phonetic cues. Thus, if the user or application has, for example, set the pitch range very narrowly, the emphasis effect may be achieved by manipulation of segmental duration or even relative amplitude. The implementation of emphasis by a TTS engine for a given word may involve manipulation (e.g., de-accentuation) of surrounding words as much as it involves heightening the pitch or volume, or stretching the phone durations, of the target word itself. In most cases the main phonetic and perceptual effect of emphasis or accentuation is heard on the lexically main stressed syllable of the word, but this can be violated under special conditions of semantic focus, e.g., "*I didn't say employer, I said employee.*" This would require a more powerful emphasis specification than is currently provided in most TTS systems, but alternatively it could be specified using phone input commands such as "*The <emp>truth</emp>, the <emp>whole truth</emp>, and nothing <emp> but</emp> the truth.*" For more control, future TTS systems may support degree emphasis: "*... nothing <emp level="strong">but</emp> the truth*" or even deemphasis: "*... nothing <emp level= "reduced">but</emp> the truth*". Emphasis is related to prominence, discussed in Section 15.6.1.2.

15.8. PROSODY EVALUATION

Evaluation of a complete TTS system is discussed in Chapter 16. We limit ourselves here to evaluating the prosody component. We assume that the text analysis module has done a perfect job, and that the synthesis module does a perfect job, which cannot be done in general, so that approximations need to be made.

Evaluation can be done automatically or by using listening tests with human subjects. In both cases it's useful to start with some natural recordings with their associated text. We start by replacing the natural prosody with the system's synthetic prosody. In the case of automatic evaluation, we can compare the enriched prosodic representations described in Section 15.2 for both the natural recording and the synthetic prosody. The reference en-

Amazon/VB Assets
Exhibit 1011
Page 810

riched prosodic representation can be obtained either manually or by using a pitch tracker and a speech recognizer.

Automated testing of prosody involves the following:

- **Duration**. It can be performed by measuring the average squared difference between each phone's actual duration in a real utterance and the duration predicted by the system.
- **Pitch** contours. It can be performed by using standard statistical measures over a system contour and a natural one. When this is done, duration and phoneme identity should be completely controlled. Measures such as root-mean-square error indicate the characteristic divergence between two contours, while correlation indicates the similarity in shape across difference pitch ranges. In general, RMSE scores of 15 Hz or less for male speech over a long sentence, with correlation of .8 or above, indicate quality that may be close to perceptually identical to the natural reference utterance. In general, such exactness of match is useful only during model training and testing and cannot be expected during training on entirely new utterances from random text.

Listening tests can be performed to evaluate a prosody module. This involves subjects listening to the natural recording and the synthetic speech, or to synthetic speech generated with two different prosody modules. This can lead to a more precise evaluation, as humans are the final consumer of this technology. However, such tests are more expensive to carry out. Furthermore, this method results in testing both the prosody module and the synthesis components together. To avoid this, the original waveform can be modified to have the synthetic prosody using the signal processing techniques described in Chapter 16. Since such techniques introduce some distortions, this measuring method is still somewhat biased. In practice, it has been shown that its effect is much smaller than that of the synthetic prosody [43].

It is shown that synthesizing pitch is more difficult than duration [43]. Subjects scored significantly higher utterances that had natural pitch and synthetic duration than utterances with synthetic pitch and natural duration. In fact, using only synthetic duration had a score quite close to that of the original recording. While duration modeling is not a solved problem, this indicates that generation of pitch contours is more difficult.

15.9. Historical Perspective and Further Reading

Prosodic methods have been incorporated within the traditional fields of rhetoric and elocution for centuries. In ancient Greece, at the time of Plato, written documentation in support of claims in legal disputes was rare. To help litigants plead their cases persuasively, systematic programs of rhetorical instruction were established, which included both content and form of verbal argument. This 'prescriptive' tradition of systematic instruction in verbal

Amazon/VB Assets
Exhibit 1011
Page 811

style uncovered issues that remain central to the descriptively oriented prosodic research of today. A masterful and entertaining discussion of this tradition and its possible relevance to the task of teaching computers to *plead a case* can be found in [64]. The Greeks were particularly concerned about an issue that, as usual, is still important for us today: the separation of rhetorical effectiveness from considerations of truth. If you are interested in this, you cannot do better than to begin with Plato's dialog *Phaedrus* [42].

Modern linguists have also considered a related, but more narrowly formulated question: Should prosody be treated as a logical, categorical analog to phonological and syntactic processes? The best discussion of these issues from a prosodic (as opposed to strictly neurological) point of view is found in [7, 8]. If you are interested in the neurological side, you can begin with [13]. For emotional modeling, before slogging through the scattered and somewhat disjointed papers on emotion in speech that have appeared sporadically for years, the reader would be well advised to get a basic grounding in some of the issues related to emotion in computation, as treated in [38].

Going in the other direction, there are many subtle interactions in the phonetics of prosody: the various muscles, their joint possibilities of operation in phonation and articulation, as well as the acoustics properties of the vocal chambers. For an excellent introduction to the whole field, start with [27].

The most complete and accessible overview of modern prosodic analysis as embedded in mainstream linguistic theory is Ladd's *Intonational Phonology* [26], which covers the precursors, current standard practice, and remaining unsolved issues of the highly influential auto segmental theory of intonational phonology, from which ToBI has arisen. ToBI was devised by speech scientists who wanted a prosodic transcription standard to enable sharing of databases [4]. For most practical purposes, the ToBI definitions are sufficient as a starting point for both research and applications, but for those who prefer alternative annotation systems aligned with the British tradition, conversion guidelines have been attempted [45]. Another major phonological approach to English intonation has been the British school described in [11]. Bridging the two is IViE, a labeling system that is philosophically aligned with ToBI but may be more appropriate for non-U.S. dialects of English [18].

The first prosodic synthesis by rule was developed by Ignatius Mattingly in 1968 in Haskins Laboratories. In 1971, Fujisaki [15] developed his superposition model that has been used for many years. The development of the ToBI in 1992 [55] marked a milestone in automatic prosody generation. The application of statistical techniques, such as CART, for phoneme durations during the 1990s constituted a significant step beyond the rule-based methods. Finally, the development of the CHATR system in the mid-1990s ignited interest in the indexing of massive databases. It is possible to attempt smoothing over both the index space and the resulting prosodic data tracks by means of generalized learning methods, such as neural nets or HMMs. These models have built-in generalization over unseen inputs, and built-in smoothing over the concatenated outputs of unit selection. The network described in [57] codes every syllable in a training database in terms of perceived prominence (human judged), a number from 1 to 31, as well as the syllable's phonemes, rising/falling boundary type for phrase-edge syllables, and distance from preceding and following phrase bounda-

Amazon/VB Assets
Exhibit 1011
Page 812

ries, for all syllables. When tested with reasonably simple text material of similar type, these networks yielded high-quality simulations.

A potential research area for future generalizations of this system is to increase the degree and accuracy of automation in labeling the training features of the recordings, such as perceived prominence. Another area is to either expand the inventory of model types, or to determine adequate generalization mechanisms. By training HMMs on accented syllables of differing phonetic structure, some of this fine alignment information can be automatically captured [16]. Another approach consists in generating pitch contours directly from a hidden Markov model, which is run in generation mode [66].

Recently, just as in speech synthesis for voice, there has been a realization that the *direct* and *parametric* prosodic models have a great deal in common. Direct models require huge databases of indexed exemplars for unmediated concatenation and playback of contours, in addition to generalized back-off methods, while parametric models are generalized for any input, but also require phonetic databases of sufficient variety to support statistical learning of parameter settings for high quality. We can, therefore, expect to see increasing numbers of hybrid systems. One such system is described in [47], which could be viewed as a stochastic realization of Fujisaki's superposition model without the phrase controls, where the accents are given by ToBI labels and the smoothing is done by means of a dynamical system.

While this chapter has focused on U.S. English, many similar issues arise in prosodic modeling of other languages. An excellent survey of the prosodic systems of every major European language, as well as Arabic and several major East Asian languages, can be found in [21].

Though not explicitly covered in this chapter, analysis of prosody for speech recognition is a small but growing area of study. Anyone who has digested this chapter should be prepared to approach the more specialized work of [25, 37] and the speech recognition prosody studies collected in [48]. Those with a psycholinguistic bent can begin with [12].

REFERENCES

[1] Allen, J., M.S. Hunnicutt, and D.H. Klatt, *From Text to Speech: the MITalk System*, 1987, Cambridge, UK, University Press.

[2] Anderson, M.D., J.B. Pierrehumbert, and M.Y. Liberman, "Synthesis by Rule of English Intonation Patterns," *Proc. of Int. Conf. on Acoustics, Speech and Signal Processing*, 1984, pp. 2.8.1-2.8.4.

[3] Arnfield, S., "Word Class Driven Synthesis of Prosodic Annotations," *Proc. of the Int. Conf. on Spoken Language Processing*, 1996, Philadelphia, PA, pp. 1978-1981.

[4] Beckman, M.E. and G.M. Ayers, *Guidelines for ToBI Labelling*, 1994, http://www.ling.ohio-state.edu/phonetics/ToBI/main.html.

[5] Black, A. and A. Hunt, "Generating F0 Contours from ToBI labels using Linear Regression," *Proc. of the Int. Conf. on Spoken Language Processing*, 1996, pp. 1385-1388.

Amazon/VB Assets
Exhibit 1011
Page 813

[6] Bolinger, D., "Accent is predictable (if you're a mind-reader)," *Language*, 1972, **48**, pp. 633-44.

[7] Bolinger, D., *Intonation and its parts*, 1986, Stanford, Stanford University Press.

[8] Bolinger, D., *Intonation and its uses*, 1989, Stanford, Stanford University Press.

[9] Campbell, N., "CHATR: A High-Definition Speech Re-sequencing System," *ASA/ASJ Joint Meeting*, 1996, Honolulu, Hawaii, pp. 1223-1228.

[10] Carroll, L., *Alice in Wonderland, Unabridged ed.*, 1997, Penguin USA.

[11] Crystal, D., *Prosodic Systems and Intonation in English*, 1969, Cambridge University Press.

[12] Crystal, D., "Prosody and Parsing," P. Warren, Editor, 1996, Lawrence Erlbaum Associates.

[13] Emmorey, K., "The Neurological Substrates for Prosodic Aspects of Speech," *Brain and Language*, 1987, **30**, pp. 305-320.

[14] Fujisaki, H., "Prosody, Models, and Spontaneous Speech" in *Computing Prosody*, Y. Sagisaka, N. Campbell, N. Higuchi, Editors, 1997, New York, Springer.

[15] Fujisaki, H. and H. Sudo, "A Generative Model of the Prosody of Connected Speech in Japanese," *Annual Report of Eng. Research Institute*, 1971, **30**, pp. 75-80.

[16] Fukada, T., *et al.*, "A Study on Pitch Pattern Generation Using HMM-based Statistical Information," *Proc. Int. Conf. on Spoken Language Processing*, 1994, Yokohama, Japan, pp. 723-726.

[17] Goldsmith, J., "English as a Tone Language" in *Phonology in the 1980's*, D. Goyvaerts, Editor 1980, Ghent, Story-Scientia.

[18] Grabe, E., F. Nolan, and K. Farrar, "IViE - a Comparative Transcription System for Intonational Variation in English," *Proc. of the Int. Conf. on Spoken Language Processing*, 1998, Sydney, Australia.

[19] Hirose, H. and H. Fujisaki, "Analysis and Synthesis of Voice Fundamental Frequency Contours of Spoken Sentences," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1982, pp. 950-953.

[20] Hirschberg, J., "Pitch Accent in Context: Predicting Intonational Prominence from Text," *Artificial Intelligence*, 1993, **63**, pp. 305-340.

[21] Hirst, D., A.D. Cristo, and A. Cruttenden, *Intonation Systems: A Survey of Twenty Languages*, 1998, Cambridge, U.K., Cambridge University Press.

[22] Hirst, D.J., "The Symbolic Coding of Fundamental Frequency Curves: from Acoustics to Phonology," *Proc. of Int. Symposium on Prosody*, 1994, Yokohama, Japan.

[23] Huang, X., *et al.*, "Whistler: A Trainable Text-to-Speech System," *Int. Conf. on Spoken Language Processing*, 1996, Philadephia, PA, pp. 2387-2390.

[24] Klasmeyer, G. and W.F. Sendlmeier, "The Classification of Different Phonation Types in Emotional and Neutral Speech," *Forensic Linguistics*, 1997, **4**(1), pp. 104-125.

[25] Kompe, R., *Prosody in Speech Understanding Systems*, 1997, Berlin, Springer.

Amazon/VB Assets
Exhibit 1011
Page 814

[26] Ladd, R.D., *Intonational Phonology*, Cambridge Studies in Linguistics, 1996, Cambridge, Cambridge University Press.

[27] Ladefoged, P., *A Course in Phonetics*, 1993, Harcourt Brace Jovanovich.

[28] Liberman, M., *The Intonation System of English*, PhD Thesis in Linguistics and Philosophy, 1975, MIT, Cambridge.

[29] Liberman, M. and J. Pierrehumbert, "Intonational Invariance under Changes in Pitch Range and Length" in *Language and Sound Structure*, M. Aronoff, Oerhle, R., ed., 1984, Cambridge, MA, MIT Press, pp. 157-233.

[30] Lopez-Gonzalo, E., and J.M. Rodriguez-Garcia, "Statistical Methods in Data-Driven Modeling of Spanish Prosody for Text to Speech," *in Proc. ICSLP 1996*, 1996, pp. 1373-1376.

[31] Malfrere, F., T. Dutoit, and P. Mertens, "Automatic Prosody Generation Using Supra-Segmental Unit Selection," *Third ESCA/COCOSCA Workshop on Speech Synthesis*, 1998, Jenolan Caves, Australia, pp. 323-328.

[32] Mason, J., *An Essay on Elocution*, 1st ed, 1748, London.

[33] Möbius, B., "Analysis and Synthesis of German F0 Contours by Means of Fujisaki's Model," *Speech Communication*, 1993, **13**(53-61).

[34] Murray, I. and J. Arnott, "Toward the Simulation of Emotion in Synthetic Speech: A Review of the Literature on Human Vocal Emotion," *Journal Acoustical Society of America*, 1993, **93**(2), pp. 1097-1108.

[35] Öhman, S., *Word and Sentence Intonation: A Quantitative Model*, 1967, KTH, pp. 20-54.

[36] Ostendorf, M., and N. Veilleux, "A Hierarchical Stochastic Model for Automatic Prediction of Prosodic Boundary Location," *Computational Linguistics*, 1994, **20**(1), pp. 27-54.

[37] Ostendorf, M., "Linking Speech Recognition and Language Processing Through Prosody," *Journal for the Integrated Study of Artificial Intelligence, Cognitive Science and Applied Epistemology*, 1998, **15**(3), pp. 279-303.

[38] Picard, R.W., *Affective Computing*, 1997, MIT Press.

[39] Pierrehumbert, J., *The Phonology and Phonetics of English Intonation*, PhD Thesis in *Linguistics and Philosophy* 1980, MIT, Cambridge, MA.

[40] Pierrehumbert, J., and M. Beckman, *Japanese Tone Structure*, 1988, Cambridge, MA, MIT Press.

[41] Pierrehumbert, J. and J. Hirschberg, "The Meaning of Intonational Contours in the Interpretation of Discourse" in *Intentions in Communication*, P.R. Cohen, J. Morgan, and M. E. Pollack, ed., 1990, Cambridge, MA, MIT Press.

[42] Plato, *The Symposium and The Phaedrus: Plato's Erotic Dialogues*, 1994, State University of New York Press.

[43] Plumpe, M. and S. Meredith, "Which is More Important in a Concatenative Text-to-Speech System: Pitch, Duration, or Spectral Discontinuity," *Third ESCA/COCOSDA Int. Workshop on Speech Synthesis*, 1998, Jenolan Caves, Australia, pp. 231-235.

Amazon/VB Assets
Exhibit 1011
Page 815

[44] Prevost, S. and M. Steedman, "Specifying Intonation from Context for Speech Synthesis," *Speech Communication*, 1994, **15**, pp. 139-153.

[45] Roach, P., "Conversion between Prosodic Transcription Systems: 'Standard British' and ToBI," *Speech Communication*, 1994, **15**, pp. 91-97.

[46] Ross, K. and M. Ostendorf, "Prediction of Abstract Prosodic Labels for Speech Synthesis," *Computer, Speech and Language*, 1996, **10**, pp. 155-185.

[47] Ross, K. and M. Ostendorf, "A Dynamical System Model for Generating Fundamental Frequency for Speech Synthesis," *IEEE Trans. on Speech and Audio Processing*, 1999, **7**(3), pp. 295-309.

[48] Sagisaka, Y., W.N. Campbell, and N. Higuchi, *Computing Prosody*, 1997, Springer-Verlag.

[49] Santen, J.V., "Contextual Effects on Vowel Duration," *Speech Communication*, 1992, **11**(6), pp. 513-546.

[50] Selting, M., *Prosodie im Gespräch*, 1995, Max Niemeyer Verlag.

[51] Shattuck-Hufnagel, S. and M. Ostendorf, "Stress Shift and Early Pitch Accent Placement in Lexical Items in American English," *Journal of Phonetics*, 1994, **22**, pp. 357-388.

[52] Shen, X.-n.S., *The Prosody of Mandarin Chinese*, 1990, Berkeley, University of California Press.

[53] Sheridan, T., *Lectures on the Art of Reading*, 3rd ed, 1787, London, Dodsley.

[54] Silverman, K., *The Structure and Processing of Fundamental Frequency Contours*, Ph.D. Thesis, 1987, University of Cambridge, Cambridge, UK.

[55] Silverman, K., "ToBI: A Standard for Labeling English Prosody," *Int. Conf. on Spoken Language Processing*, 1992, Banff, Canada, pp. 867-870.

[56] Sokal, A.D., "Transgressing the Boundaries: Towards a Transformative Hermeneutics of Quantum Gravity," *Social Text*, 1996, **46/47**, pp. 217-252.

[57] Sonntag, G., T. Portele, and B. Heuft, "Prosody Generation with a Neural Network: Weighing the Importance of Input Parameters," *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, 1997, pp. 930-934.

[58] Steedman, M., "Information Structure and the Syntax-Phonology Interface," *Linguistic Inquiry*, 2000.

[59] Stevens, K.N., "Control Parameters for Synthesis by Rule," *Proc. of the ESCA Tutorial Day on Speech Synthesis*, 1990, pp. 27-37.

[60] Taylor, P.A., "The Tilt Intonation Model," *Proc. Int. Conf. on Spoken Language Processing*, 1998, Sydney, Australia.

[61] van Santen, J., "Assignment of Segmental Duration in Text-to-Speech Synthesis," *Computer Speech and Language*, 1994, **8**, pp. 95-128.

[62] van Santen, J., "Segmental Duration and Speech Timing" in *Computing Prosody*, Y. Sagisaka, N. Campbell, and N. Higuchi, eds., 1997, New York, Springer, pp. 225-250.

[63] van Santen, J. and J. Hirschberg, "Segmental Effects of Timing and Height of Pitch Contours," *Proc. of the Int. Conf. on Spoken Language Processing*, 1994, pp. 719-722.

Amazon/VB Assets
Exhibit 1011
Page 816

[64] Vanderslice, R.L., *Synthetic Elocution: Considerations in Automatic Orthographic-to-Phonetic Conversion of English with Special Reference to Prosody*, PhD Thesis, 1968, UCLA, Los Angeles.

[65] W3C, *Speech Synthesis Markup Requirements for Voice Markup Languages*, 2000, http://www.w3.org/TR/voice-tts-reqs/.

[66] Yoshimura, T., *et al.*, "Simultaneous Modeling of Spectrum, Pitch and Duration in HMM-Based Speech Synthesis," *EuroSpeech*, 1999, Budapest, Hungary, pp. 2347-2350.

Amazon/VB Assets
Exhibit 1011
Page 817

CHAPTER 16

Speech Synthesis

The speech synthesis module of a TTS system is the component that generates the waveform. The input of traditional speech synthesis components is a phonetic transcription with its associated prosody. The input can also include the original text with tags, as this may help in producing higher-quality speech.

Speech synthesis can be classified into three types according to the model used in the speech generation. *Articulatory synthesis*, described in Section 16.2.4, uses a physical model of speech production that includes all the articulators described in Chapter 2. *Formant synthesis* uses a source-filter model, where the filter is characterized by slowly varying formant frequencies; it is the subject of Section 16.2. *Concatenative synthesis* generates speech by concatenating speech segments and is described in Section 16.3. To allow more flexibility in concatenative synthesis, a number of prosody modification techniques are described in Sections 16.4 and 16.5. Finally, a guide to evaluating speech synthesis systems is included in Section 16.6.

Amazon/VB Assets
Exhibit 1011
Page 819

Speech synthesis can also be classified according to the degree of manual intervention in the system design into *synthesis by rule* and *data-driven synthesis*. In the former, a set of manually derived rules is used to drive a synthesizer, and in the latter the synthesizer's parameters are obtained automatically from real speech data. Concatenative systems are, thus, data driven. Formant synthesizers have traditionally used synthesis by rule, since the evolution of formants in a formant synthesizer has been done with hand-derived rules. Nonetheless, formant transitions can also be trained from data, as we show in Section 16.2.3.

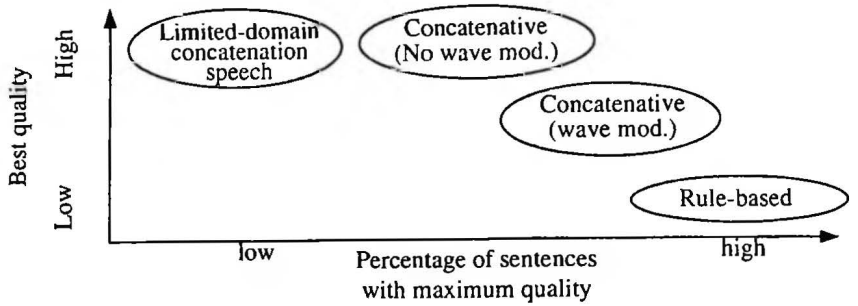


Figure 16.1 Quality and task-independence in speech synthesis approaches.

16.1. ATTRIBUTES OF SPEECH SYNTHESIS

The most important attribute of a speech synthesis system is the quality of its output speech. It is often the case that a single system can sound beautiful on one sentence and terrible on the next. For that reason we need to consider the quality of the best sentences and the percentage of sentences for which such quality is achieved. This tradeoff is illustrated in Figure 16.1, where we compare four different families of speech generation approaches:

- *Limited-domain waveform concatenation.* For a given limited domain, this approach can generate very high quality speech with only a small number of recorded segments. Such an approach, used in most interactive voice response systems, cannot synthesize arbitrary text. Many concept-to-speech systems, described in Chapter 17, use this approach.
- *Concatenative synthesis with no waveform modification.* Unlike the previous approach, these systems can synthesize speech from arbitrary text. They can achieve good quality on a large set of sentences, but the quality can be mediocre for many other sentences where poor concatenations take place.

Amazon/VB Assets
Exhibit 1011
Page 820

- *Concatenative systems with waveform modification.* These systems have more flexibility in selecting the speech segments to concatenate because the waveforms can be modified to allow for a better prosody match. This means that the number of sentences with mediocre quality is lower than in the case where no prosody modification is allowed. On the other hand, replacing natural with synthetic prosody can hurt the overall quality. In addition, the prosody modification process also degrades the overall quality.
- *Rule-based systems.* Such systems tend to sound uniformly across different sentences, albeit with quality lower than the best quality obtained in the systems above.

Best-quality and quality variability are possibly two of the most important attributes of a speech synthesis system, but not the only ones. Measuring quality, difficult to do in an objective way, is the main subject of Section 16.6. Other attributes of a speech synthesis system include:

- *Delay.* The time it takes for the synthesizer to start speaking is important for interactive applications and should be less than 200 ms. This delay is composed of the algorithmic delays of the front end and of the speech synthesis module, as well as the computation involved.
- *Memory resources.* Rule-based synthesizers require, on the average, less than 200 KB, so they are a widely used option whenever memory is at a premium. However, required RAM can be an issue for concatenative systems, some of which may require over 100 MB of storage.
- *CPU resources.* With current CPUs, processing time is typically not an issue, unless many channels need to run in the same CPU. Nonetheless, some concatenative synthesizers may require a large amount of computation when searching for the optimal sequence.
- *Variable speed.* Some applications may require the speech synthesis module to generate variable speed, particularly fast speech. This is widely used by blind people who need TTS systems to obtain their information and can accept fast speech because of the increased throughput. Fast speech is also useful when skimming material. Concatenative systems that do not modify the waveform cannot achieve variable speed control, unless a large number of segments are recorded at different speeds.
- *Pitch control.* Some spoken language systems require the output speech to have a specific pitch. This is the case if you want to generate voice for a song. Again, concatenative systems that do not modify the waveform cannot do this, unless a large number of speech segments are recorded at different pitch.

Amazon/VB Assets
Exhibit 1011
Page 821

- *Voice characteristics.* Other spoken language systems require specific voices, such as that of a robot, that cannot be recorded naturally, or some, such as monotones, that are tedious to record. Since rule-based systems are so flexible, they are able to do many such modifications.

The approaches described in this chapter assume as input a phonetic string, durations, a pitch contour, and possibly volume. Pauses are signaled by the default phoneme SIL with its corresponding duration. If the parsed text is available, it is possible to do even better in a concatenative system by conducting a matching with all the available information.

16.2. FORMANT SPEECH SYNTHESIS

As discussed in Chapter 6, we can synthesize a stationary vowel by passing a glottal periodic waveform through a filter with the formant frequencies of the vocal tract. For the case of unvoiced speech we can use white random noise as the source instead. In practice, speech signals are not stationary, and we thus need to change the pitch of the glottal source and the formant frequencies over time. The so-called *synthesis-by-rule* refers to a set of rules on how to modify the pitch, formant frequencies, and other parameters from one sound to another while maintaining the continuity present in physical systems like the human production system. Such a system is described in the block diagram of Figure 16.2.

In Section 16.2.1 we describe the second block of Figure 16.2, the formant synthesizer that generates a waveform from a set of parameters. In Section 16.2.2 we describe the first block of Figure 16.2, the set of rules that can generate such parameters. This approach was the one followed by Dennis Klatt and his colleagues [4, 30]. A data-driven approach to this first block is studied in Section 16.2.3. Finally, articulatory synthesis is the topic of Section 16.2.4.

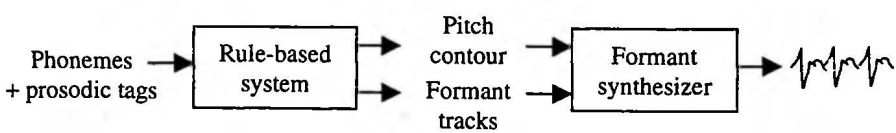


Figure 16.2 Block diagram of a synthesis-by-rule system. Pitch and formants are listed as the only parameters of the synthesizer for convenience. In practice, such system has about 40 parameters.

Amazon/VB Assets
Exhibit 1011
Page 822

16.2.1. Waveform Generation from Formant Values

To be able to synthesize speech by rule, a simple model for the filter is needed. Most rule-based synthesizers use the so-called *formant synthesis*, which is derived from models of speech production. The model explicitly represents a number of formant resonances (from 2 to 6). A formant resonance can be implemented (see Chapter 6) with a second-order IIR filter

$$H_i(z) = \frac{1}{1 - 2e^{-\pi b_i}\cos(2\pi f_i)z^{-1} + e^{-2\pi b_i}z^{-2}} \tag{16.1}$$

with $f_i = F_i / F_s$ and $b_i = B_i / F_s$, where F_i, B_i, and F_s are the formant's center frequency, formant's bandwidth, and sampling frequency, respectively, all in Hz. A filter with several resonances can be constructed by cascading several such second-order sections (*cascade model*) or by adding several such sections together (*parallel model*). Formant synthesizers typically use the parallel model to synthesize fricatives and stops and the cascade model for all voiced sounds.

Unlike the cascade model, the parallel model requires gains to be specified for each second-order section, which often are chosen proportional to the formant's frequency and inversely proportional to the formant's bandwidth. The cascade model results in an all-pole filter, whereas the parallel model has zeros in addition to poles. Such a combination is shown in Figure 16.3, where R1 through R6 are the resonances 1 to 6 and each one represents a second-order IIR filter like that in Eq. (16.1). RNP represents the nasal resonance, and RNZ is an FIR filter with the nasal zero. A1 through AB are the gains for each filter, used for the parallel model. Switch SW controls whether the cascade model or parallel model is used.

For voiced sounds the excitation model consists of an impulse train driving a low-pass filter RGP and then a bandpass filter created by the parallel of RGZ and RGS. For unvoiced sounds the excitation consists of white noise driving a low-pass filter LPF. The excitation for voiced fricatives is a combination of the two sources. In practice, this mixed excitation is used for all voiced sounds to add some breathiness. Klatt [30] showed that this model could reproduce quite faithfully a natural recording if the parameters had been manually selected.

Amazon/VB Assets
Exhibit 1011
Page 823

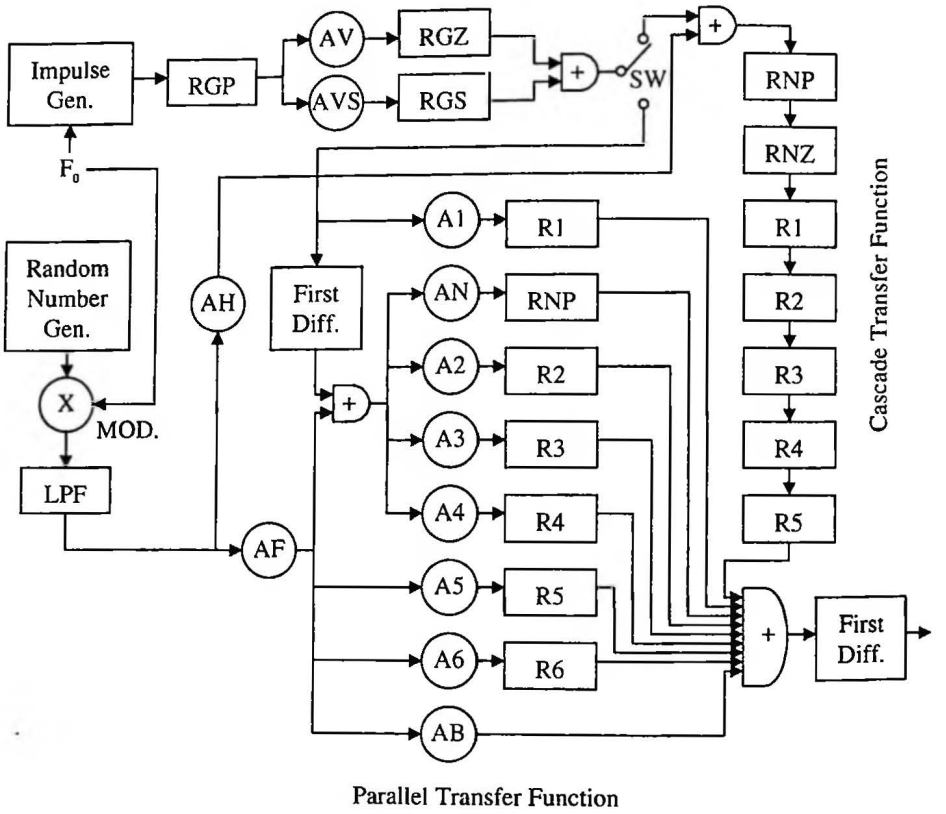


Figure 16.3 Block diagram of the Klatt formant synthesizer (after Allen [4]).

The parameter names and their minimum and maximum values are listed in Table 16.1, where the switch SW can be in voiced (V) or consonant (C) mode. For example, in Figure 16.3, R2 is the resonator corresponding to the second formant, whose center frequency F2 and bandwidth B2 are given in Table 16.1. In addition to the six resonances associated to the six formants, there are other resonances: RGP, RGZ, RGS, RNP, and RNZ. Other source models are also possible [43].

Amazon/VB Assets
Exhibit 1011
Page 824

Table 16.1 Parameter values for Klatt's cascade/parallel formant synthesizer with the parameter symbol, full name, minimum, maximum, and typical values (after Allen [4]).

N	Symbol	Name	Min	Max	Typ
1	AV	Amplitude of voicing (dB)	0	80	0
2	AF	Amplitude of frication (dB)	0	80	0
3	AH	Amplitude of aspiration (dB)	0	80	0
4	AVS	Amplitude of sinusoidal voicing (dB)	0	80	0
5	F0	Fundamental frequency (Hz)	0	500	0
6	F1	First formant frequency (Hz)	150	900	500
7	F2	Second formant frequency (Hz)	500	2500	1500
8	F3	Third formant frequency (Hz)	1300	3500	2500
9	F4	Fourth formant frequency (Hz)	2500	4500	3500
10	FNZ	Nasal zero frequency (Hz)	200	700	250
11	AN	Nasal formant amplitude (Hz)	0	80	0
12	A1	First formant amplitude (Hz)	0	80	0
13	A2	Second formant amplitude (Hz)	0	0	0
14	A3	Third formant amplitude (Hz)	0	80	0
15	A4	Fourth formant amplitude (Hz)	0	80	0
16	A5	Fifth formant amplitude (Hz)	0	80	0
17	A6	Sixth formant amplitude (Hz)	0	80	0
18	AB	Bypass path amplitude (Hz)	0	80	0
19	B1	First formant bandwidth (Hz)	40	500	50
20	B2	Second formant bandwidth (Hz)	40	500	70
21	B3	Third formant bandwidth (Hz)	40	500	110
22	SW	Cascade/parallel switch	0	1	0
23	FGP	Glottal resonator 1 frequency (Hz)	0	600	0
24	BGP	Glottal resonator 1 bandwidth (Hz)	100	2000	100
25	FGZ	Glottal zero frequency (Hz)	0	500	1500
26	BGZ	Glottal zero bandwidth (Hz)	100	9000	6000
27	B4	Fourth formant bandwidth (Hz)	100	500	250
28	F5	Fifth formant frequency (Hz)	3500	4900	3850
29	B5	Fifth formant bandwidth (Hz)	150	700	200
30	F6	Sixth formant frequency (Hz)	4000	4999	4900
31	B6	Sixth formant bandwidth (Hz)	200	2000	100
32	FNP	Nasal pole frequency (Hz)	200	500	250
33	BNP	Nasal pole bandwidth (Hz)	50	500	100
34	BNZ	Nasal zero bandwidth (Hz)	50	500	100
35	BGS	Glottal resonator 2 bandwidth (Hz)	100	1000	200
36	SR	Sampling rate (Hz)	500	20000	10000
37	NWS	Number of waveform samples per chunk	1	200	50
38	G0	Overall gain control (dB)	0	80	48
39	NFC	Number of cascaded formants	4	6	5

Amazon/VB Assets
Exhibit 1011
Page 825

16.2.2. Formant Generation by Rule

As described in Chapter 2, formants are one of the main features of vowels. Because of the physical limitations of the vocal tract, formants do not change abruptly with time. Rule-based formant synthesizers enforce this by generating continuous values for $f_i[n]$ and $b_i[n]$, typically every 5–10 milliseconds. Continuous values can be implemented through the above structures if a lattice filter is used, because it allows its reflection coefficients to vary at every sample (see Chapter 6). In practice, the values can be fixed within a frame as long as frames are smaller than 5 ms.

Rules on how to generate formant trajectories from a phonetic string are based on the *locus theory* of speech production. The locus theory specifies that formant frequencies within a phoneme tend to reach a stationary value called the *target*. Targets for formant frequencies and bandwidths for a male speaker are shown in Table 16.2 (nonvocalic segments) and Table 16.3 (vocalic segments). This target is reached if either the phoneme is sufficiently long or the previous phoneme's target is close to the current phoneme's target. The maximum slope at which the formants move is dominated by the speed of the articulators, determined by physical constraints. Since each formant is primarily caused by the position of a given articulator, formants caused by the body of the tongue do not vary as rapidly as formants caused by the tip of the tongue or the lips. Thus, rule-based systems store targets for each phoneme as well as maximum allowable slopes and transition times.

For example, a transition between a vowel and a sonorant can follow the rule shown in Figure 16.4 with a_1 being the target of unit 1 and a_2 the target of unit 2. The values of T_{cb} and T_{cf} are 40 and 80 ms, respectively, and $a_b = a_2 + 0.75(a_1 - a_2)$. The time $T_{cb} + T_{cf}$ specifies how rapid the transition is. While linear interpolation could be used, a_b and the ratio T_{cb}/T_{cf} are sometimes used to further refine the shape of the formant transition.

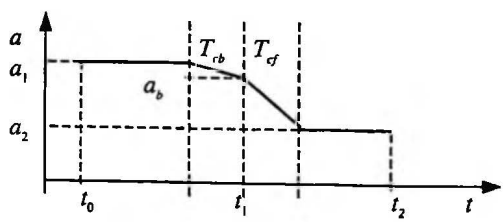


Figure 16.4 Transition between two vowels in a formant synthesizer.

Other rules can allow a discontinuity, for example, when a transition out of an unvoiced segment takes place. To improve naturalness, all these parameters can be made dependent on the immediate phonetic context.

Amazon/VB Assets
Exhibit 1011
Page 826

Table 16.2 Targets used in the Klatt synthesizer: formant frequencies and bandwidths for non-vocalic segments of a male speaker. Note that in addition to the phoneme set used in Chapter 2, there are several additional phonetic segments here such as *axp, dx, el, em, en, gp, hx, kp, lx, qq, rx, tq, wh*, that allow more control (after Allen [4]).

	F1	F2	F3	B1	B2	B3
axp	430	1500	2500	120	60	120
b	200	900	2100	65	90	125
ch	300	1700	2400	200	110	270
d	200	1400	2700	70	115	180
dh	300	1150	2700	60	95	185
dx	200	1600	2700	120	140	250
el	450	800	2850	65	60	80
em	200	900	2100	120	60	70
en	200	1600	2700	120	70	110
ff	400	1130	2100	225	120	175
g	250	1600	1900	70	145	190
gp	200	1950	2800	120	140	250
h	450	1450	2450	300	160	300
hx	450	1450	2450	200	120	200
j	200	1700	2400	50	110	270
k	350	1600	1900	280	220	250
kp	300	1950	2800	150	140	250
l	330	1050	2800	50	100	280
lx	450	800	2850	65	60	80
m	480	1050	2100	40	175	120
ng	480	1600	2050	160	150	100
n	480	1400	2700	40	300	260
p	300	900	2100	300	190	185
qq	400	1400	2450	120	140	250
r	330	1060	1380	70	100	120
rx	460	1260	1560	60	60	70
sh	400	1650	2400	200	110	280
sil	400	1400	2400	120	140	250
s	400	1400	2700	200	95	220
th	400	1150	2700	225	95	200
tq	200	1400	2700	120	140	250
t	300	1400	2700	300	180	220
v	300	1130	2100	55	95	125
wh	330	600	2100	150	60	60
w	285	610	2150	50	80	60
y	240	2070	3020	40	250	500
zh	300	1650	2400	220	140	250
z	300	1400	2700	70	85	190

Amazon/VB Assets
Exhibit 1011
Page 827

Table 16.3 Targets used in the Klatt synthesizer: formant frequencies and bandwidths for vocalic segments of a male speaker. Note that in addition to the phoneme set used in Chapter 2, there are several additional phonetic segments here such as *axr, exr, ix, ixr, oxr, uxr, yu* that allow more control (after Allen [4]).

	F1	F2	F3	B1	B2	B3
aa	700	1220	2600	130	70	160
ae	620	1660	2430	70	130	300
ah	620	1220	2550	80	50	140
ao	600	990	2570	90	100	80
aw	640	1230	2550	80	70	110
ax	550	1260	2470	80	50	140
axr	680	1170	2380	60	60	110
ay	660	1200	2550	100	120	200
eh	530	1680	2500	60	90	200
er	470	1270	1540	100	60	110
exr	460	1650	2400	60	80	140
ey	480	1720	2520	70	100	200
ih	400	1800	2670	50	100	140
ix	420	1680	2520	50	100	140
ixr	320	1900	2900	70	80	120
iy	310	2200	2960	50	200	400
ow	540	1100	2300	80	70	70
oxr	550	820	2200	60	60	60
oy	550	960	2400	80	120	160
uh	450	1100	2350	80	100	80
uw	350	1250	2200	65	110	140
uxr	360	800	2000	60	60	80
yu	290	1900	2600	70	160	220

Klatt showed that for a given natural utterance, he could manually obtain a sequence of formant tracks $f_i[n]$ and $b_i[n]$, such that the synthesized utterance not only had good quality but also sounded very similar to the original. This shows that the formant synthesizer of Section 16.2.1 appears to be sufficient for generation. On the other hand, when the formant tracks are obtained automatically through rules such as that of Figure 16.4 and Table 16.2 and Table 16.3, the output speech does not sound that natural, and the voice does not resemble the voice of the original recording.

Formant synthesis is very flexible because it can generate intelligible speech with relatively few parameters (about 40). The use of context-dependent rules can improve the quality of the synthesizer at the expense of a great deal of manual tuning. The synthesized speech is, by design, smooth, although it may not resemble any given speaker and may not sound very natural.

Amazon/VB Assets
Exhibit 1011
Page 828

Because of their flexibility, formant synthesizers can often generate many different voices and effects. While not as flexible, voice effects can also be produced in concatenative speech systems (see Section 16.5.3).

16.2.3. Data-Driven Formant Generation

While, in general, formant synthesis assumes the formant model of Section 16.2.1 driven by parameter values generated by rules, as in Section 16.2.2, data-driven methods to generate the formant values have also been proposed [3]. An HMM running in generation mode emits three formant frequencies and their bandwidths every 10 ms, and these values are used in a cascade formant synthesizer similar to that described in Section 16.2.1. Like the speech recognition counterparts, this HMM has many decision-tree context-dependent triphones and three states per triphone. A Gaussian distribution per state is used in this work. The baseline system uses a six-dimensional vector that includes the first three formant frequencies and their bandwidths. Initially it is assumed that the input to the synthesizer includes, in addition to the duration of each phoneme, the duration of each state. In this case, the maximum likelihood formant track is a sequence of the state means and, therefore, is discontinuous at state boundaries.

The key to obtaining a smooth formant track is to augment the feature vector with the corresponding delta formants and bandwidths (the difference between the feature at time t and that feature at time $t - 1$) to complete a twelve-dimensional vector. The maximum likelihood solution now entails solving a tridiagonal set of linear equations (see the discussion on statistical formant tracking in Chapter 6). The resulting formant track is smooth, as it balances formant values that are close to the state means with delta values that are also within the state means. In addition, the synthesized speech resembles that of the donor speaker. More details on the analysis and model training can be found in the formant tracking section of Chapter 6.

16.2.4. Articulatory Synthesis

Articulatory synthesis is another model that has been used to synthesize speech by rule, by using parameters that model the mechanical motions of the articulators and the resulting distributions of volume velocity and sound pressure in the lungs, larynx, and vocal and nasal tracts. Because the human speech production articulators do not have that many degrees of freedom, articulatory models often use as few as 15 parameters to drive a formant synthesizer.

The relationship between articulators and acoustics is many-to-one [17]. For example, a ventriloquist can produce speech sounds with very different articulator positions than those of normal speech. The *speech inversion* problem is therefore ill posed. By using the assumption that the articulators do not change rapidly over time, it is possible, however, to estimate

Amazon/VB Assets
Exhibit 1011
Page 829

the vocal-tract area from formant frequencies [37]. In [8] the model uses five articulatory parameters: area of lip opening, constriction formed by the tongue blade, opening to the nasal cavities, average glottal area, and rate of active expansion or contraction of the vocal tract volume behind a constriction. These five parameters are augmented with the first four formant frequencies and F0.

Those area parameters can be obtained from real speech through X-rays and magnetic resonance imaging (MRI), though positioning such sensors in the vocal tract alters the way speech is produced (such as the sensors in the lips) and impedes completely natural sounds. The relationship between articulatory parameters and acoustic values has typically been done using a nonlinear mapping such as a neural network or a codebook.

While one day this may be the best way to synthesize speech, the state-of-the-art in articulatory synthesis does not generate speech with quality comparable to that of formant or concatenative systems.

16.3. CONCATENATIVE SPEECH SYNTHESIS

While state-of-the-art synthesis by rule is quite intelligible, it sounds unnatural, because it is very difficult to capture all the nuances of natural speech in a small set of manually derived rules. In *concatenative synthesis*, a speech segment is synthesized by simply playing back a waveform with matching phoneme string. An utterance is synthesized by concatenating together several speech fragments. The beauty of this approach is that unlike synthesis-by-rule, it requires neither rules nor manual tuning. Moreover, each segment is completely natural, so we should expect very natural output.

Unfortunately, this is equivalent to assembling an automobile with parts of different colors: each part is very good yet there is a color discontinuity from part to part that makes the whole automobile unacceptable. Speech segments are greatly affected by coarticulation [42], so if we concatenate two speech segments that were not adjacent to each other, there can be spectral or prosodic discontinuities. Spectral discontinuities occur when the formants at the concatenation point do not match. Prosodic discontinuities occur when the pitch at the concatenation point does not match. A listener rates as poor any synthetic speech that contains large discontinuities, even if each segment is very natural.

Thus, when designing a concatenative speech synthesis system we need to address the following issues:

1. What type of speech segment to use? We can use diphones, syllables, phonemes, words, phrases, etc.
2. How to design the acoustic inventory, or set of speech segments, from a set of recordings? This includes excising the speech segments from the set of recordings as well as deciding how many are necessary. This is similar to the training problem in speech recognition.
3. How to select the best string of speech segments from a given library of segments, and given a phonetic string and its prosody? There may be several

Amazon/VB Assets
Exhibit 1011
Page 830

strings of speech segments that produce the same phonetic string and prosody. This is similar to the search problem in speech recognition.

4. How to alter the prosody of a speech segment to best match the desired output prosody. This is the topic of Section 16.4.

Generally, these concatenative systems suffer from great variability in quality: often they can offer excellent quality in one sentence and terrible quality in the next one. If enough good units are available, a given test utterance can sound almost as good as a recorded utterance. However, if several discontinuities occur, the synthesized utterance can have very poor quality. While synthesizing arbitrary text is still a challenge with these techniques, for restrictive domains this approach can yield excellent quality. We examine all these issues in the following sections.

We define *unit* as an abstract representation of a speech segment, such as its phonetic label, whereas we use *instance* as a speech segment from an utterance that belongs to the same unit. Thus, a system can keep several instances of a given unit to select among them to better reduce the discontinuities at the boundaries. This abstract representation consists of the unit's phonetic transcription at the minimum, in such a way that the concatenation of a set of units matches the target phonetic string. In addition to the phonetic string, this representation can often include prosodic information.

16.3.1. Choice of Unit

A number of units have been used in the field, including context-independent phonemes, diphones, context-dependent phonemes, subphonetic units, syllables, words, and phrases. A compilation of unit types for English is shown in Table 16.4 with their coverage in Figure 16.5.

Table 16.4 Unit types in English assuming a phone set of 42 phonemes. Longer units produce higher quality at the expense of more storage. The number of units is generally below the absolute maximum in theory, i.e., out of the $42^3 = 74{,}088$ possible triphones, only about 30,000 occur in practice.

Unit length	Unit type	#Units	Quality
Short	Phoneme	42	Low
	Diphone	~1500	
	Triphone	~30K	
	Demisyllable	~2000	
	Syllable	~15K	
	Word	100K–1.5M	
	Phrase	∞	
Long	Sentence	∞	High

Amazon/VB Assets
Exhibit 1011
Page 831

The issues in choosing appropriate units for synthesis are similar to those in choosing units for speech recognition (described in Chapter 9):

- The unit should lead to *low concatenation distortion*. A simple way of minimizing this distortion is to have fewer concatenations and thus use long units such as words, phrases or even sentences. But since some concatenations are unavoidable, we also want to use units that naturally lead to "small" discontinuities at the concatenation points. For example, it has been observed that spectral discontinuities at vowels are much more noticeable than at fricatives, or that a discontinuity within a syllable is more perceptually noticeable than a discontinuity across syllable boundaries, and similarly for within-word and across-word discontinuities [55]. Having several instances per unit is an alternative to long units that allows the choice of instances with low concatenation distortion.
- The unit should lead to *low prosodic distortion*. While it is not crucial to have units with the same prosody as the desired target, replacing a unit with a rising pitch with another with a falling pitch may result in an unnatural sentence. Altering the pitch and/or duration of a unit is possible (see Section 16.4) at the expense of additional distortion.
- The unit should be *generalizable*, if unrestricted text-to-speech is required. If we choose words or phrases as our units, we cannot synthesize arbitrary speech from text, because it's almost guaranteed that the text will contain words not in our inventory. As an example, the use of arbitrarily long units in such a way that no concatenation between voiced sounds occurs by cutting at obstruents results in low concatenation distortion but it is shown [47] that over 180,000 such units would be needed to cover 75% of a random corpus. The longer the speech segments are, the more of them we need to be able to synthesize speech from arbitrary text. This generalization property is not needed if closed-domain synthesis is desired.
- The unit should be *trainable*. Our training data should be sufficient to estimate all our units. Since the training data is usually limited, having fewer units leads to better trainability in general. So the use of words, phrases, or sentences may be prohibitive other than for closed-domain synthesis. The other units in Table 16.4 can be considered trainable depending on the limitations on the size of our acoustic inventory.

A practical challenge is how to balance these selection criteria. In this section we compare a number of units and point out their strengths and weaknesses.

16.3.1.1. Context-Independent Phonemes

The most straightforward unit is the phoneme. Having one instance of each phoneme, independent of the neighboring phonetic context, is very generalizable, since it allows us to gen-

Amazon/VB Assets
Exhibit 1011
Page 832

erate every word/sentence. It is also very trainable and we could have a system that is very compact. For a language with N phonemes, say $N = 42$ for English, we would need only N unit instances. The problem is that using *context-independent* phones results in many audible discontinuities. Such a system is not intelligible.

16.3.1.2. Diphones

A type of subword unit that has been extensively used is the so-called *dyad* or *diphone* [41]. A diphone `s-ih` includes from the middle of the `s` phoneme to the middle of the `ih` phoneme, so diphones are, on the average, one phoneme long. The word `hello` */hh ax l ow/* can be mapped into the diphone sequence: */sil-hh/*, */hh-ax/*, */ax-l/*, */l-ow/*, */ow-sil/*. If our language has N phonemes, there are potentially N^2 diphones. In practice, many such diphones never occur in the language, so that a smaller number is sufficient. For example, the phonetic alphabet of Chapter 2 has 42 phonemes for English, and only about 1300 diphones are needed. Diphone units were among the first type of unit used in concatenative systems because they yield fairly good quality. While diphones retain the transitional information, there can be large distortions due to the difference in spectra between the stationary parts of two units obtained from different contexts. For example, there is no guarantee that the spectra of */ax-l/* and */l-ow/* will match at the junction point, since the instance */ax-l/* could have been excised from a very different right context than */ow/* or the instance */l-ow/* could have been excised from a very different left context than */ax/*.

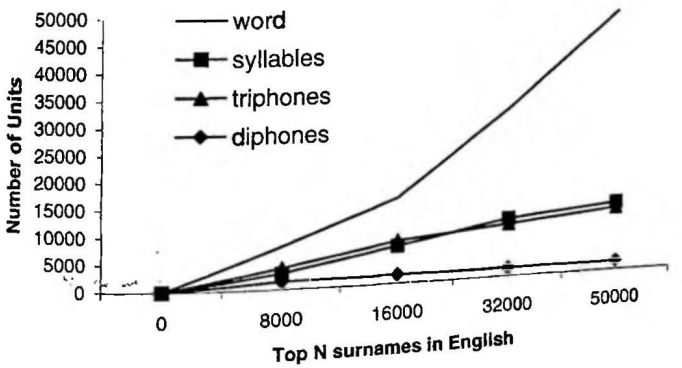


Figure 16.5 Coverage with different number of units displays the number of units of different types required to generate the top N surnames in the United States [34].

For this reason, many practical diphone systems are not purely diphone based: they do not store transitions between fricatives, or between fricatives and stops, while they store

Amazon/VB Assets
Exhibit 1011
Page 833

longer units that have a high level of coarticulation [48]. If only one representative of a dyad is available, there are pitch discontinuities. Prosody modification techniques, such as those described in Section 16.4, can be applied to correct this problem. Otherwise many instances of each diphone are needed for good prosodic coverage. Diphones are trainable, generalizable and offer better quality than context-independent phones.

16.3.1.3. Context-Dependent Phoneme

Another subword unit used in the literature [24] is the *context-dependent* phoneme. If the context is limited to the immediate left and right phonemes, the unit is known as *triphone*. As in speech recognition, not all N^3 need to be stored, because not all combinations will occur in practice. For English, typically there can be in excess of 25,000 triphones: 12,000 within-word triphones and another 12,000 across-word triphones. Because of the increased number of units, more contextual variations can be accommodated this way. Drawing from experience in speech recognition, we know that many different contexts have a similar effect on the phoneme; thus, several triphones can be clustered together into a smaller number of *generalized triphones*, typically between 500 and 3000. All the clustering procedures described in Chapter 9 can be used here as well. In particular, decision-tree clustered phones have been successfully used. Because a larger number of units can be used, discontinuities can be smaller than in the case of diphones while making the best use of the available data. In addition to only considering the immediate left and right phonetic context, we could also add stress for the current phoneme and its left and right context, word-dependent phones (where phones in particular words are considered distinct context-dependent phones), quinphones (where two immediate left and right phones are used), and different prosodic patterns (pitch ranges and/or durations). As in speech recognition, clustered context-dependent triphones are trainable and generalizable.

Traversing the tree for a given phoneme is equivalent to following the answers for the branching nodes from root to leaves, which determines the clusters for similar context-dependent phones. The decision trees are generated automatically from the analysis database to obtain minimum within-unit distortion (or entropy) for each split. Therefore, one must be able to acquire a large inventory of context-dependent phone HMMs with a decent coverage of the contexts one wishes to model. All the context-dependent phone units can be well replaced by any other units within the same cluster. This method generalizes to contexts not seen in the training data, because the decision tree uses questions involving broad phonetic categories of neighboring contexts, yet provides detailed models for contexts that are represented in the database. Given the assumption that these clustering decision trees should be consistent across different speakers, the use of ample speaker-independent databases instead of limited speaker-dependent databases allows us to model more contexts as well as deeper trees to generate a high-quality TTS voice. These techniques also facilitate the creation of acoustic inventories with a scalable number of units that trade off size with quality. Thus, we can use questions (about the immediate left/right phonetic contexts, stress, pitch, duration,

Amazon/VB Assets
Exhibit 1011
Page 834

word, etc.) in the decision-tree clustering methods of Chapter 4 to reduce all the possible combinations to a manageable number.

16.3.1.4. Subphonetic Unit

Subphonetic units, or senones, have also been used with some success [13]. Typically, each phoneme can be divided into three states, which are determined by running a speech recognition system in forced alignment mode. These states can also be context dependent and can also be clustered using decision trees like the context-dependent phonemes. The HMM state has proved to be more effective than the context-dependent phone in speech recognition, also trainable and generalizable, but for synthesis it means having more concatenations and thus possibly more discontinuities. If multiple instances per subphonetic unit are used, higher quality can be obtained.

A *half phone* goes either from the middle of a phone to the boundary between phones or from the boundary between phones to the middle of the phone. This unit offers more flexibility than a phone and a diphone and has been shown useful in systems that use multiple instances of the unit [7].

16.3.1.5. Syllable

It has been observed that discontinuities across syllables stand out more than discontinuities within syllables [55], so syllables are natural units. There are more than 10,000 syllables in English, depending on the exact definition of syllable, so even a context-independent syllable system needs to store at least as many if one instance per syllable is needed for full generalizability. There will still be spectral discontinuities, though hopefully not too noticeable. More than one instance per unit may be needed to account for varying acoustic contexts or varying prosodic patterns, particularly if no waveform modification is to be used.

16.3.1.6. Word and Phrase

The unit can be as large as a word or even a phrase. While using these long units can increase naturalness significantly, generalizability and trainability are poor, so that it is difficult to have all the instances desired to synthesize an output utterance. One advantage of using a word or longer unit over its decomposition in phonemes, as in the above units, is that there is no dependence on a phonetically transcribed dictionary. It is possible that the phoneme string associated to a word by our dictionary is not correct, or not fluent enough, so that using a whole-word model can solve this problem. Of course, the system may have a combination of all units: a set of the most frequent words, sentences, or phrases for best quality some percentage of the time, and some smaller units for full generalizability and trainability.

Amazon/VB Assets
Exhibit 1011
Page 835

16.3.2. Optimal Unit String: The Decoding Process

The goal of the decoding process is to choose the optimal string of units for a given phonetic string that best matches the desired prosody. Sometimes there is only one possible string, so that this process is trivial, but in general there are several strings of units that result in the same phonetic string yet some of them sound better than others. The goal is to come up with an objective function that approximates this sound quality that allows us to select the best string. The quality of a unit string is typically dominated by spectral and pitch discontinuities at unit boundaries. Discontinuities can occur because of:

1. *Differences in phonetic contexts.* A speech unit was obtained from a different phonetic context than that of the target unit.
2. *Incorrect segmentation.* Such segmentation errors can cause spectral discontinuities even if they had the same phonetic context.
3. *Acoustic variability.* Units can have the same phonetic context and be properly segmented, but variability from one repetition to the next can cause small discontinuities. A unit spoken in fast speech is generally different from another in slow or normal speech. Different recording conditions (amplitude, microphone, sound card) can also cause spectral discontinuities.
4. *Different prosody.* Pitch discontinuity across unit boundaries is also a cause for degradation.

The severity of such discontinuities generally decreases as the number of units increases. More importantly, the prosody of the concatenation has, in general, no resemblance with the prosody specified by the TTS front-end unless we have several instances of each unit, each with a different prosody, or use a prosody modification algorithm (see Section 16.4).

16.3.2.1. Objective Function

Our goal is to come up with a numeric measurement for a concatenation of speech segments that correlates well with how well they sound. To do that we define unit cost and transition cost between two units.

Let θ be a speech segment with phonetic transcription $p = p(\theta)$. Let $\Theta = \{\theta_1, \theta_2, \cdots, \theta_N\}$ be a concatenation of N speech segments whose combined phonetic transcription is $P = \{p_1, p_2, \cdots, p_N\}$. P is a string of M phonemes, and since each segment has at least one phoneme, it holds that $M \geq N$.

For example, the phonetic string $P =$ "*hh ax l ow*" corresponding to the word *hello* has $M = 4$ phonemes and can be decomposed in $N = 4$ segments $\Theta_1 = \{\theta_1, \theta_2, \theta_3, \theta_4\}$, where $p(\theta_1) = /hh/$, $p(\theta_2) = /ax/$, $p(\theta_3) = /l/$, $p(\theta_4) = /ow/$, each segment being a phoneme. Or it can be decomposed into $N = 2$ segments $\Theta_2 = \{\theta_5, \theta_6\}$, where $p(\theta_5) = /hh\ ax/$,

Amazon/VB Assets
Exhibit 1011
Page 836

$p(\theta_k) = /l\ ow/$, so that each segment has two phonemes. There are 8 possible such decompositions for this example (in general there are 2^{M-1} possible decompositions[1]).

The distortion or cost function between the segment concatenation Θ and the target T can be expressed as a sum of the corresponding unit costs and transition costs [27, 46] as follows:

$$d(\Theta,T) = \sum_{j=1}^{N} d_u(\theta_j,T) + \sum_{j=1}^{N-1} d_t(\theta_j,\theta_{j+1}) \tag{16.2}$$

where $d_u(\theta_j,T)$ is the *unit cost* of using speech segment θ_j within target T and $d_t(\theta_j,\theta_{j+1})$ is the transition cost in concatenating speech segments θ_j and θ_{j+1}. The optimal speech segment sequence of units $\hat{\Theta}$ can be found as the one that minimizes the overall cost

$$\hat{\Theta} = \arg\min_{\Theta} d(\Theta,T) \tag{16.3}$$

over sequences with all possible numbers of units. Transition and unit costs are described in Sections 16.3.2.2 through 16.3.2.5.

Let's analyze the second term in the sum of Eq. (16.2), also shown in Figure 16.6. If all transition costs were identical, the word string with fewest units would have lowest distortion. In practice transition costs are different and, thus, the string with fewest units is not necessarily the best, though there is clearly a positive correlation.

When a large number of speech segments are available, finding the segment sequence with lowest cost is a search problem like those analyzed in Chapter 12. Often a Viterbi algorithm is needed to make this efficient.

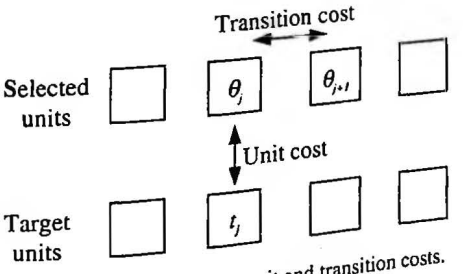


Figure 16.6 Tradeoff between unit and transition costs.

[1] This assumes that one instance per unit is available. If there are several instances per unit, the number of decompositions grows exponentially.

Amazon/VB Assets
Exhibit 1011
Page 837

The art in this procedure is in the exact definition of both transition and unit costs, for which no standard has been defined that works best to date. In Sections 16.3.2.2 and 16.3.2.3 we present an approach for which both transition and unit costs are empirically set after perceptual experiments. Such a system is easy to build, and study of those costs gives insight into the perceptual effects.

Costs obtained using a data-driven criterion are described in Sections 16.3.2.4 and 16.3.2.5. While more complicated than that of empirical costs, this method addresses the shortcomings of the previous method. Finally, we need to estimate some weights to combine the different costs for spectrum and prosody, which can be done empirically or by regression [26].

16.3.2.2. Empirical Transition Cost

If spoken in succession, two speech segments have a zero transition cost. But, when they are excised from separate utterances, their concatenation can have varying degrees of naturalness. The transition cost incorporates two types of continuity measures: coarticulatory and prosodic.

An approximation to the prosodic continuity measure is to make it proportional to the absolute difference of the F0 or log F0 at the boundaries, if the boundary is voiced for both units. If we use the prosody modification techniques of Section 16.4, this cost could be set to a small value to reflect the fact that prosody modification is not a perfect process. More sophisticated cost functions can be used to account for prosody mismatches [10].

Regarding the coarticulatory effect, it has been empirically observed that a concatenation within a syllable is more perceptible than when the concatenation is at the syllable boundary. Yi [55] proposed an empirical cost matrix for the concatenation of two speech segments when that concatenation occurs within a syllable (Table 16.5) or at a syllable boundary (Table 16.6). Phonemes are grouped by manner of articulation: vowel/semivowels, fricatives, stops, and nasals. The rows represent the left side of the transition and the columns represent the right side, and NA represents a case that does not occur. These costs reflect perceptual ratings by human listeners to unit concatenations between different phonemes. Values of 10, 2000, 5000, 7500, and 10,000 were used to indicate different degrees of goodness from very good to very bad concatenations.

Table 16.5 Cost matrix for intrasyllable concatenations (after Yi [55]). The rows represent the left side of the transition and the columns represent the right side, and NA represents a case that does not occur.

	vowel	semivowel	nasal	obstruent	/h/
vowel	10,000	10,000	7500	10	NA
semivowel	10,000	7500	7500	10	NA
nasal	5000	10	NA	10	NA
/h/	5000	NA	NA	NA	NA
obstruent	10	10	10	10,000	NA

Amazon/VB Assets
Exhibit 1011
Page 838

Table 16.6 Cost matrix for intersyllable concatenations (after Yi [55]). The rows represent the left side of the transition and the columns represent the right side, and NA represents a case that does not occur.

	vowel	semivowel	nasal	obstruent	/h/	silence
vowel	NA	7500	5000	10	5000	10
semivowel	7500	7500	2000	10	10	10
nasal	2000	10	10	10	10	10
obstruent	10	10	10	5000	10	10
/h/	NA	NA	NA	NA	NA	NA
silence	10	10	10	10	10	10

16.3.2.3. Empirical Unit Cost

The unit cost is generally a combination of the coarticulation cost and the prosodic cost. Prosodic mismatches can be made proportional to the F0 difference between the candidate unit and the target unit or set to a fixed low value if the prosody modification techniques of Section 16.4 are used.

A way of determining the cost associated with replacing a phonetic context with another was proposed by Yi [55], who empirically set cost matrices for phone classes by listening to concatenations where such contexts were replaced. These ad hoc values also bring some sense of where the coarticulation problems are. Replacing a vowel or semivowel by another with a context that has a different place of articulation or nasalization results in audible discontinuities. The rows represent the context class of the target phoneme and the columns represent the context class of the proposed unit. Table 16.7, Table 16.8, Table 16.9, Table 16.10, Table 16.11, and Table 16.12 include an empirical set of costs for such mismatches between the target's context and a candidate unit's context for the case of vowel/semivowels, fricatives, stops, and nasals. These costs reflect human listeners' perceptual ratings of speech units with an incorrect phonetic context. Values of 10, 100, 500, and 1000 were used to indicate very good, good, bad, and very bad units. These values are chosen to match the values for transition costs of Section 16.3.2.2.

Table 16.7 Unit coarticulation cost matrix (after Yi [55]) for left and right context replacements for vowels and semivowels.

	labial	alv/den/pal	velar	m	n	ng	front	back	none
labial	10	1000	1000	1000	1000	1000	1000	1000	1000
alv/den/pal	1000	10	1000	1000	1000	1000	1000	1000	1000
velar	1000	1000	10	1000	1000	1000	1000	1000	1000
m	1000	1000	1000	10	1000	1000	1000	1000	1000
n	1000	1000	1000	1000	10	1000	1000	1000	1000
ng	1000	1000	1000	1000	1000	10	1000	1000	1000
front	1000	1000	1000	1000	1000	1000	10	1000	1000
back	1000	1000	1000	1000	1000	1000	1000	10	1000
none	1000	1000	1000	1000	1000	1000	1000	1000	10

Amazon/VB Assets
Exhibit 1011
Page 839

Table 16.8 Unit coarticulation cost matrix (after Yi [55]) for left and right context replacements for fricatives.

	retroflex	round	sonorant	other
retroflex	10	100	100	100
round	100	10	100	100
sonorant	100	100	10	100
other	100	100	100	10

Table 16.9 Unit coarticulation cost matrix (after Yi [55]) for left context replacements for stops.

	front	back	retroflex	round	other
front	10	10	10	10	10
back	10	10	10	10	10
retroflex	10	10	10	10	10
round	10	10	10	10	10
other	500	500	500	500	10

Table 16.10 Unit coarticulation cost matrix (after Yi [55]) for right context replacements for stops.

	front	back	retroflex	round	schwa	other
front	10	100	100	100	500	100
back	100	10	100	100	500	100
retroflex	100	100	10	100	500	100
round	100	100	100	10	500	100
schwa	500	500	500	500	10	500
other	100	100	100	100	500	10

Table 16.11 Unit coarticulation cost matrix (after Yi [55]) for left context replacements for nasals.

	obstruent	sonorant
obstruent	10	1000
sonorant	1000	10

Amazon/VB Assets
Exhibit 1011
Page 840

Table 16.12 Unit coarticulation cost matrix (after Yi [55]) for right context replacements for nasals.

	voiced	unvoiced	sonorant
voiced	10	100	1000
unvoiced	100	10	1000
sonorant	1000	1000	10

16.3.2.4. Data-Driven Transition Cost

The empirical transition costs of Section 16.3.2.2 do not necessarily mean that a spectral discontinuity will take place, only that one is likely, and that if it occurs within a syllable it will have a larger perceptual effect than if it occurs across syllable boundaries. While that method can result in a good system, the cost is done independently of whether there is a true spectral discontinuity or not. Thus, it has been also proposed to use a measurement of the spectral discontinuity directly. This is often estimated as:

$$d_t(\theta_i,\theta_j) = \left|\mathbf{x}_i(l(\theta_i)-1) - \mathbf{x}_j(0)\right|^2 \tag{16.4}$$

the magnitude squared of the difference between the cepstrum at the last frame of θ_i and the first frame of θ_j. The quantity $l(\theta_i)$ denotes the number of frames of speech segment θ_i, and $\mathbf{x}_i(k)$ the cepstrum of segment θ_i at frame k.

This technique can effectively measure a spectral discontinuity in a region with slowly varying spectrum, but it can fail when one of the segments is a nasal, for example, for which a sharp spectral transition is expected and desired. A better way of measuring this discontinuity is shown in Figure 16.7, in which we measure the cepstral distance in an overlap region:[2] the last frame of segment 1 and the first frame before the beginning of segment 2:

$$d_t(\theta_i,\theta_j) = \left|\mathbf{x}_i(l(\theta_i)-1) - \mathbf{x}_j(-1)\right|^2 \tag{16.5}$$

When many speech segments are considered, a large number of cepstral distances need to be computed, which in turn may result in a slow process. To speed it up an approximation can be made where all possible cepstral vectors at the boundaries are vector quantized first, so that the distances between all codebook entries can be precomputed and stored in a table.

[2] This means extra frames need to be stored.

Amazon/VB Assets
Exhibit 1011
Page 841

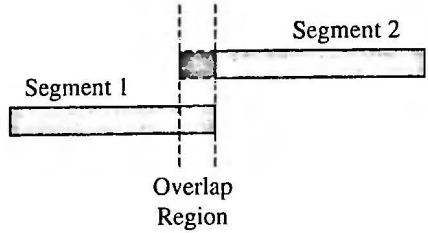


Figure 16.7 Measurement of the spectral discontinuity in the overlap region. The dark gray area is the speech region that precedes segment 2 and does not form part of segment 2. This area should match the last part of the segment 1.

A spectral discontinuity across, say, fricatives is perceptually not as important as if it happens across vowels [48]. For this reason, the cepstral distance described above does not correlate well with perceptual distances. To solve this problem, it is possible to combine both methods, for example by weighting the spectral/cepstral distance by different values.

Even if no spectral discontinuity is present, a phase discontinuity may take place. The pitch periodicity may be lost at the boundary. This can be generally solved by fine adjustment of the boundary using a correlation approach as described in Section 16.4. You need to keep in mind that such methods are not perfect.

16.3.2.5. Data-Driven Unit Cost

Spectral discontinuities across concatenations are often the result of using a speech segment with a different phonetic context than the target. One possibility is, then, to consider only speech segments where the phonetic contexts to the left and right match exactly. For example, if we use a phoneme as the basic speech segment, a perfect match would require on the order of at least 25,000 different segments. In this case, the coarticulation unit cost is zero if the target and candidate segment have the same phonetic context and infinite otherwise. When longer segments are desired, this number explodes exponentially. The problem with this approach is that it severely reduces the number of potential speech segments that can be used.

Generalized triphones, as described in Section 16.3.1.3, are used in [24]. In this approach, if the speech segments have the same generalized triphone contexts as the target utterance, the unit cost is zero, otherwise the cost is infinite. The technique allows us to use many more possible speech segments than the case above, yet it eliminates those speech segments that presumably have context mismatches that in turn lead to unnatural concatenations. When using a large training database, it was found that bringing the number of triphones from 25,000 down to about 2000 did not adversely impact the quality, whereas some degradation was perceived when using only 500 phoneme-length segments. Thus, this technique allows us to reduce the size of the speech segment inventory without severely degrading the voice quality.

Amazon/VB Assets
Exhibit 1011
Page 842