6

Speech Analysis

6.1 INTRODUCTION

Earlier chapters examined the production and perception of natural speech, and described speech-signal properties important for communication. Most applications of speech processing (e.g., coding, synthesis, recognition) exploit these properties to accomplish their tasks. This chapter describes how to extract such properties or features from a speech signal $s(n)$—a process called *speech analysis*. This involves a transformation of $s(n)$ into another signal, a set of signals, or a set of parameters, with the objective of simplification and data reduction. The relevant information in speech for different applications can often be expressed very compactly; e.g., a 10 s utterance (requiring 640,000 bits in basic coding format) typically contains about 120 seconds and 20–30 words (codable as text in a few hundred bits). In speech analysis, we wish to extract features directly pertinent for different applications, while suppressing redundant aspects of the speech. The original signal may approach optimality from the point of view of human perception, but it has much repetitive data when processed by computer; eliminating such redundancy aids accuracy in computer applications and makes phonetic interpretation simpler. We concentrate here on methods that apply to several applications; those that are particular to only one will be examined in later chapters.

For speech storage or recognition, eliminating redundant and irrelevant aspects of the speech waveform simplifies data manipulation. An efficient representation for speech recognition would be a set of parameters which is consistent across speakers, yielding similar values for the same phonemes uttered by various speakers, while exhibiting reliable variation for different phonemes. For speech synthesis, the continuity of parameter values in time is important to reconstruct a smooth speech signal; independent evaluation of parameters frame-by-frame is inadequate. Synthetic speech must replicate perceptually crucial properties of natural speech, but need not follow aspects of the original speech that are due to free variation.

This chapter investigates methods of speech analysis, both in the time domain (operating directly on the speech waveform) and in the frequency domain (after a spectral transformation of the speech). We want to obtain a more useful representation of the speech signal in terms of parameters that contain relevant information in an efficient format. Section 6.2 describes the tradeoffs involved in analyzing speech as a time-varying signal. Analyzers

periodically examine a limited time range (*window*) of speech. The choice of duration and shape for the window reflects a compromise in time and frequency resolution. Accurate time resolution is useful for segmenting speech signals (e.g., locating phone boundaries) and for determining periods in voiced speech, whereas good frequency resolution helps to identify different sounds. Section 6.3 deals with time-domain analysis, and Section 6.4 with spectral analysis. The former requires relatively little calculation but is limited to simple speech measures, e.g., energy and periodicity, while spectral analysis takes more effort but characterizes sounds more usefully.

Simple parameters can partition phones into manner-of-articulation classes, but discriminating place of articulation requires spectral measures. We distinguish speech *parameters* that are obtained by simple mathematical rules but have relatively low information content (e.g., Fourier coefficients) and *features* that require error-prone methods but yield more compact speech representations (e.g., formants, F0). Many speech analyzers extract only parameters, thus avoiding controversial decisions (e.g., deciding whether a frame of speech is voiced or not). *Linear predictive analysis* does both: the major effort is to obtain a set of about 10 parameters to represent the spectral envelope of a speech signal, but a voicing (feature) decision is usually necessary as well. Section 6.5 is devoted to the analysis methods of linear predictive coding (LPC), a very important technique in many speech applications.

The standard model of speech production (a source exciting a vocal tract filter) is implicit in many analysis methods, including LPC. Section 6.6 describes another method to separate these two aspects of a speech signal, and Section 6.7 treats yet other spectral estimation methods. The excitation is often analyzed in terms of periodicity (Section 6.8) and amplitude, while variations in the speech spectrum are assumed to derive from vocal tract variations. Finally, Section 6.9 examines how continuous speech parameters can be derived from (sometimes noisy) raw data.

The analysis technique in this chapter can be implemented digitally, either with software (programs) or special-purpose hardware (microprocessors and chips). Analog processing techniques, using electronic circuitry, can perform most of the tasks, but digital approaches are prevalent because of flexibility and low cost. Analog circuitry requires specific equipment, rewiring, and calibration for each new application, while digital techniques may be implemented and easily modified on general-purpose computers. Analyses may exceed *real time* (where processing time does not exceed speech duration) on various computers, but advances in VLSI and continued research into more efficient algorithms will render more analyses feasible without computational delay.

6.2 SHORT-TIME SPEECH ANALYSIS

Speech is dynamic or time-varying: some variation is under speaker control, but much is random; e.g., a vowel is not truly periodic, due to small variations (from period to period) in the vocal cord vibration and vocal tract shape. Such variations are not under the active control of the speaker and need not be replicated for intelligibility in speech coding, but they make speech sound more natural. Aspects of the speech signal directly under speaker control (e.g., amplitude, voicing, F0, and vocal tract shape) and methods to extract related parameters from the speech signal are of primary interest here.

During slow speech, the vocal tract shape and type of excitation may not alter for durations up to 200 ms. Mostly, however, they change more rapidly since phoneme durations average about 80 ms. Coarticulation and changing F0 can render each pitch period different

from its neighbor. Nonetheless, speech analysis usually assumes that the signal properties change relatively slowly with time. This allows examination of a *short-time window* of speech to extract parameters presumed to remain fixed for the duration of the window. Most techniques yield parameters averaged over the course of the time window. Thus, to model dynamic parameters, we must divide the signal into successive windows or *analysis frames*, so that the parameters can be calculated often enough to follow relevant changes (e.g., due to dynamic vocal tract configurations). Slowly changing formants in long vowels may allow windows as large as 100 ms without obscuring the desired parameters via averaging, but rapid events (e.g., stop releases) require short windows of about 5–10 ms to avoid averaging spectral transitions with steadier spectra of adjacent sounds.

6.2.1 Windowing

Windowing is multiplication of a speech signal $s(n)$ by a window $w(n)$, which yields a set of speech samples $x(n)$ weighted by the shape of the window. $w(n)$ may have infinite duration, but most practical windows have finite length to simplify computation. By shifting $w(n)$, we examine any part of $s(n)$ through the movable window (Figure 6.1).

Many applications prefer some speech averaging, to yield an output parameter contour (vs time) that represents some slowly varying physiological aspects of vocal tract movements. The amount of the desired smoothing leads to a choice of window size trading off three factors: (1) $w(n)$ short enough that the speech properties of interest change little within the window, (2) $w(n)$ long enough to allow calculating the desired parameters (e.g., if additive noise is present, longer windows can average out some of the random noise), (3) successive windows not so short as to omit sections of $s(n)$ as an analysis is periodically repeated. The last condition reflects more on the *frame rate* (number of times per second that speech analysis is performed, advancing the window periodically in time) than on window size. Normally, the frame rate is about twice the inverse of the $w(n)$ duration, so that successive windows overlap (e.g., by 50%), which is important in the common case that $w(n)$ has a shape that de-emphasizes speech samples near its edges (see Section 6.4).

The size and shape of $w(n)$ depend on their effects in speech anlaysis. Typically $w(n)$ is smooth, because its values determine the weighting of $s(n)$ and *a priori* all samples are equally relevant. Except at its edges, $w(n)$ rarely has sudden changes; in particular, windows

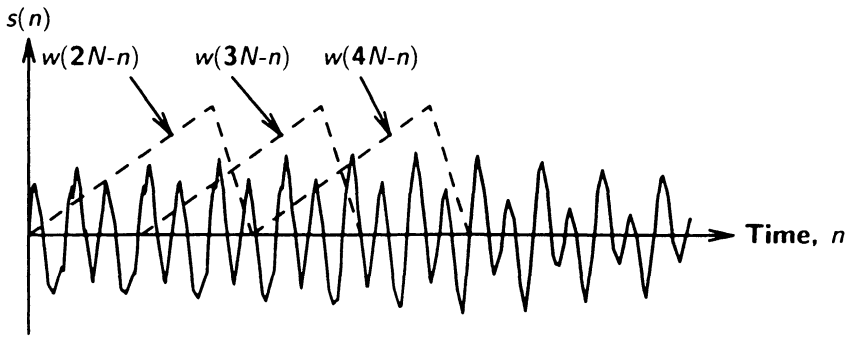

Figure 6.1 Speech signal $s(n)$ with three superimposed windows, offset from the time origin by $2N$, $3N$, and $4N$ samples. (An atypical asymmetric window is used for illustration.)

rarely contain zero- or negative-valued points since they would correspond to unutilized or phase-reversed input samples. The simplest common window has a rectangular shape $r(n)$:

$$w(n) = r(n) = \begin{cases} 1 & \text{for } 0 \leq n \leq N-1 \\ 0 & \text{otherwise.} \end{cases} \tag{6.1}$$

This choice provides equal weight for all samples, and just limits the analysis range to N consecutive samples. Many applications trade off window duration and shape, using larger windows than strictly allowed by stationarity constraints but then compensating by emphasizing the middle of the window (Figure 6.2); e.g., if speech is quasi-stationary over 10 ms, a 20 ms window can weight the middle 10 ms more heavily than the first and last 5 ms. Weighting the middle samples more than the edge relates to the effect that window shape has on the output speech parameters. When $w(n)$ is shifted to analyze successive frames of $s(n)$, large changes in output parameters can arise when using $r(n)$; e.g., a simple energy measure obtained by summing $s^2(n)$ in a rectangular window could have large fluctuations as $w(n)$ shifts to include or exclude large amplitudes at the beginning of each pitch period. If we wish to detect pitch periods, such variation would be desired, but more often the parameters of interest are properties of vocal tract shape, which usually vary slowly over several pitch periods. A common alternative to Equation (6.1) is the Hamming window, a raised cosine pulse:

$$w(n) = h(n) = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2\pi n}{N-1}\right) & \text{for } 0 \leq n \leq N-1 \\ 0 & \text{otherwise.} \end{cases} \tag{6.2}$$

or the very similar Hanning window. Tapering the edges of $w(n)$ allows its periodic shifting (at the frame rate) along $s(n)$ without having effects on the speech parameters due to pitch period boundaries.

6.2.2 Spectra of Windows: Wide- and Narrow-band Spectrograms

While a window has obvious limiting effects in the time domain, its effects on speech spectra are also important. Due to its slowly varying waveform, $w(n)$ has a frequency response of a lowpass filter (Figure 6.3). As example windows, the smooth Hamming $h(n)$ concentrates more energy at low frequencies than does $r(n)$, which has abrupt edges. This

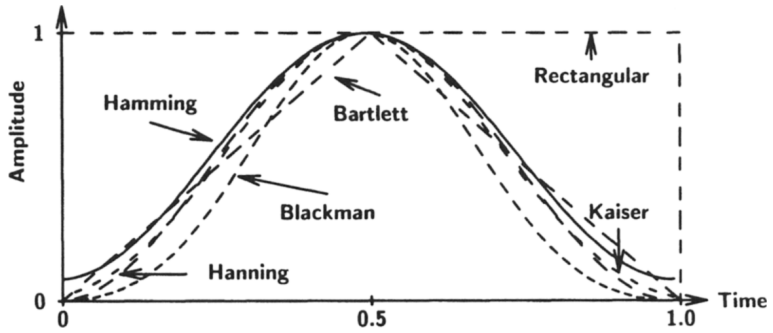

Figure 6.2 Common time windows, with durations normalized to unity.

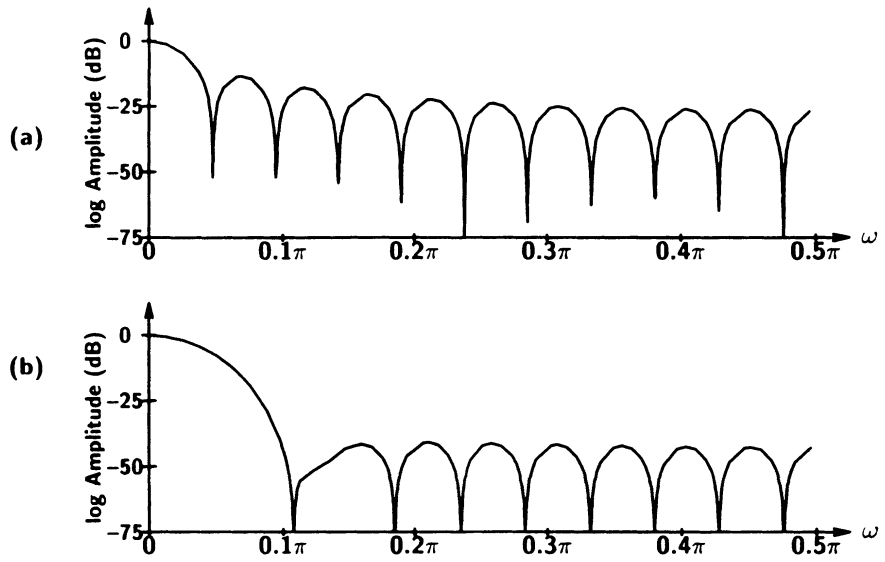

Figure 6.3 Magnitude of Fourier transforms for (a) rectangular window, (b) Hamming window.

concentration helps preserve the integrity of spectral parameters obtained from windowed signals, since $x(n) = s(n)w(n)$ corresponds to a convolution of spectra:

$$X(e^{j\omega}) = \frac{1}{2\pi}\int_{\theta=0}^{2\pi} S(e^{j\theta})W(e^{j(\omega-\theta)})d\theta. \tag{6.3}$$

To minimize distortion in the output spectral representation, $W(e^{j\omega})$ should have a limited frequency range and a smooth shape; e.g., an ideal lowpass filter (rectangular frequency pulse) strictly limits the frequency range and has a constant value. The output spectrum $X(e^{j\omega})$ is a smoothed version of $S(e^{j\omega})$, where each frequency sample is the average of its neighbors over a range equal to the bandwidth of the lowpass filter. A window with a rectangular spectrum has (usually undesirable) edge effects in frequency (as did $r(n)$ above for time); e.g., for voiced speech, $X(e^{j\omega})$ fluctuates as harmonics are included/excluded in the convolution process, depending on the interaction between the filter bandwidth and the speech F0.

An ideal lowpass filter is not a feasible window, due to its infinite duration. Practical windows however are flawed in not having strictly limited frequency ranges: each sample in $X(e^{j\omega})$ is not only the (desired) average of a range of $S(e^{j\omega})$ but also has contributions from many other frequencies. This undesirable behavior can be limited by concentrating most of $W(e^{j\omega})$ in a *main lobe* centered at zero frequency. Since the Hamming $H(e^{j\omega})$ is closer to an ideal lowpass filter than $R(e^{j\omega})$ (Figure 6.3), the former yields a better $X(e^{j\omega})$, more closely approximating the original $S(e^{j\omega})$. For a given window duration (a critical factor in computation and time resolution), however, $h(n)$ acts as a lowpass filter with twice the bandwidth of the rectangular $r(n)$ and thus smooths the speech spectrum over a range twice as wide (thus reducing the spectral detail) (Figure 6.4).

A properly smoothed output spectrum is often preferred; e.g., wideband spectrograms and formant detectors need spectral representations that smooth the fine structure of the

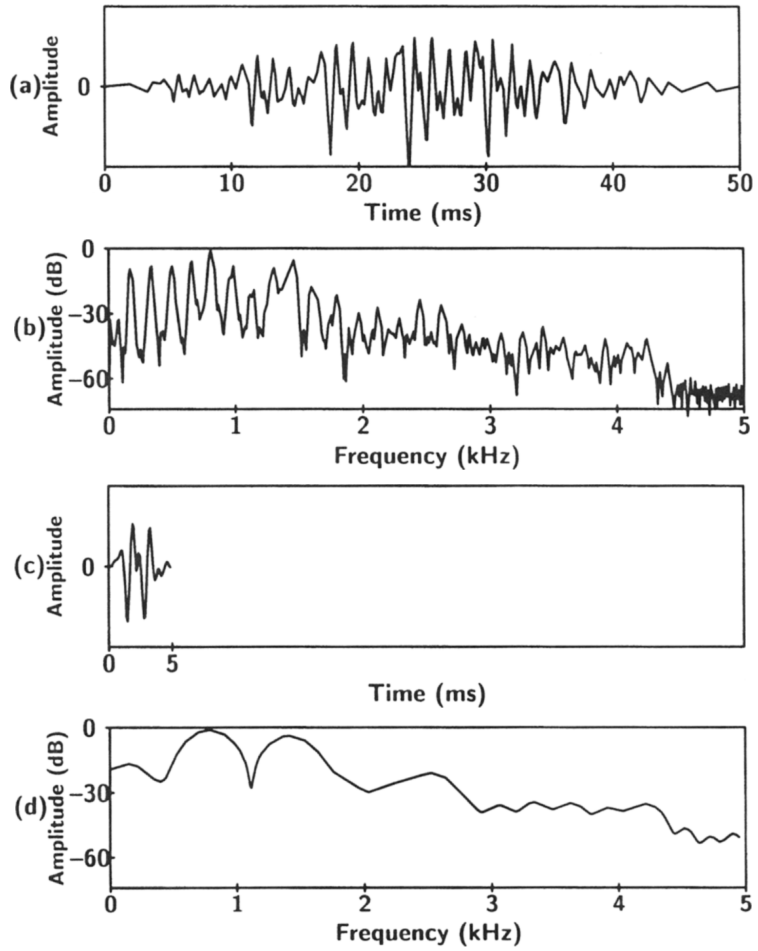

Figure 6.4 Time signals and spectra of a vowel: (a) signal multiplied by a 50 ms Hamming window; (b) the corresponding spectrum (note that harmonic structure is strongest at low frequencies); (c) signal multiplied by a 5 ms Hamming window; (d) its corresponding spectrum.

harmonics while preserving formant structure, which varies more slowly with frequency. For a given shape of window its duration is inversely proportional to its spectral bandwidth; the choice of window duration trades off time and frequency resolution. Traditional *wideband spectrograms* use a window of about 3 ms (fine time resolution, showing amplitude variations within each pitch period), which corresponds to a bandwidth of 300 Hz and smooths the harmonic structure (unless $F0 > 300$ Hz) (Figure 6.4).

Narrowband spectrograms, on the other hand, use a window with a 45 Hz bandwidth and thus a duration of about 20 ms. This allows a resolution of individual harmonics (since $F0 > 45$ Hz) (Figure 6.4) but smooths the signal in time over a few pitch periods. The latter spectral displays are good for F0 estimation, while wideband representations are better for viewing vocal tract parameters, which can change rapidly and do not need fine frequency resolution.

For windowing of voiced speech, a rectangular window with a duration of one pitch period (and centered on the period) produces an output spectrum close to that of the vocal tract impulse response, to the extent that each pitch period corresponds to such an impulse response. (This works best for low-F0 voices, where the pitch period is long enough to permit the signal to decay to low amplitude before the next vocal cord closure.) Unfortunately, it is often difficult to reliably locate pitch periods for such *pitch-synchronous* analysis, and system complexity increases if window size must change dynamically with F0. Furthermore, since most pitch periods are indeed shorter than the vocal tract impulse response, a one-period window truncates, resulting in spectral degradation.

For simplicity, most speech analyses use a fixed window size of longer duration, e.g., 25 ms. Problems of edge effects are reduced with longer windows; if the window is shifted in time without regard for pitch periods in the common *pitch-asynchronous* analysis, the more periods under the window the less the effects of including/excluding the large-amplitude beginning of any individual period. Windows well exceeding 25 ms smooth rapid spectral changes (relevant in most applications) too much. For F0 estimation, however, windows must typically contain at least two pitch periods; so pitch analysis uses a long window—often 30–50 ms.

Recent attempts to address the drawbacks of a fixed window size include more advanced frequency transforms (e.g., wavelets—see below), as well as simpler modifications to the basic DFT approach (e.g., the 'modulation spectrogram' [1], which emphasizes slowly varying speech changes around 4 Hz, corresponding to approximate syllable rates, at the expense of showing less rapid detail).

6.3 TIME-DOMAIN PARAMETERS

Analyzing speech in the time domain has the advantage of simplicity in calculation and physical interpretation. Several speech features relevant for coding and recognition occur in temporal analysis, e.g., energy (or amplitude), voicing, and F0. Energy can be used to segment speech in automatic recognition systems, and must be replicated in synthesizing speech; accurate voicing and F0 estimation are crucial for many speech coders. Other time features, e.g., zero-crossing rate and autocorrelation, provide inexpensive spectral detail without formal spectral techniques.

6.3.1 Signal Analysis in the Time Domain

Time-domain analysis transforms a speech signal into a set of parameter signals, which usually vary much more slowly in time than the original signal. This allows more efficient storage or manipulation of relevant speech parameters than with the original signal; e.g., speech is usually sampled at 6000–10,000 samples/s (to preserve bandwidth up to 3–5 kHz), and thus a typical 100 ms vowel needs up to 1000 samples for accurate representation. The information in a vowel relevant to most speech applications can be represented much more efficiently: energy, F0, and formants usually change slowly during a vowel. A parameter signal at 40–100 samples/s suffices in most cases (although 200 samples/s could be needed to accurately track rapid changes such as stop bursts). Thus, converting a speech waveform into a set of parameters can decrease sampling rates by two orders of magnitude. Capturing the relevant aspects of speech, however, requires several parameters sampled at the lower rate.

While time-domain parameters alone are rarely adequate for most applications, a combined total of 5–15 time- and frequency-domain parameters often suffice.

Most short-time processing techniques (in both time and frequency) produce parameter signals of the form

$$Q(n) = \sum_{m=-\infty}^{\infty} T[s(m)]w(n-m). \tag{6.4}$$

The speech signal $s(n)$ undergoes a (possibly nonlinear) transformation T, is weighted by the window $w(n)$, and is summed to yield $Q(n)$ at the original sampling rate, which represents some speech property (corresponding to T) averaged over the window duration. $Q(n)$ corresponds to a convolution of $T[s(n)]$ with $w(n)$. To the extent that $w(n)$ represents a lowpass filter, $Q(n)$ is a smoothed version of $T[s(n)]$.

Since $Q(n)$ is the output of a lowpass filter (the window) in most cases, its bandwidth matches that of $w(n)$. For efficient manipulation and storage, $Q(n)$ may be decimated by a factor equal to the ratio of the original sampled speech bandwidth and that of the window; e.g., a 20 ms window with an approximate bandwidth of 50 Hz allows sampling of $Q(n)$ at 100 samples/s (100:1 decimation if the original rate was 10,000 samples/s). As in most decimation operations, it is unnecessary to calculate the entire $Q(n)$ signal; for the example above, $Q(n)$ need be calculated only every 10 ms, shifting the analysis window 10 ms each time. For any signal $Q(n)$, this eliminates much (mostly redundant) information in the original signal. The remaining information is in an efficient form for many speech applications.

In addition to the common rectangular and Hamming windows, the Bartlett, Blackman, Hann, Parzen, or Kaiser windows [2, 3] are used to smooth aspects of speech signals, offering good approximations to lowpass filters while limiting window duration (see Figure 6.2). Most windows have finite-duration impulse responses (FIR) to strictly limit the analysis time range, to allow a discrete Fourier transform (DFT) of the windowed speech and to preserve phase. An infinite-duration impulse response (IIR) filter is also practical if its z transform is a rational function; e.g., a simple IIR filter with one pole at $z = a$ yields a recursion:

$$Q(n) = aQ(n-1) + T[s(n)]. \tag{6.5}$$

IIR windows typically need less computation than FIR windows, but $Q(n)$ must be calculated at the original (high) sampling rate before decimating. (In real-time applications, a speech measure may be required at every sample instant anyway). FIR filters, having no recursive feedback, permit calculation of $Q(n)$ only for the desired samples at the low decimated rate. Most FIR windows of N samples are symmetric in time; thus $w(n)$ has linear phase with a fixed delay of $(N-1)/2$ samples. IIR filters do not permit simple delay compensation.

6.3.2 Short-Time Average Energy and Magnitude

$Q(n)$ corresponds to short-time energy or amplitude if T in Equation (6.4) is a squaring or absolute magnitude operation, respectively (Figure 6.5). Energy emphasizes high amplitudes (since the signal is squared in calculating $Q(n)$), while the amplitude or magnitude measure avoids such emphasis and is simpler to calculate (e.g., with fixed-point arithmetic, where the dynamic range must be limited to avoid overflow). Such measures can help segment speech into smaller phonetic units, e.g., approximately corresponding to syllables or phonemes. The large variation in amplitude between voiced and unvoiced speech, as well as smaller variations between phonemes with different manners of articulation, permit segmentations based on energy $Q(n)$ in automatic recognition systems. For isolated word recognition,

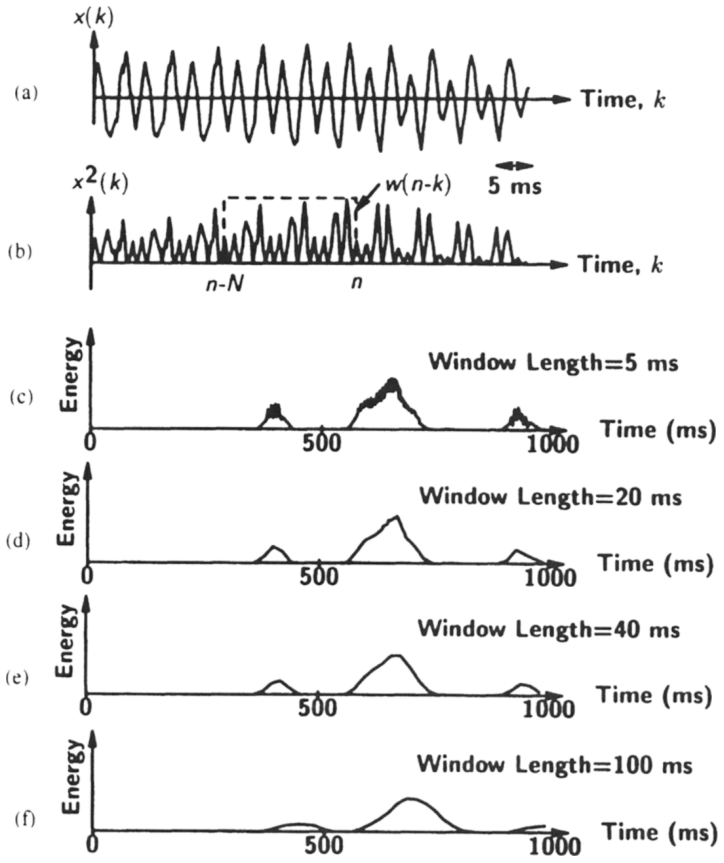

Figure 6.5 Illustration of the computation of short-time energy: (a) 50 ms of a vowel, (b) the squared version of (a), with a superimposed window of length N samples delayed n samples, (c–f) energy function for a 1 s utterance, using rectangular windows of different lengths.

such $Q(n)$ can aid in accurate determination of the endpoints of a word surrounded by pauses. In speech transmission systems that multiplex several conversations, this $Q(n)$ can help detect the boundaries of speech, so that pauses need not be sent.

6.3.3 Short-Time Average Zero-crossing Rate (ZCR)

Normally, spectral measures of speech require a Fourier or other frequency transformation or a complex spectral estimation (e.g., linear prediction). For some applications, a simple measure called the zero-crossing rate (ZCR) provides adequate spectral information at low cost. In a signal $s(n)$ such as speech, a *zero-crossing* occurs when $s(n) = 0$, i.e., the waveform crosses the time axis or changes algebraic sign. For narrowband signals (e.g., sinusoids), ZCR (in zero-crossings/s) is an accurate spectral measure; a sinusoidal has two zero-crossings/period, and thus its F0 = ZCR/2.

For discrete-time signals with ZCR in zero-crossings/sample,

$$\text{F0} = (\text{ZCR} * F_s)/2, \tag{6.6}$$

for F_s sample/s.

The ZCR can be defined as $Q(n)$ in Equation (6.4), with

$$T[s(n)] = 0.5|\text{sgn}(s(n)) - \text{sgn}(s(n-1))| \tag{6.7}$$

where the algebraic sign of $s(n)$ is

$$\text{sgn}(s(n)) = \begin{cases} 1 & \text{for } s(n) \geq 0 \\ -1 & \text{otherwise,} \end{cases} \tag{6.8}$$

and $w(n)$ is a rectangular window scaled by $1/N$ (where N is the duration of the window) to yield zero-crossings/sample, or by F_s/N to yield zero-crossings/s. This $Q(n)$ can be heavily decimated since the ZCR changes relatively slowly with the vocal tract movements.

The ZCR can help in voicing decisions. Most energy in voiced speech is at low frequency, since the spectrum of voiced glottal excitation decays at about −12 dB/oct. In unvoiced sounds, broadband noise excitation excites mostly higher frequencies, due to effectively shorter vocal tracts. While speech is not a narrowband signal (and thus the sinusoid example above does not hold), the ZCR correlates well with the average frequency of major energy concentration. Thus high and low ZCR correspond to unvoiced and voiced speech, respectively. A suggested boundary is 2500 crossings/s, since voiced and unvoiced speech average about 1400 and 4900 crossings/s, respectively, with a larger standard deviation for the latter (Figure 6.6).

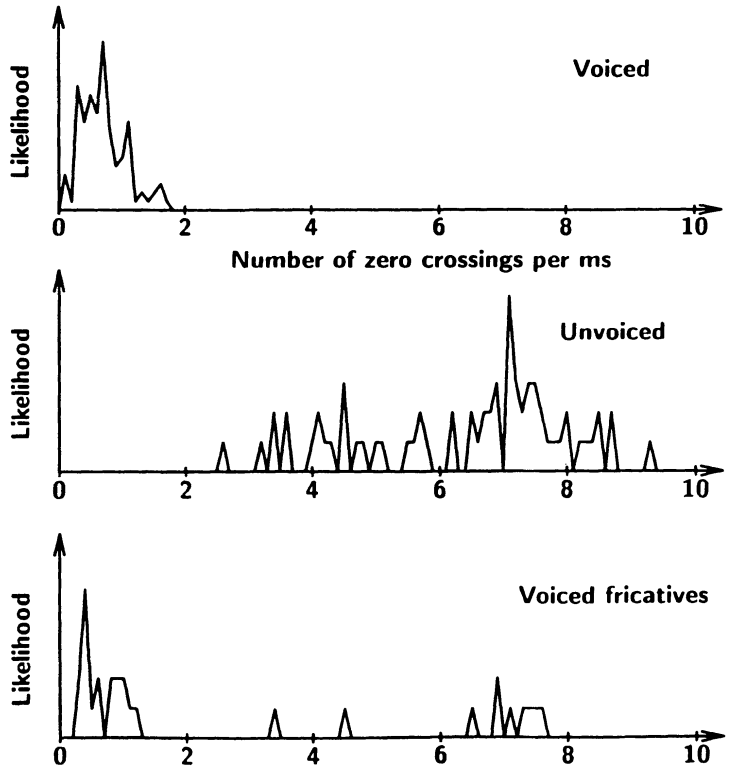

Figure 6.6 Typical distribution of zero-crossings for voiced sonorants, for unvoiced frication, and for voiced frication.

For vowels and sonorants, the ZCR corresponds mostly to F1, which has more energy than other formants. Interpreting ZCR is harder for voiced fricatives, which have both periodic energy in the voice bar at very low frequency and unvoiced energy at high frequency. This, of course, is a problem for all voiced/unvoiced determination methods; a binary decision using a simple threshold test on the ZCR is inadequate. Depending on the balance of periodic and aperiodic energy in voiced fricatives, some are above the threshold (e.g., the strident /z/) and others (e.g., /v/) are below. This problem is also language-dependent; e.g., English appears to have relatively weak voice bars, while French has strong ones.

Unlike short-time energy, the ZCR is highly sensitive to noise in the recording environment (e.g., 60 Hz hum from a power supply) or in analog-to-digital (A/D) conversion. Since energy below 100 Hz is largely irrelevant for speech processing, it may be desirable to highpass filter the speech in addition to the normal lowpass filtering before A/D conversion.

The ZCR can be applied to speech recognition. If speech is first passed through a bank of bandpass filters, each filter's output better resembles a narrowband signal, whose frequency of major energy concentration the ZCR easily estimates. Such a frequency could be a single harmonic (for filter bandwidths less than F0) or a formant frequency (for bandwidths of about 300–500 Hz). A bank or eight filters covering the 0–4 kHz range provides a simple set of eight measures, which could replace a more complex spectral representation (e.g., a DFT) in some applications.

6.3.4 Short-Time Aurocorrelation Function

The Fourier transform $S(e^{j\omega})$ of speech $s(n)$ provides both spectral magnitude and phase. The time signal $r(k)$ for the inverse Fourier transform of the energy spectrum ($|S(e^{j\omega})|^2$) is called the *autocorrelation* of $s(n)$. $r(k)$ preserves information about harmonic and formant amplitudes in $s(n)$ as well as its periodicity, while ignoring phase (as do many applications), since phase is less important perceptually and carries much less communication information than spectral magnitude. $r(k)$ has applications in F0 estimation, voiced/unvoiced determination, and linear prediction.

The autocorrelation function is a special case of the cross-correlation function,

$$\phi_{sy}(k) = \sum_{m=-\infty}^{\infty} s(m)y(m-k), \tag{6.9}$$

which measures the similarity of two signals $s(n)$ and $y(n)$ as a function of the time delay between them. By summing the products of a signal sample and a delayed sample from another signal, the cross-correlation is large if at some delay the two signals have similar waveforms. The range of summation is usually limited (i.e., windowed), and the function can be normalized by dividing by the number of summed samples.

When the same signal is used for $s(n)$ and $y(n)$, Equation (6.9) yields an autocorrelation. It is an even function ($r(k) = r(-k)$), it has maximum value at $k = 0$, and $r(0)$ equals the energy in $s(n)$ (or average power, for random or periodic signals). If $s(n)$ is periodic in P samples, then $r(k)$ also has period P. Maxima in $r(k)$ occur for $k = 0, \pm P, \pm 2P$, etc., independently of the absolute timing of the pitch periods; i.e., the window does not have to be placed synchronously with the pitch periods.

The *short-time* autocorrrelation function is obtained by windowing $s(n)$ and then using Equation (6.9), yielding

$$R_n(k) = \sum_{m=-\infty}^{\infty} s(m)w(n-m)s(m-k)w(n-m+k). \tag{6.10}$$

Equivalently, the product of speech $s(n)$ with its delayed version $s(n-k)$ is passed through a filter with response $w(n)w(n+k)$ (time index n indicates the position of the window). Equation (6.10) is evaluated for different values of k depending on the application. For linear prediction (Section 6.5), $R_n(k)$ for k ranging from 0 to 10–16 are typically needed, depending on the signal bandwidth. In F0 determination, $R_n(k)$ is needed for k near the estimated number of samples in a pitch period; if no suitable prior F0 estimate is available, $R_n(k)$ is calculated for k from the shortest possible period (perhaps 3 ms for a female voice) to the longest (e.g., 20 ms for men). With a sampling rate of 10,000 samples/s, the latter approach can require up to 170 calculations of $R_n(k)$ for each speech frame, if a pitch period resolution of 0.1 ms is desired.

Short windows minimize calculation: if $w(n)$ has N samples, $N-k$ products are needed for each value of $R_n(k)$. Proper choice of $w(n)$ also helps; e.g., using a rectangular window reduces the number of multiplications; symmetries in autocorrelation calculation can also be exploited (see LPC below). While the duration of $w(n)$ is almost directly proportional to the calculation (especially if $N \gg k$), there is a conflict between minimizing N to save computation and having enough speech samples in the window to yield a valid autocorrelation function: longer $w(n)$ give better frequency resolution. For F0 estimation, $w(n)$ must include more than one pitch period, so that $R_n(k)$ exhibits periodicity and the corresponding energy spectrum $|X_n(e^{j\omega})|^2$ resolves individual harmonics of F0 (see Figure 6.4). Spectral estimation applications (e.g., LPC) permit short windows since harmonic resolution is unimportant and the formant spectrum can be found from a portion of a pitch period.

For F0 estimation, an alternative to using autocorrelation is the average magnitude difference function (AMDF) [4]. Instead of multiplying speech $s(m)$ by $s(m-k)$, the

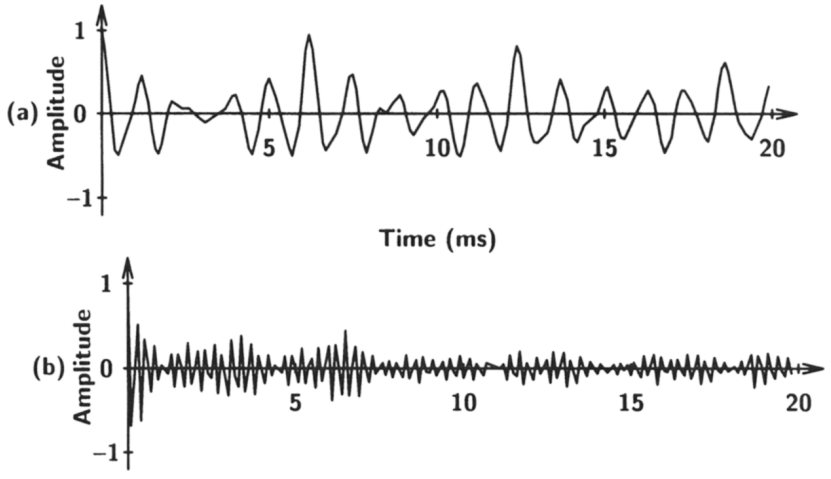

Figure 6.7 Typical autocorrelation function for (a) voiced speech and (b) unvoiced speech, using a 20 ms rectangular window ($N = 201$).

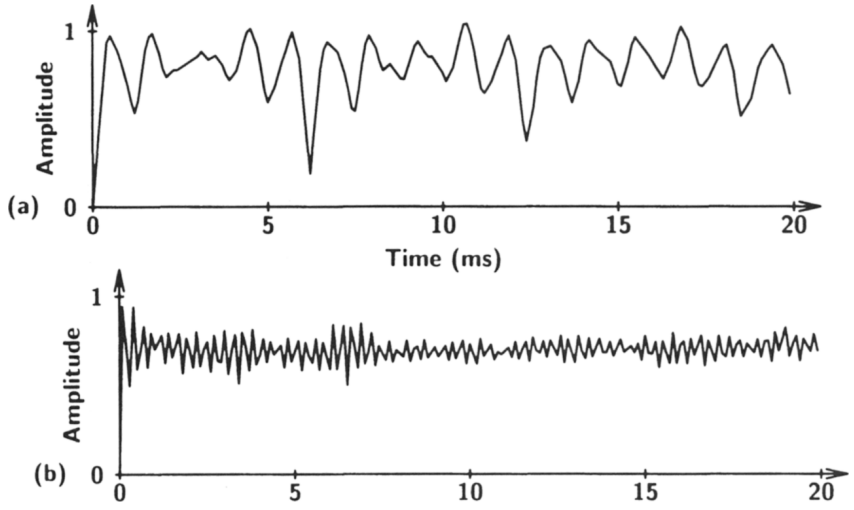

Figure 6.8 AMDF function (normalized to 1.0) for the same speech segments as in Figure 6.7.

magnitude of their difference is taken:

$$\text{AMDF}(k) = \sum_{m=-\infty}^{\infty} |s(m) - s(m-k)|. \tag{6.11}$$

Since subtraction and rectification are much simpler operations than multiplication, the AMDF is considerably faster. Where $R_n(k)$ peaks for values of k near multiples of the pitch period (Figure 6.7), the AMDF has minima (Figure 6.8).

Some speech recognition applications have used a simplified version of the autocorrelation [5]:

$$\psi(k) = \sum_{m=-\infty}^{\infty} \text{sgn}(s(m))s(m-k). \tag{6.12}$$

Replacing $s(m)$ by its sign in Equation (6.9) eliminates the need for multiplications and reduces the emphasis that $r(k)$ normally places on the high-amplitude portions of $s(n)$.

6.4 FREQUENCY-DOMAIN (SPECTRAL) PARAMETERS

The frequency domain provides most useful parameters for speech processing. Speech signals are more consistently and easily analyzed spectrally than in the time domain. The basic model of speech production with a noisy or periodic waveform that excites a vocal tract filter corresponds well to separate spectral models for the excitation and for the vocal tract. Repeated utterances of a sentence by a speaker often differ greatly temporally while being very similar spectrally. Human hearing appears to pay much more attention to spectral aspects of speech (e.g., amplitude distribution in frequency) than to phase or timing aspects. Thus, spectral analysis is used to extract most parameters from speech.

6.4.1 Filter-Bank Analysis

One spectral analysis method (popular due to real-time, simple, and inexpensive implementations) uses a *filter bank* or set of bandpass filters (either analog or digital), each analyzing a different range of frequencies of the input speech. Filter banks are more flexible than DFT analysis since the bandwidths can be varied to follow the resolving power of the ear, rather than being fixed, as in DFTs. Furthermore, many applications require a small set of parameters describing the spectral distribution of energy, especially the spectral envelope. The amplitude outputs from a bank of 8–12 bandpass filters provide a more efficient spectral representation than a more detailed DFT. Filters often follow the bark scale, i.e., equally spaced, fixed-bandwidth filters up to 1 kHz, and then logarithmically increasing bandwidth. One-third-octave filters are also common. Certain speech recognition systems use two levels of spectral analysis, a coarse filter bank with only a few filters for preliminary classification of sounds, followed where necessary by a more detailed analysis using a larger set of narrower filters.

6.4.2 Short-Time Fourier Transform Analysis

As the traditional spectral technique, Fourier analysis provides a speech representation in terms of amplitude and phase as a function of frequency. Viewing the vocal tract as a linear system, the Fourier transform of speech is the product of the transforms of the glottal (or noise) excitation and of the vocal tract response. For steady-state vowels of fricatives, the basic (infinite-time) Fourier transform could be used by extending or repeating sections or pitch periods of the speech ad infinitum. However, speech is not stationary, and thus short-time analysis using windows is necessary.

The short-time Fourier transform of a signal $s(n)$ is often defined as

$$S_n(e^{j\omega}) = \sum_{m=-\infty}^{\infty} s(m)e^{-j\omega m}w(n-m). \tag{6.13}$$

If ω is considered fixed, the transform has an interpretation as $Q(n)$ in Equation (6.4), where the transformation T corresponds to multiplication by a complex exponential of frequency ω, which has the spectral effect of rotating energy through a frequency shift of ω rad. Assuming $w(n)$ acts as a lowpass filter, $S_n(e^{j\omega})$ is a time signal (a function of n), describing the amplitude and phase of $s(n)$ within a bandwidth equivalent to that of the window but centered at ω rad. Repeating the calculation of $S_n(e^{j\omega})$ at different ω of interest yields a two-dimensional representation of the input speech: an array of time signals indexed on frequency, each noting the speech energy in a limited bandwidth about the chosen frequency.

A second interpretation of $S_n(e^{j\omega})$ views n as fixed, thus yielding the Fourier transform of $s(m)w(n-m)$, the windowed version of $s(m)$ using a window shifted to a time n with respect to the speech. This calculation could be repeated for successive n to produce an array of Fourier transforms index on time n, each expressing the spectrum of the speech signal within a window centered at time n.

For computational purposes, the DFT is used instead of the standard Fourier transform, so that the frequency variable ω only takes on N discrete values (N = the window duration, or *size*, of the DFT):

$$S_n(k) = \sum_{m=0}^{N-1} s(m)e^{-j2\pi km/N}w(n-m). \tag{6.14}$$

(In practice, each frame of speech samples $s(m)$ is shifted by the time delay n to align with a start at $m = 0$, allowing a simple N-sample window $w(m)$ to replace $w(n - m)$, and the fast Fourier transform or FFT is used to implement the DFT [6]. Since the Fourier transform is invertible, no information about $s(n)$ during the window is lost in the representation $S_n(e^{j\omega})$, as long as the transform is sampled in frequency sufficiently often (i.e., at N equally spaced values of ω) and the window $w(n)$ has no zero-valued samples among its N samples. The choice of N is thus crucial for short-time Fourier analysis. Low values for N (i.e., short windows and DFT's of few points) give poor frequency resolution since the window lowpass filter is wide, but they yield good time resolution since the speech properties are averaged only over short time intervals (see Figure 6.4). Large N, on the other hand, gives poor time resolution and good frequency resolution.

Assuming a rectangular window $w(n) = r(n)$ and viewing the main spectral lobe of $R(e^{j\omega})$ as its bandwidth, common choices are a 3.3 ms *wideband* window (300 Hz bandwidth) for good time resolution or a 22 ms *narrowband* window (45 Hz bandwidth) for good frequency resolution. The time–frequency tradeoff in resolution is related to window shape. Finite-duration windows theoretically have energy at infinitely high frequencies, but most is concentrated in a lowpass bandwidth. The abrupt $r(n)$ in particular has much of its energy beyond the main lobe of the lowpass filter. While the problem is reduced for other windows, frequency range and window duration cannot be completely limited simultaneously. Viewed as a time signal, $S_n(e^{j\omega})$ primarily notes energy components around frequency $\omega F_s/2\pi$ Hz but has contributions beyond the main lobe bandwidth in varying degree, depending on the window shape.

Alternatives to the rectangular $r(n)$ are common in spectral analysis due to $r(n)$'s high proportion of energy outside the main lobe and despite its narrow main lobe, which provides good frequency resolution for a short-time window. It is preferable to use another window with an appropriate increase in window duration to achieve the same frequency resolution, rather than accept the frequency distortion due to poor lowpass filtering. The allowable window duration is limited by the desired time resolution, though, which usually corresponds to the rate at which spectral changes occur in speech (e.g,. as rapidly as 5–10 ms). Any single spectral representation usually does not contain enough information for all speech processing applications. Short windows serve for formant analysis and segmentation, where good time resolution is important and where the smoothing of spectral harmonics into wider-frequency formants is desirable. Long windows are good for harmonic analysis and F0 detection, where individual harmonics must be resolved.

Because it retains sufficient information to completely reconstruct the windowed speech $x(n)$, the short-time Fourier transform is not economical for representing speech, in terms of the number of data samples. For fixed ω, $S_n(e^{j\omega})$ is a time signal of bandwidth roughly equal to that of the window and must be sampled at the Nyquist rate of twice the highest frequency. For fixed-time n, $S_n(e^{j\omega})$ is a Fourier transform to which an appropriate sampling rate in *frequency* may be calculated by applying the Nyquist theorem through the duality of the Fourier transform and its inverse. Since common windows are strictly "timelimited," $S_n(e^{j\omega})$ must be sampled at twice the window's "time width"; e.g., with a rectangular window of N samples and speech at F_s samples/s, the main spectral lobe occupies the range $0-F_s/N$ Hz. Thus each time function $S_n(e^{j\omega})$ must be sampled at $2F_s/N$ samples/s, and N time functions must be retained at N uniformly spaced frequencies from $\omega = 0$ to 2π. Since speech $s(n)$ is real, $S_n(e^{j\omega})$ is conjugate symmetric, and therefore the latter function need be retained only for $\omega = 0$ to π. However, since the Fourier transform is complex-valued, the net requirements are $2F_s$ real-valued samples/s, which is twice the original sampling rate. With the Hamming

and other windows, coding rates are even higher because of the larger bandwidths for the same window durations.

The short-time Fourier transform is thus not directly used for efficient coding, but as an alternative speech representation that has simpler interpretation in terms of the speech production and perception processes. Chapter 7 will explore coding applications that exploit data reduction of $S_n(e^{j\omega})$ while limiting speech quality degradation. More economical representation of speech parameters is achieved when the transform is subsampled below the Nyquist rate. This does not permit exact reconstruction of the speech waveform, but the ear is very tolerant of certain changes in speech signals that are more easily exploited in spectral form than in the time domain.

6.4.3 Spectral Displays

For decades a major speech analysis tool has been the *spectrogram*, or sound spectrograph, which provides a three-dimensional representation of short speech utterances (typically 2–3 s). The short-time Fourier transform $S_n(e^{j\omega})$ is plotted with time n on the horizontal axis, with frequency ω (from 0 to π) on the vertical axis (i.e., $0-F_s/2$ in Hz), and with magnitude indicated as degrees of shading (weak energy below one threshold shows as white, while very strong energy is black; the range between the two displays a varying amount of gray) (Figure 6.9). Since the transform phase is often of little interest, only the magnitude of the complex-valued $S_n(e^{j\omega})$ is displayed, typically on a logarithmic scale (following the dynamic range of audition). In the past, spectrograms used analog filtering, transferring electrical energy to Teledeltos paper through an electromechanical operation [7]; the dynamic range of such paper was only about 12 dB, which nonetheless was adequate to study most formant behavior. Recent computer-generated spectrograms are much more flexible.

Wideband spectrograms display individual pitch periods as vertical striations corresponding to the large speech amplitude each time the vocal cords close (Figure 6.9a). Voicing can be easily detected visually by the presence of these periodically spaced striations. Fine time resolution here permits accurate temporal location of spectral changes corresponding to

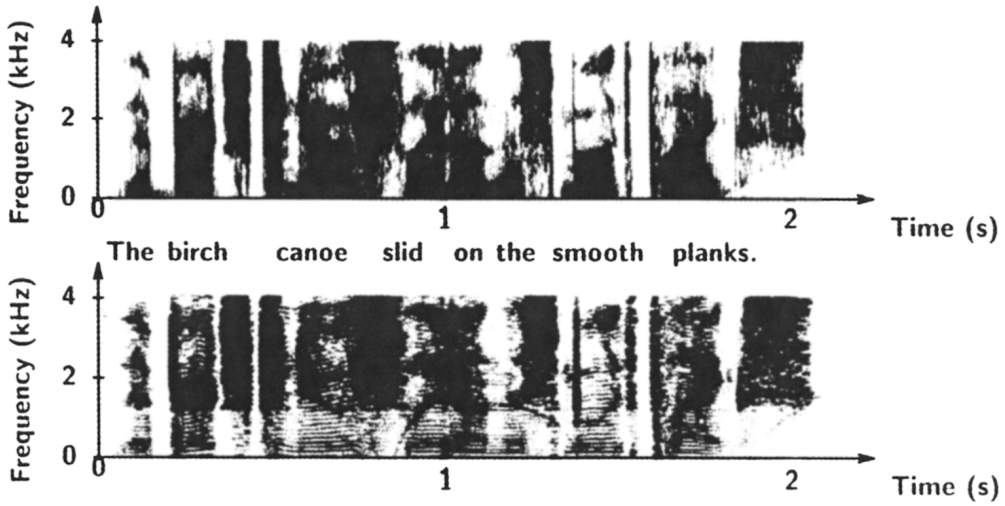

Figure 6.9 (a) Wideband and (b) narrowband spectrograms of a sentence.

vocal tract movements. The wide filter bandwidth smooths the harmonic amplitudes under each formant across a range of 300 Hz, displaying a band of darkness (of width proportional to the formant's bandwidth) for each formant. The center of each band is a good estimate of formant frequency.

Narrowband spectrograms display separate harmonics instead of pitch periods, and are less useful for segmentation because of poorer time resolution (Figure 6.9b). Instead they aid anlaysis of F0 and vocal tract excitation. A traditional, but tedious, way to estimate F0 is to divide a low-frequency range (e.g., 0–2 kHz, chosen due to the presence of strong formants) by the number of harmonics there. Due to limited range on spectrograms or to filtering of the speech (e.g., in the telephone network), however, harmonics are often invisible (i.e., their weak energy shows as white).

Since the amplitude of voiced speech falls off at about −6 dB/oct, dynamic range is often compressed prior to spectral analysis so that details at weak, high frequencies may be visible. *Pre-emphasizing* the speech, either by differentiating the analog speech $s_a(t)$ prior to A/D conversion or by differencing the discrete-time $s(n) = s_a(nT)$, compensates for falloff at high frequencies. If speech is to be reconstructed later using data from pre-emphasized speech, the final synthesis stage requires the inverse operation of *de-emphasis* or integration, which restores the proper dynamic range. The most common form of pre-emphasis is

$$y(n) = s(n) - As(n-1), \tag{6.15}$$

where A typically lies between 0.9 and 1.0 and reflects the degree of pre-emphasis. Effectively, $s(n)$ passes through a filter with a zero at $z = A$. The closer the zero to $z = 1$, the greater the pre-emphasis effect. The attenuation at frequencies below 200 Hz can be large, but such low frequencies are rarely of interest in spectral analysis applications.

The −6 dB/oct falloff applies only to voiced speech, since unvoiced speech tends to have a flat spectrum at high frequencies. Ideally, pre-emphasis should be applied only to voiced speech. In practice, however, the slightly degraded analysis of pre-emphasized unvoiced speech does not warrant limiting pre-emphasis only to voiced speech. Most applications use pre-emphasis throughout the entire speech signal and limit its effects on unvoiced speech by choosing a compromise value for A (e.g., 0.9).

6.4.4 Formant Estimation and Tracking

An assumption for much speech analysis is that the signal can be modeled as a source exciting a time-varying vocal tract filter. The source is either the quasi-periodic puffs of air passing through the glottis or broadband noise generated at a constriction in the vocal tract. The vocal tract filter response normally varies slowly because of constraints on movements of the tongue and lips, but it can change rapidly at articulator discontinuities (e.g., when a vocal tract passage closes or opens). The spectrum of voiced speech is the product of a line spectrum (harmonics spaced at F0 Hz) and the vocal tract spectrum. The latter is a slowly varying function of frequency, with an average of one formant peak/kHz.

The behavior of the first 3–4 formants is of crucial importance in many applications, e.g., formant vocoders (voice coders), some speech recognizers, and speech analysis leading to formant-based synthesis. Typical methods to estimate formant center frequencies and their bandwidths involve looking for peaks in spectral representations from short-time Fourier transforms, filter bank outputs, or linear prediction [8–11]. Such *peak-picking* methods appear to be accurate to within ±60 Hz for the first and second formants, but simple Fourier

transforms allow an accuracy of only ±110 Hz for F3 [12]. This compares to errors of ±40 Hz for manual measurements of spectrograms. For dynamic formants, wideband spectrograms (e.g., with a 6 ms analysis window) allow accurate tracking, especially pitch-synchronously [13].

The automatic tracking of formants is difficult, despite the typical spacing of formants every 1 kHz (for a vocal tract 17 cm long), the limited range of possible bandwidths (30–500 Hz), and the generally slow formant changes. Occasional rapid spectral changes limit the assumption of formant continuity. In oral vowel and sonorant sequences, formants smoothly rise and fall, and are readily followed via spectral peak-picking. Acoustic coupling of the oral and nasal cavities during nasals causes abrupt formant movements as well as the introduction of extra formants. Zeros in the glottal source excitation or in the vocal tract response for lateral or nazalized sounds also tend to obscure formants. Many sounds have two formants close enough that they appear as one spectral peak (e.g., F1–F2 in /o, ɑ/, F2–F3 in /i/). Continuity constraints can often resolve these problems but are frequently thwarted by nasal and obstruent consonants, which interrupt the formants and abruptly alter the spectral distribution of energy. During obstruents, the sound source excites only a forward portion of the vocal tract; thus F1 and often F2 have little energy.

One way to track formants is to estimate speech $S(z)$ in terms of a ratio of z polynomials, solve directly for the roots of the denominator, and identify each root as a formant if it has a narrow bandwidth at a reasonable frequency location. This process can be precise but expensive since the polynomial often has order greater than 10 to represent 4–5 formants (see however a recent fixed-point algorithm [14]). Another approach [15–17] uses phase information to decide whether a spectral peak is a formant. In evaluating $S(z)$ along the unit circle $z = \exp(j\omega)$, a large negative phase shift occurs when ω passes a pole close to the unit circle. Since formants correspond to complex–conjugate pairs of poles with relatively narrow bandwidths (i.e., near the unit circle), each spectral peak having such a phase shift is a formant. The phase shift approaches $-180°$ for small formant bandwidths.

When two formants may appear as one broad spectral peak, a modified DFT can resolve the ambiguity. The *chirp z transform* (CZT) (named after a *chirp*, or signal of increasing frequency) calculates the z transform of the windowed speech on a contour inside the unit circle. Whereas the DFT samples $S(z)$ at uniform intervals on the unit circle, the CZT can follow a spiral contour anywhere in the z plane. It is typically located near poles corresponding to a spectral peak of interest and is evaluated only for a small range of frequency samples. Such a contour can be much closer to the formant poles than for the DFT; thus the CZT can resolve two poles (for two closely spaced formants) into two spectral peaks (Figure 6.10). Because formant bandwidths tend to increase with frequency, the spiral contour often starts near $z = \alpha$, just inside the unit circle (e.g., $\alpha = 0.9$), and gradually spirals inward with increasing frequency $\omega_k = 2\pi k/N$ ($z_k = \alpha\beta^k \exp(j\omega_k)$, with β just less than 1). This contour follows the expected path of the formant poles and eliminates many problems of merged peaks in DFT displays. CZT algorithms can reduce the amount of calculation necessary, approaching that of the DFT, by taking advantage of the spiral nature of the contour in the z plane [18].

Formant trackers have great difficulty when F0 exceeds formant bandwidths, e.g., F0 > 250 Hz [12], as in children's voices. Harmonics in such speech are so widely separated that only one or two constitute each formant. Thus, most spectral analyzers tend to label the most prominent harmonic as the formant, which is erroneous when the center frequency is not a multiple of F0. An analysis using critical band filters, rather than formants, has been more successful in classifying children's vowels [19].

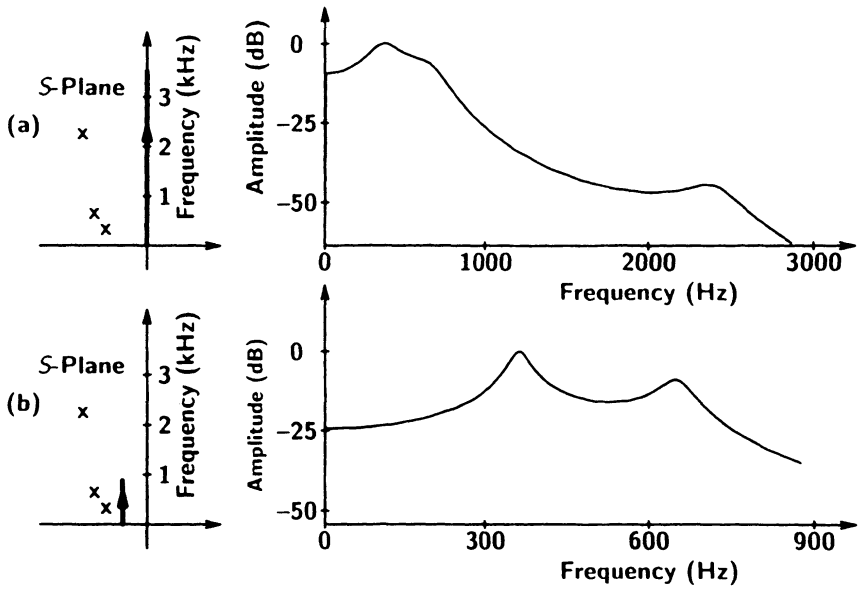

Figure 6.10 Improved frequency resolution obtained by using the chirp z-transform (After Schafer and Rabiner [9].)

6.4.5 Other Spectral Methods (‡)

While formants are widely viewed as important spectral measures for much of speech processing, the difficulty of reliably tracking them has led to related measures. One recent analysis method passes speech through a bank of bandpass filters, and then calculates an autocorrelation of each bandpass power spectrum; following the mel scale, the subband filters have increasing bandwidth with frequency [20]. Another method uses principal components analysis on 16 filter outputs, reducing speech information to as little as two dimensions, which correspond roughly to F1–F2 (but do not require formant tracking) [21].

6.4.6 Energy Separation (‡)

Following recent evidence of significant amplitude and frequency modulations (AM and FM) within pitch periods (due to nonlinear air flows in the vocal tract), an Energy Separation Algorithm (ESA) was developed to analyze these modulations [22]. Each formant is viewed as an AM–FM signal $x(t) = a(t)\cos(\phi(t))$ with AM $a(t)$ and a time-varying frequency $f(t) = (1/2\pi)d\phi(t)/dt = f_c + f_m(t)$, with oscillation $f_m(t)$ around the formant center f_c. An "energy operator" is defined as $\Psi(x(t)) = (dx(t)/dt)^2 - x(t)(d^2x(t)/dt^2)$ (in discrete time: $\Psi_d(x(n)) = (x(n))^2 - x(n-1)x(n+1)$). Under some reasonable assumptions on bandwidths and deviations, it can be shown that $f(t) \approx (1/2\pi)\sqrt{\Psi(dx(t)/dt)/\Psi(x(t))}$ and $|a(t)| \approx \Psi(x(t))/\sqrt{\Psi(dx(t)/dt)}$ [22]. In discrete time, estimations of the envelope and instantaneous frequency only require simple manipulations of a 5-sample moving window [23].

An iterative ESA converges quickly if given good initial estimates for f_c, but requires another form of formant tracker for the initial values. A simulation of what humans do for

estimating formants from narrowband spectrograms uses local minima and maxima in harmonic amplitudes over a range of frequencies B, where B is 250 Hz at low frequency and extends to 750 Hz for the high formants [23]. This method has shown significant AM and FM in formants within pitch periods, presumably due to nonlinearities in vocal tract air flow. The ESA method also provides reliable formant tracking, including good bandwidth estimates [11].

6.5 LINEAR PREDICTIVE CODING (LPC) ANALYSIS

As a model for speech, a popular alternative to the short-time Fourier transform is linear predictive coding (LPC). LPC provides an accurate and economical representation of relevant speech parameters that can reduce transmission rates in speech coding, increase accuracy and reduce calculation in speech recognition, and generate efficient speech synthesis. Chapter 7 examines applications of linear prediction in adaptive differential pulse-code modulation (ADPCM) systems and LP coders, and Chapters 9 and 10 show man–machine applications of LPC.

LPC is the most common techniques for low-bit-rate speech coding and is a very important tool in speech analysis. The popularity of LPC derives from its compact yet precise representation of the speech spectral magnitude as well as its relatively simple computation. LPC has been used to estimate F0, vocal tract area functions, and the frequencies and bandwidths of spectral poles and zeros (e.g., formants), but it primarily provides a small set of speech parameters that represent the configuration of the vocal tract. LPC estimates each speech sample based on a linear combination of its p previous samples; a larger p enables a more accurate model. The weighting factors (or LPC *coefficients*) in the linear combination can be directly used in digital filters as multiplier coefficients for synthesis or can be stored as templates in speech recognizers. LPC coefficients can be transformed into other parameter sets for more efficient coding. We examine below how to calculate the parameters, and also examine spectral estimation via LPC.

LPC has drawbacks: to minimize analysis complexity, the speech signal is usually assumed to come from an all-pole source; i.e., that its spectrum has no zeros. Since actual speech has zeros due to the usual glottal source excitation and due to multiple acoustic paths in nasals and unvoiced sounds, such a model is a simplification, which however does not cause major difficulties in most applications. Nonetheless, some efforts have been made to modify all-pole LPC to model zeros as well.

6.5.1 Basic Principles of LPC

LPC provides an analysis–synthesis system for speech signals [24, 25]. The synthesis model consists of an excitation source $U(z)$ providing input to a spectral shaping filter $H(z)$, yielding output speech $\hat{S}(z)$. Following certain constraints, $U(z)$ and $H(z)$ are chosen so that $\hat{S}(z)$ is close (in some sense) to the original speech $S(z)$. To simplify the modeling problem, $U(z)$ is chosen to have a flat spectral envelope so that most relevant spectral detail lies in $H(z)$. A flat spectrum is a reasonable assumption for $U(z)$ since the vocal tract excitation for unvoiced sounds resembles white noise. For voiced sounds, the source is viewed as a uniform sample train, periodic in N samples (the pitch period), having a line spectrum with uniform-area harmonics (below we discuss problems of viewing a uniform line spectrum as "flat"). The vocal cord puffs of air, which are normally viewed as the excitation for the vocal tract in

voiced speech, can be modeled as the output of a glottal filter whose input is the sample train. The spectral shaping effects of the glottis and the vocal tract are thus combined into one filter $H(z)$.

To simplify obtaining $H(z)$ given a speech signal $s(n)$, we assume the speech to be stationary during each window or frame of N samples. This allows the $H(z)$ filter to be modeled with constant coefficients (to be updated with each frame of data). $H(z)$ is assumed to have p poles and q zeros in the general *pole-zero* case, i.e., a synthetic speech sample $\hat{s}(n)$ can be modeled by a linear combination of the p previous output samples and $q + 1$ previous input samples of an LPC synthesizer:

$$\hat{s}(n) = \sum_{k=1}^{p} a_k \hat{s}(n-k) + G \sum_{l=0}^{q} b_l u(n-l), \tag{6.16}$$

where G is a gain factor for the input speech (assuming $b_0 = 1$). Equivalently,

$$H(z) = \frac{\hat{S}(z)}{U(z)} = G \frac{1 + \sum_{l=1}^{q} b_l z^{-l}}{1 - \sum_{k=1}^{p} a_k z^{-k}}. \tag{6.17}$$

Most LPC work assumes an all-pole model (also known as an *autoregressive*, or AR, model), where $q = 0$. (Any zeros at $z = 0$ are ignored here, because such zeros do not change the spectral magnitude and add only linear phase, since they result from simple time delays.) An all-zero model ($p = 0$) is called a *moving average* (MA) model since the output is a weighted average of the q prior inputs. The more general, but less popular, LPC model with both poles and zeros ($q > 0$) is known as an autoregressive moving average (ARMA) model. We assume here the AR model. If speech $s(n)$ is filtered by an inverse or *predictor* filter (the inverse of an all-pole $H(z)$)

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k}, \tag{6.18}$$

the output $e(n)$ is called an *error* or *residual* signal:

$$e(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k). \tag{6.19}$$

The unit sample response for $A(z)$ has only $p + 1$ samples and comes directly from the set of LPC coefficients: $a(0) = 1$, $a(n) = -a_n$ for $n = 1, 2, \ldots, p$. To the extent that $H(z)$ adequately models the vocal tract system response, $E(z) \approx U(z)$. Since speech production cannot be fully modeled by a p-pole filter $H(z)$, there are differences between $e(n)$ and the presumed impulse train $u(n)$ for voiced speech (Figures 6.11 and 6.12). If $s(n)$ has been recorded without phase distortion [26] and if the inverse filtering is done carefully (e.g., pitch-synchronously), an estimate of the actual glottal waveform can be obtained after appropriate lowpass filtering of $e(n)$ (to simulate the smooth shape of the glottal puff of air) [27, 28].

6.5.2 Least-squares Autocorrelation Method

Two approaches are often used to obtain a set of LPC coefficients a_k characterizing an all-pole $H(z)$ model of the speech spectrum. The classical *least-squares* method chooses a_k to minimize the mean energy in the error signal over a frame of speech data, while the *lattice*

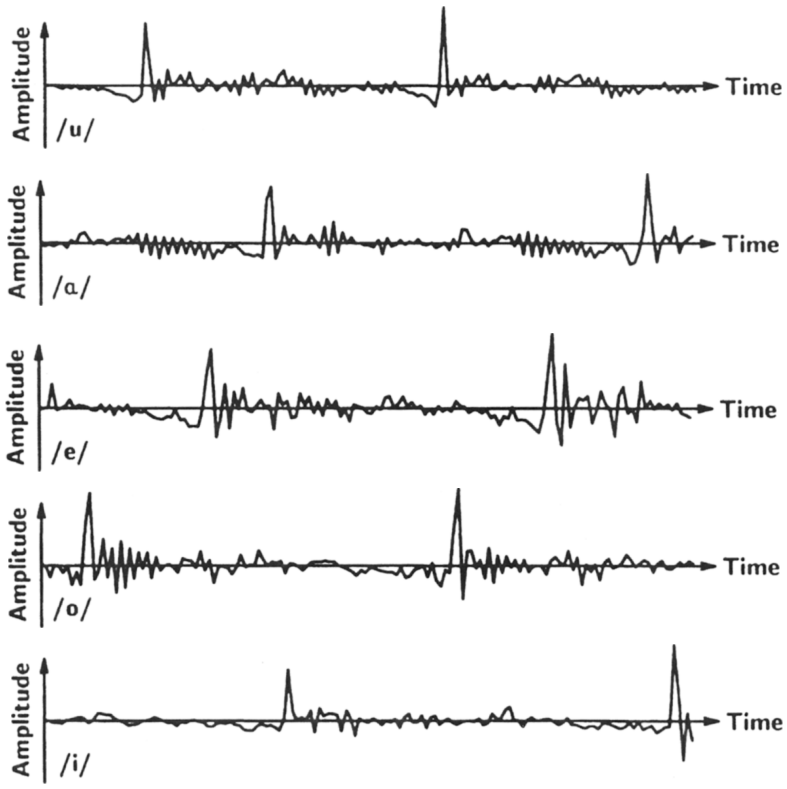

Figure 6.11 Examples of pre-emphasized speech signals and their corresponding prediction error signals for five vowels /u, ɑ, e, o, i/.

approach permits instantaneous updating of the coefficients (at the expense of extra computation). In the former technique, either $s(n)$ or $e(n)$ is windowed to limit the extent of the speech under analysis. The first of two least-squares techniques is the data-windowing or *autocorrelation* method, which multiplies the speech by a Hamming or similar time window

$$x(n) = w(n)s(n) \tag{6.20}$$

so that $x(n)$ has finite duration (N samples, typically corresponding to 20–30 ms). Thus $x(n) = 0$ outside the range $0 \leq n \leq N - 1$. As in other speech analyses, $s(n)$ is assumed to be stationary during each window. LPC equally considers all speech samples within each frame; thus for nonstationary speech, the LPC coefficients describe a smoothed average of the signal.

Let E be the error energy:

$$E = \sum_{n=-\infty}^{\infty} e^2(n) = \sum_{n=-\infty}^{\infty} \left[x(n) - \sum_{k=1}^{p} a_k x(n-k) \right]^2, \tag{6.21}$$

where $e(n)$ is the residual corresponding to the windowed signal $x(n)$. The values of a_k that minimize E are found by setting $\partial E / \partial a_k = 0$ for $k = 1, 2, 3, \ldots, p$. This yields p linear

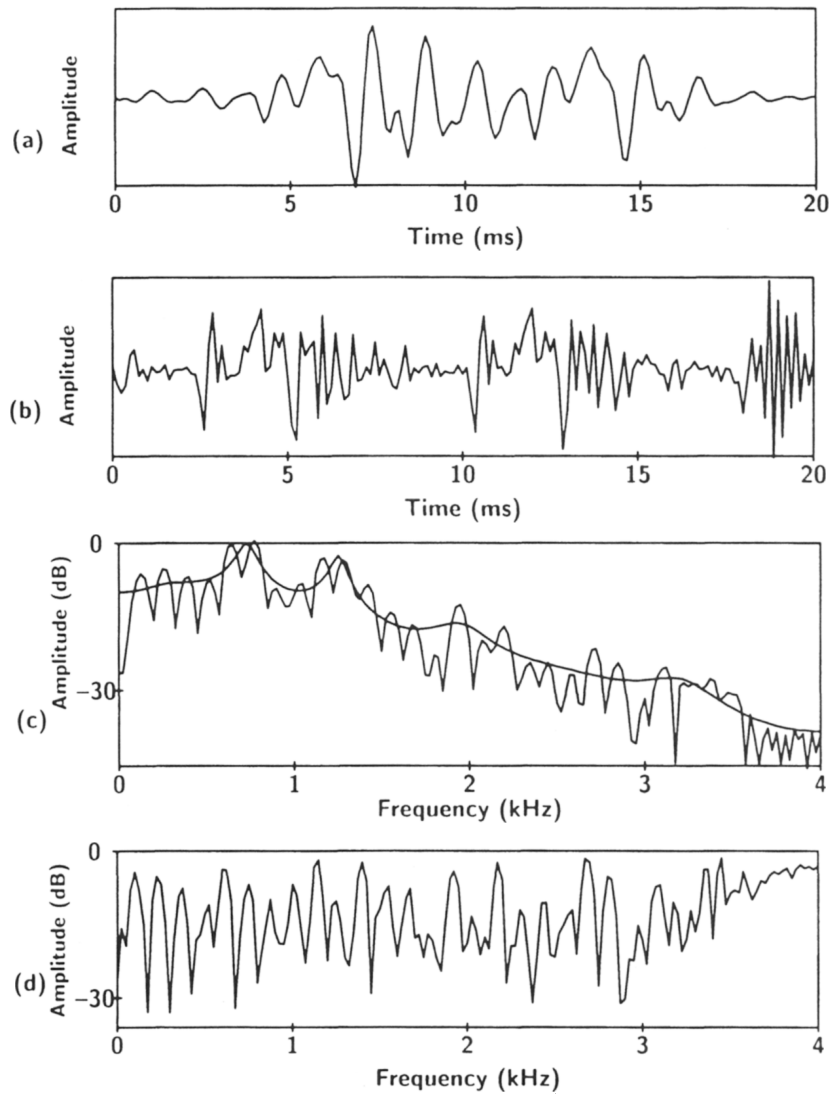

Figure 6.12 Signals and spectra in LPC via the autocorrelation method using 12 poles: (a) 20 ms of an /ɑ/ vowel from a male speaker at 8000 samples/s (using a Hamming window); (b) residual error signal obtained by inverse LPC filtering the speech (magnified about 3 times); (c) speech spectrum with the smooth LPC spectrum superimposed; and (d) spectrum of the error signal (note the different amplitude scales for parts c and d).

equations

$$\sum_{n=-\infty}^{\infty} x(n-i)x(n) = \sum_{k=1}^{p} a_k \sum_{n=-\infty}^{\infty} x(n-i)x(n-k), \qquad \text{for } i = 1, 2, 3, \ldots, p, \tag{6.22}$$

in p unknowns a_k. Recognizing the first term as the autocorrelation $R(i)$ of $x(n)$ and taking advantage of the finite duration of $x(n)$, we have

$$R(i) = \sum_{n=i}^{N-1} x(n)x(n-i), \qquad \text{for } i = 1, 2, 3, \ldots, p, \tag{6.23}$$

so that Equations (6.22) reduce to

$$\sum_{k=1}^{p} a_k R(i-k) = R(i), \qquad \text{for } i = 1, 2, 3, \ldots, p. \tag{6.24}$$

The autocorrelation could be calculated for all integers i, but since $R(i)$ is an even function, it need be determined only for $0 \leq i \leq p$. From Equations (6.21) and (6.24), the minimum residual energy or *prediction error* E_p for a p-pole model is

$$E_p = R(0) - \sum_{k=1}^{p} a_k R(k), \tag{6.25}$$

where the first term $R(0)$ is simply the energy in $x(n)$. For synthesis, setting $G^2 = E_p$ in Equation (6.16) yields an energy match between the original windowed speech and the synthesized version. The match can be imprecise when output pitch periods overlap significantly (yielding only slight speech degradation), but may cause overflows [29] when implementing LPC synthesis in fixed-point arithmetic [30].

The conventional least-squares method is equivalent to a *maximum likelihood* (*ML*) approach to parameter estimation; it simplifies computation, but ignores certain information about speech production. Alternative *maximum a posteriori* (*MAP*) methods exploit better the redundancies in the speech signal, but at a high cost. Constraints on the MAP estimation process (e.g., smooth time contours) which aid speech applications are feasible [31].

6.5.3 Least-Squares Covariance Method

An alternative least-squares technique of LPC analysis, the *covariance* method, windows the error $e(n)$ instead of $s(n)$:

$$E_p = \sum_{n=-\infty}^{\infty} e^2(n)w(n). \tag{6.26}$$

Setting $\partial E/\partial a_k = 0$ again to zero leads to p linear equations

$$\sum_{k=1}^{p} a_k \phi(i,k) = \phi(0,i), \qquad 1 \leq i \leq p, \tag{6.27}$$

where

$$\phi(i,k) = \sum_{n=-\infty}^{\infty} s(n-k)s(n-i)w(n) \tag{6.28}$$

is the covariance function for $s(n)$. Usually the error is weighted uniformly in time via a simple rectangular window of N samples, and Equation (6.28) reduces to

$$\phi(i, k) = \sum_{n=0}^{N-1} s(n-k)s(n-i), \qquad \text{for } 0 \leq (i, k) \leq p. \tag{6.29}$$

The autocorrelation R and covariance ϕ functions are quite similar, but they differ in the windowing effects. The autocorrelation method uses N (Hamming) windowed speech samples, whereas the covariance method uses no window on the speech samples. The former thus introduces distortion into the spectral estimation since windowing corresponds to convolving the original short-time $S(e^{j\omega})$ with the frequency response of the window $W(e^{j\omega})$. Since most windows have lowpass spectra, the windowed speech spectrum is a smoothed version of the original, with the extent and type of smoothing dependent on the window shape and duration. The covariance method avoids this distortion, but requires knowledge of $N + p$ speech samples ($s(n)$ for $-p \leq n \leq N - 1$ in Equation (6.29)).

6.5.4 Computational Considerations

In the autocorrelation method, the p linear equations (Equation (6.24)) to be solved can be viewed in matrix form as $\mathbf{R}A = \mathbf{r}$, where $\mathbf{R}$ is a $p \times p$ matrix of elements $\mathbf{R}(i, k) = R(|i - k|)$, $(1 \leq i, k \leq p)$, $\mathbf{r}$ is a column vector $(R(1), R(2), \ldots, R(p))^T$, and A is a column vector of LPC coefficients $(a_1, a_2, \ldots, a_p)^T$. Solving for the LPC vector requires inversion of the $\mathbf{R}$ matrix and multiplication of the resultant $p \times p$ matrix with the $\mathbf{r}$ vector. A parallel situation occurs for the covariance approach if we replace the autocorrelation matrix $\mathbf{R}$ with the $p \times p$ covariance matrix ¶ of elements $¶(i, k) = \phi(i, k)$ and substitute the $\mathbf{r}$ vector with a ϕ vector $(\phi(0, 1), \phi(0, 2), \ldots, \phi(0, p))$. Calculation of the minimum residual error E_p can also be expressed in vector form as the product of an extended LPC vector

$$\mathbf{a} = (1, -a_1, -a_2, \ldots, -a_p), \tag{6.30}$$

with either the $\mathbf{r}$ or ϕ vector augmented to include as its first element the speech energy ($R(0)$ or $\phi(0, 0)$, respectively). The extended LPC vector contains the $p + 1$ coefficients of the LPC inverse filter $A(z)$.

Redundancies in the $\mathbf{R}$ and ¶ matrices allow efficient calculation of the LPC coefficients without explicitly inverting a $p \times p$ matrix. Both matrices are symmetric (e.g., $\phi(i, k) = \phi(k, i)$); however, $\mathbf{R}$ is also Toeplitz (all elements along a given diagonal are equal), whereas ¶ is not. As a result, the autocorrelation approach is simpler ($2p$ storage locations and $O(p^2)$ math operations) than the basic covariance method ($p^2/2$ storage locations and $O(p^3)$ operations, although this can be reduced to $O(p^2)$ operations [32]). ($O(p)$ means "of the order of p" and indicates approximation.) If $N \gg p$ (often true in speech processing), then computation of the $\mathbf{R}$ or ¶ matrix ($O(pN)$ operations) dominates the overall calculation. (N often exceeds 100, while p is about 10.) Assuming the ¶ matrix is positive definite (generally true for speech input), its symmetry allows solution through the square root or Cholesky decomposition method [33], which roughly halves the computation and storage needed for direct matrix inversion techniques.

The additional redundancy in the Toeplitz **R** matrix allows the more efficient Levinson–Durbin recursive procedure [25, 33], in which the following set of ordered equations is solved recursively for $m = 1, 2, \ldots, p$:

$$k_m = \frac{R(m) - \sum_{k=1}^{m-1} a_{m-1}(k)R(m-k)}{E_{m-1}}, \tag{6.31a}$$

$$a_m(m) = k_m, \tag{6.31b}$$

$$a_m(k) = a_{m-1}(k) - k_m a_{m-1}(m-k) \qquad \text{for } 1 \le k \le m-1, \tag{6.31c}$$

$$E_m = (1 - k_m^2)E_{m-1}, \tag{6.31d}$$

where initially $E_0 = R(0)$ and $a_0 = 0$. At each cycle m, the coefficients $a_m(k)$ (for $k = 1, 2, \ldots, m$) describe the optimal mth-order linear predictor, and the minimum error E_m is reduced by the factor $(1 - k_m^2)$. Since E_m, a squared error, is never negative, $|k_m| \le 1$. This condition on the *reflection coefficients* k_m, which can be related to acoustic tube models, also guarantees a stable LPC synthesis filter $H(z)$ since all the roots of $A(z)$ are then inside (or on) the unit circle in the z plane. The negatives of the reflection coefficients are called *partial correlation*, or PARCOR, coefficients. The k_m rarely have magnitude equal to unity since that would terminate the recursion with $E_m = 0$ and yield $H(z)$ poles on the unit circle, a marginally stable situation. Unlike the covariance method, the autocorrelation method, even when not calculating the reflection coefficients directly, guarantees a stable synthesis filter when using infinite-precision calculation.

One radical way to reduce calculation in LPC analysis is to center-clip and infinite-peak clip the speech signal before LPC processing. Clipping is useful for F0 estimation as a means to simplify the speech signal, eliminating formant detail while preserving periodicity. If the clipping level is lowered to about 20% of its value in F0 estimation, formant detail is also preserved, yet the signal may be simplified to contain only values of -1, 0, and $+1$. Calculating the autocorrelation matrix for LPC using such a signal involves no multiplications, which greatly reduces computation. Some supplementary multiplications must be done, however, to find the LPC gain since clipping destroys energy information. The cost for such efficiency is that synthetic LPC spectra differ from the original by about 2 dB [34], which can be significant.

6.5.5 Spectral Estimation via LPC

Parseval's theorem for the energy E of a discrete-time signal (e.g., the error signal $e(n)$) and its Fourier transform is

$$E = \sum_{n=-\infty}^{\infty} e^2(n) = \frac{1}{2\pi}\int_{\omega=-\pi}^{\pi} |E(e^{j\omega})|^2 \, d\omega. \tag{6.32}$$

Since $e(n)$ can be obtained by passing speech $s(n)$ through its inverse LPC filter $A(z) = G/H(z)$, the residual error can be expressed as

$$E_p = \frac{G^2}{2\pi}\int_{\omega=-\pi}^{\pi} \frac{|S(e^{j\omega})|^2}{|H(e^{j\omega})|^2} d\omega. \tag{6.33}$$

Obtaining the LPC coefficients by minimizing E_p is equivalent to minimizing the average ratio of the speech spectrum to its LPC approximation. Equal weight is given to all

frequencies, but $|H(e^{j\omega})|$ models the peaks in $|S(e^{j\omega})|$ better than its valleys (Figure 6.12c) because the contribution to the error E_p at frequencies where the speech spectrum exceeds its LPC approximation is greater than for the opposite condition. The LPC all-pole spectrum $|H(e^{j\omega})|$ is limited, by the number p of poles used, in the degree of spatial detail it can model in $|S(e^{j\omega})|$. For a typical $p = 10$, at most five resonances can be represented accurately. A short-time voiced-speech spectrum, with rapid frequency variation due to the harmonics as well as the slower variations due to the formant structures, cannot be completely modeled by such an $|H(e^{j\omega})|$. Locating the (smooth) LPC spectrum well below the (ragged) speech spectrum (to model spectral valleys well) would cause large contributions to the overall error at spectral peaks. $|H(e^{j\omega})|$ tends to follow the spectral envelope of $|S(e^{j\omega})|$ just below the harmonic peaks, which balances small errors at peak frequencies with larger errors in valleys (which contribute less to E_p). Thus, the valleys between harmonics are less well modeled than the harmonic peaks, and valleys between formants (including those due to zeros in the vocal tract transfer function) are less accurately modeled than formant regions. The importance of good formant modeling has been underlined recently by suggested modifications to LPC to emphasize narrow bandwidth components in the spectral model [35].

6.5.5.1 Pre-emphasis. Many analysis methods concentrate on the high-energy portions of the speech spectrum. It is nonetheless clear that relatively weak energy at high frequencies is often important in many applications. To help model formants of differing intensity equally well, input speech energy is often raised as a function of frequency prior to spectral analysis (e.g., LPC) via pre-emphasis. The degree of pre-emphasis is controlled by a constant α, which determines the cutoff frequency of the single-zero filter through which speech effectively passes. This reduces the dynamic range (i.e., "flattens" the speech spectrum) by adding a zero to counteract the spectral falloff due to the glottal source in voiced speech. The pre-emphasis and radiation zeros approximately cancel the falloff, giving formants of similar amplitudes. In speech coding, the final stage of synthesis must contain a *de-emphasis* filter $1/(1 - \beta z^{-1})$ to undo the pre-emphasis. With values of α and β of typically about 0.94, pre-emphasis acts as a differentiator, while de-emphasis performs integration. In addition to making spectral analysis more uniform in frequency, pre-emphasis reduces a signal's dynamic range, facilitating some fixed-point implementations [24].

Usually β is chosen equal to α so that the de-emphasis exactly cancels the pre-emphasis effects, but when α is near unity, a slight mismatch often yields higher-quality speech. Such common high values for α locate the pre-emphasis zero at very low frequency, causing significant attenuation in the region below F1, which in turn is poorly matched by the usual LPC analysis. Frequently, LPC spectra overestimate gain at these low frequencies. By allowing $\beta < \alpha$ (e.g., 0.74 and 0.94, respectively), this mismatch can be reduced, while having little effect on the formant spectra [36]. Whereas intelligibility depends little on frequencies below F1, much energy is present there in voiced speech, and a proper spectral match is important for naturalness.

6.5.5.2 Order of the LPC model. In the LPC model, the choice of the order p is a com-promise among spectral accuracy, computation time/memory, and transmission bandwidth (the last being relevant only for coding applications). In the limit as $p \to \infty$, $|H(e^{j\omega})|$ matches $|S(e^{j\omega})|$ exactly (Figure 6.13), but at the cost of memory and computation. In general, poles are needed to represent all formants (two poles per resonance) in the signal bandwidth plus an additional 2–4 poles to approximate possible zeros in the spectrum and general spectral shaping (e.g., the standard for 8 kHz sampled speech is 10 poles [37]). The

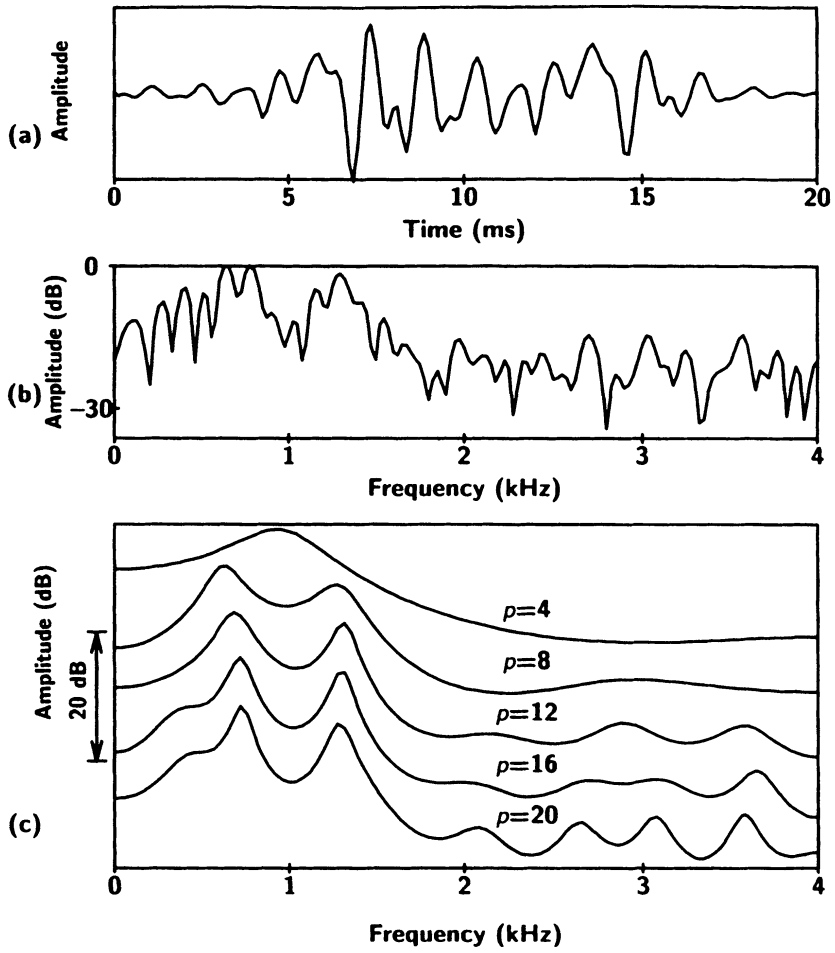

Figure 6.13 Signals and spectra in LPC for 20 ms of an /α/ vowel at 8000 samples/s: (a) time waveform, (b) speech spectrum, (c)–(g) LPC spectra using 4, 8, 12, 16, and 20 poles, respectively.

latter effects come mostly from the spectra of the glottal waveform and lip radiation, but zeros also arise from nasalized and unvoiced sounds. It is usually unnecessary to add more poles to the model for nasals, despite the extra nasal formants in such speech, since high-frequency formants in nasals have wide bandwidths and so little energy that their accurate spectral modeling is unimportant.

The all-pole LPC model can handle zeros indirectly; e.g., a zero at $z = a$ ($|a| < 1$) can be exactly represented by an infinite number of poles:

$$(1 - az^{-1}) = \frac{1}{1 - \sum_{n=1}^{\infty}(az^{-1})^n}. \tag{6.34}$$

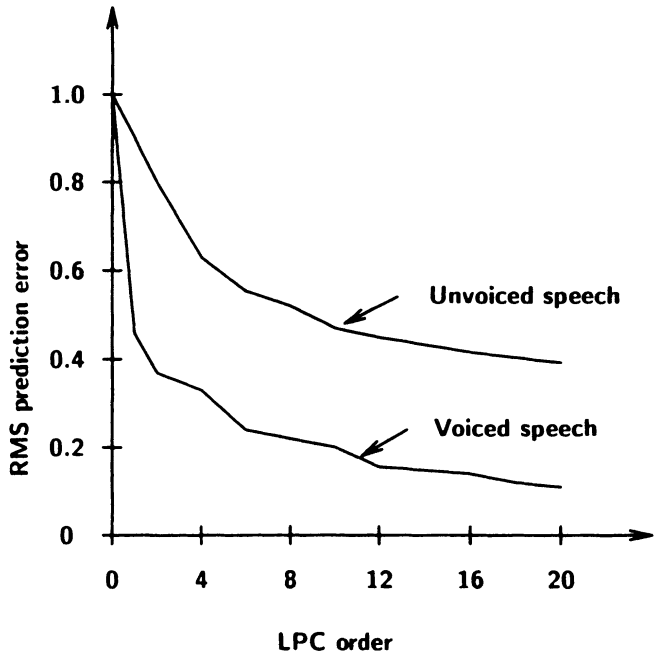

Figure 6.14 Normalized prediction error as a function of the LPC model order. (After Atal and Hanauer [39].)

Evaluating on the unit circle ($z = e^{j\omega}$), we can approximate the infinite-order denominator with a finite number of terms (e.g., M) and hence a finite number of poles. The high-order terms in Equation (6.34) can be ignored if $a^M \ll 1$. Wide-bandwidth zeros (i.e., those with small $|a|$) are more accurately modeled with a few poles than are zeros whose bandwidths are comparable to those of the formants. It is generally (but not universally [38]) assumed that 2–4 poles can handle the zeros and other glottal effects, given the ear's greater sensitivity to spectral peaks than valleys.

The prediction error energy E_p is often used as a measure of the accuracy of an LPC model. The *normalized prediction error* (i.e., divided by the speech energy), $V_p = E_p/R(0)$ (see Equation (6.31d)), decreases monotonically with predictor order p (Figure 6.14) (i.e., each additional pole improves the model). For voiced speech, after having enough poles to model the formant structure (e.g., $p = 10$), additional poles do little to improve the spectral fit (as measured by V_p), but they add significantly to the computation (and to bit rate, for vocoders). Unvoiced speech yields larger V_p because its excitation signal is spread out in time. The usual calculation of LPC coefficients ignores $u(n)$ in Equation (6.16). (The effects of $u(n)$ for voiced speech are small for a small analysis frame located in the middle of a pitch period, but this requires a period detector to find the F0 epochs before LPC analysis.) Unvoiced $u(n)$ has relatively constant energy over the analysis frame; in voiced speech, $u(n)$ has energy concentrated at the start of each pitch period (primarily when the vocal cords close), allowing $u(n)$ to be ignored for most of the speech samples. Thus, the LPC model is a better fit to voiced speech because ignoring $u(n)$ is valid for more time samples in Equation (6.21) for voiced speech. Some algorithms exploit this distinction by basing voiced–unvoiced decisions on the relative V_p (high, unvoiced; low, voiced).

A recent proposal to account for $u(n)$ in the LPC representation of voiced speech suggests modifying the spectral coefficients at the synthesis stage, to make the original speech and the corresponding synthetic speech more similar. Specifically, the first $p + 1$ autocorrelation coefficients $R(i)$ should be identical for the two signals, except for the interference of multiple excitations in $u(n)$ for the speech within the frame of analysis. In the case of voices with high F0 (leading to more excitations per frame), two iterations modifying the LPC coefficients to guarantee the $R(i)$ match lead to significant improvements in synthetic speech quality [40].

6.5.6 Updating the LPC Model Sample by Sample

We earlier described the *block estimation* approach to LPC analysis, where spectral coefficients are obtained for each successive frame of data. Alternatively, LPC parameters can be determined sample by sample, updating the model for each speech sample. For real-time implementation (e.g., echo cancellation in the telephone network [41]), this reduces the delay inherent in the block approach, where typical frame lengths of 20–30 ms cause 10–15 ms delays. Chapter 7 notes that ADPCM with feedback adaptation requires an instantaneous method for updating its predictor, based only on transmitted residual samples. Feedforward ADPCM allows block LPC estimation, but feedback ADPCM with its minimal delay and lack of side information does not. In instantaneous LPC estimation, a recursive procedure is necessary to minimize computation. Each new sample updates some intermediate speech measure (e.g., a local energy or covariance measure), from which the LPC parameters are revised. Recalculating and inverting the covariance matrix for each speech sample, as in the block methods, is unnecessary.

6.5.7 Transversal Predictors

The two basic ways to implement a linear predictor are the *transversal* form (i.e., direct-form digital filter) and the *lattice* form. The transversal predictor derives directly from Equation (6.16) and updates N LPC coefficients $a_k(n)$ (the kth spectral coefficient at time n) as follows:

$$a_k(n+1) = va_k(n) + (1-v)a_k^* + G_k(n+1)e(n+1), \tag{6.35}$$

where a^* is a target vector of coefficients that is approached exponentially in time (depending on the damping factor v) during silence (i.e., when the LPC error $e(n) = 0$) and G is an "automatic gain control" vector (based on the N previous speech samples) that controls the model adaptation. The *gradient* or *least-mean-square* (LMS) approach assigns simple values to G:

$$G_k(n) = \frac{\hat{s}(n-k)}{C + \sum_{i=0}^{N-1} w^i \hat{s}^2(n-i-1)}, \tag{6.36}$$

where the denominator is simply a recent speech energy estimate (with weighting controlled by a damping factor w) and C is a constant to avoid division by zero during silence. Alternatives to the gradient approach, e.g., the *Kalman algorithm*, trade more computation for G against more accurate LPC coefficients [2, 42].

6.5.8 Lattice LPC Models

The *lattice* method for LPC typically involves both a *forward* and a *backward* prediction [43]. (These should not be confused with feedforward and feedback adaptation of waveform coders.) Block LPC analysis uses only forward prediction (i.e., the estimate $\hat{s}(n)$ is based on p prior samples of $s(n)$), but the estimation can be done similarly from p ensuing samples in a form of backward "prediction." Consider $a_m(n)$ to be the unit-sample response of a fixed $A_m(z)$, the inverse LPC filter for a block of data at the mth stage of the Durbin recursion (Equation (6.31)) (i.e., for an m-pole model). The usual (forward) error signal $f_m(n)$ is the convolution of $s(n)$ with $a_m(n)$. Applying Equation (6.31c),

$$f_m(n) = s(n) * a_{m-1}(n) - k_m s(n) * a_{m-1}(m - n), \tag{6.37}$$

whose first term is the forward error from an $(m - 1)$th predictor and whose second term is a parallel backward error. Assigning $b_m(n)$ to this backward error yields a recursion formula:

$$f_m(n) = f_{m-1}(n) - k_m b_{m-1}(n - 1), \tag{6.38}$$

where

$$b_m(n) = s(n) * a_m(m - n) = \sum_{l=n-m}^{n} s(l)a_m(m - n + l). \tag{6.39}$$

Shifting index l by $n - m$ samples and noting that $a_m(0) = 1$ (from Equation (6.18)), we obtain

$$b_m(n) = s(n - m) - \sum_{l=1}^{m} a_m(l)s(n - m + l), \tag{6.40}$$

which has the interpretation of predicting sample $s(n - m)$ from m ensuing samples of $s(n)$ (note the similarity to Equation (6.19)). The same set of $m + 1$ samples is involved in both the forward prediction of $s(n)$ and the backward prediction of $s(n - m)$. The recursion formula for the mth stage of backward prediction can be derived in similar fashion:

$$b_m(n) = b_{m-1}(n - 1) - k_m f_{m-1}(n). \tag{6.41}$$

The recursion Equations (6.38) and (6.41) lead to the lattice flow diagram of Figure 6.15(a), with initial conditions of $f_0(n) = b_0(n) = s(n)$; i.e., using no predictor gives an "error" equal to the speech signal itself. The corresponding synthesis filter in Figure 6.15(b) can be derived directly from the same recursion equations by viewing Equation (6.38) as

$$f_{m-1}(n) = f_m(n) - k_m b_{m-1}(n - 1). \tag{6.42}$$

The lattice synthesizer has the same form as one of the vocal tract models in Chapter 3, viewed as a lossless acoustic tube of p sections of equal length with uniform cross-sectional area A_m within each section. The reflection coefficients k_m could specify the amount of plane wave reflection at each section boundary:

$$k_m = \frac{A_m - A_{m-1}}{A_m + A_{m-1}}. \tag{6.43}$$

Efforts to relate these k_m (obtained from speech) to corresponding vocal tract areas, however, have met with only limited success because (a) natural vocal tracts have losses, and (b) standard models using k_m must locate all losses at the glottal or labial ends [44]. If glottal

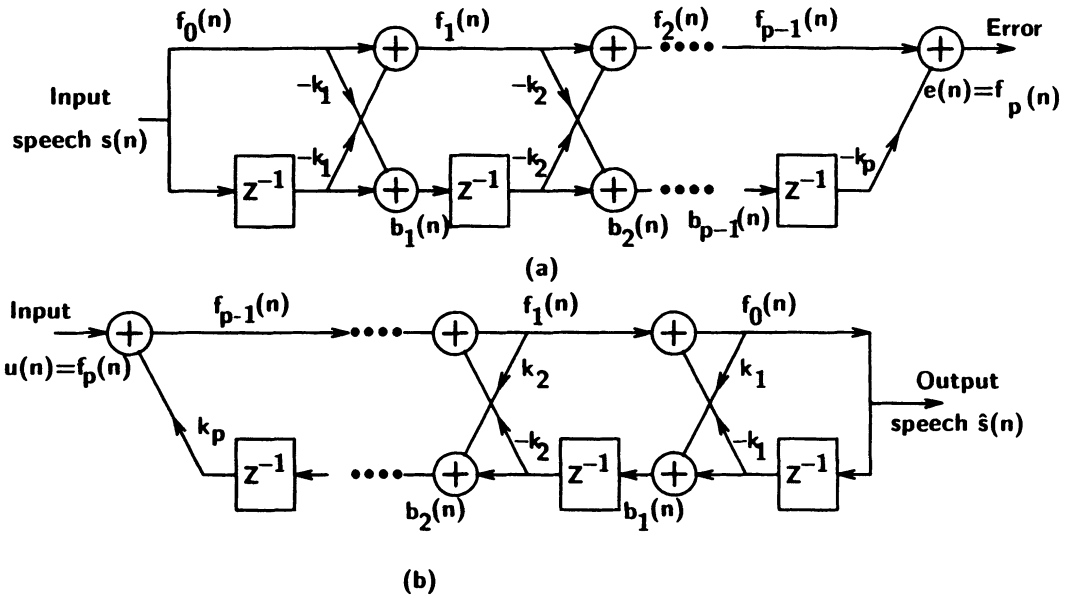

Figure 6.15 Lattice filters: (a) inverse filter $A(z)$, which generates both forward and backward error signals at each stage of the lattice; (b) synthesis filter $1/A(z)$.

pressure can be measured (e.g., through skin accelerometers attached to the throat) in addition to the speech signal, then accurate vocal tract shapes can be determined automatically [45].

Applying z transforms to Equation (6.37), we have

$$F_m(z) = S(z)[A_{m-1}(z) - k_m z^{-m} A_{m-1}(z^{-1})]. \tag{6.44}$$

If $S(z)$ is temporarily considered as unity (i.e., to find the filter's unit-sample response, using $s(n) = \delta(n)$), then $F_m(z) = A_m(z)$, yielding a recursion formula for the mth stage of the LPC inverse filter via the reflection coefficients:

$$A_m(z) = A_{m-1}(z) - k_m z^{-m} A_{m-1}(z^{-1}). \tag{6.45}$$

Minimizing the forward energy over an appropriate time window, Equation (6.38) gives

$$k_{m+1}^f = \frac{E[f_m(n) b_m(n-1)]}{E[b_m^2(n-1)]}, \tag{6.46}$$

where $E[\ ]$ means expectation (averaging), k_m^f denotes the reflection coefficient obtained using forward error minimization at the mth stage of LPC lattice analysis, and k_{m+1}^f is equal to the ratio of the cross-correlation between the forward and backward errors to the backward error energy. Equivalently, minimizing the backward error energy leads to

$$k_{m+1}^b = \frac{E[f_m(n) b_m(n-1)]}{E[f_m^2(n)]}, \tag{6.47}$$

the ratio of the cross-correlation to the forward error energy. The disadvantage of both approaches is that neither guarantees that $k_m < 1$ for all m, although it can be shown that either k_m^f or k_m^b must be so bounded for each m.

For instantaneous adaptation of LPC coefficients obtained via the lattice approach, the Itakura and Burg methods [33, 46] are popular. The Itakura method follows directly from the Levinson–Durbin recursion and defines the reflection coefficients as

$$k_m = \frac{E[f_{m-1}(n)b_{m-1}(n-1)]}{\{E[f_{m-1}^2(n)]E[b_{m-1}^2(n-1)]\}^{1/2}}, \tag{6.48}$$

i.e., the partial correlation between forward and backward error signals, normalized by their energies. As PARCOR coefficients, the $k_m \leq 1$, thus guaranteeing stable synthesis filters (even when using quantized coefficient values and finite-wordlength computation [24]).

Windowing the error instead of the speech signal suggests an adaptive method to update the model sample by sample. The Burg technique minimizes

$$E_m = \sum_{n=-\infty}^{\infty} w(n)[f_m^2(n) + b_m^2(n)], \tag{6.49}$$

where the $w(n)$ error window could be rectangular (as in the Itakura method) or shaped so that more recent speech samples are weighted more heavily, e.g., simple real-pole filters of the form

$$W(z) = \frac{1}{(1-\beta z^{-1})^L}. \tag{6.50}$$

(Good-quality speech results when $L = 3$ and $\beta = 1 - (100L/F_s)$ [47].) This leads to reflection coefficients involving the ratio of the cross-correlation between the forward and backward errors to the average of the two error energies:

$$k_m = \frac{\sum_{n=-\infty}^{\infty} w(n)f_{m-1}(n)b_{m-1}(n-1)}{\frac{1}{2}\sum_{n=-\infty}^{\infty} w(n)[f_{m-1}^2(n) + b_{m-1}^2(n-1)]}. \tag{6.51}$$

Coefficient magnitudes are bounded by unity if $w(n) > 0$ (over its finite duration). Figure 6.16 illustrates how the reflection coefficients for time $n + 1$ can be obtained from the immediately prior error samples. The filter memories for $W(z)$ retain the necessary information about earlier speech samples in the window. See [2, 48] for alternative *least-squares* (LS) approaches. Two-bit (16 kbit/s) ADPCM with fourth-order adaptive prediction performs best with the LS lattice approach, which yields SEGSNR of 15 dB, about 1–2 dB better than nonadaptive or other adaptive methods [2].

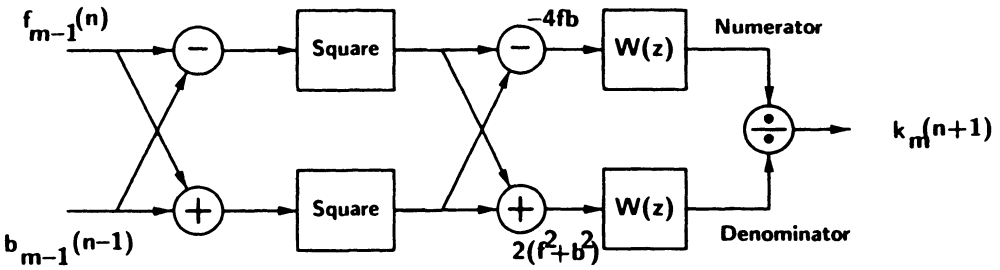

Figure 6.16 Adaptive estimation of reflection coefficients (only the mth stage of p identical stages is shown).

Traditional lattice methods, updating the LPC parameters every sample, require $5p$ multiplication operations per speech sample (where p is the LPC order), compared to p multiplies/sample in calculating the autocorrelation or covariance matrices in the block approaches. (This assumes that each sample is used in one matrix calculation, i.e., in nonoverlapping blocks of data, and ignores the matrix inversion for $N \gg p$, which adds $O(p^2)$ or $O(p^3)$ multiplies.) Similarly, three memory locations per sample (to store the forward and backward errors, and the speech) are needed in the lattice approach, compared with one location per sample for the other methods. More efficient techniques, however, exist, for block-lattice LPC analysis, making that approach computationally comparable to other block estimation LPC [46].

6.5.9 Window Considerations

Both window size N and order p should be small to minimize calculation in LPC analysis. However, since p is usually specified by the speech bandwidth, only N allows any flexibility to trade off spectral accuracy and computation. Due to windowing distortion, the autocorrelation LPC window must include at least two pitch periods for accurate spectral estimates (20–30 ms typically, to guarantee two periods even at low F0). In the lattice and covariance methods, the lack of signal windowing theoretically allows windows as short as $N = p$, but spectral accuracy usually increases with larger N. The major difficulty with short windows concerns the unpredictability of the speech excitation signal $u(n)$. The LPC model predicts a speech sample based on p prior samples, assuming that an all-pole vocal tract filter describes the signal. It makes no attempt to deconvolve $s(n)$ into $h(n)$ and $u(n)$ and cannot distinguish vocal tract resonances and excitation effects. The poles of the LPC model correspond primarily to vocal tract resonances but also account for the excitation disturbance.

Most LPC analysis is done pitch-asynchronously, i.e., without regard for F0; e.g., adaptive lattice techniques evaluate for every sample, and block methods usually examine the sets of N samples which are shifted periodically by N or $N/2$ samples. This leads to poorer spectral estimation when pitch epochs (the large initial samples of periods, which are unpredictable for small p) are included during an analysis frame. The problem is worse when N is small because some analysis frames are then dominated by poorly modeled excitation effects. Spectral accuracy improves if N is large enough to contain a few pitch periods, because the LPC model is good for speech samples in each period after the first p (i.e., after the first p samples, $s(n)$ is based on prior samples that all include the effects of the major pitch excitation). Use of a rectangular window to evaluate the error signal pitch-asynchronously leads to fluctuating spectral estimates, with the size of the variations inversely related to window length. They can be reduced by using a smooth (e.g., Hamming) window to weight the error in Equation (6.28) [49], at the cost of some increased computation. Another possible solution which trades off computation for improved spectral estimates, is to eliminate from the analysis window those speech samples $s(n)$ that lead to values of $e(n)$ exceeding a specified threshold [50]. These large-error samples (usually near pitch epochs) degrade the spectral estimates the most. This approach does not need a pitch epoch locator as in pitch-synchronous methods, but uses less efficient algorithms than the standard autocorrelation or covariance techniques do.

These problems can be partly avoided by *pitch-synchronous analysis*, where each analysis window is fully within a period. This, however, requires an accurate F0 estimator

since short windows yield poor spectral estimation for frames improperly placed. The extra computation required for (sometimes unreliable) epoch location has deterred most LPC analysis from using short windows. For rapidly changing speech, however, accurate estimates require pitch-synchronous techniques [51]. Here, the covariance method is typically used since (unlike the autocorrelation method) it requires no explicit window that would distort the signal significantly over short frame analyses of less than one pitch period. The standard Burg method does not perform well with short windows because its use of both forward and backward errors presumes similar energy in the two residuals [52]. A modified Burg technique, which weights the instantaneous LPC error with a tapered window prior to error minimization, yields spectral estimates approaching those of the covariance method, except that formant bandwidths are underestimated [53].

6.5.10 Modifications to Standard LPC

The standard forms of all-pole LPC analysis, minimizing the energy in the error signal over a time window, are simple and computationally efficient. However, their spectral estimates are flawed due to inherent limitations in the procedure. Zeros in the speech spectrum can only be approximately modeled by poles, and their presence causes the pole estimates to deviate from actual formant values. Not accounting for vocal tract excitation in pitch-asynchronous LPC analysis leads to vocal tract estimates that vary with the fine structure of the speech spectrum. Such structure depends on environmental noise and the choice of analysis window as well as on the actual vocal tract excitation. Placement of the analysis window not aligned with a pitch period causes variation in LPC parameters even during stationary speech [54], which can cause warbling in LPC speech.

The problem of poor spectral estimation is especially acute for high-F0 voices, where several pitch impulses occur in a typical analysis frame and few harmonics are available to define the center frequencies and bandwidths of the crucial F1–F2 formants. When one harmonic dominates a formant, LPC often incorrectly places a pole frequency to coincide with the harmonic. Synthesis based on such parameters is usually poor when F0 deviates from its original values, as when F0 is quantized or when an alternate F0 counter is used.

Basic LPC analysis weighs high-amplitude frequencies (e.g., harmonic peaks) more than spectral valleys, which corresponds well with perceptual resolution. For high-F0 voices, however, the weighting could be adjusted to improve spectral estimates and vocoder speech quality. One could compress the amplitude of the speech spectrum before LPC analysis (e.g., by taking its cube root) [55]. This, however, may require a preliminary DFT and inverse DFT on the windowed speech signal (before and after the root operation, respectively) to obtain a transformed autocorrelation signal for LPC analysis. In a vocoder application, two more DFTs in the synthesis stage would be needed to compensate for the spectral distortion of the analysis stage.

A less costly way is to identify the harmonic peaks (through F0 estimation and a peak picking operation on the speech spectrum) and to transform the ragged speech spectrum $|S(e^{j\omega})|$ (with ripples due to the fine structure of the harmonics) into a smooth approximation of the vocal tract spectrum $|H(e^{j\omega})|$ via parabolic interpolation of the peaks [55]. Such a transformation preserves the basic shape of the spectral envelope, eliminating most of the fine-structure interference. After an inverse DFT, the resulting autocorrelation function can be

used as input to standard autocorrelation method LPC analysis. For high-F0 voices, this approach improves spectral estimation and speech quality at the cost of extra computation.

Another approach, called *Perceptual Linear Prediction* (PLP), is useful for speech recognition. It follows some auditory phenomena in modifying basic LPC, e.g., using a critical-band power spectrum with a logarithmic amplitude compression. The spectrum is multiplied by the equal-loudness curve and raised to the power 0.33 to simulate the power law of hearing [56, 57]. Seventeen critical-band (CB) filters equally-spaced in Bark z,

$$z = 6 \log\left(\frac{f}{600} + \sqrt{\left(\frac{f}{600}\right)^2 + 1}\right),$$

map the range 0–5 kHz into 0–17 Bark. Each CB is simulated by a spectral weighting,

$$c_k(z) = \begin{cases} 10^{z-y_k} & \text{for } z \le y_k, \\ 1 & \text{for } y_k < z < y_k + 1, \\ 10^{-2.5(z-y_k-1)} & \text{for } z \ge y_k + 1, \end{cases}$$

where z_k are the center frequencies (roughly, $1, 2, \ldots 17$ Bark) and $y_k = z_k - 0.5$ (the zeroth filter is arbitrarily set equal to the first filter). The 10 dB/Bark roll-off for low frequencies and −25 dB/Bark roll-off for high frequencies matches typical CB filters. A fifth-order PLP can suppress speaker-dependent aspects of the speech spectrum, leading to improved speech recognition.

A related auditory-based analysis method called *Ensemble-Interval Histograms* (EIH) models synchrony phenomena with 85 cochlear filters equally spaced in log-frequency in the 200–3200 Hz range. It has seen some success in speech recognition. Another similar technique is the *correlogram*, which shows a series of short-time autocorrelations of auditory-neuron firing rates [58]. The ERB-scale (equivalent rectangular bandwidth) is yet another frequency scale of practical relevance, motivated by auditory phenomena [59].

In noisy conditions, doing LP analysis on part of the autocorrelation vector (rather than on the speech itself) has been shown to yield more robust parameters. One version of this method [60] models the magnitude spectrum of a one-sided (i.e., causal) autocorrelation (this involves the Hilbert transform); the benefit comes from an enhancement of peaks in the spectrum, at the expense of noise-corrupted valleys. Basic LPC models peaks rather than valleys, due to the use of the mean square error as a criterion; this method raises the spectrum to the second power, emphasizing the peaks even more. It and a related method (short-time modified coherence) [61] require more computation than basic LPC.

6.5.11 Emphasizing Low Frequencies

Standard LPC weighs all frequencies in the speech spectrum equally, although lower frequencies are better resolved by the ear and are more important for speech intelligibility than are higher frequencies. LPC modeling could be improved by combining subband coding (see Chapter 7) with LPC, which allows LPC analyses of different orders to model different frequency ranges according to their perceptual importance. *Selective linear prediction* [25] models the F1–F3 frequency range with the standard 2 poles/kHz (plus 2–3 poles for general shaping) and relies on only a few poles for the higher frequencies, where formant structure is of less importance (Figure 6.17). The filtering problems of subband coding, however, reduce the advantages of this approach.

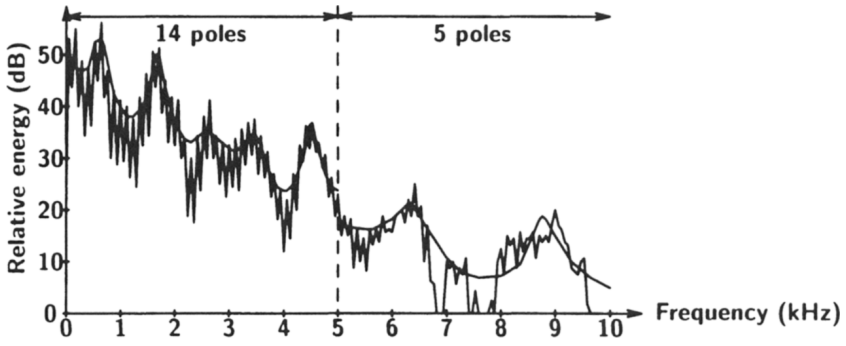

Figure 6.17 Speech spectrum (ragged line) and LPC spectrum (smooth line), corresponding to a 14-pole LPC analysis in the 0–5 kHz region and a 5-pole analysis in the 5–10 kHz region. (After Makhoul [25] © IEEE.)

Other ways to emphasize low frequencies in LPC analysis include modifying the error function or warping the frequency axis (as in PLP) following the perceptually based mel or Bark scale. Minimizing the standard LPC error treats all frequencies equally; attempts to utilize a frequency-weighted error without substantially increasing computation have been partially successful [54, 62]. If frequency is warped with an all-pass transformation (a standard procedure in designing digital filters [6]), DFTs and inverse DFTs are not needed, thus reducing additional computation [63]. Such frequency-warping, however, appears to yield improved speech quality only for very-low-order LPC vocoders (e.g., $p < 8$ for 4.8 kHz bandwidth speech). In standard LPC analysis, too low an order causes perceptually important lower formants to be inadequately modeled; but in frequency-warping LPC, the low-frequency range takes on increased significance in proportion to the degree of the warping. Frequency warping appears useful only where bit rate constraints impose a low order on the LPC model.

In some applications, the behavior of the LPC inverse filter $A(e^{j\omega})$ is of concern at high frequencies, where very high gain is possible if most of the energy near $\omega = \pi$ has been eliminated during the lowpass filtering of the original analog speech (prior to A/D conversion). In some coders, a predictor $P(e^{j\omega}) = 1 - A(e^{j\omega})$ is usually placed in a feedback loop around a quantizer, which filters coarsely quantized speech samples. The quantization adds broadband noise (including energy around $\omega = \pi$) to the input of the predictor filter, which yields an output with unwarranted large gains at high frequencies. One solution is to add a small amount of highpass noise to the digitized speech for input to the LPC analysis (ideally the noise spectrum should be the complement of the A/D lowpass filter spectrum) [49]. The noise is used only in determining $A(z)$, not in the actual waveform coding.

6.5.12 Pole-Zero LPC Models

Almost all applications of LPC use the all-pole (AR) model. By not modeling zeros directly, the analysis equations (e.g., Equation (6.24)) are linear and have symmetries that reduce computation. Pole-zero (ARMA) models require solution of nonlinear equations to obtain the optimal set of parameters [64]. Since the quality of ARMA speech is only slightly better than AR speech [65], pole-zero modeling is rarely used for coding. ARMA has however shown good results in formant and voicing estimation [66], and can handle lossy

vocal tract models [67]. It has also been suggested for processing noisy speech [68], since the all-pole AR model is less valid in noise. Faster special-purpose hardware may increase use of ARMA in the future [69].

Approaches to ARMA modeling may trade off strict optimality in spectral representation and computation time. For instance, one method involves a two-step procedure that locates the poles first by a standard AR technique and then models the spectral *inverse* of the residual signal with a second AR model [70]. The residual after the first AR model presumably contains the effects of the zeros; thus, inverting its spectrum provides the input for a second all-pole modeling. Solving for the poles and zeros sequentially is efficient but does not guarantee that the pole-zero locations obtained would be those of a simultaneous, optimal solution.

A major difficulty in ARMA modeling is determining the order of the model, i.e., how many poles and zeros to use. A poor choice of model order leads to inaccurate estimation of both poles and zeros [71]. One study of 4.8 kHz male speech suggests using ten poles for voiced sounds and only five poles for unvoiced consonants, plus three zeros for the latter and five zeros for nasals [72]. The magnitude spectrum of any type of speech with 4 kHz bandwidth can be very accurately modeled with a *high-order* all-pole model, e.g., 40–50 poles [73]. In such a model, strong harmonics are directly approximated by pairs of poles. One may decompose such a high-order model $C(z)$ into low-order polynomials:

$$C(z) = \sum_{k=1}^{r} c_k z^{-k} = \frac{Q(z)}{P(z)}, \tag{6.52}$$

where $Q(z)$ represents the zeros and $P(z)$ represents the poles, both with order much less than $r \approx 45$. Assuming p poles and q zeros, the problem reduces to solving $p + q$ linear equations involving the high-order LPC coefficients c_k, with a cross-correlation between the high-order residual signal and the original speech, and autocorrelations of the speech and the error signals [73]. Requiring 4–7 times as much computation as AR analysis, this ARMA method yields accurate results.

6.6 CEPSTRAL ANALYSIS

In speech analysis we usually estimate parameters of an assumed speech-production model. The most common model views speech as the output of a linear, time-varying system (the vocal tract) excited by either quasi-periodic pulses or random noise. Since the easily observable speech signal is the result of convolving excitation with vocal tract sample response, it would be useful to separate or "deconvolve" the two components. While unfeasible in general, such deconvolution works for speech because the convolved signals have very different spectra.

One step in *cepstral* deconvolution transforms a product of two spectra into a sum of two signals. If the resulting summed signals are sufficiently different spectrally, they may be separated by linear filtering. The desired transformation is logarithmic, in which $\log(EV) = \log(E) + \log(V)$, where E is the Fourier transform of the excitation waveform and V is the vocal tract response. Since the formant structure of V varies slowly in frequency compared to the harmonics or noise in E, contributions due to E and V can be linearly separated after an inverse Fourier transform.

6.6.1 Mathematical Details of Cepstral Analysis

Consider a simple signal $x(n) = a^n u(n)$ and its z transform $X(z) = 1/(1 - az^{-1})$, with a pole at $z = a$ and a zero at $z = 0$. For $\log(X(z))$ in a power series,

$$\log(X(z)) = \sum_{n=1}^{\infty} \frac{a^n}{n} z^{-n}, \qquad \text{if } |z| > |a|. \tag{6.53}$$

(A similar expansion holds for $n < 0$, if the region of convergence is $|z| < |a|$.) The *complex cepstrum* $\hat{x}(n)$ (the circumflex notation is often used to denote cepstra) is the inverse transform of $\log(X(z))$. In this example,

$$\hat{x}(n) = -\frac{a^n}{n} u(n-1) \tag{6.54}$$

by simple inverse z transform, term by term. Thus $x(n)$ retains its exponential form in $\hat{x}(n)$, except for a more rapid decay due to the $1/n$ factor.

Because the logarithm of a product equals the sum of the individual log terms, a more complicated z transform consisting of several first-order poles and zeros results in a complex cepstrum that is the sum of exponential terms, each decaying with the extra $1/n$ factor. Since $\log(1/A) = -\log(A)$, the only difference between the effect of a pole and that of a zero in the complex cepstrum is the sign of the power series. The equal treatment of poles and zeros is an advantage for cepstral modeling (vs all-pole LPC).

For a more general $X(z)$ (e.g., speech) that converges on the unit circle, poles p_k and zeros z_k inside the unit circle contribute linear combinations of p_k^n/n and $-z_k^n/n$, for $n > 0$, while corresponding poles a_k and zeros b_k outside the unit circle contribute summed terms of the form $-p_k^n/n$ and z_k^n/n, for $n < 0$. The complex cepstrum is of infinite extent, even if $x(n)$ has finite duration. However, given a stable, infinite-duration $x(n)$, $\hat{x}(n)$ decays more rapidly in time than the original $x(n)$:

$$|\hat{x}(n)| < \alpha \frac{\beta^{|n|}}{|n|}, \qquad \text{for } |n| \to \infty, \tag{6.55}$$

where α is a constant and β is the maximum absolute value among all $p_k, z_k, 1/a_k$, and $1/b_k$ (which corresponds to the closest pole or zero to the unit circle).

For speech, the closest pole involves F1, which has relatively narrow bandwidth and dominates the rate of amplitude decay in most pitch periods. While many periods in speech $s(n)$ have time constants about 10–30 ms, the $1/n$ factor in Eq. (6.54) causes $\hat{s}(n)$ to decay rapidly within a few milliseconds of $n = 0$. This is in distinct contrast to the excitation component $e(n)$ of voiced speech $s(n)$, which may be viewed as the convolution of a sample train $e(n)$ (where N = pitch period) and the vocal tract response $v(n)$ (including glottal effects). Using Fourier transforms (for convergence when using impulse signals), recall that $E(e^{j\omega})$ is a uniform train of impulses with frequency spacing of $2\pi/N$. Taking the logarithm of the Fourier transform affects only the areas of the impulses, not their spacing. Thus the complex cepstrum $\hat{e}(n)$ retains the same form as $e(n)$, i.e., a sample train of period N. Since $S(z) = E(z)V(z)$, $\log(X(z)) = \log(E(z)) + \log(V(z))$ and $\hat{s}(n) = \hat{e}(n) + \hat{v}(n)$. With $\hat{v}(n)$ decaying to near zero over its first few milliseconds and $\hat{e}(n)$ being nonzero only at $n = 0, \pm N, \pm 2N, \pm 3N, \ldots$, the two functions are easily separated via a rectangular window. A suitable boundary for that window would be the shortest possible pitch period, e.g., 3–4 ms.

The duration and shape of the speech analysis window $w(n)$ can have a significant effect on the cepstrum. The simple discussion above must be modified, since the signal under analysis is

$$x(n) = s(n)w(n) = [e(n) * v(n)]w(n). \tag{6.56}$$

Much research has assumed that Equation (6.56) can be approximated by

$$x(n) \approx [e(n)w(n)] * v(n), \tag{6.57}$$

which is valid for impulsive $e(n)$ (as in voiced speech) only if

$$w(n) \approx w(n + M), \tag{6.58}$$

where M is the effective duration of $v(n)$. Unfortunately, typical cepstral analysis windows tend to violate Equation (6.58). As a result, the vocal tract contribution to the cepstrum is repeated every pitch period and is subject to a double sinc-like distortion [74]. Applications using the windowed cepstrum should compensate for this distortion and should employ a cepstral window no larger than half the expected pitch period to avoid aliasing.

6.6.2 Applications for the Cepstrum

Applications for cepstral analysis occur in speech vocoders, spectral displays, formant tracking (Figure 6.18), and F0 detection [75]. Samples of $\hat{x}(n)$ in its first 3 ms describe $v(n)$ and can be coded separately from the excitation. The later is viewed as voiced if $\hat{x}(n)$ exhibits sharp pulses spaced at intervals typical of pitch periods, and the interval is then deemed to be 1/F0. If no such structure is visible in $\hat{x}(n)$, the speech is considered unvoiced. The Fourier transform of $\hat{v}(n)$ provides a "cepstrally smoothed" spectrum, without the interfering effects of $e(n)$ (see Figure 6.18).

In practice, the complex cepstrum is not needed; the real cepstrum suffices, defined as the inverse transform of the logarithm of the speech magnitude spectrum:

$$c(n) = \frac{1}{2\pi}\int_{\omega=0}^{2\pi} \log|X(e^{j\omega})|e^{j\omega n}\, d\omega. \tag{6.59}$$

For real signals $x(n)$, $c(n)$ is the even part of $\hat{x}(n)$ because

$$\hat{X}(e^{j\omega}) = \log(X(e^{j\omega})) = \log|X(e^{j\omega})| + j\arg[X(e^{j\omega})] \tag{6.60}$$

and the magnitude is real and even, while the phase is imaginary and odd. In cepstral speech coding, as in other coding techniques (see Chapter 7), the phase may be discarded for economy, at the risk of some degradation in output speech quality.

To render the cepstrum suitable for digital algorithms, the DFT must be used in place of the general Fourier transform in Equation (6.59):

$$c_d(n) = \frac{1}{N}\sum_{k=0}^{N-1} \log|X(k)|e^{j2\pi kn/N} \qquad \text{for } n = 0, 1, \ldots, N-1. \tag{6.61}$$

Replacing $X(e^{j\omega})$ with $X(k)$ is equivalent to sampling the Fourier transform (multiplication by an impulse train) at N equally spaced frequencies from $\omega = 0$ to 2π. Including the inverse DFT, the net effect is to convolve the original $c(n)$ with a uniform sample train of period N:

$$c_d(n) = \sum_{i=-\infty}^{\infty} c(n + iN). \tag{6.62}$$

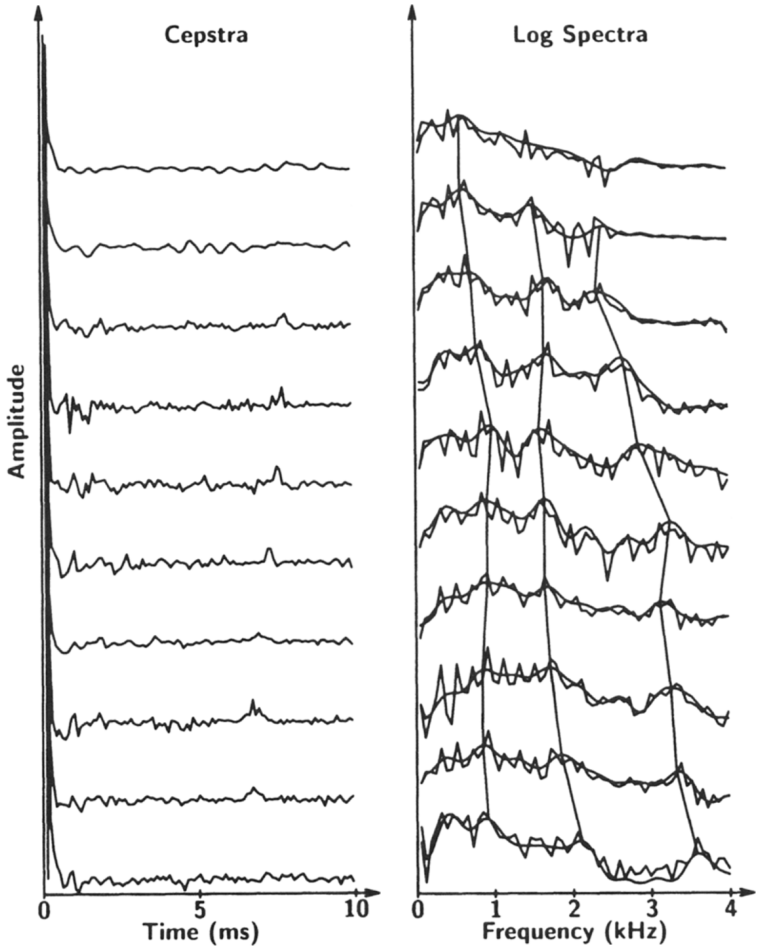

Figure 6.18 Automatic formant estimation from cepstrally smoothed log spectra. (After Schafer and Rabiner [9].)

Thus, the "digital" version $c_d(n)$ of the cepstrum contains copies of $c(n)$, at intervals of N samples. The resulting aliasing is not important if N, the duration of the DFT analysis window, is large enough. N is usually a few hundred samples, which more than suffices to eliminate any aliasing problem in the $\hat{v}(n)$ part of the cepstrum. The $\hat{e}(n)$ components for voiced speech extend into the high time range of $c(n)$, though, and may cause aliasing problems for $c_d(n)$. Typically, however the analysis frame contains a few pitch periods that are sufficiently nonidentical to cause the impulses in $\hat{e}(n)$ to be of lower amplitude as n increases. This minimizes the interference of aliased copies of $c(n)$ on the copy of interest near $n = 0$.

The cepstrum has not been popular for speech coding due to its computational complexity. Two DFTs and a logarithm operation are needed to obtain $c(n)$, which is windowed to separate $\hat{v}(n)$ and $\hat{e}(n)$. Then inverse operations (two more DFTs and an exponential) reconstruct $v(n)$, which is convolved at the synthesis stage with a synthetic $e(n)$ to generate output speech. In addition, good-quality speech has required coding up to 3 ms of

$c(n)$, which involves more stored samples/s of speech than other coding approaches, for comparable quality.

Cepstral analysis is more practical for F0 or formant estimation (especially in speech recognition) since response reconstruction is unnecessary. A further application for cepstra is the elimination of fixed-time echos in speech signals, e.g., in the telephone network. An echo in a speech signal $x(n)$ can be modeled as convolution with $\delta(n) + A\delta(n - N)$, where A is the percentage of echo and N is the number of samples in the echo delay. The effect is similar to that of periodic $e(n)$ samples exciting $v(n)$, with the echo introducing into $c(n)$ a set of impulses, decaying in amplitude with time n and spaced at intervals of N samples. A comb filter, with notches located at multiples of N, can eliminate the echo effects in $c(n)$, which can then be converted back into speech, much as in the manner of a cepstral speech coder. This procedure works best when the echo delay N is fixed (so that the comb filter need not be dynamic) and when N is outside the range where $c(n)$ would be significantly nonzero; e.g., an echo at a pitch period interval would result in whispered output speech after comb filtering.

Several modifications to basic cepstral analysis have been suggested. One claims to overcome a tendency for the cepstrum to overestimate formant bandwidths, and accounts for some auditory traits in a way similar to PLP [76]. "Root cepstral analysis" also has some potential advantages [77]. Lastly, an LPC spectrum is often used in Equation (6.61) instead of a DFT, to eliminate F0 effects.

6.6.3 Mel-Scale Cepstrum

The most popular analysis method for automatic speech recognition uses the cepstrum, with a nonlinear frequency axis following the Bark or mel scale. Such *mel-frequency cepstral coefficients* c_n (MFCCs) provide an alternative representation for speech spectra which incorporates some aspects of audition. An LPC or DFT magnitude spectrum S of each speech frame is frequency-warped (to follow the bark or critical-band scale) and amplitude-warped (logarithmic scale), before the first 8–14 coefficients c_n of an inverse DFT are calculated. A common approach [78] simulates critical-band filtering with a set of 20 triangular windows (Figure 6.19), whose log-energy outputs are designated X_k; if M cepstral coefficients are desired, they are

$$c_n = \sum_{k=1}^{20} X_k \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{20}\right] \qquad \text{for } n = 1, 2, \ldots, M. \tag{6.63}$$

These windows are sometimes called filters, but they simply weight spectral $S(i)$ values across a frequency index i (i.e., they do not filter time signals).

The initial c_0 coefficient represents the average energy in the speech frame and is often discarded (amplitude normalization); c_1 reflects the energy balance between low and high frequencies, positive values indicating sonorants and negative values for frication. (This is due to the cosine weighting in the final IDFT of the cepstral calculation: for c_1, the one-period cosine weights the lower half of the log spectrum positively and the upper half negatively.) For $i > 1$, c_i represent increasingly fine spectral detail (as the cosine with i periods weights shorter frequency ranges (corresponding to $0.25F_s/i$ Hz) alternately positively and negatively). As with LPC a_k, no simple relationship exists between c_i and formants; e.g., in speech with four formants, a high c_2 suggests high energy in F1 and F3 and low energy in F2 and F4, but such a relationship is only approximate when the formants deviate from their average positions.

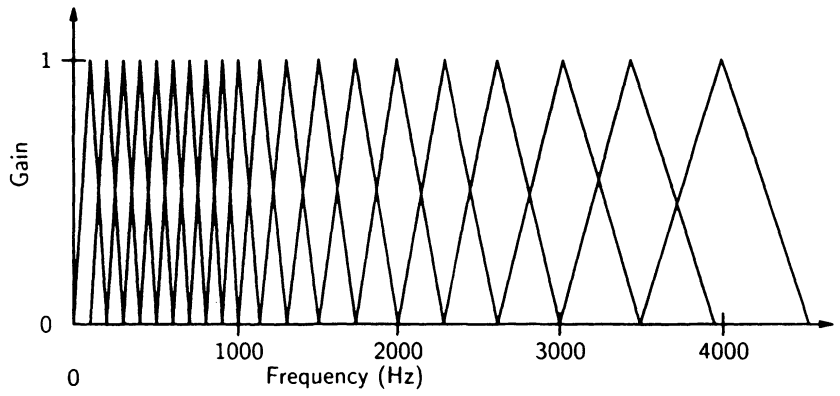

Figure 6.19 Filter bank for generating mel-based cepstral coefficients. (After Davis and Mermelstein [78] © IEEE.)

6.7 OTHER SPECTRAL ESTIMATION METHODS (‡)

Since the spectral distribution of speech energy as a function of both time and frequency is widely considered to be the most important factor in speech production and perception, analysis applications search for efficient time–frequency representations (TFRs). The Fourier Transform (FT) and related spectral measures, such as LPC and the cepstrum, are by far the most common TFRs. This is due to their mathematical simplicity, ease of computation, and easy interpretation. Spectrographic displays show clearly where energy is concentrated in two-dimensional time–frequency plots. However, the use of a fixed window for each display leads to an obligatory compromise for resolution between time and frequency (e.g., good time and poor frequency resolution in wideband spectrograms). Some alternative TFRs have been explored for speech analysis to avoid this compromise, at the cost of some loss of ease of interpretation.

6.7.1 Karhunen–Loeve Transform (KLT)

The objective of speech analysis is usually to reduce the dimensionality of an input signal vector, while retaining the pertinent information of the vector for applications such as coding or recognition. Relatively simple transformations such as the FT or LPC are most common. However, as computers increase in capacity, more complex algorithms become feasible. The KLT has the advantage of being optimal in compressing a vector, but is expensive in computation. For KLT, an N-dimensional speech vector $\mathbf{X}$ is converted to a set of N eigenvectors ϕ_j and N corresponding weights (eigenvalues) λ_j. The λ_j can be rank-ordered in terms of the relative energy (and thus perceptual importance) of each ϕ_j (the ϕ_j act as basis functions). The FT is a special case of the KLT, where ϕ_j are harmonically related sinusoids (where F0 corresponds to the frame length). The more general KLT allows different ϕ_j for each frame of data, which usually yields a much more efficient choice of basis functions. As a result, the speech frame can be very compactly represented by a few eigenvectors. As an example, suppose the input vector is one pitch period of a voiced speech signal; the KLT would likely select damped sinusoids (corresponding to the formants) as eigenvectors, and the eigenvalues would decrease in value with frequency (that for F1 being

highest), following the usual spectral tilt of voiced speech. Since formants and F0 are so dynamic, the simpler FT uses undamped harmonically related sinusoids usually with no correlation to F0; as such, the spectral information is widely spread throughout the Fourier coefficients. The KLT, while costly in determining a new set of ϕ_j for each frame, has a much more compact representation. Due to its need for much calculation, it has not had wide application in speech processing [79].

6.7.2 Wavelets

The logarithmic scale seems to play an important role in speech production and especially in perception. The decibel, mel, and semitone scales correlate better with perception than do linear scales for energy and frequency. The ear's resolution decreases as energy and frequency increase. The decibel scale easily handles the nonlinearity for energy, but most spectral displays retain a fixed analysis bandwidth for simplicity. The *wavelet transform* (WT) replaces the fixed bandwidth of the FT with one proportional to frequency (i.e., constant Q), which allows better time resolution at high frequencies than the FT (especially for brief sounds). The resulting loss of frequency resolution as frequency increases is acceptable in most applications. The discrete WT for a speech signal $s(n)$ is

$$WT_n(k) = \sum_{m=-\infty}^{\infty} s(m)\gamma(n, m, k). \tag{6.64}$$

The WT simply replaces the frequency-shifted lowpass filter $e^{-j2\pi km/N}w(n-m)$ of the FT with a wavelet $\gamma(n, m, k)$. Both the FT and WT preserve time shifts (i.e., a delayed speech signal simply delays the spectral representation), but the WT replaces the FT's preservation of frequency shifts with one of time scaling instead. The most common wavelet has the form $\gamma(n, m, k) = \gamma((m-n)k)$; $\gamma(n)$ is usually a bandpass function (often Gaussian-shaped) centered in time around $n = 0$. Since the WT suffers the same basic problem of trading time and frequency as does the FT, its application to speech has been limited [80–82] (see however a recent coder [83]).

6.7.3 Wigner Distribution

While the linear properties of the FT and WT are very useful, their inherent time–frequency trade-off is often a problem. The choice of the length of the FT fixes a constant time and frequency resolution; the WT gives good time resolution at high frequencies and good frequency resolution at low frequencies (which may correspond better to human perception). The basic trade-off nonetheless remains. A class of quadratic TFRs, of which the *Wigner* (or *Wigner–Ville*) *distribution* (WD) is prominent, is able to show fine resolution in time and frequency simultaneously, but at the cost of significant interference terms (ITs) in the representation, as well as negative values, which hinder their interpretation. Linear TFRs (e.g., the FT) have the advantage that the distortion of ITs for multicomponent signals (e.g., speech, with its many harmonics) is limited to the time and frequency ranges where the components overlap. Energy in spectrograms is smeared in time or frequency over a range depending on the analysis bandwidth, but the smearing is local and often easily tolerated in visual or algorithmic interpretation. In quadratic TFRs, ITs go beyond simple smearing to appear at distant time and frequency locations (typically at points corresponding to averages in time or frequency). With *a priori* knowledge about the nature of the components in an input signal (e.g., harmonics), distant ITs may sometimes be attenuated, but in general the

interpretation and use of quadratic TFRs are more complicated than for the common linear TFRs.

Given the importance of energy in speech analysis, a quadratic TFR is desirable, but the power or energy spectrum contains cross-terms that do not occur in the linear FT. For a signal with N components $s(n) = \sum_{i=1}^{N} s_i(n)$, the power spectrum has N desired terms (the FT of each $s_i^2(n)$) and $(N^2 - N)/2$ (undesired) cross-terms (the FT of $2s_i(n)s_j(n)$ for each $i \neq j$). The large number of cross-ITs hinders use of quadratic TFRs. There is a general compromise between good time–frequency concentration (e.g., in the Wigner distribution) and small ITs (e.g., in the FT). The WD has many desirable mathematical properties [81], e.g., it is real-valued, preserves time and frequency shifts, and can be viewed as a two-dimensional display of energy over the time–frequency plane:

$$W(t,f) = \int_{\tau} s(t + \tau/2)s^*(t - \tau/2)\exp(-j2\pi ft)d\tau = \int_{v} S(f + v/2)S^*(f - v/2)\exp(j2\pi vt)dv. \tag{6.65}$$

The ITs of the WD are oscillatory in nature, and thus can be attenuated by smoothing. A smoothed WD trades decreased ITs for some broadening of time–frequency concentration; the Choi–Williams distribution is a popular version. It is doubtful that wavelets or quadratic TFRs will displace the FT [13, 84], which remains the primary basis of speech analysis. A recent application to speech (*minimum cross-entropy time–frequency distribution*) shows promise, but at high cost [85].

6.7.4 Other Recent Techniques

The search for better analysis methods has led to nonlinear techniques [86], which sacrifice simplicity and efficient calculation to obtain more compact or useful representations of speech. One motivation for nonlinear analysis is that, as a random process, speech is not Gaussian, and thus has non-zero third-order (and higher) moments (Gaussians are fully described by their mean and variance). Relatively little speech research has explicitly exploited third- and higher-order statistics ("cumulants") [87]. Linear models cannot handle higher-order statistics of this sort. In particular, apparently random aspects of speech (e.g., in the residuals of LPC) may be due to *chaos* (and hence predictable with nonlinear models) [88]. In most cases, the added cost of nonlinear methods and the small modeling gains (e.g., 2–4 dB in SNR [88]) have limited their application (e.g., nonlinear prediction is much less used than LPC [89]). Recent analysis methods from other domains (e.g., fractals [90]) often do not apply easily to speech, although a recent coder reports good results [91].

6.8 F0 ("PITCH") ESTIMATION

Determining the fundamental frequency (F0) or "pitch" of a signal is important in many speech applications. (Although pitch is perceptual, and what is being measured is actually F0, the estimators are commonly called "pitch detectors.") In voiced speech the vocal cords vibrate; "pitch" refers to the percept of the fundamental frequency of such vibration or the resulting periodicity in the speech signal. It is the primary acoustic cue to intonation and stress in speech, and is crucial to phoneme identification in tone languages. Most low-rate voice coders requires accurate F0 estimation for good reconstructed speech, and some medium-rate coders use F0 to reduce transmission rate while preserving high-quality speech.

F0 patterns are useful in speaker recognition and synthesis (in the latter, natural intonations must be simulated by rule). Real-time F0 displays can also give feedback to the deaf learning to speak.

F0 determination is fairly simple for most speech, but complete accuracy has eluded the many published algorithms, owing to speech's nonstationary nature, irregularities in vocal cord vibration, the wide range of possible F0 values, interaction of F0 with vocal tract shape, and degraded speech in noisy environments [26, 92, 93]. Instrumental methods can estimate F0 using information other than the speech signal, e.g., by measuring the impedance of the larynx as the vocal cords open and close through the use of contact microphones or accelerometers attached to the body, or via ultrasound or actual photography of the vocal cords. Most F0 detectors, however, are algorithms using only the speech signal as input. They often yield a *voicing decision* as part of the process, in which up to four classes of speech can be distinguished; voiced, unvoiced, combined (e.g., /z/), and nonspeech (silence, or background noise). Unlike F0 estimation, voicing determination (involving discrete categories) appears well suited to pattern recognition techniques [94, 95]. Voicing estimates can be accurate to about 95% if SNR exceeds 10 dB, but fail for SNR below 0 dB [96]. While voicing decisions are often a by-product of F0 estimators, better accuracy can result with separate algorithms.

F0 can be determined either from periodicity in time or from regularly spaced harmonics in frequency. Time-domain F0 estimators have three components: a preprocessor (to filter and simplify the signal via data reduction), a basic F0 extractor (to locate pitch epochs in the waveform), and a postprocessor (to correct errors). The algorithms try to locate one or more of the following aspects in the speech signal: the fundamental harmonic, a quasi-periodic time structure, an alternation of high and low amplitudes, or points of discontinuities. Harmonics and periodicities usually provide good results but fail in certain instances. The F0 algorithms trade complexity in one component for that in another; e.g., harmonic extraction requires a complex filter as preprocessor but allows an elementary basic extractor that may simply count zero-crossings of the filtered speech. Nonzero thresholds and hysteresis are used in postprocessing to eliminate irrelevant zero-crossings. The preprocessor is often a simple lowpass filter, but problems in choosing its cutoff frequency arise due to the large range of possible F0 values from different speakers.

Frequency-domain methods for F0 estimation involve correlation, maximum likelihood, and other spectral techniques where speech is examined over a short-term window. Autocorrelation, average magnitude difference, cepstrum, spectral compression, and harmonic matching methods are among the varied spectral approaches [92]. They generally have higher accuracy than time-domain methods, but need more computation.

Real-time F0 estimators must produce values with little delay. Since most frequency-domain methods require a buffer of speech samples prior to the spectral transformation, they are not as fast as those operating directly on the time waveform. Some F0 detectors can be modified for speed, but lose the timing of pitch periods; e.g., periodicity (and the duration of the period) can be evaluated more quickly than finding the actual locations of periods. Such F0 estimators do not output period times (useful for segmentation purposes) but yield period durations suitable for applications such as voice coders.

6.8.1 Time-Domain F0 Techniques

F0 estimation seems simple; humans, especially trained phoenticians, can easily segment most speech into successive pitch periods. Since the major excitation of the vocal

tract for a pitch period occurs when the vocal cords close, each period tends to start with high amplitude (referred to as an *epoch*) and then to follow a decaying-amplitude envelope. Since voiced speech is dominated by first-formant energy, the rate of decay is usually inversely proportional to the F1 bandwidth. Except when speech has short periods or a narrow F1, sufficient decay allows epoch location by simple peak-picking, with some basic constraints on how long periods may be. If speakers may range from an infant or soprano singer to a deep baritone, possible pitch periods extend from less than 2 ms to more than 20 ms, i.e., a range >18 ms, although typical ranges are smaller, e.g., about 6 ms for adult males. The rate of F0 change is limited; in a voiced section of speech, F0 usually changes slowly with time, rarely by more than an octave over 100 ms. Before applying such continuity constraints, one must find reliable pitch periods within each voiced section since F0 can change greatly during unvoiced speech (i.e., the resumption of voicing after a silence or obstruent can have F0 very different from that at the end of the previous voiced section).

Most F0 estimation difficulties occur at voiced–unvoiced boundaries, where continuity constraints are less useful and where pitch periods are often irregular. Other problems are due to sudden amplitude and formant changes that may occur at phone boundaries. To aid peak-picking and other methods, the input speech is normally lowpass-filtered in a preprocessing stage to retain only F1 (e.g., the 0–900 Hz range). This removes the influence of other formants (which confound F0 estimation) while still retaining enough strong harmonics to yield a "cleaner" signal for peak-picking. One approach chooses candidates for epochs with a variable-amplitude threshold: since all periods exceed 2 ms, the threshold remains high for 2 ms after each estimated epoch, ignoring all signal excursions right after the start of a period, and then the threshold decays exponentially at a rate typical of pitch periods [97].

A more direct approach filters out all speech energy except the fundamental harmonic and then detects zero crossings (which occur twice every period for a sinusoid such as the fundamental). A major difficulty is determining the cutoff for the lowpass filter: high enough to allow one harmonic from a high-F0 voice yet low enough to reject the second harmonic of a low-F0 voice. Secondly, many applications use bandpass-filtered speech (e.g., telephone speech, which eliminates the 0–300 Hz range), and the fundamental harmonic is often not present. One solution to this latter problem is to reconstruct the fundamental from higher harmonics via a nonlinear distortion, e.g., passing speech through a rectifier, which generates energy at all harmonics.

F0 estimation in the time domain has two advantages: efficient calculation, and specification of times for the pitch epochs. The latter is useful when pitch periods must be manipulated (e.g., for pitch synchronous analysis, or to reconstruct the glottal waveform) [27]. F0 values alone suffice for many analysis applications, e.g., vocoders. However, systems that vary speaking rate (speeding or slowing, depending on preferred listening rates) often delete or duplicate pitch periods, splicing at epoch times to minimize waveform discontinuities. Knowledge of period locations is crucial here, as well as for types of speech synthesis and coding which concatenate periods. Spectral F0 estimators do not provide such information, but normally yield more reliable F0 estimates.

6.8.2 Short-Time Spectral Techniques

The second class of F0 estimators operates on a block (short-time frame) of speech samples, transforming them spectrally to enhance the periodicity information in the signal. Periodicity appears as peaks in the spectrum at the fundamental and its harmonics. While peaks in the time signal are often due to formant (especially F1) interaction with the glottal

excitation, spectral peaks are usually easier to relate to F0. In these systems, one can view the spectral transformation as a preprocessor and a spectral peak detector as the basic F0 estimator; a postprocessor then examines estimates from successive frames to correct obvious errors. These errors could be major, e.g., F0 doubling or halving, which result from confusing F0 with its first harmonic. Doubling tends to occur when the energy level in the fundamental is weak compared to adjacent harmonics. (F0 halving is more common in time-domain methods, when two periods are mistaken as one.) Since F0 cannot physically change one octave during a frame (typically 10–30 ms), a postprocessor applies continuity constraints to smooth any estimates out of line with the neighboring F0 contour. *Fine* F0 errors of a few hertz are more difficult to deal with than *coarse*, major errors (e.g., doubling) and tend to arise when analysis frames are too short (not containing enough information to specify F0 accurately) or too long (if F0 changes rapidly within the frame). Since many systems evaluate F0 independently for each frame, fairly simple postprocessing can often significantly improve performance [26].

These examples illustrate the tradeoffs in choosing frame length. As in other windowing applications, the best speech parameters are obtained if the signal is stationary during the frame. Thus the frame must be short, a few pitch periods at most, since F0 may change rapidly. Abrupt spectral changes at phone boundaries can also affect spectral F0 estimators. The frame must nonetheless contain at least two periods to provide periodicity information. The precision of F0 measurement is proportional to the number of samples in the analysis frame; thus short frames inherently are more vulnerable to fine pitch errors. The single F0 estimate from each analyzed frame provides an average F0 value for that frame.

One complication in F0 estimation is caused by phase distortion (found in many transmission media, e.g., telephony) and by phase differences among harmonics. Since speech spectral phase has a shift of 180° near each formant, the harmonics in the 200–900 Hz range of F1 have phase differences that complicate the waveform and can obscure periodicity for time-domain F0 estimators. One solution is to eliminate phase effects by peak-picking, not directly on the filtered speech signal, but on its short-time autocorrelation $\phi(k)$ [98]. Recall that $\phi(k)$ is the inverse Fourier transform of the energy spectrum (i.e., $|X(e^{j\omega})|^2$) and thus sets the phase of each squared harmonic to zero. Although a time-domain signal, $\phi(k)$ cannot locate pitch epochs because of phase loss in the short-time analysis.

Since F0 estimation, and not faithful reproduction of the power spectrum, is the objective here, the speech signal is often distorted during preprocessing before autocorrelation to reduce calculation and to enhance periodicity parameters. *Center clipping* $s(n)$ (Figure 6.20), in which low-amplitude samples are set to zero and the magnitude of high-amplitude samples is reduced, is sometimes used to improve F0 estimation [99]. (Such clipping may, however, hurt F0 detection in noisy speech [100].) A variable clipping threshold, typically 30% of the maximum $|s(n)|$, must be used to adapt to different speech intensities. *Infinite peak clipping*, which reduces $s(n)$ to a zero-crossing signal, also yields good F0 estimation through autocorrelation and significantly reduces calculation, since all multiplications involve only zeros and ones. As an alternative to clipping, the signal can be raised to a high power (while preserving the algebraic sign of each speech sample) in order to highlight peaks in $s(n)$.

Estimating F0 directly by trying to locate the fundamental spectral peak is often unreliable because the speech signal may have been bandpass filtered (e.g., in the telephone network) or the fundamental may have low energy if F1 is high. The harmonic structure (spectral peaks at multiples of F0) is a more reliable indicator of F0; the frequency of the greatest common divisor of the harmonics provides a good F0 estimate. Female speech, with its widely spaced harmonics, often yields more reliable F0 estimates than male speech (sometimes female speech is so dominated by one harmonic as to appear almost sinusoidal).

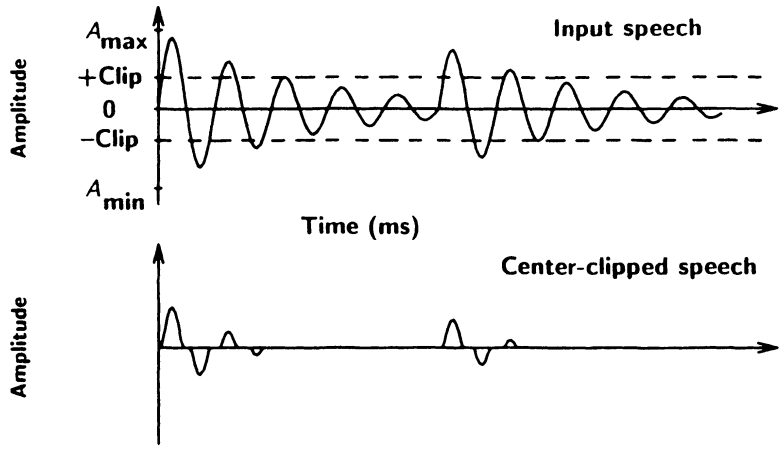

Figure 6.20 An example showing how center clipping affects two pitch periods. (After Sondhi [99] © IEEE.)

One approach measures the separation of adjacent harmonics; an alternative is to compress the spectrum by integer factors (i.e., compress the frequency scale by factors of two, three, four, etc.): a sum of these spectra has its strongest peak at F0 due to reinforcement of harmonics shifted down [26, 101].

Another variation is the *harmonic-sieve* F0 estimator. Rather than shift the speech spectrum, a spectral "sieve" with equally spaced holes is aligned with the spectrum; the frequency spacing at which the most harmonics line up with holes of the sieve is considered to be F0. One implementation [102] processes narrowband DFT spectra to simulate the ear's frequency and temporal resolution in identifying harmonics.

Maximum-likelihood methods provide another F0 estimator, which behaves especially well for noisy speech [103]. One way to determine the period of $s(n)$ in background noise is to add a delayed version, $s(n - D)$, to the original. When $D = 1/\text{F0}$, $s(n) + s(n - D)$ is strong, while the noise (out of phase due to the delay) tends to cancel. Finally, a recent F0 estimator with good results (especially for noisy speech) is based on auditory models [104].

Spectral F0 detectors give more accurate estimates than time-domain methods but require about 10 times more calculation due to the spectral transformation. The transformation focuses information about speech periodicity in ways that time-domain analysis cannot. Assuming that voicing determination is part of F0 detection, the performance of different systems can be rated objectively in terms of four types of errors: gross F0 errors (e.g., doubling), fine F0 errors, mistaking a voiced speech frame for unvoiced, and vice versa. No algorithm is superior in all four categories [93]. Alternatively, the detectors can be evaluated perceptually by using them in speech vocoders that represent excitation in terms of F0 and voicing decisions. No one type of objective F0 error correlates well with the subjective quality of coded speech, but voiced-to-unvoiced errors appear to be the most objectionable since they lead to harsh, noisy sounds where periodic sounds are expected [105]. While subjective and objective measures of F0 performance are not well correlated, there does not seem to be a large range of variation in coded speech quality using different major F0 algorithms. A good choice is probably the computationally simple AMDF (see Section 6.3), which ranks high in subjective tests, both for speech coders that crucially rely on voicing decisions and for those more concerned with F0 errors [100].

6.9 ROBUST ANALYSIS

A speech signal contains information from multiple sources: speaker, recording environment, and transmission channel. We are usually interested in extracting information about what is being said (for speech coding or recognition) or who is saying it (for speaker recognition). Most analysis methods, however, cannot easily distinguish the desired speaker signal from the unwanted effects of background noise, competing speakers, and the channel. Many analysis methods degrade as noise increases (e.g., LPC; F0 estimation [98]). Chapter 8 will examine ways to enhance speech signals and Chapter 10 will deal with recognition of noisy signals. Here we briefly discuss some analysis methods to suppress undesired components in speech signals.

Some of the "noise" in speech concerns variability on the speaker's part and must be handled on a stochastic basis, examining much "training" speech to obtain reliable models of speakers' voices (see Chapter 10). Distortions due to the communication channel and recording medium usually vary slowly compared to the dynamics of speech. Background noise (e.g., clicks, pops), on the other hand, often varies more rapidly than vocal tract movements. Suppressing spectral components of a speech signal that vary more quickly or slowing than the desired speech can improve the quality of speech analysis.

Often a mean spectrum or cepstrum is subtracted from that of each speech frame (e.g., blind deconvolution), to eliminate channel effects. The mean may require a long-term average for efficiency, which is difficult for real-time applications. Alternatively, the mean is estimated from a prior section of the input signal thought to be silent; this requires a speech detector and assumes that pauses occur regularly in the speech signal. If the channel changes with time, the mean must be updated periodically.

The RASTA (RelAtive SpecTrAl) method of speech processing has been successfully applied to enhancement and recognition. It bandpasses spectral parameter signals to eliminate steady or slowly varying components (including environmental effects and speaker characteristics) and rapid noise events. The bandpass range is typically 1–10 Hz, with a sharp zero at 0 Hz and a time constant of about 160 ms [57, 106]. Events changing more slowly than once a second (e.g., most channel effects) are thus eliminated by the highpass filtering. The lowpass cutoff is more gradual, smoothing parameter tracks over about 40 ms, to preserve most phonetic events, while suppressing impulse noise. When speech is degraded by convolutional noise, the J-RASTA method replaces the logarithm operation with $Y_i = \log(1 + JX_i)$, where i is a critical band index, J depends on the noise level, and X and Y are the input and output [106].

Another recent analysis method with application to speech recognition is the *dynamic cepstrum* [107], which does a two-dimensional (time–frequency) smoothing to incorporate a forward masking, enhance rapid formant transitions, and suppress slowly varying properties (e.g., channel effects; speaker-dependent global spectral shape). Thus it is similar to RASTA in emphasizing spectral change, but unlike RASTA also includes time–frequency interaction and does not completely eliminate static spectral components. There are other recent time–frequency analysis methods (with application to coding and recognition) [108, 109].

6.10 REDUCTION OF INFORMATION

In both coding and recognition applications, a major objective of speech analysis is to efficiently represent information in the signal while retaining parameters enough to recon-

struct or identify the speech. In coding we wish to reduce the storage or transmission rate of speech while maximizing the quality of reconstructed speech in terms of intelligibility, naturalness, and speaker identifiability. Thus an economical representation of the crucial aspects of the speech signal is paramount. In recognitioin systems, the storage question is secondary to recognition accuracy. Nonetheless, recognizers perform faster when the network information or stored templates occupy less memory. Furthermore, small, efficient templates often yield better results, e.g., templates of sampled speech waveforms require much storage but give much worse accuracy than spectral templates.

In analysis, eliminating redundant information in the speech signal is important. Whether information is superfluous depends on the application: speaker-dependent aspects of speech are clearly relevant for speaker identification, but those aspects are often superfluous for identification of the textual message in automatic speech recognition. It is not always clear which speech aspects can be sacrificed. Acceptable speech can be synthesized with rates under 600 bit/s (100 times lower than that for a simple digital representation). However, as bit rate is reduced, distortion is gradually introduced into the reconstructed speech; e.g., signal aspects relating to speaker identity tend to be lost at low storage rates. Synthesis models emphasize spectral and timing aspects of speech that preserve intelligibility, often at the expense of naturalness.

6.10.1 Taking Advantage of Gradual Vocal Tract Motion

Viewing speech as a sequence of phones linked via intonation patterns, most speech analysis attempts to extract parameters related to the spectral and timing patterns that distinguish individual phonemes. Speech is often transformed into a set of parameter signals that are closely related to movements of the vocal tract articulators. These signals may follow one particular articulator (e.g., the F0 "parameter" follows vocal cord vibration) or may result from several articulators acting together; e.g., the output from a bandpass filter or a DFT spectral sample relates to formant position and amplitude, which in turn are specified by the overall vocal tract configuration.

The vocal tract moves slowly compared to most speech sampling rates; e.g., typical phonetic events last more than 50 ms (although some, like stop bursts, are shorter), while speech may be sampled every 0.1 ms. Thus, speech parameters usually vary slowly and allow decimation; e.g., the short-time DFT (examined at a fixed frequency) is a time signal of bandwidth equal to that of the window used in the spectral analysis. Without loss of information, it may be decimated to a rate of twice the window's bandwidth; e.g., for wideband spectra, the 300 Hz window allows 600 samples/s. Since window bandwidth is not strictly limited in practical applications, small amounts of distortion are introduced in the decimation.

For the vast majority of speech samples, events are slowly varying. Rapid spectral changes are limited to stop onsets and releases or to phone boundaries involving a change in manner of articulation (i.e., when speech switches among the classes of fricatives, nasals, sonorants, and silence). In terms of the total number of analysis frames at a typical 10 ms update rate, only a very small fraction involve sudden changes. Thus, it is inefficient to output spectral parameters at rates up to 600 samples/s. Practical coding and recognition algorithms use parameters at about 25–200 samples/s, depending on the application. This sacrifices accuracy during rapid spectral changes; performance is not greatly degraded (e.g., smoothing rapid changes may not be perceptually noticeable in the reconstructed speech), while parameter storage is greatly reduced.

6.10.2 Smoothing: Linear and Nonlinear

To be able to subsample parameter signals at rates as low as 25/s, the signals should first be lowpass filtered to obey the Nyquist rate. In some cases, an analysis produces an appropriate signal for decimation (e.g., one can choose the window bandwidth in the short-time DFT to match the desired parameter sampling rate P_s). Occasionally, however, a slowly varying parameter is interrupted by rapid fluctuations. In yet other situations, small fine temporal variation may be superimposed on a slowly varying base pattern. Assuming that the slowly varying contour is the desired component for storage, transmission, or further analysis, smoothing is necessary so that subsampling does not give spurious results.

The basic approach is linear lowpass filtering to eliminate energy in the parameter signal above half the desired P_s. This has the advantage of smoothing rapid parameter transitions, e.g., which can be of use in phone segmentation for speech recognizers. If the parameter is simply to be subsampled at a fixed rate, then linear filtering may be best. However, other ways to represent parameter signals in a reduced-data format are often more successful with nonlinear smoothing. Linear filtering is particularly inappropriate for F0 patterns, in which F0 is traditionally considered to be zero during unvoiced sections of speech. Voiced–unvoiced transitions are abrupt, and linear smoothing yields poor F0 values. Linear filters are also suboptimal for signals with discrete values (e.g., unlike continuous parameters such as energy, discrete parameters classify speech into one of a finite set of states, such as phonemes).

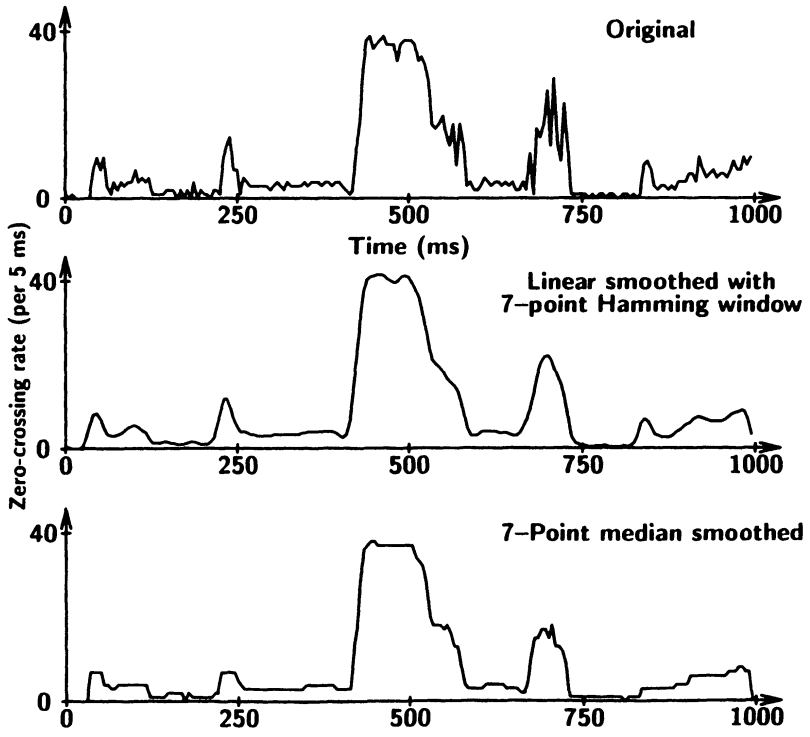

Figure 6.21 Example of smoothing applied to a zero-crossing parameter signal.

Another difficulty with linear filtering is its behavior when mistakes occur in parameter extraction. Formant and F0 estimators are notorious for producing erroneous isolated estimates or *outliers*, deviating from the rest of the parameter contour. Such mistakes should be corrected in postprocessing, but some errors may persist in the output. Linear filters give equal weight to all signal samples, propagating the effect of a mistake into adjacent parts of the smoothed output parameter contour.

One alternative to linear filtering is *median smoothing* [110], which preserves sharp signal discontinuities while eliminating fine irregularities and outliers. Most smoothers operate on a finite time window of the input signal, but linear smoothers linearly combine the windowed samples to produce the smoothed output sample, whereas median smoothing chooses a single value from among the window samples. In each set of windowed data, the samples are ordered in amplitude without regard to timing within the window. The output sample is the median, i.e., the $((N + 1)/2)$nd of N ordered samples (for odd N). Sudden discontinuities are preserved because no averaging occurs. Up to $(N - 1)/2$ outlier samples, above or below the main contour, do not affect the output (Figure 6.21).

Median smoothers do well in eliminating outliers and in global smoothing, but do not provide very smooth outputs when dealing with noisy signals. Thus they are often combined with elementary linear smoothers to yield a compromise smoothed output, with sharp transitions better preserved than with only linear filtering and with a smoother output signal than would be possible with only median smoothing.

6.11 SUMMARY

This chapter presented an introduction to speech analysis methods, from the viewpoint of transforming the speech signal into a set of parameters that more economically represent its pertinent information. Time-domain analysis yields simple speech parameters, especially suitable for energy and segmentation, whereas spectral analysis provides the more common approach to an efficient representation of speech information.

Since most speech applications use LPC and/or cepstral parameters, let us finish with a summary of the most common steps in speech analysis. After A/D conversion to $s(n)$ with typically 16 bits/sample at F_s samples/s, preemphasis (Equation (6.15)) may be applied (e.g., $x(n) = s(n) - 0.95s(n - 1)$). The $x(n)$ samples are then buffered into frames of N samples at a time (e.g., 25 ms units, overlapped and updated every 10 ms) and multiplied by a Hamming window) (Equation (6.2)). For each weighted frame, an autocorrelation matrix is calculated, and then the LPC reflection coefficients are computed (Equation (6.24)). At this point, most coders have the required spectral parameters for their synthesis filters.

For speech recognition, the cepstral coefficients require more computation. They can be obtained directly from the LP parameters, but this way does not include the popular mel-scale mapping. For that, we instead use an FFT on the speech frame (skipping LP analysis), getting $X(k)$ (Eq. (6.14)), then the take the log-magnitude $\log X(k)$, multiply by the critical-band triangular filters (Eq. (6.63)), and take the inverse FFT (Eq. (2.17)). The low-order 10–16 parameters are the static mel-scale cepstral coefficients, from which the delta parameters are simply differenced values between two neighboring frames. These cepstral parameters are often weighted by a raised-sine pulse (to de-emphasize low-order values that may relate to channel conditions, as well as high-order values that correspond to less relevant fine spectral detail—see Chapter 10).

PROBLEMS

P6.1. Consider time windows for speech analysis.
(a) What are the advantages and disadvantages of short and long windows?
(b) To what type of filter should the spectrum of a window correspond?
(c) Explain how the bandwidth of an analysis window affects spectrographic estimation of formants and F0.
(d) How is placement of a window on the speech signal important?

P6.2. Consider a pitch detection scheme that lowpass filters the speech to 900 Hz and then calculates an autocorrelation function.
(a) Why is the speech first lowpass filtered?
(b) How is the autocorrelation function used to generate a pitch estimate?
(c) How is an LPC residual useful to find pitch?

P6.3. Consider a steady vowel with formants at 500, 1500, 2500, ... Hz, lowpass filtered to 4000 Hz, and then sampled at the Nyquist rate.
(a) Draw a detailed block diagram of a system to generate a good version of this (already sampled) signal at 10,000 sample/s.
(b) Within the range $|\omega| < \pi$, at which "digital" frequencies ω_k would the formants be for the 10,000 sample/s signal?

P6.4. "Time windowing" is a basic operation in speech analysis.
(a) Explain how the durations of the window affects the output of the analysis for the discrete Fourier transform (e.g., spectrograms).
(b) Instead of multiplying speech by a window, we may convolve the two signals. How is this useful? What features should the window have?

P6.5. A simple FIR filter has one tap (with multiplier coefficient a).
(a) For what values of a does the filter act as a simple time window for speech analysis? Explain.
(b) Is this window useful for wideband or narrowband spectrograms?
(c) What advantages would there be to use a longer time window?

P6.6. One measure of a speech signal is its zero-crossing rate. What information about the speech spectrum is available from this measure? Specifically, what information concerning formants and manner of articulation can be found in the zero-crossing rate?

P6.7. A vowel has formants at 500, 1500, 2,500, ... Hz, etc., and F0 = 200 Hz.
(a) Which harmonic has the highest amplitude? Explain.
(b) Suppose the time waveform of the vowel is sharply lowpass filtered so that no energy remains above 4 kHz. Then the waveform is sampled at 6000 sample/s. If the speech is played back now through a digital-to-analog (D/A) converter, how would the signal be different from that before the sampling? Have the formants changed? Explain.
(c) Suppose instead that the waveform had been properly sampled at the Nyquist rate. Describe in detail a way to change the sampling rate to 6000 sample/s without corrupting the signal.

P6.8. Consider $x(n) = \sin(\omega n)$, where ω is a fixed frequency, and suppose $x(n)$ is input to a 3-level center clipper whose output is

$$y(n) = \begin{cases} 1 & \text{for } x(n) > C, \\ 0 & \text{for } |x(n)| \leq C, \\ -1 & \text{for } x(n) < -C. \end{cases}$$

(a) Sketch $y(n)$ for $C = 0.5$ and $C = \sqrt{3}/2$.
(b) Sketch the autocorrelation function $\phi(k)$ for the two waveforms in part (a).

(c) How would a simple pitch detector determine an F0 estimate of $x(n)$ based on $\phi(k)$ in part (b)?

P6.9. With all-pole LPC analysis:
(a) How many poles are needed in the synthesizer to model well speech of 3 kHz bandwidth?
(b) Why does the analysis window have to be larger when the analysis is done without a pitch detector?

P6.10. In LPC analysis, we can vary the order p of the model (p = number of poles), the length M of the analysis frame window, and the time L between parameter updates.
(a) If $N = 2$ with LPC coefficients a_1 and a_2, explain how the coefficients would vary for different speech sounds (e.g., a vowel and a fricative).
(b) Explain the criteria for choosing the window size M; what are the advantages and disadvantages of using a large M?
(c) Explain the criteria for choosing the update interval L.
(d) For which set of phonemes is the LPC residual error signal large?

P6.11. Explain the advantages and disadvantages of using wavelets for speech analysis, instead of a Fourier transform.

P6.12. How can pre-emphasis help speech analysis?