Linear Predictive Speech Processing

In a variety of applications, it is desirable to compress a speech signal for efficient transmission or storage. For example, to accommodate many speech signals in a given bandwidth of a cellular phone system, each digitized speech signal is compressed before transmission. In the case of a digital answering machine, to save a memory space, a message is digitized and compressed. For medium or low bit-rate speech coders, linear predictive coding (LPC) is most widely used. Redundancy in a speech signal is removed by passing the signal through a speech analysis filter. The output of the filter, termed the residual error signal, has less redundancy than original speech signal and can be quantized by smaller number of bits than the original speech. The residual error signal along with the filter coefficients are transmitted to the receiver. At the receiver, the speech is reconstructed by passing the residual error signal through the synthesis filter. To model a human speech production system, all-pole model (also known as the linear prediction model) is used. In this chapter, human speech production system, spectrogram, speech analysis and speech synthesis using linear prediction are explained.

7.1 Speech Production

When a person speaks, his or her lungs work like a power supply of the speech production system. The glottis supplies the input with the certain pitch frequency (F_0). The vocal tract, which consists of the pharynx and the mouth and nose cavities, works like a musical instrument to produce a sound. In fact, different vocal tract shape would generate a different sound. To form different vocal tract shape, the mouth cavity plays the major role. To produce nasal sounds, nasal cavity is often included in the vocal tract. The nasal cavity is connected in parallel with the mouth cavity. The simplified vocal tract is shown in Fig. 7.1.

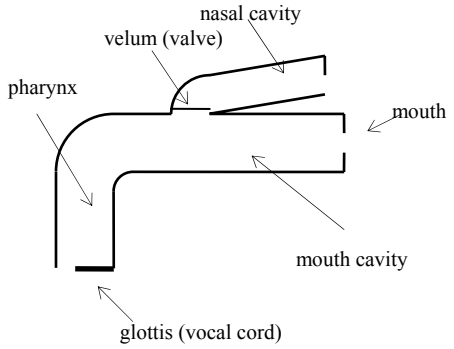
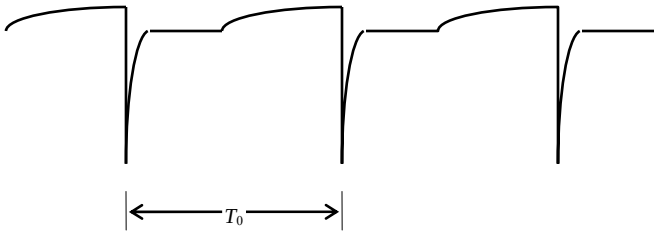


Fig. 7.1 Simplified view of a vocal tract

The glottal pulse generated by the glottis is used to produce vowels or voiced sounds. And the noise-like signal is used to produce consonants or unvoiced sounds. These are shown in Fig. 7.2.

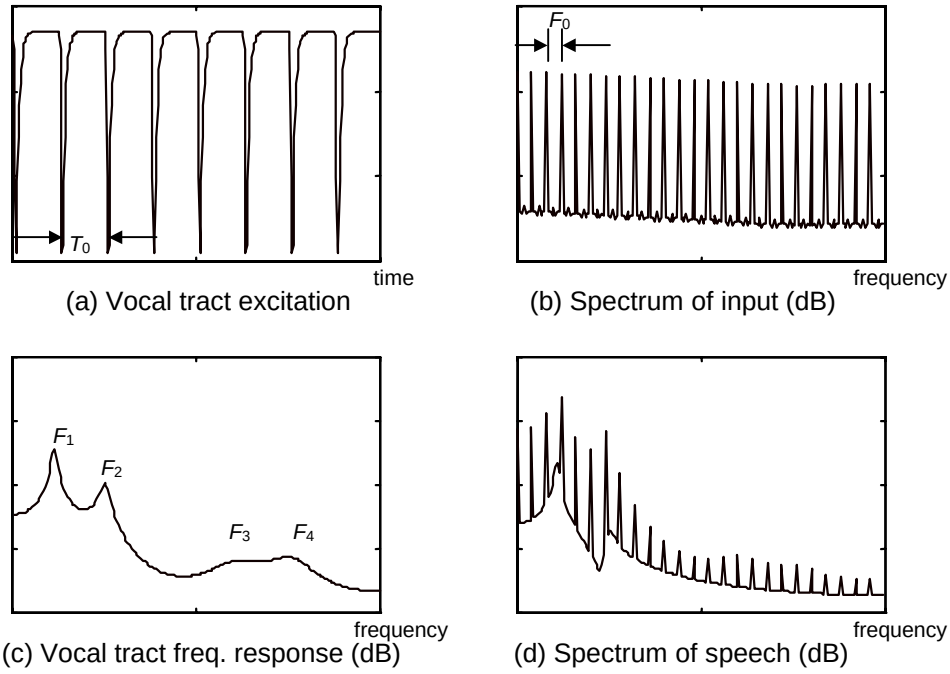


(a) glottal pulse excitation for a voiced sound.

(b) hiss (white noise) input for a unvoiced sound.

Fig. 7.2 Two kinds of inputs used to generate sound (T_0 : pitch period).

Pitch frequency F_0 ($1/T_0$) varies in different people. A little child's pitch frequency can go as high as 400 Hz. Adult male's pitch frequency is as low as 100 Hz. Adult female's pitch frequency is between 200 Hz and 300 Hz range. This glottal pulse excites a vocal tract cavity and produces a vowel (or voiced) sound. Some characteristics of a typical vowel sound are shown below.



(a) Vocal tract excitation

(b) Spectrum of input (dB)

(c) Vocal tract freq. response (dB)

(d) Spectrum of speech (dB)

Fig. 7.3 Characteristics of a typical vowel sound

As shown in Fig. 7.3 (c), there are at least four resonant peaks visible. The frequencies at which the resonant peaks occur are called the *formant frequencies* (or simply *formants*). Formant frequencies, F_1 and F_2, are very distinct from the plot. Formants F_3 and F_4 are not quite distinct. These formant frequencies are often used for speech recognition. From the spectrum of the speech, one can see that there are many harmonics of F_0, pitch frequency. Some of them have higher amplitude according to formants. From the spectrum of speech, it is not easy to extract formants because of many harmonious peaks. On the other hand, by inputting a noise-like signal to the vocal tract, unvoiced sound such as plosives and fricatives are produced

To analyze or examine a speech signal, a *spectrogram* is widely used. Spectrogram is similar to a musical score as shown in Fig. 7.4 (a).

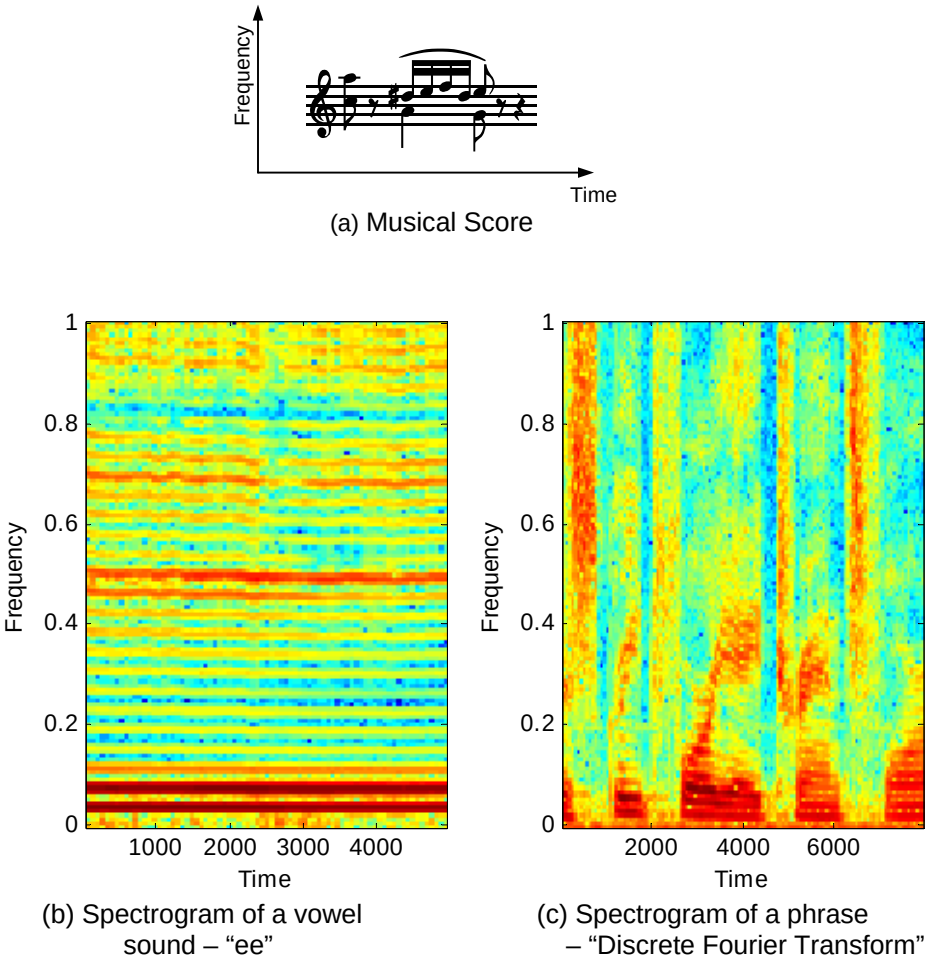


Fig. 7.4 Musical score, spectrograms of a vowel sound and one short phrase.

From the musical score, the first top note is high "A" whose frequency is 440 Hz. The note is played for a half beat followed by a half-beat pause. The first bottom note is

"C" and is played together with "A" note. Obviously, the "C" note has lower frequency than "A" note. The vertical axis in the musical score is for frequency, and the horizontal axis is for time. Fig. 7.4 (b) shows the spectrogram of a vowel sound – "ah." It is, in fact, a three-dimensional plot on a two-dimensional space: high amplitude is indicated by dark spot and low amplitude is indicated by light spot. The dark spots at low frequencies indicate the presence of strong harmonics at low frequencies. Because one vowel sound was uttered over the given time interval, there is almost no change in spectrogram. Fig 7.4 (c) shows the spectrogram of the utterance – "This is a test." Locations of strong harmonics change over time for vowels. However, for consonants, harmonics are not quite visible.

The following are some of the important properties of speech.

- Fricatives (s, sh, f, th) are produced when the vocal tract is constricted at some location and air is forced through that constriction.
- Plosives (p, k, t) are produced when the end of the vocal tract is constricted or closed momentarily while air pressure built up, then pressure is suddenly released.
- There are about 40 phonemes (sound elements) in English (16 vowels, 24 consonants).
- In normal speech, 10 to 15 phonemes are spoken in one second.

7.2 Linear Prediction Model

In this section, an all-pole system (or the linear prediction system) is used to model a vocal tract as shown in Fig. 7.5.

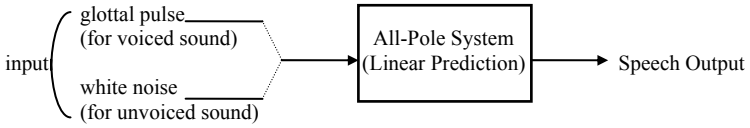


Figure 7.5 Simplified model of the speech production

An efficient algorithm known as the Levinson-Durbin algorithm is used to estimate the linear prediction coefficients from a given speech waveform.

Assume that the present sample of the speech is predicted by the past M samples of the speech such that

$$\tilde{x}(n) = a_1 x(n-1) + a_2 x(n-2) + \cdots + a_M x(n-M) = \sum_{i=1}^{M} a_i x(n-i) \qquad (7.1)$$

where $\tilde{x}(n)$ is the prediction of $x(n)$, $x(n-i)$ is the i-th step previous sample, and $\{a_i\}$ are called the linear prediction coefficients. The error between the actual sample and the predicted one can be expressed as

$$\varepsilon(n) = x(n) - \tilde{x}(n) = x(n) - \sum_{i=1}^{M} a_i x(n-i) \,. \tag{7.2}$$

The sum of the squared error to be minimized is expressed as

$$E = \sum_n \varepsilon^2(n) = \sum_n \left(x(n) - \sum_{i=1}^{M} a_i x(n-i) \right)^2 . \tag{7.3}$$

We would like to minimize the sum of the squared error. By setting to zero the derivative of E with respect to a_i (using the chain rule), one obtains

$$2\sum_n x(n-k)\left(x(n) - \sum_{i=1}^{M} a_i x(n-i) \right) = 0 \qquad \text{for } k = 1, 2, 3, \cdots, M. \tag{7.4}$$

Equation (7.4) results in M unknowns in M equations such that

$$a_1 \sum_n x(n-k)x(n-1) + a_2 \sum_n x(n-k)x(n-2) + \cdots + a_M \sum_n x(n-k)x(n-M)$$
$$= \sum_n x(n-k)x(n) \qquad \text{for } k = 1, 2, 3, \cdots, M. \tag{7.5}$$

Example 7.2.1 Write equations (7.2) through (7.5) for $N = 5$ and $M = 2$.

From equation (7.2), there are five equations for $n = 0, 1, 2, 3, 4$.

$$\begin{aligned}
\varepsilon(0) &= x(0) \\
\varepsilon(1) &= x(1) - a_1 x(0) \\
\varepsilon(2) &= x(2) - a_1 x(1) - a_2 x(0) \\
\varepsilon(3) &= x(3) - a_1 x(2) - a_2 x(1) \\
\varepsilon(4) &= x(4) - a_1 x(3) - a_2 x(2)
\end{aligned}$$

The last three equations can be written as

$$\begin{bmatrix} \varepsilon(2) \\ \varepsilon(3) \\ \varepsilon(4) \end{bmatrix} = \begin{bmatrix} x(2) \\ x(3) \\ x(4) \end{bmatrix} - \begin{bmatrix} x(1) & x(0) \\ x(2) & x(1) \\ x(3) & x(2) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

or

$$\boldsymbol{\varepsilon} = \boldsymbol{x} - \boldsymbol{X}\boldsymbol{a}$$

To minimize the squared error, $\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon}$, we need to set to zero the derivative of it with respect to $\boldsymbol{a}$, i.e.

$$\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} = \left(\boldsymbol{x} - \boldsymbol{X}\boldsymbol{a}\right)^T \left(\boldsymbol{x} - \boldsymbol{X}\boldsymbol{a}\right)$$
$$\frac{\partial \boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon}}{\partial \boldsymbol{a}} = -2\boldsymbol{X}^T\left(\boldsymbol{x} - \boldsymbol{X}\boldsymbol{a}\right) = \boldsymbol{0}.$$

Now we need to solve the following 2 by 2 matrix equation to find a_1 and a_2.

$$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{a} = \boldsymbol{X}^T\boldsymbol{x}$$

which is

$$\begin{bmatrix} \sum_{n=1}^{3} x^2(n) & \sum_{n=0}^{2} x(n)x(n+1) \\ \sum_{n=0}^{2} x(n)x(n+1) & \sum_{n=0}^{2} x^2(n) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_{n=1}^{3} x(n)x(n+1) \\ \sum_{n=0}^{2} x(n)x(n+2) \end{bmatrix}$$

This method is called the covariance method.

Let us assume that a speech signal is divided into many segments (or frames) each with N samples. If the length of each segment (or frame) is short enough, the speech signal in the segment may be stationary. In other words, the vocal tract model is fixed over the time period of one segment. The length of each segment is usually chosen as 20-30 [ms]. If a speech signal is sampled at the rate of 8,000 samples/s (as in telephone application) and the length of each segment is 20 ms, then the number of samples in each segment will be 160. If the length is 30 ms, then the number of samples is going to be 240.

If there are N samples in the sequence indexed from 0 to N–1 such that $\{x(n)\} = \{x(0), x(1), x(2), \cdots, x(N-2), x(N-1)\}$, Equation (7.5) can be approximately expressed in terms of matrix equation. (See Example 7.2.1 for an accurate matrix expression for $M = 2$.)

$$\begin{bmatrix} r(0) & r(1) & \cdots & r(M-2) & r(M-1) \\ r(1) & r(0) & \cdots & r(M-3) & r(M-2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ r(M-2) & r(M-3) & \cdots & r(0) & r(1) \\ r(M-1) & r(M-2) & \cdots & r(1) & r(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{M-1} \\ a_M \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(M-1) \\ r(M) \end{bmatrix} \qquad (7.6)$$

$$\boldsymbol{R}\,\boldsymbol{a} \quad = \quad \boldsymbol{r}.$$

where

$$r(k) = \sum_{n=0}^{N-1-k} x(n)x(n+k). \tag{7.7}$$

This is called the autocorrelation method. There is a difference between this autocorrelation method and the covariance method. However, when N is large enough there is not much difference. Also there is an advantage with the autocorrelation method as the synthesis filter is always stable. To solve the matrix equation (7.6), Gauss elimination, iteration method, or QR decomposition can be used. In any case, an order of M^3 multiplications is required to solve the equation. However, because of the special characteristics of the matrix, the number of multiplications can be reduced to the order of M^2 with the Levinson-Durbin algorithm that will be introduced in the next section.

Once the linear prediction coefficients $\{a_i\}$ are found, Equation (7.2) can be used to compute the error sequence $\varepsilon(n)$. The implementation of Equation (7.2), where $x(n)$ is the input and $\varepsilon(n)$ is the output, is called the analysis filter and shown in Figure 7.6.

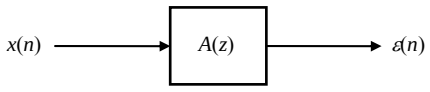


Fig. 7.6 Speech Analysis Filter

The transfer function is given by

$$A(z) = 1 - \sum_{i=1}^{M} a_i z^{-i}. \tag{7.8}$$

Because $\varepsilon(n)$, residual error, has less standard deviation and less correlated than speech itself, smaller number of bits is needed to quantize the residual error sequence.

Equation (7.2) can be rewritten as the difference equation of a digital filter whose input is $\varepsilon(n)$ and output is $x(n)$ such that

$$x(n) = \sum_{i=1}^{M} a_i x(n-i) + \varepsilon(n). \tag{7.9}$$

The implementation of equation (7.9) is called the synthesis filter and is shown in Figure 7.7.

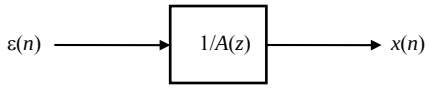


Fig. 7.7 Speech Synthesis Filter

If both the linear prediction coefficients and the residual error sequence are available, the speech signal can be reconstructed using the synthesis filter. In practical speech coders, linear prediction coefficients and residual error samples need to be compressed before transmission. Instead of quantizing the residual error, sample by sample, several important parameters such as pitch period, code for a particular excitation, etc are transmitted. At the receiver, the residual error is reconstructed from the parameters.

7.3 Levinson-Durbin Recursive Method

In this section the Levinson-Durbin method is introduced to solve equation (7.6) recursively. The Levinson-Durbin method is efficient, as it needs only the order of M^2 multiplications to compute the linear prediction coefficients.

The sum of squared errors of the M-th order prediction (or simply the M-th order prediction error) in equation (7.3) can be rewritten as

$$E_M = \sum_n x(n)\varepsilon(n) - \sum_n \left(\sum_{i=1}^{M} a_i x(n-i) \right) \varepsilon(n) \tag{7.10}$$

where subscript M of E_M denotes the order of prediction. Equation (7.4) can be rewritten as

$$\sum_n x(n-i)\varepsilon(n) = 0 \qquad \text{for } i = 1, 2, 3, \cdots, M. \tag{7.11}$$

Because of equation (7.11), the second summation of equation (7.10) is zero. Thus, the final expression of the prediction error becomes

$$E_M = \sum_n x(n) \left(x(n) - \sum_{i=1}^{M} a_i x(n-i) \right) \tag{7.12}$$

$$= r(0) - a_1 r(1) - a_2 r(2) - \cdots - a_{M-1} r(M-1) - a_M r(M) = r(0) - \sum_{i=1}^{M} a_i r(i) .$$

We now want to develop a recursive method to solve equation (7.6). Let us start from the order $m = 0$ and increase it until the desired order reaches.

***m*=0:** When $m = 0$ (i.e., when no prediction is made), the error is expressed from equation (7.12).

$$E_0 = r(0). \tag{7.13}$$

***m*=1:** When $m = 1$,

$$E_1 = r(0) - a_{11}r(1) \tag{7.14}$$

where the second subscript 1 of a_{11} indicates that the prediction order m in this case is 1. The solution to equation (7.6) is

$$a_{11} = r(1)/r(0) = \kappa_1 \tag{7.15}$$

where κ_1 is termed the reflection coefficient. Note that magnitude of κ_1 is less than 1 ($|\kappa_1|<1$) as $|r(1)|$ is less than $r(0)$. Now the prediction error for $m = 1$ becomes

$$E_1 = r(0) - \kappa_1 r(1) = r(0)[1-\kappa_1^2] = E_0[1-\kappa_1^2]. \tag{7.16}$$

One can easily show that the prediction error E_1 is smaller than E_0.

***m*=2:** When $m = 2$, equations (7.12) and (7.6) can be combined in a single matrix equation

$$\begin{bmatrix} r(0) & r(1) & r(2) \\ r(1) & r(0) & r(1) \\ r(2) & r(1) & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ -a_{12} \\ -a_{22} \end{bmatrix} = \begin{bmatrix} E_2 \\ 0 \\ 0 \end{bmatrix} \tag{7.17}$$

Assume that the solution can be found recursively as shown below.

$$\begin{bmatrix} 1 \\ -a_{12} \\ -a_{22} \end{bmatrix} = \begin{bmatrix} 1 \\ -a_{11} \\ 0 \end{bmatrix} - \kappa_2 \begin{bmatrix} 0 \\ -a_{11} \\ 1 \end{bmatrix} \tag{7.18}$$

where κ_2 is the reflection coefficient. The subscript 2 of a_{12} and a_{22} indicates that these are the second order linear prediction coefficients. When the prediction order $m = 1$, one can easily show that

$$\begin{bmatrix} r(0) & r(1) \\ r(1) & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ -a_{11} \end{bmatrix} = \begin{bmatrix} E_1 \\ 0 \end{bmatrix}.$$

Now equation (7.17) becomes

$$\begin{bmatrix} r(0) & r(1) & r(2) \\ r(1) & r(0) & r(1) \\ r(2) & r(1) & r(0) \end{bmatrix} \left\{ \begin{bmatrix} 1 \\ -a_{11} \\ 0 \end{bmatrix} - \kappa_2 \begin{bmatrix} 0 \\ -a_{11} \\ 1 \end{bmatrix} \right\} = \begin{bmatrix} E_1 \\ 0 \\ q_2 \end{bmatrix} - \kappa_2 \begin{bmatrix} q_2 \\ 0 \\ E_1 \end{bmatrix} = \begin{bmatrix} E_2 \\ 0 \\ 0 \end{bmatrix} \tag{7.20}$$

where

$$q_2 = r(2) - a_{11} r(1). \tag{7.21}$$

Because $q_2 - \kappa_2 E_1 = 0$ from equation (7.20), the reflection coefficient becomes

$$\kappa_2 = q_2 / E_1. \tag{7.22}$$

The new prediction error for $M = 2$ becomes

$$E_2 = E_1 - \kappa_2 q_2 = E_1[1 - \kappa_2^2]. \tag{7.23}$$

The linear prediction coefficients can be obtained using Equation (7.18) such that

$$\begin{aligned} a_{12} &= a_{11} - \kappa_2 a_{11} \\ a_{22} &= \kappa_2 \end{aligned} \tag{7.24}$$

***m*=3:** When $m = 3$, one can show that

$$\begin{aligned} q_3 &= r(3) - a_{12} r(2) - a_{22} r(1) \\ \kappa_3 &= q_3 / E_2 \\ E_3 &= E_2[1 - \kappa_3^2] \end{aligned} \tag{7.25}$$

with the following assumption.

$$\begin{bmatrix} 1 \\ -a_{13} \\ -a_{23} \\ -a_{33} \end{bmatrix} = \begin{bmatrix} 1 \\ -a_{12} \\ -a_{22} \\ 0 \end{bmatrix} - \kappa_3 \begin{bmatrix} 0 \\ -a_{22} \\ -a_{12} \\ 1 \end{bmatrix} \tag{7.26}$$

Now the linear coefficients can be obtained from (7.26).

$$\begin{aligned} a_{i3} &= a_{i2} - \kappa_3 a_{(3-i)2} \qquad \text{for } i = 1, 2. \\ a_{33} &= \kappa_3. \end{aligned} \tag{7.27}$$

Recursive Algorithm:

Now the recursive solution method for any prediction order M is described below.

Initial values:

$E_0 = r(0)$

$a_{11} = \kappa_1 = r(1)/E_0$

$E_1 = E_0(1-\kappa_1^2)$.

With $m \geq 2$, the following recursion is performed

(i) $q_m = r(m) - \sum_{i=1}^{m-1} a_{i(m-1)} r(m-i)$

(ii) $\kappa_m = \dfrac{q_m}{E_{(m-1)}}$

(iii) $a_{mm} = \kappa_m$

(iv) $a_{im} = a_{i(m-1)} - \kappa_m a_{(m-i)(m-1)}$ for $i = 1, \cdots, m-1$

(v) $E_m = E_{m-1}[1-\kappa_m^2]$.

(vi) If $m < M$, then increase m to $m+1$ and go to (i). If $m = M$, then stop.

In the recursion, there are $2m+1$ multiplications are involved for each m. Thus, the total number of multiplications to estimate prediction coefficients for the prediction order, M, becomes

$$\#\text{ multiplications} = \sum_{m=1}^{M}(2m+1) = M(M+2). \tag{7.28}$$

Readers may wonder what kind of prediction order is used in practice. When the sampling rate is 8 kHz, 4 kHz is the maximum frequency. There usually is one resonant peak per one kHz of bandwidth. That means there are 4 resonant peaks in speech signal. To fit 4 resonant peaks 8 poles are required. In addition a couple of extra poles may be necessary to take care of some zeros. Thus, the order of prediction, M, is usually chosen to be 10. Another reason is that the prediction error does not decrease much beyond $M = 10$.

7.4 Lattice Implementation of LPC Filters

Linear prediction coefficients are computed recursively using the Levinson-Durbin algorithm. The first order prediction coefficient a_{11} is the same as the reflection coefficient κ_1. The m^{th} order linear prediction coefficients are obtained from the $(m-1)^{\text{th}}$ order prediction coefficients and the reflection coefficient κ_m. Thus, M linear prediction coefficients are equivalent to M reflection coefficients. If reflection coefficients are given, the corresponding linear prediction coefficients can be obtained or vice versa. Quantization of reflection coefficients is easier because of the well-defined range of values that they take on. Note that the absolute value of reflection coefficients is never greater than one (see Prob. 7.2). This is the reason why reflection coefficients instead of linear prediction coefficients are often used to represent a vocal tract filter. In this section, linear predictive coding (LPC) filters are implemented in a lattice form using reflection coefficients.

The prediction error for the m^{th} order prediction is rewritten as

$$\varepsilon_m(n) = x(n) - \sum_{i=1}^{m} a_{im} x(n-i) \tag{7.28}$$

where $\varepsilon_m(n)$ indicates that this error is the forward prediction error. Let us assume that the backward linear prediction of $x(n–m)$ is made based on $x(n)$, $x(n–1)$, ⋯, and $x(n–m+1)$. The backward prediction error $\beta_m(n)$ is defined as follows.

$$\beta_m(n) = x(n-m) - \sum_{i=1}^{m} a_{im} x(n-m+i) \tag{7.29}$$

Now the $(m–1)^{\text{th}}$ forward prediction error is given by

$$\varepsilon_{m-1}(n) = x(n) - \sum_{i=1}^{m-1} a_{i(m-1)} x(n-i) \tag{7.30}$$

and the $(m–1)^{\text{th}}$ backward prediction error is

$$\beta_{m-1}(n) = x(n-m+1) - \sum_{i=1}^{m-1} a_{i(m-1)} x(n-m+1+i)\,. \tag{7.31}$$

Because the recursive formula for linear prediction coefficients is given by

$$a_{im} = a_{i(m-1)} - \kappa_m a_{(m-i)(m-1)} \qquad \text{for } i = 1, 2, \cdots, m-1 \tag{7.32a}$$

$$a_{mm} = \kappa_m, \tag{7.32b}$$

It can be shown that

$$\begin{aligned} \varepsilon_m(n) &= \varepsilon_{m-1}(n) - \kappa_m \beta_{m-1}(n-1) \\ \beta_m(n) &= \beta_{m-1}(n-1) - \kappa_m \varepsilon_{m-1}(n) \end{aligned} \tag{7.33}$$

for $m = 1, 2, \cdots, M$.
The initial values are given by

$$\varepsilon_0(n) = \beta_0(n) = x(n)\,. \tag{7.34}$$

The final value is given by

$$\varepsilon(n) = \varepsilon_M(n). \tag{7.35}$$

Note

Proof of Equation (7.33)

Let $A_m(z) = 1 - \sum_{i=1}^{m} a_{im} z^{-i}$ and $A_{m-1}(z) = 1 - \sum_{i=1}^{m-1} a_{i(m-1)} z^{-i}$.

From equations (7.32a) and (7.32b),

$$A_m(z) = A_{m-1}(z) - \kappa_m A_{m-1}(z^{-1}) z^{-m}.$$

Equations (7.28) and (7.29) respectively are expressed in terms of their z-transforms

$$\varepsilon_m(z) = A_m(z) X(z)$$

and

$$\beta_m(z) = A_m(z^{-1}) z^{-m} X(z).$$

Likewise from equations (7.30) and (7.31), the following are obtained.

$$\varepsilon_{m-1}(z) = A_{m-1}(z) X(z)$$

$$\beta_{m-1}(z) = A_{m-1}(z^{-1}) z^{-m+1} X(z)$$

Thus,

$$\begin{aligned} \varepsilon_m(z) &= \left[A_{m-1}(z) - \kappa_m A_{m-1}(z^{-1}) z^{-m} \right] X(z) \\ &= \varepsilon_{m-1}(z) - \kappa_m \beta_{m-1}(z) z^{-1} \end{aligned}$$

and

$$\begin{aligned} \beta_m(z) &= A_m(z^{-1}) z^{-m} X(z) \\ &= \left[A_{m-1}(z^{-1}) - \kappa_m A_{m-1}(z) z^{m} \right] z^{-m} X(z) \\ &= \beta_{m-1}(z) z^{-1} - \kappa_m \varepsilon_{m-1}(z) \end{aligned}$$

Q.E.D.

Thus, the analysis filter can be implemented as shown in Figure 7.8 where the input is the speech sequence and the output is the forward prediction error.

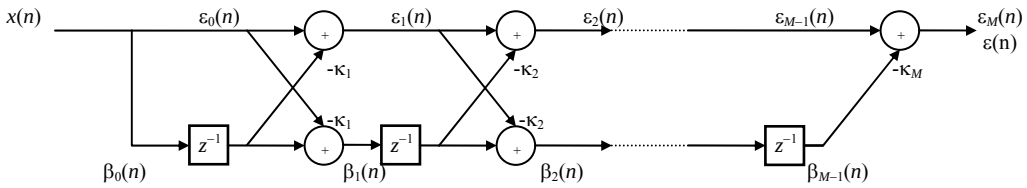


Figure 7.8 Lattice Implementation of the LPC analysis filter using reflection coefficients

From each frame of speech samples, *M* reflection coefficients are computed. Because important information about the vocal tract model is extracted in the form of reflection coefficients, the output of the LPC analysis filter using reflection coefficients will have less redundancy than the original speech. Thus, less number of bits is required to quantize this so-called *residual error*. This quantized residual error along with the quantized reflection coefficients are transmitted or stored. To play back, a lattice implementation of the LPC synthesis filter is required. In this case, the input is the residual error and the output is the reconstructed speech. By reversing all the arrows in the top part of the analysis filter, one can implement the synthesis filter as shown in Fig. 7.9.

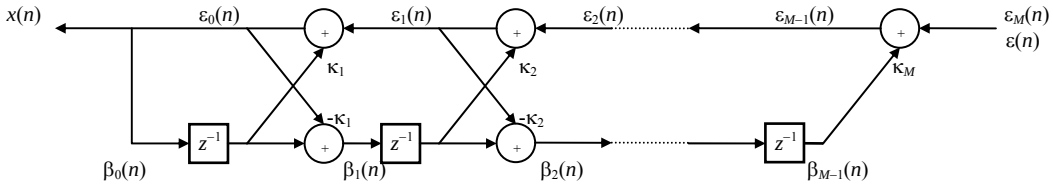


Figure 7.9 Lattice Implementation of the LPC synthesis filter using reflection coefficients

In the synthesis filter, the initial value is

$$\varepsilon_M(n) = \varepsilon(n), \tag{7.36}$$

the final values are

$$\varepsilon_0(n) = \beta_0(n) = x(n), \tag{7.37}$$

and

$$\varepsilon_{m-1}(n) = \varepsilon_m(n) + \kappa_m \beta_{m-1}(n-1)$$
$$\beta_m(n) = \beta_{m-1}(n-1) - \kappa_m \varepsilon_{m-1}(n) \tag{7.38}$$

for $m = M, M-1, M-2, \cdots, 2, 1$.

7.5 Line Spectral Frequencies (LSF)

Phonemes are recognized by their own resonant frequencies. Thus, maintaining original speech spectrum is very important after the quantization of parameters. Quantization of linear prediction coefficients or reflection coefficients will result in error in the speech spectrum. Problem is that error in each of linear prediction coefficients or reflection coefficients is not directly related to error in any of resonant frequencies. However, each resonant frequency in speech spectrum is directly related to corresponding line spectral frequencies (LSF). If there is error in a resonant frequency because of quantization, the error is localized.

Let $F_m(z)$ be the z-transform of the sequence $\{1, -a_{1m}, -a_{2m}, \cdots, -a_{mm}, 0\}$ such that

$$F_m(z) = 1 - a_{1m} z^{-1} - a_{2m} z^{-2} - \cdots - a_{mm} z^{-m} = A(z) \tag{7.39}$$

and $G_m(z)$ be the z-transform of the reversed sequence $\{0, -a_{mm}, -a_{(m-1)m}, \cdots, -a_{2m}, -a_{1m}, 1\}$ such that

$$G_m(z) = -a_{mm} z^{-1} - a_{(m-1)m} z^{-2} - \cdots - a_{2m} z^{-(m-1)} - a_{1m} z^{-m} + z^{-(m+1)}. \tag{7.40}$$

Then the Levinson-Durbin recursive formula for $m = 1, 2, \cdots, M$ becomes

$$F_m(z) = F_{m-1}(z) - \kappa_m G_{m-1}(z) \tag{7.41}$$

and

$$G_m(z) = z^{-1}[G_{m-1}(z) - \kappa_m F_{m-1}(z)] \tag{7.42}$$

where initial conditions $F_0(z) = 1$ and $G_0(z) = z^{-1}$.

Now using $F_M(z)$ and $G_M(z)$, two types of $F_{M+1}(z)$, $P(z)$ and $Q(z)$, can be constructed under the conditions $\kappa_{M+1} = -1$ and $\kappa_{M+1} = 1$, respectively. The condition $|\kappa_{M+1}| = 1$ is necessary so that the $(M+1)^{\text{th}}$ order polynomials $P(z)$ and $Q(z)$ have roots on the unit circle. If roots are on the unit circle, only the angles or normalized frequencies are needed to represent the roots of $P(z)$ and $Q(z)$.

Using equation (7.41), $P(z)$ and $Q(z)$ can be represented as

$$P(z) = F_M(z) + G_M(z) \tag{7.43}$$

and

$$Q(z) = F_M(z) - G_M(z). \tag{7.44}$$

Using equations (7.43) and (7.44), one obtains

$$F_M(z) = \frac{P(z) + Q(z)}{2}. \tag{7.45}$$

A pair of complex conjugate roots of $P(z)$ or $Q(z)$ will give the following expression if they are on the unit circle.

$$(1 - e^{j\theta_i} z^{-1})(1 - e^{-j\theta_i} z^{-1}) = 1 - 2z^{-1}\cos\theta_i + z^{-2} \tag{7.46}$$

It has been proved[1] that the roots of the two polynomials are located on the unit circle and interlaced if the original synthesis filter, $H(z) = 1/A(z) = 1/F_M(z)$, is stable. If M is even, $P(z)$ and $Q(z)$ are factored as

$$P(z) = (1 + z^{-1}) \prod_{i=1,3,\cdots,M-1} (1 - 2z^{-1}\cos\theta_i + z^{-2}) \tag{7.47}$$

and

$$Q(z) = (1 - z^{-1}) \prod_{i=2,4,\cdots,M} (1 - 2z^{-1}\cos\theta_i + z^{-2}) \tag{7.48}$$

The coefficients $\{\theta_i\}$ are referred to as the *line spectral frequencies* (LSF). The parameters $\{\theta_i\}$ are ordered as

$$0 < \theta_1 < \theta_2 < \cdots < \theta_{M-1} < \theta_M < \pi. \tag{7.49}$$

Odd-indexed $\{\theta_i\}$ and even-indexed $\{\theta_i\}$ are interlaced.

Using equation (7.45), the square of the magnitude response of the synthesis filter can be represented as

$$\begin{aligned} |H(\theta)|^2 &= \frac{1}{\left|F_M(\theta)\right|^2} \\ &= 2^2 |P(\theta) + Q(\theta)|^{-2} \\ &= 2^{-M} \left[\cos^2\frac{\theta}{2} \prod_{i=1,3,\cdots,M-1} (\cos\theta - \cos\theta_i)^2 + \sin^2\frac{\theta}{2} \prod_{i=2,4,\cdots,M} (\cos\theta - \cos\theta_i)^2 \right]^{-1} \end{aligned} \tag{7.50}$$

[1] F. K. Soong and B.-H. Juang, "Line Spectrum Pair and Speech Compression," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, San Diego, Calif., vol. 1, pp.1.10.1-4, 1984.

The first term inside the parentheses in Equation (7.50) approaches 0 when θ approaches π or one of the $\{\theta_i\}$ ($i = 1,3,\cdots,M-1$), and the second term approaches 0 when θ approaches 0 or one of the $\{\theta_i\}$ ($i = 2,4,\cdots,M$). Therefore, when two LSF parameters, θ_i and θ_j, are close together, the gain of $H(z)$ becomes large and resonance occurs. Strong resonance occurs when two or more θ_i's are concentrated. That is, the speech spectrum is directly related to LSF parameters $\{\theta_i\}$. Thus, if there is error in the parameters because of quantization, the error is localized. Typical magnitude plots of $P(z)$ and $Q(z)$ are shown in Fig. 7.10.

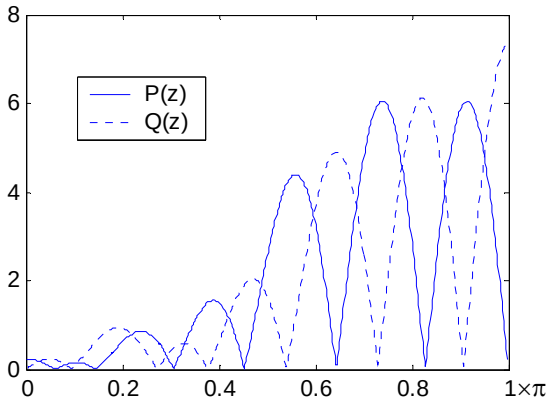


Fig.7.10 Magnitude plots of $P(z)$ and $Q(z)$. Prediction order M is chosen to be 12 in this case.

Roots of the polynomials $P(z)$ and $Q(z)$ can be obtained using a root finding technique via companion matrix or Newton's iteration method. Because at each LSF parameter the magnitude response of either $P(z)$ or $Q(z)$ is zero, LSF parameters can be found alternatively by the following procedure.

1. Find linear prediction coefficients.
2. Form $P(z)$ and $Q(z)$.
3. Estimate the magnitude response of $P(z)$ and $Q(z)$. FFT may be used for this.
4. Frequencies at which the local minima occur are LSF parameters.

The procedure for searching for the roots is largely reduced using the relationship $0 < \theta_1 < \theta_2 < \cdots < \theta_{M-1} < \theta_M < \pi$.

The magnitude spectrum based on the LPC parameters can be compared to the magnitude spectrum computed based on the LSF estimation. When the FFT length is 128, there usually is a large difference between the two spectra as shown in Fig. 7.11.

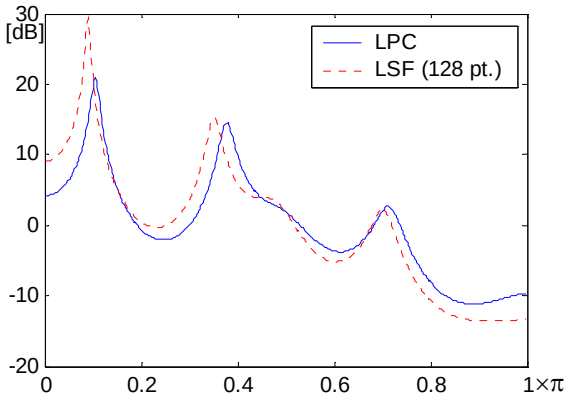


Fig.7.11. Magnitude spectra computed based on the LPC and the LSF estimation when FFT length is 128.

A typical difference between the magnitude spectra when 256-point FFT is used is given in Fig. 7.12.

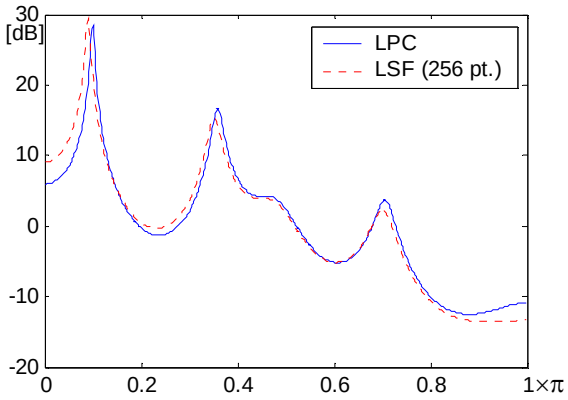


Fig.7.12 Magnitude spectra computed based on the LPC and the LSF estimation when FFT length is 256.

When 512-point FFT is used, the difference is negligible.

7.6 LSF Analysis-Synthesis System

In LSF speech synthesis, a digital filter $H(z)$ can be constructed using the LSF parameters $(\theta_1, \theta_2, \cdots, \theta_M)$. Since $H(z) = 1/F_M(z)$, this transfer function can be realized by inserting a filter having a transfer function of $F_M(z) - 1$ into a negative feedback path in the signal flow graph. Based on equations (7.45), (7.47), and (7.48), when P is even, one can show that

$$F_M(z)-1=\frac{1}{2}\left[\left(P(z)-1\right)+\left(Q(z)-1\right)\right] \tag{7.51}$$

$$=\frac{z^{-1}}{2}\left[\left(c_1+z^{-1}\right)+\sum_{\substack{i=3\\ i\,\text{odd}}}^{M-1}\left(c_i+z^{-1}\right)\prod_{\substack{j=1\\ j\,\text{odd}}}^{i-2}\left(1+c_j z^{-1}+z^{-2}\right)+\prod_{\substack{i=1\\ i\,\text{odd}}}^{M}\left(1+c_i z^{-1}+z^{-2}\right)\right]$$

$$+\frac{z^{-1}}{2}\left[\left(c_2+z^{-1}\right)+\sum_{\substack{i=4\\ i\,\text{even}}}^{M}\left(c_i+z^{-1}\right)\prod_{\substack{j=2\\ j\,\text{even}}}^{i-2}\left(1+c_j z^{-1}+z^{-2}\right)-\prod_{\substack{i=2\\ i\,\text{even}}}^{M}\left(1+c_i z^{-1}+z^{-2}\right)\right]$$

where

$c_i = -2\cos\theta_i$ for $i = 1, 2, \cdots, M$.

$F_M(z) - 1$ can thus be constructed by a pair of trunk circuit as shown in Figure 7.13. Each trunk circuit is a $M/2$-stage cascade connection of quadratic circuits: $1 - 2\cos\theta_i z^{-1} + z^{-2}$. The outputs at the middle of each stage on each trunk are successively summed up, and the outputs at the final stage are added or subtracted from the former value.

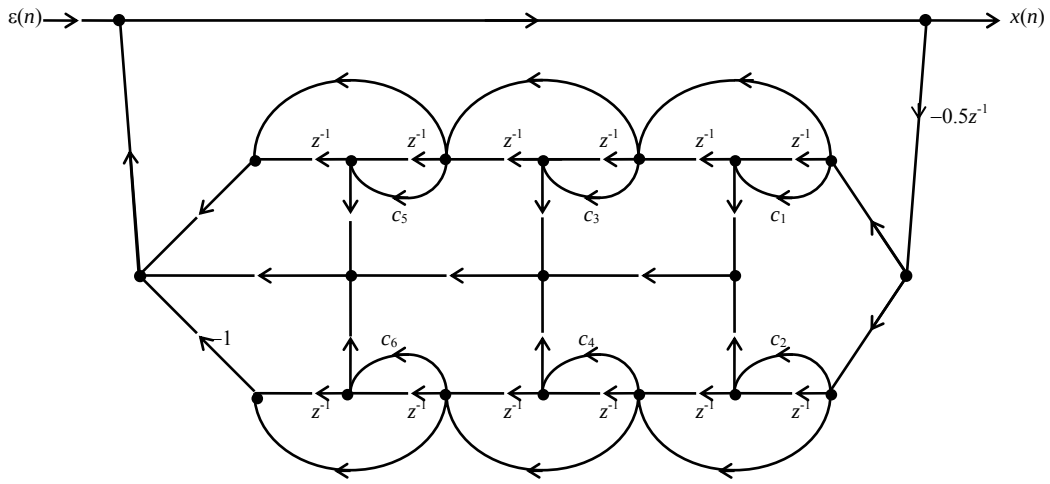


Figure 7.13 Signal flow graph of LSF synthesis filter (with $M = 6$).

MATLAB Example

```
x = wavread('ah.wav');
x = x(1:240);
a = lpc(x, 10);
```

```
poly2rc(a) % This is to convert lpc to reflection coefficients
poly2lsf(a) % Convert lpc to line spectral frequencies
```

7.7 Pitch Estimation

There are several ways to estimate the pitch of a voiced sound: autocorrelation method, average magnitude difference function and cepstrum.

1. Autocorrelation

The autocorrelation of a stationary sequence $x(n)$ is defined as

$$R_x(\tau) = \langle x(n)x(n+\tau)\rangle = \frac{1}{N}\sum_{n=0}^{N-1} x(n)x(n+\tau). \tag{7.52}$$

where τ is termed the lag. Auto means self or from one signal, and correlation means relation between two samples. An autocorrelation is the average correlation between two samples from one signal that are separated by τ samples. It should be noted that the upper limit in the summation will be less than $N-1$ when τ is positive, and the lower limit will be greater than 0 when τ is negative. Thus, the autocorrelation can be rewritten as

$$R_x(\tau) = \frac{1}{N}\sum_{n=0}^{N-1-|\tau|} x(n)x(n+|\tau|) \tag{7.53}$$

Because the number of items in the summation decreases as τ increases, the envelope of the autocorrelation decreases linearly as τ increases. In some cases, to prevent this tapering, autocorrelation is defined alternatively as

$$\hat{R}_x(\tau) = \frac{1}{N-|\tau|}\sum_{n=0}^{N-1-|\tau|} x(n)x(n+|\tau|). \tag{7.54}$$

MATLAB Example

```
plot(xcorr(x,'unbiased'))
```

2. Average Magnitude Difference Function (AMDF)

The average magnitude difference function uses the following property. Suppose that a signal $x(n)$ is periodic with period T. Then the difference between two samples

$$Diff(k) = x(n) - x(n+k) \tag{7.55}$$

will be zero for $k = 0, \pm T, \pm 2T$ and so on. Because a voiced sound is not exactly periodic, the short time average magnitude difference function (AMDF) is defined as

$$\text{AMDF}(k) = \frac{1}{N-k} \sum_{n=0}^{N-1-k} \left| x(n) - x(n+k) \right| \text{ for positive } k. \tag{7.56}$$

MATLAB Example

```
for k = 1:240,
      amdf(k) = 0;
      for n = 1:240-k+1,
            amdf(k) = amdf(k) + abs(x(n)-x(n+k-1));
      end
      amdf (k) = amdf(k)/(240-k+1);
end
plot(amdf)
```

3. Cepstrum

There are two kinds of cepstra: the real cepstrum and the complex cepstrum. Only the real cepstrum is explained here. Suppose that $x(n)$ is a speech signal. The magnitude spectrum $|X(k)|$ is obtained by computing the magnitude of the DFT of $x(n)$. The real cepstrum is defined as the inverse discrete Fourier transform of the logarithm of the magnitude response, i.e,

$$c(n) = \text{IDFT}\{\log|X(k)|\}.$$

where $c(n)$ is termed the cepstrum and log is the natural logarithm.

MATLAB Example

```
X = fft(x);
X = log(abs(X));
c = ifft(X);
plot(real(c))
% Four commands above are equivalent to the command below
plot(rceps(x))
```

7.8 Voiced/Unvoiced Detection

A simplified speech production system is shown in Fig. 7.14.

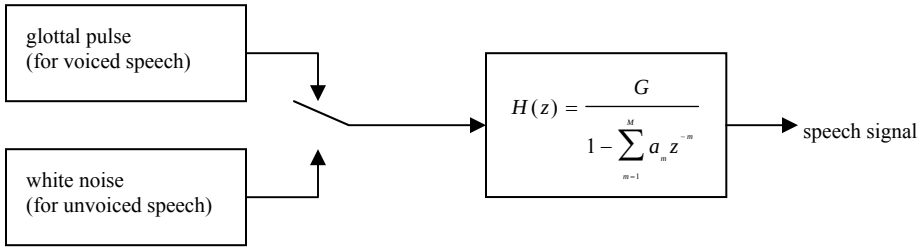


Figure 7.14 A simplified speech production system.

One important task is segmentation and labeling of each segment as voiced of unvoiced. To identify whether the speech segment is voiced or unvoiced speech, spectral flatness measure, energy, and zero crossing rate are most widely used.

1. Spectral Flatness Measure

One of the methods for detecting the voiced/unvoiced sections of speech is the spectral flatness measure. The spectral flatness makes use of the property that the spectrum of pure noise is expected to be flat. In other words, the spectrum of unvoiced section is flat and the spectrum of voiced section is less flat. The spectral flatness measure (SFM) is given by

$$SFM = \frac{G_m}{A_m}$$

where G_m is the geometric mean of the magnitude spectrum and is determined by multiplying all the spectral lines together and raising the final product to one over the total number of spectral lines. A_m is the arithmetic mean of the magnitude spectrum and is obtained by taking the sum of the spectral lines divided by the number of spectral lines.

$$SFM = \frac{\left(\prod_{k=0}^{N-1} X_j(k) \right)^{\frac{1}{N}}}{\frac{1}{N} \sum_{n=0}^{N-1} X_j(k)}$$

where $X_j(k)$ is the magnitude of the N-point DFT of the j^{th} frame of the speech signal. The spectral flatness measure ranges from 0.9 for a white noise to 0.1 for a voiced signal. The threshold is usually chosen to be 0.35 ~ 0.48.

MATLAB Example

```
X = abs(fft(x));  % x is 240-point long vowel sound
am = mean(X);
gm = 1;
for n=1:240,
      gm = gm*X(n);
end
gm = gm^(1/240);
sfm = gm/am

y = randn(240,1);  % y is 240-point long white noise
Y = abs(fft(y));
am = mean(Y);
gm = 1;
for n=1:240,
      gm = gm*Y(n);
end
gm = gm^(1/240);
sfm = gm/am
```

2. Energy and Zero-Crossing Rate

Energy of the *j*-th frame of the speech signal is calculated by the following:

$$E = \sum_{n=0}^{N-1} x_j^{\,2}(n)$$

where $x_j(n)$ is the *n*-th speech sample in the *j*-th frame. Usually the energy of a voiced speech frame is large than that of an unvoiced speech frame.

Zero-crossing rate is obtained by counting the sign changes (either from positive to negative or from negative to positive) in successive speech samples. The ZCR of the voiced sound is lower than the ZCR of the unvoiced sound.

MATLAB Assignment

1. Implement an LPC analysis filter. Choose $M = 10$ and find a new set of coefficients and the variance of the prediction error for every 30 ms-long frame. The input is your own speech signal of about 2 seconds long. Choose 8 kHz as the sampling frequency. Plot and listen to the output residual error sequence.

2. Implement an LPC synthesis filter. Use a new set of parameters for every 30 ms-long frame. The input is the normalized residual error sequence we obtained in part 1. Plot and listen to the reconstructed speech sequence.

3. Quantize the normalized residual error with four fixed quantization levels. Reconstruct the speech using this quantized residual error. Listen to the reconstructed speech.

4. Quantize the original speech signal with four fixed quantization levels. Listen to this quantized speech.

5. Compare the quality of the reconstructed speeches of 4 and 5.

Problems

7.1 Is the autocorrelation method accurate? Under what condition the solution to the autocorrelation method will be the true solution?

7.2 Show that $E_m \le E_{m-1}$ for any m. In other words, show that $|\kappa_m| < 1$.

7.3 Show equation (7.50).

7.4 Show equation (7.51).

Computer Assignment 7.1

1. From the first 240-point long speech sequence, find the linear prediction coefficients using the efficient Levinson-Durbin recursive algorithm. Use $M = 10$.

2. Plot the magnitude response of the linear prediction model you obtained in part 1.

3. Repeat parts 1 & 2 for the second 240-point long speech sequence.

Computer Assignment 7.2

1. Write a C program for a lattice implementation of the LPC analysis filter using equations (7.33) and (7.34) or Fig. 7.8. Use $M = 10$ and a new set of parameters every 30 ms (240 samples). The input is the speech signal. Plot the output residual error sequence.

2. Write a C program for a lattice implementation of the LPC synthesis filter using Fig. 7.9. Use $M = 10$ and a new set of parameters every 30 ms (240 samples). The input is the residual error sequence we obtained in part 1. Plot the reconstructed speech sequence.

Computer Assignment 7.3

1. Using the first 240 speech samples, find 10 LSF parameters via the FFT method (512 points).

2. Compute and plot the magnitude response of the vocal tract model using equation (7.50). Compare the result to the one you obtained in the previous assignment.

3. Construct an LSF analysis filter and compute the 240-point error sequence.

4. Construct an LSF synthesis filter and obtain the reconstructed speech.