APPENDIX 6

ANALYSIS ALGORITHMS

The speech analysis software provided with this book is implemented in MATLAB using a graphic user interface. The software provides basic display tools as well as editing tools and also has features for glottal inverse filtering.

A6.1 TIME DOMAIN ANALYSIS

For time domain analysis, the energy and zero-crossing rate can be calculated as well as the biased autocorrelation function. For our use, all data are windowed. A frame is a windowed segment of data. In speech research, we use short-time analysis of data, that is, the data are windowed. Thus, we use an analysis data frame. Data frames can be overlapped or not. The result of the frame analysis of data is usually a number, or perhaps a small set of numbers, that is less than the number of data samples within the data window. This set of numbers varies with time, that is, it forms a waveform. So we analyze data to reduce the number of data samples in a given data waveform to a small set of values or features. These features can be used to determine characteristics of speech, for example, if voicing or non-voicing is present in the speech segment being analyzed.

The window option under time (domain) analysis illustrates how frames of data are generated. The reader should experiment with this option.

One of the most common short-time analysis methods is the short-time average energy of a signal. If w(n) is the data window for $0 \leq n \leq N-1$ and is zero otherwise, and x(n) is the data record, then the short-time average energy is

$$E_n = \sum_m [x(m)w(n-m)]^2 = \sum_m x^2(m)h(n-m) \qquad \text{(A6.1.1)}$$

where $h(n) = w^2(n)$. Thus, h(n) is a window (filter) that affects the short-time energy function. If this window is very short, then its bandwidth is large and there is little filtering of the data by the window. If the window is very long, then the window bandwidth is narrow, and there is a great deal of filtering of the data. We select the window parameters (or type of window) to provide "good" data smoothing. The type of window selected can depend on the application or task but for speech, the window used is usually the hamming window.

An illustration of windowing the data using the window option under time analysis in the software provided with this book is shown in Figure A6.1. In this figure, the energy at point n1 of the data is calculated and labeled as E_{n1}. The point n1 is at the end of the window shown directly above the windowed data. Then the window is shifted to point n2 and the energy is calculated and labeled as E_{n2}. This procedure is repeated for point n3. A plot of these values is called the energy contour. Note in Figure A6.1 that the x-axis scales are not perfectly aligned with one another. The scales for the data and the windowed data align. However, the scale of the window is slightly off. This is a problem in MATLAB. This explains why the "notch" in the windowed data does not occur where two successive windows overlap. One can see that the energy contour is a considerable data reduction over that of the actual data. This is quite useful for certain applications.

Since the squaring of the data in the calculation of the energy can make small data values even smaller, the average magnitude function is often used instead of the average energy function. The

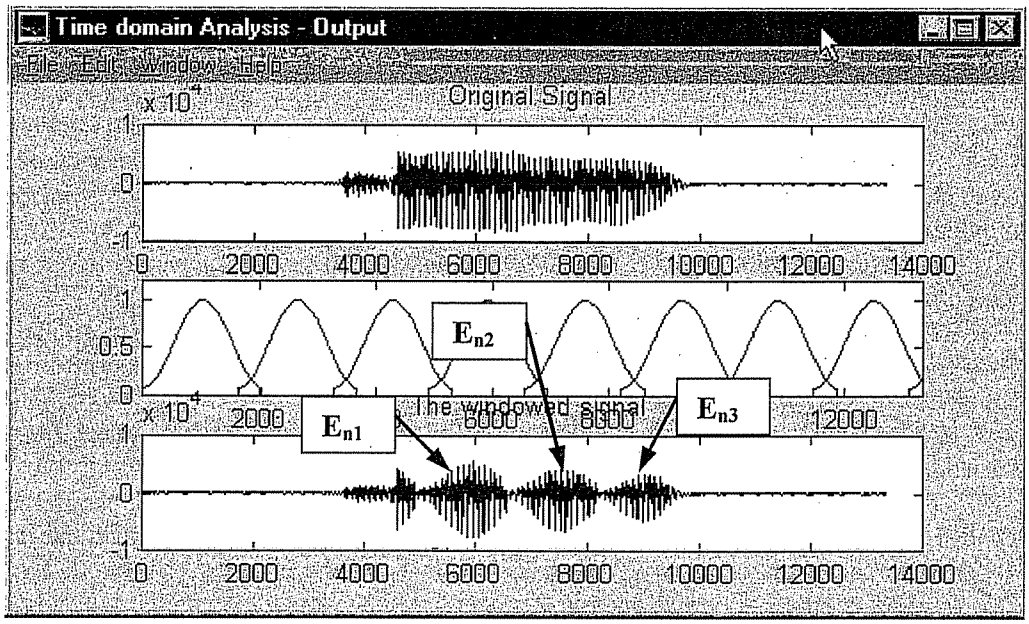

FIGURE A6.1 An illustration of calculating the average energy.

average magnitude function is

$$M_n = \sum_m |x(m)|w(n-m) \tag{A6.1.2}$$

The advantage of these short-time functions is that they have a smaller bandwidth than the signal, so they can be sampled less frequently. For example, a speech signal sampled at 10,000 samples/sec can be reduced to approximately 100 samples/sec using the short-time energy function for some applications.

The short-time average zero-crossing rate is used to estimate the frequency of voicing. Suppose we are given a sampled sinewave of frequency F_0 that is sampled at rate F_s. The number of samples in a period is $N = F_s/F_0$. For a periodic signal, we have two zero-crossings per period, which is $2F_0/T$, where $T = 1/F_s$. To estimate F_0, count the number of zero-crossings in the sampled data over some time interval, T_1. Then T_1 is the number of samples in T_1 divided by F_s, that is, $T_1 = N_1/F_s$ or $N_1 = F_sT_1$. The average number of zero-crossings per sample is $2F_0/F_s$, thus $F_0 =$ (average number of zero-crossings)/$(2T_1)$. So the zero-crossing rate algorithm is the following.

- Count the number of samples in the data between sign changes. This interval (number of samples) represents $\frac{1}{2}$ the period or $\frac{1}{2F_0}$.
- The average number of samples between sign changes is N_1.
- $N_1/F_s = \frac{1}{2F_0}$
- $F_0 = \frac{F_s}{2N_1}$

The short-time energy and zero-crossing functions are useful for estimating word and phoneme boundaries.

The short-time autocorrelation function is

$$R_n(k) = \sum_m x(m)w(n-m)x(m+k)w(n-k-m) \tag{A6.1.3}$$

which is the autocorrelation function for the windowed data. This function is useful for estimating the fundamental frequency of voicing for voiced speech segments.

A6.2 FREQUENCY DOMAIN ANALYSIS

Linear prediction (LP) analysis is based on an all pole model of the data, as discussed in Chapter 5. This analysis technique is also known as autoregression (AR) analysis, which is one of the most popular time series modeling methods because accurate estimates of the model parameters can be calculated by solving a set of linear equations using well known algorithms. The LP spectral estimator provides less bias and has less variability than Fourier-based spectral estimators.

The autocorrelation method (Yule–Walker) is used to estimate the LP parameters by minimizing an estimate of the prediction error power.

The covariance method estimates the LP parameters by minimizing an estimate of the prediction error power as well. However, the summation over the observed data samples differs from that for the autocorrelation method.

The modified covariance method estimates the LP parameters by minimizing the average of the estimated forward and backward prediction error powers.

There are three methods for calculating an estimate of the order of the LP model, namely, final prediction error (FPE) due to Akaike, the Akaike information criterion (AIC), and the criterion autoregressive transfer function (CAT) due to Parzen. However, these, and other, methods for estimating the LP model order are not particularly useful in practical situations. One rule to follow is that the LP model order, p, should be such that $N/3 < p < N/2$, where N is the data record length. For further discussion of these issues see Kay (1988).

In contrast to the above methods, which estimate the LP parameters directly, the Burg method estimates the reflection coefficients first, and then uses the Levinson recursion to obtain the LP parameter estimates. The reflection coefficient estimates are obtained by minimizing estimates of the prediction error power for different order predictors in a recursive manner. For more information, see Kay (1988), where a discussion of the recursive maximum likelihood estimation procedure can also be found.

The perceptual linear prediction (PLP) analysis technique (Hermansky, 1990), is based on well established psychophysical concepts of hearing. The speech signal is filtered by a critical band filter bank followed by an equal loudness pre-emphasis, and an intensity to loudness adjustment using the intensity-loudness power law. The auditory spectrum is then modeled by an LP model. The PLP analysis yields an auditory spectrum with relatively low-frequency resolution. Furthermore, the frequency resolution of the auditory spectrum is nonuniform. In the higher frequency range it has less resolution, which agrees with the characteristics of the human auditory system. The PLP method is more consistent with human hearing than the conventional LP method. The PLP method is computationally efficient and yields a low dimensional representation of speech. The block diagram of PLP analysis is shown in Figure A6.2.

A moving average (MA) process has a power spectral density with wide peaks and/or sharp nulls. Therefore, the MA spectral estimator is not a high resolution spectral estimator for processes with narrowband spectral features. When the underlying process is an MA process, the MA estimator is more accurate than conventional Fourier-based estimators. Durbin's algorithm (maximum likelihood estimation) for the estimation of the MA parameters of an MA(q) process uses the data $\{x[O], x[l], \ldots, x[N-1]\}$ to fit a large order AR model to the data by the autocorrelation method (see Kay, 1988).

ARMA spectral estimation is a more general model than AR. Since nearly all data are corrupted by some amount of observation noise, the ARMA model is nearly always the appropriate one. However, optimal estimators based on maximum likelihood for an ARMA model require the solution of nonlinear equations. As a result, suboptimal but easily implementable algorithms have been emphasized. In this software, the Akaike approximate MLE method, modified Yule–Walker equations, least squares modified Yule–Walker equations, and the Mayne–Firoozan method are included. For more detail on these algorithms, refer to (Kay, 1988).

The spectrogram is calculated using the MATLAB function specgram.

The menu window for the spectrum option includes FFT analysis, periodogram estimation, Blackman–Tukey analysis, the MUSIC spectrum estimation (uses a MATLAB function), and the ESPRIT spectral estimation method. The periodogram is an inconsistent estimator in that even though the average value converges to the true value as the data record length becomes large, the variance is

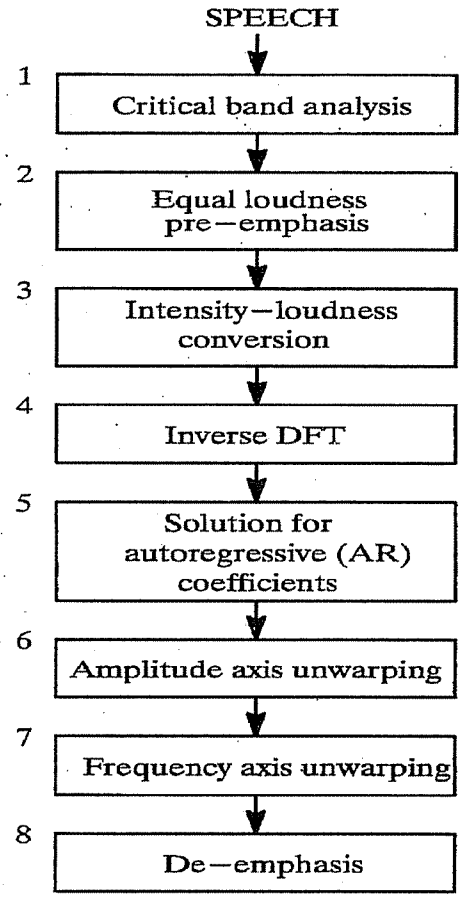

FIGURE A6.2 Block diagram of PLP analysis.

a constant. To circumvent this problem, we can reduce the variance by segmenting the data by non-overlapping blocks and averaging the periodograms of each block. However, this method increases the bias. The Blackman–Tukey spectral estimator also has a bias to variance trade off. The trade off is affected by the choice of the lag window. For more information on the MUSIC estimator and ESPRIT estimator, refer to Kay (1988) and Therrien (1992).

A6.3 GLOTTAL INVERSE FILTERING

The manual inverse filtering option is implemented to assist the analysis of the speech signal with an automatic glottal inverse filtering algorithm. The filter coefficients of the inverse filter can be adjusted by hand using keyboard entry or with a mouse selection so that the glottal volume velocity has a maximally flat spectrum. To construct the inverse filter manually, the vocal tract filter is modeled with a cascade model. First, by controlling the five formant frequencies and bandwidths, the spectrum of the cascade vocal tract model is matched to the spectrum of the pre-emphasized speech. Then the inverse filter transfer function is the reciprocal of the transfer function of the cascade model. The software program provides initial estimates of the formant frequencies and bandwidths. The filter parameters can be calculated pitch synchronously as well as pitch asynchronously.

The pole-zero plot and spectrum of the speech signal can be displayed along with the result obtained by glottal inverse filtering. Using these tools, accurate values of formant frequencies and bandwidths along with the glottal excitation signal can be obtained. The estimates can be saved for further research, such as speech synthesis and coding, and other uses.

Alku (1992) suggested a new glottal wave analysis algorithm PSIAIF (pitch synchronous iterative adaptive inverse filtering) based on IAIF (iterative adaptive inverse filtering). In the PSIAIF

method, the glottal pulse is computed by applying the IAIF twice to the same speech signal. The first analysis gives a glottal excitation that spans several pitch periods, which is used to determine the positions and lengths of frames for pitch synchronous analysis. The final result for the glottal waveform is obtained by analyzing the original speech pitch period by pitch period.

The IAIF method can be summarized as follows.

Step 1: The effect of the glottal pulse on the speech is estimated by a first-order LPC analysis.

Step 2: Remove the effect of the glottal pulse by inverse filtering.

Step 3: Estimate the vocal tract transfer function by applying LPC analysis to the output of Step 2.

Step 4: Inverse filter the speech to obtain the differentiated glottal waveform.

Step 5: Get a first glottal waveform estimate by canceling the lip radiation effect.

Step 6: Estimate a more accurate glottal pulse by applying LPC analysis to the output of Step 5.

Step 7: Inverse filer to remove the effect of the glottal pulse contribution to the speech.

Step 8: Do LPC analysis on the output of the previous step to get the final vocal tract model.

Step 9: Inverse filter the original speech with the output of Step 8.

Step 10: Integrate the output to cancel the lip radiation effect.

The PSIAIF method can be summarized as follows.

Step 1: Apply IAIF to get the glottal pulse and the pitch period for pitch synchronous analysis.

Step 2: Apply IAIF pitch synchronously.

One advantage of pitch synchronous analysis is that it is free of the harmonic structure of the source and, therefore, more accurate modeling is possible, especially when a vowel changes rapidly to another vowel. The iterative method seems to overcome the weak points of conventional linear predictive analysis. When the fundamental frequency is high and the first formant is low, the tract is excited by a source pulse in a way that affects the previous pulse. In other words, the effects of one excitation pulse will not be entirely attenuated before the next one occurs.

A6.4 PITCH/JITTER/FORMANT TRACKING

First, the glottal closure instants are obtained using the glottal excitation waveform, that is, the linear prediction residual signal and a peak picking algorithm. Using this information, the pitch contour can be constructed. The pitch tracking algorithm is similar to the SIFT algorithm (see Markel and Gray, 1976 and Rabiner and Schafer, 1978). The jitter contour is calculated using the pitch perturbation, order 1, method. The perturbation method is expressed as follows. Let a_i be any cyclic parameter, such as amplitude, pitch period, and so on, in the ith cycle of the waveform. The arithmetic mean over N cycles is

$$\bar{a} = \frac{1}{N}\sum_{i=1}^{N} a_i \qquad \text{(A6.4.1)}$$

The zeroth-order perturbation function is the arithmetic difference, given as

$$p_i^0 = a_i - \bar{a}, \quad i = 1, \ldots, N \tag{A6.4.2}$$

where the subscript i denotes the order of the perturbation function. Higher order perturbation functions are obtained by taking backward and forward differences of lower order functions, for example

$$\begin{aligned} p_i^k &= p_i^{k-1} - p_{i-1}^{k-1}, \quad k \text{ odd} \\ &= p_{i+1}^{k-1} - p_i^{k-1}, \quad k \text{ even} \end{aligned} \tag{A6.4.3}$$

where

$$\begin{aligned} &k_1 + 1 \le i \le N - k_2 \\ &k_1 = (k+1)/2, \quad k_2 = (k-1)/2, \quad k \text{ odd} \\ &k_1 = k/2, \quad k_2 = k/2, \quad k \text{ even} \end{aligned} \tag{A6.4.4}$$

For example, the first-order perturbation function is

$$p_i^1 = p_i^0 - p_{i-1}^0 = a_i - a_{i-1}, \quad i = 2, \ldots, N \tag{A6.4.5}$$

and the second-order perturbation function is

$$p_i^2 = p_{i+1}^1 - p_i^1 = a_{i+1} - 2a_i + a_{i-1}, \quad i = 2, \ldots, N \tag{A6.4.6}$$

The pitch and jitter contours are smoothed with a 5-point median filter.

The formant frequency is calculated using the iterative glottal inverse filtering algorithm. From the inverse filtering algorithm, accurate estimates of the vocal tract transfer function are obtained. The formant frequency and bandwidth are calculated by root solving for the AR model coefficients.

A6.5 CEPSTRUM

Historically the cepstrum is concerned with the general problem of the deconvolution of two or more signals. Bogert, Healy, and Tukey (1963) described the power cepstrum for finding echo arrival times in a composite radar signal. They showed that if you have a signal composed of a waveform and an echo of the waveform, then the effect in the log spectrum turns out to be a "ripple." The "frequency" of this ripple is easily determined by calculating the spectrum of the log spectrum wherein this "frequency" will appear as a peak. However, the units of "frequency" of this ripple in the log spectrum are in units of time; thus, the independent variable (abscissa) in the spectrum of the log spectrum is time. Other parameters were also observed to undergo similar transformations of units. To avoid confusion these authors introduced the following paraphrased terms according to a syllabic interchange rule.

frequency....quefrency
spectrum.....cepstrum
phase.....saphe
amplitude....gamnitude
filtering.....liftering
harmonic.....rahmonic
period.....repoid

among others. Today the two most common terms that remain are cepstrum and quefrency. Filtering in the cepstral domain is often not called liftering, but is simply called filtering. The remainder of this discussion is based on a paper by Childers et al. (1977).

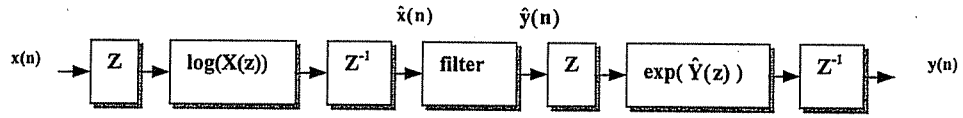

FIGURE A6.3 Calculating the complex cepstrum.

We define the complex cepstrum of a data sequence as the inverse z-transform of the complex logarithm of the z-transform of the data sequence

$$\hat{x}(n) = \frac{1}{2\pi j} \oint_c \log(X(z)) z^{n-1}\, dz \tag{A6.5.1}$$

where $\hat{x}(0) = \log[x(0)]$ and $X(z)$ is the z-transform of the data sequence $x(n)$. Frequently, $\hat{X}(z)$ is used to denote the $\log[X(z)]$. Then $\hat{x}(n)$, the complex cepstrum, is the inverse z-transform of $\hat{X}(z)$. The contour integration lies within an annular region in which $\hat{X}(z)$ has been defined as singular valued and analytic. If we have the convolution of two sequences, then

$$x(n) = f(n) * g(n) \tag{A6.5.2}$$

or

$$X(z) = F(z)G(z) \tag{A6.5.3}$$

and

$$\hat{X}(z) = \log X(z) = \log F(z) + \log G(z) \tag{A6.5.4}$$

or

$$\hat{x}(n) = \hat{f}(n) + \hat{g}(n) \tag{A6.5.5}$$

Further, if $\hat{f}$ and $\hat{g}$ occupy different quefrency ranges, then the complex cepstrum can be liftered (filtered) to remove one or the other of the convolved sequences. We will show how this applies to speech later. Since the phase information is retained, the complex cespstrum in invertible. Thus, if $\hat{g}$ is rejected from $\hat{x}$ by liftering, then $\hat{x} = \hat{f}$ and we can then z-transform, exponentiate, and inverse z-transform to obtain the sequence f, that is, f and g have been deconvolved. Figure A6.3 illustrates an overall wavelet recovery or deconvolution (filtering) system. MATLAB has functions to calculate both the real (`rceps`) and complex (`cceps`) cepstrum, which we use in our software. The real cepstrum is similar to the complex cepstrum but take the absolute magnitude of $X(z)$ first.

Examples of long-pass, short-pass, and notch lifters (filters) are shown in Figure A6.4. These lifters are analogous to high-pass, low-pass, and notch (comb) filters, respectively, in the frequency domain.

The computation of the complex cepstrum is complicated by the fact that the complex logarithm is multivalued. If the imaginary part of the logarithm is computed modulo 2π; that is, evaluated as its principal value, then discontinuities can appear in the phase curve. This is not allowed since the $\log(X(z))$ is the z-transform of $\hat{x}$ and thus must be analytic in some annular region of the z-plane. This problem may be rectified by making the following observations.

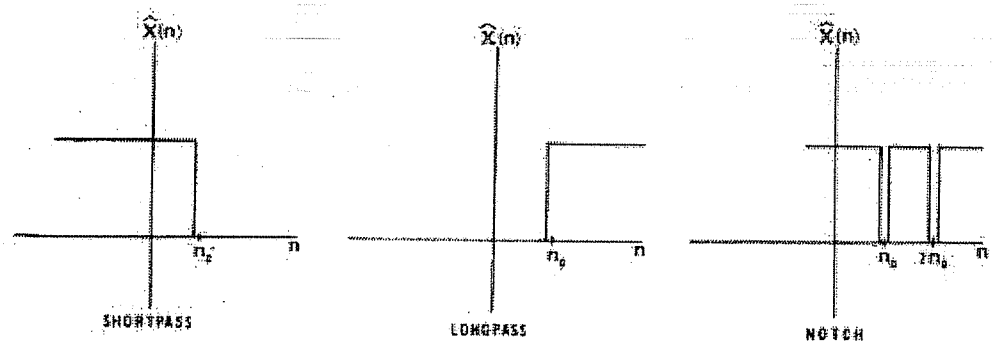

FIGURE A6.4 Example of lifters in the quefrency domain.

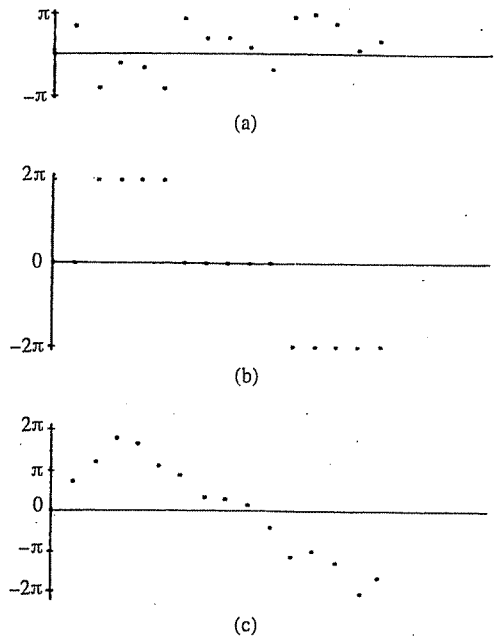

FIGURE A6.5 Phase unwrapping. (*a*) Phase modulo 2π. (*b*) C(k), the correction sequence. (*c*) Unwrapped phase.

- The imaginary part of log(X(z)) must be a continuous and periodic (evaluated on the unit circle) function of ω with period $(2\pi/T)$ since it is the z-transform of $\hat{x}$.
- Since it is required that the complex cepstrum of a real function be real, it follows that the imaginary part of log(X(z)) must be an odd function of ω.

Subject to these conditions, we may compute the unwrapped phase curve as follows (provided the phase is sampled at a rate sufficiently great to assure that it never changes by more than π between samples). A correction sequence C(k) is added to the modulo 2π phase sequence P(k) where C(k) is

$$\begin{aligned} C(0) &= 0 \\ C(k) &= C(k-1) - 2\pi, \quad \text{if } P(k) - P(k-1) > \pi \\ &= C(k-1) + 2\pi, \quad \text{if } P(k-1) - P(k) > \pi \\ &= C(k-1), \qquad\quad \text{otherwise} \end{aligned} \tag{A6.5.6}$$

This is illustrated in Figure A6.5.

Alternately, the phase can be unwrapped by computing the relative phase between adjacent samples of the spectrum. These phases may be added to achieve a cumulative (unwrapped) phase for each point. Both methods have the drawback that the computation must be done sequentially. It is also noted that if the phase never changes by more than $\frac{\pi}{2}$ between samples, the phase modulo π could be computed and unwrapped with algorithms similar to the above. This is interesting since it is slightly easier to calculate the phase modulo π than the phase modulo 2π (the arctangent algorithm is simpler) and many signals have this property (though noise generally does not). There are other phase unwrapping procedures.

Phase unwrapping is unnecessary for the class of minimum phase signals, that is, a sequence with a z-transform that has no poles or zeros outside the unit circle, which implies that $\hat{x}(n) = 0$ for $n < 0$. The complex cepstrum of such a sequence is zero at negative quefrencies. Analogously, a maximum phase sequence can be defined (the z-transform has no poles or zeros inside the unit circle). The complex cepstrum of such a sequence is zero for positive quefrencies. The signals of general interest are of mixed phase. It is difficult to properly analyze or process such signals in the presence of noise.

We now show that the impulses that appear in the complex cepstrum can be caused by the presence of a single additive echo. These impulses are non-zero on only one side of the origin and are therefore referred to as minimum or maximum phase impulse trains.

For $x(n) = f(n) * g(n)$, let

$$g(n) = \delta(n) + a\delta(n - n_0) \tag{A6.5.7}$$

then

$$x(n) = f(n) + af(n - n_0) \tag{A6.5.8}$$

or

$$X(z) = F(z)(1 + az^{-n_0}) \tag{A6.5.9}$$

Taking the log of both sides, we have

$$\hat{X}(z) = \log(F(z)) + \log(1 + az^{-n_0}) \tag{A6.5.10}$$

If $a < 1$ (corresponding to a minimum phase sequence), then we may expand the right most term above in a power series, giving

$$\hat{X}(z) = \log(F(z)) + az^{-n_0} - \frac{a^2}{2}z^{-2n_0} + \frac{a^3}{3}z^{-3n_0} - \cdots \tag{A6.5.11}$$

Inverse z-transforming, we have the complex cepstrum

$$\hat{x}(n) = \hat{f}(n) + a\delta(n - n_0) - \frac{a^2}{2}\delta(n - 2n_0) + \frac{a^3}{3}\delta(n - 3n_0) - \cdots \tag{A6.5.12}$$

Thus the complex cepstrum of the composite signal consists of the complex cepstrum of the basic wavelet, $\hat{f}$, plus a train of δ functions located at positive quefrencies at the echo delay (and its multiples) with amplitudes that are directly related to the echo amplitude. Notch liftering and interpolation (smoothing) can be performed to remove the δ functions. The basic wavelet can then be recovered by inverting the operations used to compute the complex cepstrum (see Figure A6.3). If the complex cepstra of the basic wavelet and the impulse train are sufficiently separated in quefrency, then short-pass liftering can be used to recover the basic wavelet. Analogously, the impulse train, g, can be recovered by using long-pass liftering.

If the echo amplitude is greater than or equal to unity, $a \geq 1$ (corresponding to a maximum phase sequence), then we have

$$\hat{X}(z) = \log(az^{-n_0}F(z)) + \log\left(1 + \frac{1}{a}z^{n_0}\right) \tag{A6.5.13}$$

which can be expanded to give

$$\hat{X}(z) = \log(az^{-n_0}F(z)) + \frac{1}{a}z^{n_0} - \frac{1}{2a^2}z^{2n_0} \ldots \tag{A6.5.14}$$

which can be rewritten as

$$\hat{X}(z) = \log(F(z)) - \log z^{n_0} + \log a + \frac{1}{a}z^{n_0} - \frac{1}{2a^2}z^{2n_0} \ldots \tag{A6.5.15}$$

If we remove the linear phase term ($-\log z^{n_0}$), the complex cepstrum is

$$\hat{x}(n) = \hat{f}(n) + \log(a)\delta(n) + \frac{1}{a}\delta(n + n_0) - \frac{1}{2a^2}\delta(n + 2n_0) \ldots \tag{A6.5.16}$$

Thus, the complex cepstrum again has peaks at the echo delay (and its multiples), however, these peaks now occur at negative rather than positive quefrencies and their gamnitudes (amplitudes) are related to $\frac{1}{a}$ rather than a. If these peaks are removed by liftering and the wavelet recovery procedure is followed including the reinsertion of the linear phase term, then the echo is recovered rather than the basic wavelet. The effect of liftering on the complex cepstrum is schematized in Figure A6.6 for $a > 1$ and $a < 1$.

It will be noticed that the peaks in the complex cepstrum due to the impulse train can never have an amplitude greater than unity regardless of the value of a. Furthermore, note that multiplying the original composite signal by a scale factor only changes the coefficient of the $\delta(n)$ term in the

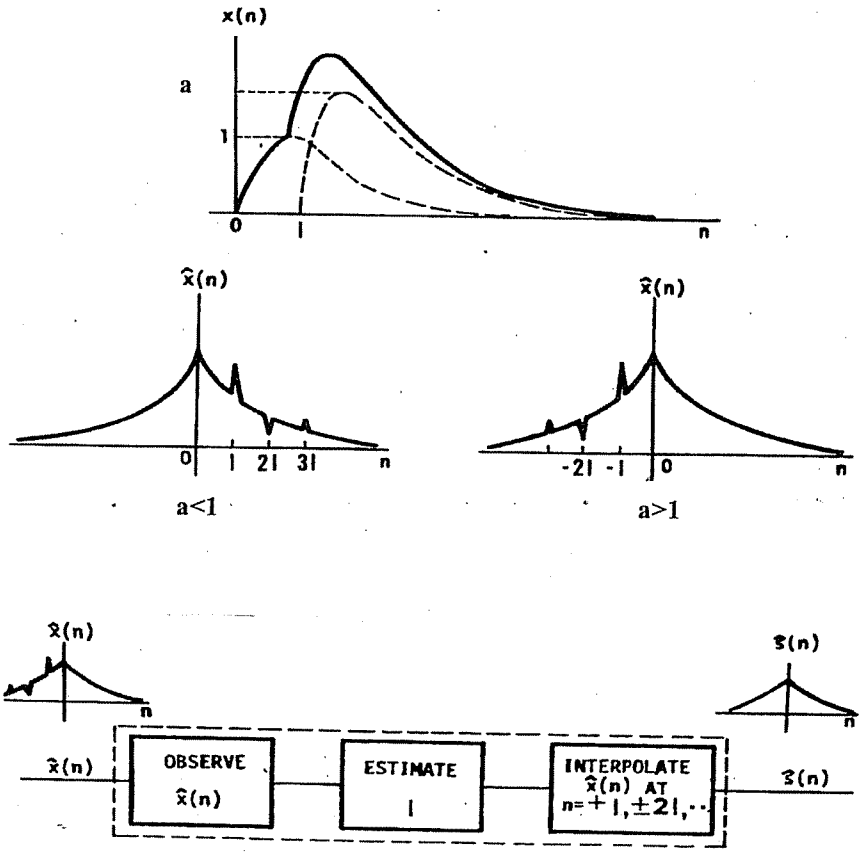

FIGURE A6.6 The superposition of two wavelets to form x(n); the complex cepstra for a < 1 and a > 1; the liftering of the complex cepstrum by notch filtering.

complex cepstrum, since the scale factor appears as a shift in the mean of the log spectrum. Therefore, the complex cepstrum does not depend on the composite signal scale factor, but does depend on the signal-to-noise ratio (SNR).

Another interesting, and perhaps, more representative example is the one with an infinite series of decaying echoes. Here we have

$$g(n) = \delta(n) + \alpha_1^{n_0}\delta(n - n_0) + \alpha_1^{2n_0}\delta(n - 2n_0) + \cdots \tag{A6.5.17}$$

where $0 \leq \alpha_1 \leq 1$. Then with the simplification $\alpha = \alpha_1^{n_0}$ we have

$$\begin{aligned} g(n) &= \delta(n) + \alpha\delta(n - n_0) + \alpha^2\delta(n - 2n_0) + \cdots \\ &= \sum_{m=0}^{\infty} \alpha^m \delta(n - mn_0) \end{aligned} \tag{A6.5.18}$$

which when convolved with f(n) will give us a minimum phase sequence. Then

$$\begin{aligned} G(z) &= 1 + \alpha z^{-n_0} + \alpha^2 z^{-2n_0} + \cdots \\ &= \frac{1}{1 - \alpha z^{-n_0}} \end{aligned} \tag{A6.5.19}$$

Thus the complex cepstrum is

$$\hat{x}(n) = \hat{f}(n) + \alpha\delta(n - n_0) + \frac{\alpha^2}{2}\delta(n - 2n_0) + \cdots \tag{A6.5.20}$$

This complex cepstrum is minimum phase and is nearly identical to that calculated previously, except that the signs of the train of pulse functions are all positive rather than alternating in sign. The remarks made previously apply here as well. When α is near unity, this example might be considered more representative of speech data for the situation of a sustained vowel phonation such as /IY/.

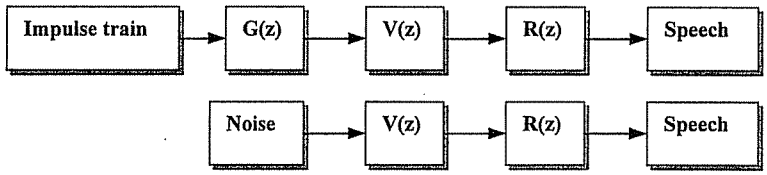

FIGURE A6.7 Two models of speech production.

In the general multiple echo case, the delays become "mixed" via the series expansion of the logarithm. This greatly complicates the proper estimation of the true echo delay times. The estimation is further complicated if aliasing is severe.

For speech production we have two basic models shown in Figure A6.7. The upper model is for voiced speech, while the lower model is for unvoiced speech. In either case, we have the convolution of a signal with transfer functions to produce speech. For voiced speech, if we are given the speech signal and asked to estimate the glottal excitation function, then we must deconvolve the radiation and vocal tract system functions from the glottal transfer function. The complex cepstrum can be applied to this task. A similar task is present for unvoiced speech.

The complex cepstrum will be affected by the vocal tract, the glottal pulse, and the radiation. We will generally have a non-minimum phase signal. This means that the complex cepstrum will be non-zero for both negative and positive quefrencies. The complex cepstrum will decay as $\frac{1}{n}$ with quefrency. The periodic excitation will cause pulses in the complex cepstrum at integer multiples of the spacing between impulses, that is, the complex cepstrum will have pulses at multiples of the fundamental period of voicing.

The impulses due to the fundamental frequency of voicing in speech tend to be separated from the other components (glottal, vocal, and radiation filters) of the complex cepstrum. Thus one can filter the complex cepstrum with a lifter defined as $h(n) = 1$ for $-n_0 < n < n_0$ (and zero elsewhere), where n_0 is less than the pitch period n_p. This lifter tends to select the glottal, vocal tract, and radiation effects rather than the low quefrency material. Therefore it is useful for formant estimation. A lifter defined such that $h(n) = 1$ for $n \geq |n_0|$ and zero elsewhere will retain the impulse excitation train. Thus, this lifter is useful for pitch period estimation. For pitch detection for voiced speech, one measures the interval between the origin and the first peak in the cepstrum. No pulses appear in the cepstrum for unvoiced speech. Thus, we can use the cepstrum to detect voiced/unvoiced segments and to estimate the pitch period.

A6.6 WRLS-VFF

The WRLS_VFF algorithm is discussed in the following paper titled "Adaptive WRLS_VFF for Speech Analysis" previously published in Childers, D. G., Principe, J. C., and Ting, Y. T. (1995). "Adaptive WRLS-VFF for Speech Analysis," *IEEE Transactions on Speech and Audio Processing*, 209–213.

A6.7 SILENT AND VOICED/UNVOICED/ MIXED EXCITATION (FOUR-WAY) CLASSIFICATION OF SPEECH

One algorithm for the four-way classification of speech using the speech and electroglottographic (EGG) signals is described in the following paper titled "Silent and Voiced/ Unvoiced/Mixed Excitation (Four-Way) Classification of Speech," previously published in Childers, D. G., Holm, M., and Larar, J. N. (1989). "Silent and Voiced/Unvoiced/Mixed Excitation (Four Way) Classification of Speech," 37, 1771–1774. The algorithm can be modified to eliminate the use of the EGG signal and rely only on the speech signal.

IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, VOL. 3, NO. 3, MAY 1995 209

Adaptive WRLS-VFF for Speech Analysis

D. G. Childers, J. C. Principe, and Y. T. Ting

Abstract— **The purpose of this correspondence is to show that an adaptive weighted recursive least squares algorithm with a variable forgetting factor (WRLS-VFF) will adjust the size of the data segment to be analyzed according to its time-varying characteristics, as during the transitions between vowels and consonants. The algorithm can accurately estimate the vocal tract formants, anti-formants, and their bandwidths, be used for glottal inverse filtering, perform voiced (V)/unvoiced (U)/silent (S) classification of speech segments, estimate the input excitation (either white noise or periodic pulse trains), and estimate the instant of glottal closure.**

I. Algorithm Description

We assume that the speech signal is generated by an autoregressive, moving average (ARMA) model

$$y_k = -\sum_{i=1}^{p} a_i(k) y_{k-i} + \sum_{j=1}^{q} b_j(k) u_{k-j} + u_k \tag{1}$$

where y_k denotes the kth sample of the speech signal, u_k is the input excitation, (p, q) are the order of the poles and zeros, respectively, and $a_i(k)$ and $b_j(k)$ are the time-varying AR and MA parameters, respectively. Measurement noise is ignored in this model but could be included [10], [11]. We assume that the values of p and q are predetermined. Note that the measured speech signal, y_k, depends on the input, u_k. The excitation, u_k, is usually considered to be white Gaussian noise. In this paper we allow u_k to be either a zero-mean, white, Gaussian noise process, u_k^w, with variance σ_k^2, or a train of periodic pulses, u_k^p. We must estimate the input excitation, either u_k^w or u_k^p, so that the ARMA parameters can be estimated accurately from y_k.

Let us define a parameter vector, θ_k, and a data vector, ϕ_k, by the following equations

$$\theta_k^t = [a_1(k), \cdots, a_p(k), b_1(k), \cdots, b_q(k)] \tag{2}$$

$$\phi_k^t = [-y_{k-1}, \cdots, -y_{k-p}, u_{k-1}, \cdots, u_{k-q}] \tag{3}$$

where the superscript t denotes transpose. The corresponding estimated quantities will be denoted by $\hat{\theta}_k$. Then

$$y_k = \phi_k^t \theta_k + u_k \tag{4}$$

$$\hat{y}_k = \hat{\phi}_k^t \hat{\theta}_k + \hat{u}_k. \tag{5}$$

Let r_k be the residual error of the ARMA process, namely,

$$r_k = y_k - \hat{y}_k = y_k - \hat{\phi}_k^t \hat{\theta}_k - \hat{u}_k. \tag{6}$$

The predicted signal, $\hat{y}_{k/k-1}$, determined from the estimated ARMA parameters at time $(k-1)$, is

$$\hat{y}_{k/k-1} = \hat{\phi}_k^t \hat{\theta}_{k-1}. \tag{7}$$

Manuscript received August 13, 1993; revised November 15, 1994. This work was supported by NIH Grant NIDCD R01 DC00577, NSF Grant IRI-9215331, the University of Florida Center of Excellence Program in Information Transfer and Processing, and the Mind Machine Interaction Research Center. The associate editor coordinating the review of this paper and approving it for publication was Dr. Amro El-Jaroudi.

D. G. Childers and J. C. Principe are with the Department of Electrical Engineering, University of Florida, Gainesville, FL 32611 USA.

Y. T. Ting is with the Chung San Institute of Science and Technology, Taiwan, Republic of China.

IEEE Log Number 9410229.

Consequently, the prediction error is

$$e_k = y_k - \hat{y}_{k/k-1} - \hat{u}_k. \tag{8}$$

Note that $\hat{u}_k$ is usually assumed to be unavailable at $(k-1)$ and is set to zero [10], [11]. We will address this issue again later and modify the algorithm accordingly. An approach to deal with time varying regression coefficients is to minimize a weighted estimation (or residual) error [10], [14]

$$E_k = w(1,k)[\hat{\theta}_k^t P_1^{-1} \hat{\theta}_k] + \sum_{i=1}^{k} w(i,k) r_i^2 \tag{9}$$

where P_1 is an arbitrary real symmetric positive definite matrix. The weighting coefficient ($w(i,k)$) is called a forgetting factor [10], [14]

$$w(i,k) = \prod_{j=i+1}^{k} \lambda_j, \qquad i = 1, 2, \cdots, k-1$$
$$= 1, \qquad i = k, \cdots \tag{10}$$

The coefficient λ_j decreases the weight of past estimation errors provided $0 < \lambda_j < 1$. Note for fixed $\lambda_j = \lambda$ that $w(i,k)$ becomes an exponentially weighted coefficient, e.g., $\lambda^{k-1}, \lambda^{k-2}, \cdots, \lambda, 1$. Consequently, the estimation error, E_k, becomes the exponentially weighted sum of squares of the estimation errors [3], [5], [14], i.e.

$$E_k = \sum_{i=1}^{k} \lambda^{k-i} (y_i - \hat{y}_i)^2 + \lambda^{k-1} [\hat{\theta}_k^t P_1^{-1} \hat{\theta}_k]. \tag{11}$$

Minimizing the least square weighted estimation error, E_k, with respect to the ARMA parameter vector, $\hat{\theta}_k$, assuming that $\hat{u}_k$ is available, gives the following algorithm [10], [11], [14].

Residual error

$$r_k = y_k - \hat{\phi}_k^t \hat{\theta}_k - \hat{u}_k. \tag{12}$$

Prediction error

$$e_k = y_k - \hat{\phi}_k^t \hat{\theta}_{k-1} - \hat{u}_k. \tag{13}$$

Gain update

$$K_k = P_{k-1} \hat{\phi}_k [\lambda_{k-1} + \hat{\phi}_k^t P_{k-1} \hat{\phi}_k]^{-1}. \tag{14}$$

Parameter update

$$\hat{\theta}_k = \hat{\theta}_{k-1} + K_k e_k. \tag{15}$$

Covariance matrix

$$P_k = \lambda_k^{-1} [P_{k-1} - K_k \hat{\phi}_k^t P_{k-1}]. \tag{16}$$

The above algorithm updates the ARMA parameters at each instant k and has been shown to be stable and to provide a unique solution [3], [5], [10], [11], [14].

E_k can be calculated recursively, thereby allowing λ to be calculated recursively. Since $w(k,k) = 1$ and $w(k-1,k) = \lambda_k$, then (9) leads to the following recursive expression

$$E_k = \lambda_k E_{k-1} + (y_k - \hat{y}_k)^2 + w(1,k) \times [\hat{\theta}_k^t P_1^{-1} \hat{\theta}_k - \hat{\theta}_{k-1}^t P_1^{-1} \hat{\theta}_{k-1}]. \tag{17}$$

Then rearranging, we have

$$\lambda_k = \frac{E_k}{E_{k-1}} - \frac{(y_k - \hat{y}_k)^2}{E_{k-1}} - \frac{w(1,k)}{E_{k-1}} \times [\hat{\theta}_k^t P_1^{-1} \hat{\theta}_k - \hat{\theta}_{k-1}^t P_1^{-1} \hat{\theta}_{k-1}]. \tag{18}$$

Using the previous expressions leads to the following approximation to compute and update λ_k

$$\lambda_k = \frac{E_k}{E_{k-1}} - \frac{e_k^2}{E_{k-1}}[1 - \hat{\phi}_k^t K_k]^2 \tag{19}$$

where we have assumed that the third term in (18) becomes negligible with increasing k since $w(1,k) = \lambda_2\lambda_3\cdots\lambda_k \ll 1$ for $0 \le \lambda_i < 1$.

In (19), λ_k depends upon the ratio of the weighted estimation errors at step k and $k-1$ (E_k and E_{k-1}), and on the prediction error at step k. A simplifying strategy to compute λ_k can be defined if we require that $E_k = E_{k-1} = \cdots = E_1$ [4]. This means that the forgetting factor compensates at each step k for the new error information in the latest measurement. This has the added benefit of normalizing the forgetting factor with respect to the same error information, yielding

$$\lambda_k = 1 - \frac{e_k^2}{E_1}[1 - \hat{\phi}_k^t K_k]^2. \tag{20}$$

The WRLS-VFF algorithm is specified by a set of equations similar to those for the WRLS algorithm. However, the weighting factor, λ_k, is estimated by (20). We recommend that the estimation of E_1 be determined using an initial block of data. The minimum length of the window should be related to the size of the ARMA model, therefore, we further limit the smallest value of λ by

$$\lambda_{\min} = 1 - \frac{1}{N_a}, \text{ if } \lambda_k < \lambda_{\min}, \text{ then } \lambda_k = \lambda_{\min}. \tag{21}$$

We have determined empirically that the value of N_a should be approximately twice the model order to obtain good results.

We now estimate the residual and prediction errors as well as the gain, ARMA parameters, the covariance matrix, and the excitation, u_k. Furthermore, we must determine whether $\hat{u}_k$ is $\hat{u}_k^p$ or $\hat{u}_k^w$. Since the previous results assumed that we had an estimate for u_k, we now modify our definitions of the residual and prediction errors to account for the fact that an estimate for the excitation, $\hat{u}_k$, is not available at k. Thus, we define the following two errors

$$r_k = y_k - \hat{y}_k = y_k - \hat{\phi}_k^t\hat{\theta}_k \tag{22}$$

$$\xi_k = y_k - \hat{y}_{k/k-1} = y_k - \hat{\phi}_k^t\hat{\theta}_{k-1}. \tag{23}$$

The equation for the forgetting factor (eq. (20)) remains as before with e_k replaced by ξ_k. Once we make a decision regarding the input excitation, we define a new error as

$$e_k = \begin{cases} \xi_k & \text{for } \hat{u}_k = u_k^w \\ \xi_k - u_k^p & \text{for } \hat{u}_k = u_k^p \end{cases} \tag{24}$$

We update the parameter estimates and the covariance matrix estimate. The data/excitation vector $\hat{\phi}_k^t$ is updated using the new speech sample y_k and $\hat{u}_k$.

From (20), we see that an increase in the prediction error, e_k, results in a decrease in λ_k. A small value of λ_k indicates that the input has undergone an abrupt change, typically indicating that a glottal pulse excitation has occurred. Hence, we can determine the time of occurrence of a pulse by determining the instant at which λ_k falls below a minimum threshold λ_0. When this occurs we set $\hat{u}_k = \hat{u}_k^p$ and $\hat{u}_k^w = 0$. The magnitude of the pulse excitation is determined from the prediction error ξ_k, at the estimated time of the input pulse by assuming that $\xi_k = \hat{u}_k^p$ [10]. Thus,

$$\hat{u}_k^p = y_k - \hat{\phi}_k^t\hat{\theta}_{k-1}. \tag{25}$$

For white noise input, λ_k is close to unity upon convergence [14]. Under this condition the residual error, r_k, of the adaptive process can be used as the estimate of the white noise input, $\hat{u}_k^w$, as indicated in Morikawa's method [12], [13], i.e., from (22)

$$r_k = y_k - \hat{y}_k = \xi_k(1 - \hat{\phi}_k^t K_k) = \hat{u}_k^w \tag{26}$$

and $\hat{u}_k^p = 0$. This is similar to estimates used previously [5], [12], while a different approach was used in [10]. However, our approach uses only one adaptive algorithm instead of two as in [10].

In order to select the threshold value, λ_0, that determines whether the speech was voiced or unvoiced, we adopted the following strategy. We compute a running average of the last M values of the forgetting factor as follows, where M is typically the number of samples in a frame

$$L_k = \frac{1}{M}\sum_{i=0}^{M-1}\lambda_{k-i}. \tag{27}$$

If $L_k < 0.9$, then $\lambda_0 = 0.99^* L_k$; if $L_k > 0.9$, then $\lambda_0 = 0.9^* L_k$; should $\lambda_0 < \lambda_{\min}$, then $\lambda_0 = \lambda_{\min}$. Thus, the threshold may be made adaptive, whereby, it is adjusted on a frame-by-frame basis.

We summarize the WRLS-VFF algorithm including input estimation in Table I. The algorithm in Table I differs from previous algorithms in that we: 1) update the variable forgetting factor at each step, 2) let the prediction error, e_k, be the estimate for the pulse magnitude, $\hat{u}_k^p$, and 3) let the residual error, r_k, be the estimate for the noise excitation, $\hat{u}_k^w$. The algorithm may be shown to be stable and to provide a unique solution following the method given in Appendix III of [10]. Several factors affect the convergence of the WRLS-VFF algorithm: 1) model order, 2) stationarity of the signal, and 3) size of the data analysis interval. We have assumed that the model order may be determined *a priori* and that the data analysis interval is sufficiently large for the algorithm to work. One can show from (22) and (23) that

$$r_k^2 = r_k^2(1 + \lambda_{k-1}^{-1}[\hat{\phi}_k^t P_{k-1}\hat{\phi}_k])^2. \tag{28}$$

If 1) the covariance matrix P_{k-1} is positive definite and 2) $[\hat{\phi}_k^t P_{k-1}\hat{\phi}_k]$ converges to zero as k goes to infinity, then the variance of the prediction error, e_k, converges to the variance of the residual error, r_k. These two conditions can be shown to be satisfied for a

TABLE I
ADAPTIVE WRLS-VFF ALGORITHM WITH INPUT ESTIMATION

Prediction error:	$\xi_k = y_k - \hat{\phi}_k{}^t\,\hat{\theta}_{k-1}$
Gain update:	$K_k = P_{k-1}\,\hat{\phi}_k\,[\lambda_{k-1} + \hat{\phi}_k{}^t\,P_{k-1}\,\hat{\phi}_k]^{-1}$
Forgetting Factor:	$\lambda_k = 1 - \xi_k{}^2\,(1 - \hat{\phi}_k{}^t\,K_k)^2\,/\,E_1$
Input estimate:	
a) Pulse input	If $\lambda_k < \lambda_0$ then $\hat{u}_k{}^w = 0$, $\hat{u}_k = \hat{u}_k{}^p = y_k - \hat{\phi}_k\,\hat{\theta}_{k-1}$
b) White noise input	If $\lambda_k > \lambda_0$ then $\hat{u}_k{}^p = 0$, $\hat{u}_k = \hat{u}_k{}^w = \xi_k\,(1 - \hat{\phi}_k{}^t\,K_k)$
Parameter update:	$\hat{\theta}_k = \hat{\theta}_{k-1} + K_k\,(y_k - \hat{\phi}_k{}^t\,\hat{\theta}_{k-1} - \hat{u}_k{}^p)$
Covariance matrix:	$P_k = \lambda_k{}^{-1}\,[P_{k-1} - K_k\,\hat{\phi}_k{}^t\,P_{k-1}]$

IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, VOL. 3, NO. 3, MAY 1995 211

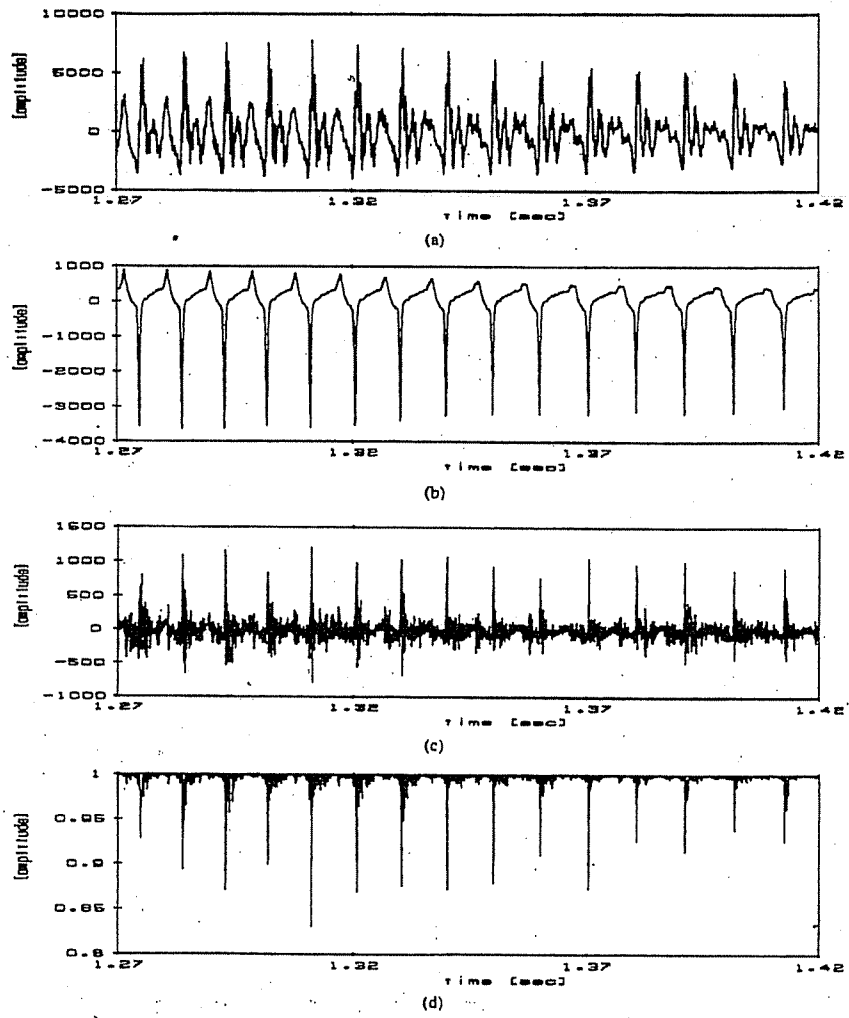

Fig. 1. Speech (a), differentiated electroglottographic (DEGG) signal (b), residual error (c), and variable forgetting factor (VFF) (d).

stationary ARMA process with white noise, zero mean excitation [9], [10], [16] under the assumption that λ_k approaches unity.

The parameter estimation error is usually large when there is a large glottal open phase during speech production. Small values of λ_k occur at the instants of glottal closure where the prediction error is maximum. Fig. 1 shows that the minima of λ_k occur at nearly the same instant as the negative peaks of the differentiated electroglottographic (DEGG) signal. Since the large negative peaks in the DEGG are known to occur at glottal closure (or the minimum glottal aperture) [1], [7], then comparisons such as those in Fig. 1 serve as a partial validation of the WRLS-VFF algorithm. With such validation we have concluded that the minima of λ_k can be used to predict the instant of glottal closure, and, therefore, the presence of voiced excitation. When λ_k does not fall below the threshold value, λ_0, then we may decide that the excitation is unvoiced.

The WRLS-VFF algorithm requires on the order of $(5(p+q)^2 + 6(p+q))$ floating point multiplications and additions (flops) per data point for an ARMA model [16]. By using the idea of shift low rank the WRLS-VFF algorithm can be implemented with $O(N(p+q))$ flops instead of $O(N(p+q)^2)$ [8]. Consequently, the WRLS-VFF algorithm can be made of the same complexity as other recursive algorithms.

II. Conclusion

We have implemented a closed phase adaptive WRLS-VFF algorithm, which is able to track formants and anti-formants for various speech sounds, e.g., vowels, diphthongs, nasals and some consonants for either isolated words or sentences. In [16] we have shown that this algorithm accomplishes these tasks with greater accuracy than other methods, such as LPC [9], Iterative Inverse Filtering (ITIF) [15], the two-stage least squares modified Yule–Walker equations (MYWE) [6], and the recursive algorithms: sequential estimation ARMA (SEARMA) [13], weighted recursive least squares (WRLS) [5], weighted least squares lattice (WLSL) [8], and modified WRLS

(MRLS) [4]. The WRLS-VFF algorithm has also been shown to estimate accurately the values of the formants and their bandwidths for vowels and fricatives [2]. The algorithm is able to perform glottal inverse filtering automatically as well as or better than other procedures that require two channels of data (e.g., speech and EGG) or that require two-passes of the data. The WRLS-VFF algorithm uses the VFF and the estimation error to determine the time at which an excitation pulse occurs and excludes that interval from parameter updating. This provides an improved estimation of the vocal tract transfer function and, consequently, an improved estimation of the glottal volume-velocity waveform.

REFERENCES

[1] D. G. Childers, D. M. Hicks, G. P. Moore, L. Eskenazi, and A. L. Lalwani, "Electroglottography and vocal fold physiology," *J. Speech and Hearing Res.*, vol. 33, pp. 245–254, June 1990.

[2] D. G. Childers and K. Wu, "Gender recognition from speech: Part II. Fine analysis," *J. Acoust. Soc. Am.*, vol. 90, no. 4, pp. 1841–1856, 1991.

[3] C. F. N. Cowan and P. M. Grant, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985, ch. 3.

[4] T. R. Fortescue, L. S. Kershenbaum, and B. E. Ydstie, "Implementation of self-regulators with variable forgetting factors," *Automatica*, vol. 17, pp. 831–835, 1981.

[5] B. Friedlander, "A recursive maximum likelihood algorithm for ARMA spectral estimation," *IEEE Trans. Inform. Theory*, vol. 28, no. 4, pp. 639–646, 1982.

[6] S. Kay, *Modern Spectrum Estimation*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[7] A. K. Krishnamurthy and D. G. Childers, "Two channel speech analysis," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, no. 4, pp. 730–743, 1986.

[8] D. T. L. Lee, M. Morf, and B. Friedlander, "Recursive least squares ladder estimation algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 29, no. 3, pp. 627–641, 1981.

[9] J. Makhoul, "Linear prediction: A tutorial review," *Proc. IEEE*, vol. 64, pp. 99–118, 1976.

[10] Y. Miyanaga, N. Miki, N. Nagai, and K. Hatori, "A speech analysis algorithm which eliminates the influence of pitch using the model reference adaptive system," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 30, no. 1, pp. 88–95, 1982.

[11] Y. Miyanaga, N. Miki, and N. Nagai, "Adaptive identification of a time-varying ARMA speech model," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, pp. 423–433, 1986.

[12] H. Morikawa and H. Fujisaki, "Adaptive analysis of speech based on a pole-zero representation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 30, no. 1, pp. 77–87, 1982.

[13] ——, "System identification of the speech production process based on a state-space representation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 32, no. 2, pp. 252–262, 1984.

[14] T. Soderstrom and P. Stoica, *System Identification*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[15] K. Steiglitz, "On the simultaneous estimation of poles and zeros in speech analysis," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 25, pp. 194–202, 1977.

[16] Y. T. Ting, "Adaptive estimation of time-varying signal parameters with applications to speech," Ph.D. dissertation, University of Florida, 1989.

IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. 37, NO. 11, NOVEMBER 1989

Silent and Voiced/Unvoiced/Mixed Excitation (Four-Way) Classification of Speech

D. G. CHILDERS, M. HAHN, AND J. N. LARAR

***Abstract*—We present an algorithm for automatically classifying speech into four categories: silent and speech produced by three types of excitation, namely, voiced, unvoiced, and mixed (a combination of voiced and unvoiced). The algorithm uses two-channel (speech and electroglottogram) signal analysis and has been tested on data from six speakers (three male and three female), each speaking five sentences. An overall correct classification accuracy of approximately 98.2 percent was achieved when compared to skilled manual classification. This is superior to previously reported automatic classification schemes. If word boundary errors, including the beginning and ending of sentences, are excluded, then the algorithm's performance improves to 99.5 percent.**

INTRODUCTION

In previous work [1], we described a two-channel, two-way (V/U–S) algorithm for automatically classifying speech. This algorithm used the speech and electroglottogram (EGG) signals. One of our objectives has been to demonstrate that two-channel-based algorithms can lead to computational and performance improvements over algorithms based on acoustic-signal-only analysis methods. We recognize that in many situations, the EGG signal is either unavailable or cannot be used. However, both the speech and EGG signals can be used in the laboratory to help benchmark the performance of numerous speech systems. We advocate this approach.

Previous research has focused on three-way speech classification, i.e., either V/U/S or V/U/M [2]–[13]. The speech classification problem is important because its solution affects other speech analysis, synthesis, and recognition problems. For example, speech classification can help reduce the number of lexical candidates in speech (word) recognition, improve speech synthesis by selecting the proper excitation, and improve the performance of phoneme boundary detection in speech analysis. Consider the large vocabulary isolated word recognition problem. By using only four-way (V/U/M/S) classification and stress analysis, one can define an equivalence class of words having the same representation or "coding" [9], [10], [14]. For example, the words speed, steep, scout, and stop all belong to the same equivalence class of U/S/streased-U/V/U. In an isolated word recognition system, the search to identify a test word among all possible candidates can be reduced by using such a simple coding technique. Following such a reduction of the lexical candidates, one may perform other, more detailed analyses to match the test word with one of the remaining words.

Some of the problems with classifying speech as V/U using acoustic-signal-based algorithms are caused by the use of a large analysis frame, a low level of voicing, or even the strength of the first formant. Classification of U/S segments is even more difficult for such algorithms. Typically, researchers have adopted sophisticated approaches to overcome these problems, using additional features, a statistical approach, or an optimized set of parameters [2], [3], [7], [12], [13].

Manuscript received March 8, 1988; revised February 13, 1989. This work was supported in part by NSF Grant ECE-8413583, NIH Grants NINCDS RO1 NS17078 and NS 27022, the University of Florida Center of Excellence Program in Information Transfer and Processing, the Florida High Technology and Industry Council under Grant 4505208-12, and Mind-Machine Interaction Research Center.

D. G. Childers and M. Hahn are with the Department of Electrical Engineering, University of Florida, Gainesville, FL 32611.

J. N. Larar was with AT&T Bell Laboratories, Murray Hill, NJ. He is now with the Department of Medicine, University of Miami, Miami, FL 33101.

IEEE Log Number 8930546.

IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, VOL. 37, NO. 11, NOVEMBER 1989 1771

A Speech-EGG-Based Algorithm for V/U/M/S Classification

A. Algorithm Overview

The properties and some applications of the EGG to speech analysis appear in [1], [8], [15]–[18]. The EGG offers advantages not readily available from a microphone, even a throat contact microphone. The EGG is not susceptible to environment noise, providing instead a direct measure of vocal fold contact [19], while the throat contact microphone provides an acoustic signal similar in form to that provided by other microphones.

The EGG amplitude varies both within and across speakers. Baseline variations in the EGG may be removed by differentiating the EGG. For voiced segments, the EGG usually has only two zero crossings per fundamental (pitch) period of voicing. One exception is vocal fry. For unvoiced segments, the electroglottograph output is a very low-level high-frequency noise-like signal generated by the internal electronics of the device that is easily distinguished from the excitation for voiced speech. Thus, V/U–S classification is achieved using a combination of EGG amplitude and level-crossing rate [1], [8]. Mixed excitation detection is accomplished by noting that the EGG signal appears similar to that for voiced sounds, but the speech signal is small in amplitude and has a high level-crossing rate (see Fig. 2 and other examples in [1]). Silent intervals are detected by observing that the EGG waveform appears as it does for unvoiced speech and that the speech signal is below a predetermined energy threshold. The two-channel, four-way speech classification algorithm appears in Fig. 1. Note that the algorithm does not use endpoint classification, but this could be added if desired [4], [11].

B. Algorithm Details

Fig. 2 depicts both illustrative results and some difficulties encountered in attempting to evaluate the level-crossing rate (LCR) and energy of the EGG signal. Some fluctuations in the EGG data may be removed by simple differentiation, a procedure which also yields a waveform with enhanced positive and negative peaks. These peaks occur approximately at the instants of glottal opening and closure, respectively [1], [8], [18]. The differentiation is implemented as a backward difference equation. The differentiated EGG is then normalized by dividing by its maximum positive value. The resulting waveform is denoted as the DEGG and is shown at the bottom of Fig. 2.

The two-way, V–M and U–S classification uses the LCR and energy information from the DEGG as follows. The DEGG is segmented into 10 ms frames of 100 samples each (10 kHz sampling rate). The energy of each frame n of the DEGG signal is given by

$$E_D(n) = \sum_{i=1}^{100} \left(D((n-1)\,100 + i)\right)^2 \qquad (1)$$

where $D(j)$ denotes the sample value of the DEGG signal. The LCR of the DEGG is calculated at the -0.5 level, also on a frame basis. The energy and LCR contours are smoothed with a three-point zero-phase filter with coefficients (0.19, 0.62, 0.19), such that the filter output is $y_n = 0.19x_{n-1} + 0.62x_n + 0.19x_{n+1}$. The energy and LCR for the speech signal are calculated in a similar manner and the contours are also smoothed.

Other calculations include the following:

1) average energy levels for both the speech and DEGG signals for voiced segments,

2) average of the rectified voiced speech,

3) average LCR for voiced segments that exceed the 10 percent level of signal calculated in step 2) above.

4) smooth the preliminary four-way classification to remove obvious errors in a string. For example, if the preliminary classification is . . . VVVUVV . . ., then "smooth" this string to give . . . VVVVVV . . . .

Various threshold values are determined empirically, but are fixed once selected. The thresholds we selected are shown in Fig. 1 enclosed in the parentheses. Samples of silent intervals are required to establish several threshold values. The other numbers in Fig. 1 were determined experimentally to establish the decision levels for steps (b), (d), and (f).

Results and Discussion

The algorithm has been tested with data from six speakers (three male and three female), each speaking five sentences. The sentences were as follows.

1) We were away a year ago. (Voiced.)

2) Early one morning a man and a woman ambled along a one mile lane. (Voiced and nasals.)

3) Should we chase those cowboys? (Fricatives and plosives.)

4) That zany van is azure. (Voiced fricatives, i.e., mixed.)

5) We saw the ten pink fish. (Unvoiced plosives and fricatives.)

This data set is more extensive than that used in [2], [3], [7], [13].

The threshold values were determined with data from two speakers (one male and one female), each speaking the first three sentences only. An example comparing the algorithm classification to manual classification is shown in Fig. 2. Note that the classification task is aided by the EGG and the DEGG.

The detailed classification results appear in Table I. The designation for the test data sets is as follows.

Complete: Refers to all the data from all six speakers.

Threshold: The subset of Complete used to establish the threshold values, one male and one female speaker, for the first three sentences only.

Nonthreshold: The subset of Complete not included in the Threshold set.

Male: The male speaker subset of Complete.

Female: The female speaker subset of Complete.

The overall correct recognition rate is 98.2 percent when compared to manual classification of the data. The recognition rate is an improvement over the overall 95 percent rate reported in [3], [7] and 88 percent reported in [2], [13]. Nearly 83 percent correct classification of the mixed excitation frames was achieved in [7], which we have increased to 89 percent.

Table II provides a breakdown of the types of errors. The most troublesome classifications for the algorithm were unvoiced and

TABLE I
Classification Results

TEST DATA SETS	TOTAL NUMBER OF FRAMES	NUMBER OF FRAMES IN ERROR	ERROR RATE (%)	CORRECT RATE (%)
COMPLETE	7785	146	1.88	98.12
THRESHOLD	1622	20	1.23	98.77
NON-THRESHOLD	6163	126	2.04	97.96
MALE	3887	47	1.21	98.79
FEMALE	3898	99	2.54	97.46

TABLE II
Error Analyses in Number of Frames

CLASSIFICATION OUTPUT / MANUAL CLASSIFICATION	V	U	M	S	CORRECT RATE (%)
V	5298	24	39	6	98.71
U	20	710	5	6	95.82
M	5	5	81	0	89.01
S	27	9	0	1550	97.73

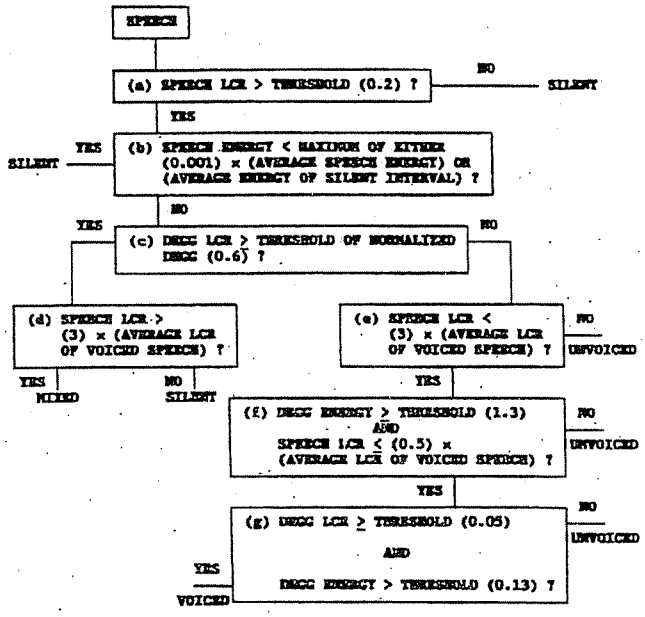

Fig. 1. Speech-EGG algorithm for V/U/M/S (four-way) classification of speech.

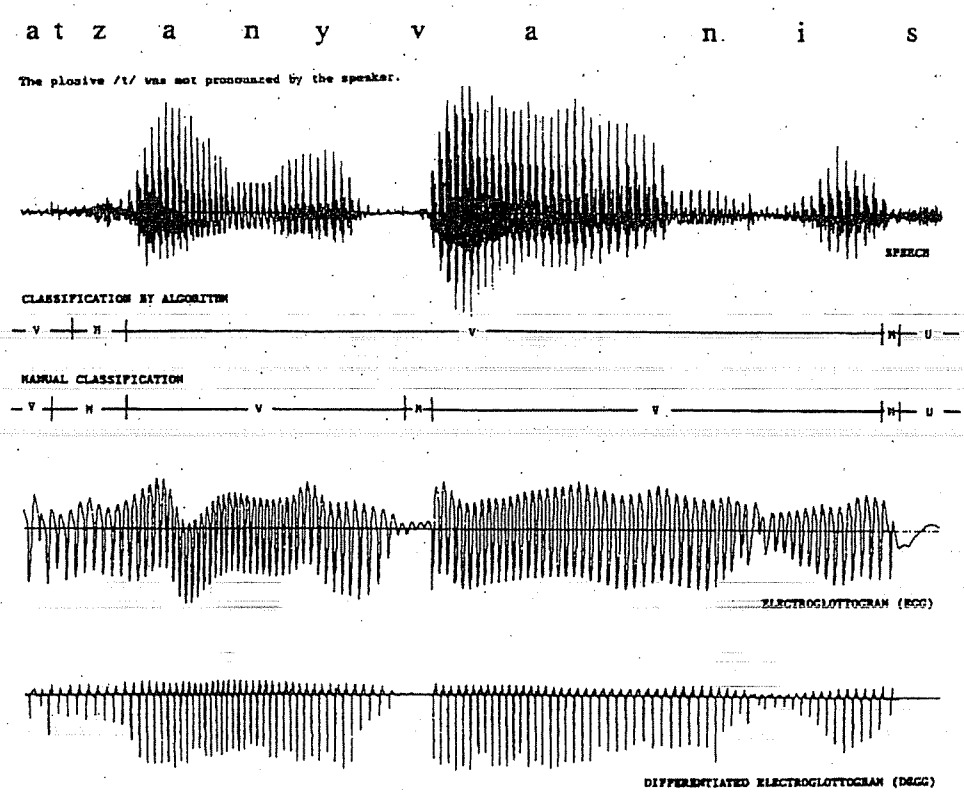

Fig. 2. Comparison of V/U/M/S classification by the algorithm and manual procedures. The speech is a segment of the sentence "That zany van is azure" spoken by a male subject.

mixed excitation frames. A large number of errors (38.7 percent) occurred at the beginning and ending of the sentences. If these errors are ignored, then the overall performance of the algorithm becomes 99.23 percent. If we further ignore the errors that occurred at the boundaries between words, then the overall performance increases to 99.5 percent. The major cause of errors at word boundaries (approximately 60 percent), including the beginning and ending of the sentences, was due to U to V and S to V misclassifications. These errors were caused by a failure to properly recognize voice-onset and voice-offset intervals. An algorithm for recognizing voice onset and offset using the EGG is described in [11] and is an extension of the one in [3], [4]. Note that there is a slight tendency for the algorithm to perform better using male speech than female speech.

Conclusions

We advocate the laboratory use of this algorithm to benchmark speech system performance. The benchmarking can be done automatically and the results compared to acoustic-signal-only-based algorithms. A useful improvement to the algorithm would be a diagnostic capability. For example, perhaps the algorithm could identify and label the frames that were particularly difficult to classify. Such information could conceivably be used to improve system designs. We believe a spectral distance metric can be used to improve the V/U (U/V) and V/S (S/V) classifications.

References

[1] A. K. Krishnamurthy and D. G. Childers, "Two-channel speech analysis," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 730–743, Aug. 1986.

[2] B. S. Atal and L. R. Rabiner, "A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 201–212, June 1976.

[3] L. R. Rabiner and M. Sambur, "Application of an LPC distance measure to the voiced-unvoiced-silence detection problems," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 338–343, Aug. 1977.

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

[5] J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech*. New York: Springer-Verlag, 1976.

[6] R. W. Schafer and J. D. Markel, Eds., *Speech Analysis*. New York: IEEE Press, 1979.

[7] L. J. Siegel and A. C. Bessey, "Voiced/unvoiced/mixed excitation classification of speech," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 451–460, June 1982.

[8] D. G. Childers and J. N. Larar, "Electroglottography for laryngeal function assessment and speech analysis," *IEEE Trans. Biomed. Eng.*, vol. BME-31, pp. 807–817, Dec. 1984.

[9] J. N. Larar, "Towards speaker independent isolated word recognition for large lexicons: A two channel, two-pass approach," Ph.D. dissertation, Univ. Florida, Gainesville, 1985.

[10] ——, "Lexical access using broad acoustic-phonetic classifications," *Comput. Speech Language*, vol. 1, pp. 47–59, 1986.

[11] N. B. Pinto, D. G. Childers, and A. L. Lalwani, "Formant speech synthesis: Improving production quality," *IEEE Trans. Acoust., Speech, Signal Processing*, to appear, Dec. 1989.

[12] F. Daaboul and J. P. Adoul, "Parametric segmentation of speech into voiced-unvoiced-silence intervals," in *Proc. IEEE Conf. Acoust., Speech, Signal Processing*, Hartford, CT, May 1977, pp. 327–331.

[13] L. R. Rabiner, C. E. Schmidt, and B. S. Atal, "Evaluation of a statistical approach to voiced-unvoiced-silence analysis for telephone quality speech," *Bell Syst. Tech. J.*, vol. 56, pp. 455–482, Mar. 1977.

[14] D. W. Shipman and V. W. Zue, "Properties of large lexicons: Implications for advanced word recognition systems," in *Proc. IEEE Conf. Acoust., Speech, Signal Processing*, Paris, France, May 1982, pp. 546–549.

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

[16] W. Hess and H. Indefrey, "Accurate pitch determination of speech signals by means of a laryngograph," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, 1984, pp. 1813.1.1–1813.1.4.

[17] ——, "Accurate time domain pitch determination of speech signals by means of a laryngograph," *Speech Commun.*, vol. 6, pp. 55–58, Mar. 1987.

[18] D. G. Childers and A. K. Krishnamurthy, "A critical review of electroglottography," *CRC Crit. Rev. Bioeng.*, vol. 12, no. 2, pp. 131–164, 1985.

[19] D. G. Childers, D. M. Hicks, G. P. Moore, and Y. A. Alsaka, "A model for vocal fold vibratory motion, contact area, and the electroglottogram," *J. Acoust. Soc. Amer.*, vol. 80, pp. 1309–1320, Nov. 1986.