8

Linear Predictive Coding of Speech

8.0 Introduction

One of the most powerful speech analysis techniques is the method of linear predictive analysis. This method has become the predominant technique for estimating the basic speech parameters, e.g., pitch, formants, spectra, vocal tract area functions, and for representing speech for low bit rate transmission or storage. The importance of this method lies both in its ability to provide extremely accurate estimates of the speech parameters, and in its relative speed of computation. In this chapter, we present a formulation of the ideas behind linear prediction, and discuss some of the issues which are involved in using it in practical speech applications.

The basic idea behind linear predictive analysis is that a speech sample can be approximated as a linear combination of past speech samples. By minimizing the sum of the squared differences (over a finite interval) between the actual speech samples and the linearly predicted ones, a unique set of predictor coefficients can be determined. (The predictor coefficients are the weighting coefficients used in the linear combination.)

The philosophy of linear prediction is intimately related to the basic speech synthesis model discussed in Chapter 3 in which it was shown that speech can be modelled as the output of a linear, time-varying system excited by either quasi-periodic pulses (during voiced speech), or random noise (during unvoiced speech). The linear prediction method provides a robust, reliable, and accurate method for estimating the parameters that characterize the linear, time-varying system.

Linear predictive techniques have already been discussed in the context of the waveform quantization methods of Chapter 5. There it was suggested that a linear predictor could be applied in a differential quantization scheme to reduce the bit rate of the digital representation of the speech waveform. In fact, the mathematical basis for an adaptive high order predictor used for DPCM waveform coding is identical to the analysis that we shall present in this chapter. In adaptive DPCM coding the emphasis is on finding a predictor that will reduce the variance of the difference signal so that quantization error can also be reduced. In this chapter we take a more general viewpoint and show how the basic linear prediction idea leads to a set of analysis techniques that can be used to estimate parameters of a speech model. This general set of linear predictive analysis techniques is often referred to as linear predictive coding or LPC.

The techniques and methods of linear prediction have been available in the engineering literature for a long time. The ideas of linear prediction have been in use in the areas of control, and information theory under the names of system estimation and system identification. The term system identification is particularly descriptive of LPC methods in that once the predictor coefficients have been obtained, the system has been uniquely identified to the extent that it can be modelled as an all-pole linear system.

As applied to speech processing, the term linear prediction refers to a variety of essentially equivalent formulations of the problem of modelling the speech waveform [1-18]. The differences among these formulations are often those of philosophy or way of viewing the problem. In other cases the differences concern the details of the computations used to obtain the predictor coefficients. Thus as applied to speech, the various (often equivalent) formulations of linear prediction analysis have been:

1. the covariance method [3]
2. the autocorrelation formulation [1,2,9]
3. the lattice method [11,12]
4. the inverse filter formulation [1]
5. the spectral estimation formulation [12]
6. the maximum likelihood formulation [4,6]
7. the inner product formulation [1]

In this chapter we will examine in detail the similarities and differences among only the first three basic methods of analysis listed above, since all the other formulations are equivalent to one of these three.

The importance of linear prediction lies in the accuracy with which the basic model applies to speech. Thus a major part of this chapter is devoted to a discussion of how a variety of speech parameters can be reliably estimated using linear prediction methods. Furthermore some typical examples of speech applications which rely primarily on linear predictive analysis are discussed here, and in Chapter 9, to show the wide range of problems to which LPC has been successfully applied.

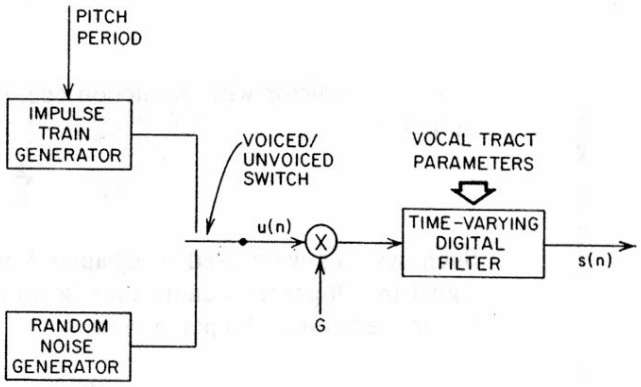


Fig. 8.1 Block diagram of simplified model for speech production.

8.1 Basic Principles of Linear Predictive Analysis

Throughout this book we have repeatedly referred to the basic discrete-time model for speech production that was developed in Chapter 3. The particular form of this model that is appropriate for the discussion of linear predictive analysis is depicted in Fig. 8.1. In this case, the composite spectrum effects of radiation, vocal tract, and glottal excitation are represented by a time-varying digital filter whose steady-state system function is of the form

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{8.1}$$

This system is excited by an impulse train for voiced speech or a random noise sequence for unvoiced speech. Thus, the parameters of this model are: voiced/unvoiced classification, pitch period for voiced speech, gain parameter G, and the coefficients $\{a_k\}$ of the digital filter. These parameters, of course, all vary slowly with time.

The pitch period and voiced/unvoiced classification can be estimated using one of the many methods already discussed in this book or by methods based on linear predictive analysis to be discussed later in this chapter. As discussed in Chapter 3, this simplified all-pole model is a natural representation of non-nasal voiced sounds, but for nasals and fricative sounds, the detailed acoustic theory calls for both poles and zeros in the vocal tract transfer function. We shall see, however, that if the order p is high enough, the all-pole model provides a good representation for almost all the sounds of speech. The major advantage of this model is that the gain parameter, G, and the filter coefficients $\{a_k\}$ can be estimated in a very straightforward and computationally efficient manner by the method of linear predictive analysis.

For the system of Fig. 8.1, the speech samples $s(n)$ are related to the excitation $u(n)$ by the simple difference equation

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n) \tag{8.2}$$

A linear predictor with prediction coefficients, α_k is defined as a system whose output is

$$\tilde{s}(n) = \sum_{k=1}^{p} \alpha_k s(n-k) \tag{8.3}$$

Such systems were used in Chapter 5 to reduce the variance of the difference signal in differential quantization schemes. The system function of a p^{th} order linear predictor is the polynomial

$$P(z) = \sum_{k=1}^{p} \alpha_k z^{-k} \tag{8.4}$$

The prediction error, $e(n)$, is defined as

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{p} \alpha_k s(n-k) \tag{8.5}$$

From Eq. (8.5) it can be seen that the prediction error sequence is the output of a system whose transfer function is

$$A(z) = 1 - \sum_{k=1}^{p} \alpha_k z^{-k} \tag{8.6}$$

It can be seen by comparing Eqs. (8.2) and (8.5) that if the speech signal obeys the model of Eq. (8.2) exactly, and if $\alpha_k = a_k$, then $e(n) = Gu(n)$. Thus, the *prediction error filter*, $A(z)$, will be an *inverse filter* for the system, $H(z)$, of Eq. (8.1), i.e.,

$$H(z) = \frac{G}{A(z)} \tag{8.7}$$

The basic problem of linear prediction analysis is to determine a set of predictor coefficients $\{\alpha_k\}$ directly from the speech signal in such a manner as to obtain a good estimate of the spectral properties of the speech signal through the use of Eq. (8.7). Because of the time-varying nature of the speech signal the predictor coefficients must be estimated from short segments of the speech signal. The basic approach is to find a set of predictor coefficients that will minimize the mean-squared prediction error over a short segment of the speech waveform. The resulting parameters are then *assumed* to be the parameters of the system function, $H(z)$, in the model for speech production.

That this approach will lead to useful results may not be immediately obvious, but it can be justified in several ways. First, recall that if $\alpha_k = a_k$, then $e(n) = Gu(n)$. For voiced speech this means that $e(n)$ would consist of a train of impulses; i.e., $e(n)$ would be small most of the time. Thus, finding α_k's that minimize prediction error seems consistent with this observation. A second motivation for this approach follows from the fact that if a signal is generated by Eq. (8.2) with non-time-varying coefficients and excited either by a single impulse or by a stationary white noise input, then it can be shown that the predictor coefficients that result from minimizing the mean squared prediction error (over all time) are identical to the coefficients of Eq. (8.2). A third

very pragmatic justification for using the minimum mean-squared prediction error as a basis for estimating the model parameters is that this approach leads to a set of linear equations that can be efficiently solved to obtain the predictor parameters. More importantly the resulting parameters comprise a very useful and accurate representation of the speech signal as we shall see in this chapter.

The short-time average prediction error is defined as

$$E_n = \sum_m e_n^2(m) \tag{8.8}$$

$$= \sum_m (s_n(m) - \tilde{s}_n(m))^2 \tag{8.9}$$

$$= \sum_m \left[s_n(m) - \sum_{k=1}^{p} \alpha_k s_n(m-k) \right]^2 \tag{8.10}$$

where $s_n(m)$ is a segment of speech that has been selected in the vicinity of sample n, i.e.,

$$s_n(m) = s(m+n) \tag{8.11}$$

The range of summation in Eqs. (8.8)-(8.10) is temporarily left unspecified, but since we wish to develop a short-time analysis technique, the sum will always be over a finite interval. Also note that to obtain an average we should divide by the length of the speech segment. However, this constant is irrevelant to the set of linear equations that we will obtain and therefore is omitted. We can find the values of α_k that minimize E_n in Eq. (8.10) by setting $\partial E_n/\partial \alpha_i = 0$, $i=1,2, \ldots, p$, thereby obtaining the equations

$$\sum_m s_n(m-i)s_n(m) = \sum_{k=1}^{p} \hat{\alpha}_k \sum_m s_n(m-i)s_n(m-k) \qquad 1 \leqslant i \leqslant p \tag{8.12}$$

where $\hat{\alpha}_k$ are the values of α_k that minimize E_n. (Since $\hat{\alpha}_k$ is unique, we will drop the caret and use the notation α_k to denote the values that minimize E_n.) If we define

$$\phi_n(i,k) = \sum_m s_n(m-i)s_n(m-k) \tag{8.13}$$

then Eq. (8.12) can be written more compactly as

$$\sum_{k=1}^{p} \alpha_k \phi_n(i,k) = \phi_n(i,0) \qquad i=1,2, \ldots ,p \tag{8.14}$$

This set of p equations in p unknowns can be solved in an efficient manner for the unknown predictor coefficients $\{\alpha_k\}$ that minimize the average squared prediction error for the segment $s_n(m)$.[1] Using Eqs. (8.10) and (8.12), the minimum mean-squared prediction error can be shown to be

[1]It is clear that the α_k's are functions of n (the time index at which they are estimated) although this dependence will not be explicitly shown. We shall also find it advantageous to drop the subscripts n on E_n, $s_n(m)$, and $\phi_n(i,k)$ when no confusion will result.

$$E_n = \sum_m s_n^2(m) - \sum_{k=1}^{p} \alpha_k \sum_m s_n(m)s_n(m-k) \tag{8.15}$$

and using Eq. (8.14) we can express E_n as

$$E_n = \phi_n(0,0) - \sum_{k=1}^{p} \alpha_k \phi_n(0,k) \tag{8.16}$$

Thus the total minimum error consists of a fixed component, and a component which depends on the predictor coefficients.

To solve for the optimum predictor coefficients, we must first compute the quantities $\phi_n(i,k)$ for $1 \leqslant i \leqslant p$ and $0 \leqslant k \leqslant p$. Once this is done we only have to solve Eq. (8.14) to obtain the α_k's. Thus, in principle, linear prediction analysis is very straightforward. However, the details of the computation of $\phi_n(i,k)$ and the subsequent solution of the equations are somewhat intricate and further discussion is required.

So far we have not explicitly indicated the limits on the sums in Eqs. (8.8)-(8.10) and in Eq. (8.12); however it should be emphasized that the limits on the sum in Eq. (8.12) are identical to the limits assumed for the mean squared prediction error in Eqs. (8.8)-(8.10). As we have stated, if we wish to develop a short-time analysis procedure, the limits must be over a finite interval. There are two basic approaches to this question, and we shall see below that two methods for linear predictive analysis emerge out of a consideration of the limits of summation and the definition of the waveform segment $s_n(m)$.

8.1.1 The autocorrelation method [1,2,5]

One approach to determining the limits on the sums in Eqs. (8.8)-(8.10) and Eq. (8.12) is to assume that the waveform segment, $s_n(m)$, is identically zero outside the interval $0 \leqslant m \leqslant N-1$. This can be conveniently expressed as

$$s_n(m) = s(m+n)\,w(m) \tag{8.17}$$

where $w(m)$ is a finite length window (e.g. a Hamming window) that is identically zero outside the interval $0 \leqslant m \leqslant N-1$.

The effect of this assumption on the question of limits of summation for the expressions for E_n can be seen by considering Eq. (8.5). Clearly, if $s_n(m)$ is nonzero only for $0 \leqslant m \leqslant N-1$, then the corresponding prediction error, $e_n(m)$, for a p^{th} order predictor will be nonzero over the interval $0 \leqslant m \leqslant N-1+p$. Thus, for this case E_n is properly expressed as

$$E_n = \sum_{m=0}^{N+p-1} e_n^2(m) \tag{8.18}$$

Alternatively, we could have simply indicated that the sum should be over all nonzero values by summing from $-\infty$ to $+\infty$ [2].

Returning to Eq. (8.5), it can be seen that the prediction error is likely to be large at the beginning of the interval (specifically $0 \leqslant m \leqslant p-1$) because we

are trying to predict the signal from samples that have arbitrarily been set to zero. Likewise the error can be large at the end of the interval (specifically $N \leqslant m \leqslant N+p-1$) because we are trying to predict zero from samples that are nonzero. For this reason, a window which tapers the segment, $s_n(m)$, to zero is generally used for $w(m)$ in Eq. (8.17).

The limits on the expression for $\phi_n(i,k)$ in Eq. (8.13) are identical to those of Eq. (8.18). However, because $s_n(m)$ is identically zero outside the interval $0 \leqslant m \leqslant N-1$, it is simple to show that

$$\phi_n(i,k) = \sum_{m=0}^{N+p-1} s_n(m-i)s_n(m-k) \qquad \begin{matrix} 1 \leqslant i \leqslant p \\ 0 \leqslant k \leqslant p \end{matrix} \tag{8.19a}$$

can be expressed as

$$\phi_n(i,k) = \sum_{m=0}^{N-1-(i-k)} s_n(m)s_n(m+i-k) \qquad \begin{matrix} 1 \leqslant i \leqslant p \\ 0 \leqslant k \leqslant p \end{matrix} \tag{8.19b}$$

Furthermore it can be seen that in this case $\phi_n(i,k)$ is identical to the short-time autocorrelation function of Eq. (4.30) evaluated for $(i-k)$. That is

$$\phi_n(i,k) = R_n(i-k) \tag{8.20}$$

where

$$R_n(k) = \sum_{m=0}^{N-1-k} s_n(m)s_n(m+k) \tag{8.21}$$

The computation of $R_n(k)$ is covered in detail in Section 4.6 and thus we shall not consider such details here. Since $R_n(k)$ is an even function, it follows that

$$\phi_n(i,k) = R_n(|i-k|) \qquad \begin{matrix} i = 1,2,\ldots,p \\ k = 0,1,\ldots,p \end{matrix} \tag{8.22}$$

Therefore Eq. (8.14) can be expressed as

$$\sum_{k=1}^{p} \alpha_k R_n(|i-k|) = R_n(i) \qquad 1 \leqslant i \leqslant p \tag{8.23}$$

Similarly, the minimum mean squared prediction error of Eq. (8.16) takes the form

$$E_n = R_n(0) - \sum_{k=1}^{p} \alpha_k R_n(k) \tag{8.24}$$

The set of equations given by Eqs. (8.23) can be expressed in matrix form as

$$\begin{bmatrix} R_n(0) & R_n(1) & R_n(2) & \cdots & R_n(p-1) \\ R_n(1) & R_n(0) & R_n(1) & \cdots & R_n(p-2) \\ R_n(2) & R_n(1) & R_n(0) & \cdots & R_n(p-3) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ R_n(p-1) & R_n(p-2) & R_n(p-3) & \cdots & R_n(0) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \cdots \\ \cdots \\ \alpha_p \end{bmatrix} = \begin{bmatrix} R_n(1) \\ R_n(2) \\ R_n(3) \\ \cdots \\ \cdots \\ R_n(p) \end{bmatrix} \tag{8.25}$$

The $p \times p$ matrix of autocorrelation values is a Toeplitz matrix; i.e., it is symmetric and all the elements along a given diagonal are equal. This special property will be exploited in Section 8.3 to obtain an efficient algorithm for the solution of Eq. (8.23).

8.1.2 The covariance method [3]

The second basic approach to defining the speech segment $s_n(m)$ and the limits on the sums is to fix the interval over which the mean-squared error is computed and then consider the effect on the computation of $\phi_n(i,k)$. That is, if we define

$$E_n = \sum_{m=0}^{N-1} e_n^2(m) \tag{8.26}$$

then $\phi_n(i,k)$ becomes

$$\phi_n(i,k) = \sum_{m=0}^{N-1} s_n(m-i)s_n(m-k) \qquad \begin{matrix} 1 \leqslant i \leqslant p \\ 0 \leqslant k \leqslant p \end{matrix} \tag{8.27}$$

In this case, if we change the index of summation we can express $\phi_n(i,k)$ as either

$$\phi_n(i,k) = \sum_{m=-i}^{N-i-1} s_n(m)s_n(m+i-k) \qquad \begin{matrix} 1 \leqslant i \leqslant p \\ 0 \leqslant k \leqslant p \end{matrix} \tag{8.28a}$$

or

$$\phi_n(i,k) = \sum_{m=-k}^{N-k-1} s_n(m)s_n(m+k-i) \qquad \begin{matrix} 1 \leqslant i \leqslant p \\ 0 \leqslant k \leqslant p \end{matrix} \tag{8.28b}$$

Although the equations look very similar to Eq. (8.19b), we see that the limits of summation are not the same. Equations (8.28) call for values of $s_n(m)$ outside the interval $0 \leqslant m \leqslant N-1$. Indeed, to evaluate $\phi_n(i,k)$ for all of the required values of i and k requires that we use values of $s_n(m)$ in the interval $-p \leqslant m \leqslant N-1$. If we are to be consistent with the limits on E_n in Eq. (8.26) then we have no choice but to supply the required values. In this case it does not make sense to taper the segment of speech to zero at the ends as in the autocorrelation method since the necessary values are made available from outside the interval $0 \leqslant m \leqslant N-1$. Clearly, this approach is very similar to what was called the modified autocorrelation function in Chapter 4. As pointed out in Section 4.6, this approach leads to a function which is not a true autocorrelation function, but rather, the cross-correlation between two very similar, but not identical, finite length segments of the speech wave. Although the differences between Eq. (8.28) and Eq. (8.19b) appear to be minor computational details, the set of equations

$$\sum_{k=1}^{p} \alpha_k \phi_n(i,k) = \phi_n(i,0) \qquad i = 1,2,\ldots,p \tag{8.29a}$$

has significantly different properties that strongly affect the method of solution

and the properties of the resulting optimum predictor. In matrix form these equations become

$$\begin{bmatrix} \phi_n(1,1) & \phi_n(1,2) & \phi_n(1,3) & \cdots & \phi_n(1,p) \\ \phi_n(2,1) & \phi_n(2,2) & \phi_n(2,3) & \cdots & \phi_n(2,p) \\ \phi_n(3,1) & \phi_n(3,2) & \phi_n(3,3) & \cdots & \phi_n(3,p) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \phi_n(p,1) & \phi_n(p,2) & \phi_n(p,3) & \cdots & \phi_n(p,p) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \cdots \\ \cdots \\ \alpha_p \end{bmatrix} = \begin{bmatrix} \phi_n(1,0) \\ \phi_n(2,0) \\ \phi_n(3,0) \\ \cdots \\ \cdots \\ \phi_n(p,0) \end{bmatrix} \tag{8.29b}$$

In this case, since $\phi_n(i,k) = \phi_n(k,i)$ (see Eq. (8.28)), the $p \times p$ matrix of correlation-like values is symmetric but *not* Toeplitz. Indeed, it can be seen that the diagonal elements are related by the equation

$$\phi_n(i+1,k+1) = \phi_n(i,k) + s_n(-i-1)s_n(-k-1) - s_n(N-1-i)s_n(N-1-k) \tag{8.30}$$

The method of analysis based upon this method of computation of $\phi_n(i,k)$ has come to be known as the *covariance method* because the matrix of values $\{\phi_n(i,k)\}$ has the properties of a covariance matrix [5].[2]

8.1.3 Summary

It has been shown that by using different definitions of the segments of the signal to be analyzed, two distinct sets of analysis equations can be obtained. For the autocorrelation method, the signal is windowed by an N-point window, and the quantities $\phi_n(i,k)$ are obtained using a short-time autocorrelation function. The resulting matrix of correlations is Toeplitz leading to one type of solution for the predictor coefficients. For the covariance method, the signal is assumed to be known for the set of values $-p \leqslant n \leqslant N-1$. Outside this interval no assumptions need be made about the signal, since these are the only values needed in the computation. The resulting matrix of correlations in this case is symmetric but not Toeplitz. The result is that the two methods of computing the correlations lead to different methods of solution of the analysis equations and to sets of predictor coefficients with somewhat different properties.

In later sections we will compare and contrast computational details and results for both these techniques as well as for another method yet to be discussed. First, however, we will show how the gain, G, in Fig. 8.1, can be determined from the prediction error expression.

8.2 Computation of the Gain for the Model [2]

It is reasonable to expect that the gain, G, could be determined by matching the

[2]This terminology, which is firmly entrenched, is somewhat confusing since the term covariance usually refers to the correlation of a signal with its mean removed.

energy in the signal with the energy of the linearly predicted samples. This indeed is true when appropriate assumptions are made about the excitation signal to the LPC system.

It is possible to relate the gain constant G to the excitation signal and the error in prediction by referring back to Eqs. (8.2) and (8.5).[3] The excitation signal, $Gu(n)$, can be expressed as

$$Gu(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \tag{8.31a}$$

whereas the prediction error signal $e(n)$ is expressed as

$$e(n) = s(n) - \sum_{k=1}^{p} \alpha_k s(n-k) \tag{8.31b}$$

In the case where $a_k = \alpha_k$, i.e., the actual predictor coefficients, and those of the model are identical, then

$$e(n) = Gu(n) \tag{8.32}$$

i.e., the input signal is proportional to the error signal with the constant of proportionality being the gain constant, G. A detailed discussion of the properties of the prediction error signal is given in Section 8.5.

Since Eq. (8.32) is only approximate (i.e., it is valid to the extent that the ideal and the actual linear prediction parameters are identical) it is generally not possible to solve for G in a reliable way directly from the error signal itself. Instead the more reasonable assumption is made that the energy in the error signal is equal to the energy in the excitation input, i.e.,

$$G^2 \sum_{m=0}^{N-1} u^2(m) = \sum_{m=0}^{N-1} e^2(m) = E_n \tag{8.33}$$

At this point we must make some assumptions about $u(n)$ so as to be able to relate G to the known quantities, e.g., the α_k's and the correlation coefficients. There are two cases of interest for the excitation. For voiced speech it is reasonable to assume $u(n) = \delta(n)$, i.e., the excitation is a unit sample at $n = 0$.[4] For this assumption to be valid requires that the effects of the glottal pulse shape used in the actual excitation for voiced speech be lumped together with the vocal tract transfer function, and therefore both of these effects are essentially modelled by the time-varying linear predictor. This requires that the predictor order, p, be large enough to account for both the vocal tract and glottal pulse effects. We will discuss the choice of predictor order in a later section. For unvoiced speech it is most reasonable to assume that $u(n)$ is a zero mean, unity variance, stationary, white noise process.

Based on these assumptions we can now determine the gain constant G by utilizing Eq. (8.33). For voiced speech, we have as input $G\delta(n)$. If we call the

[3]Note that the gain is also a function of time.

[4]Note that for this assumption to be valid requires that the analysis interval be about the same length as a pitch period.

resulting output for this particular input $h(n)$ (since it is actually the impulse response of the system with transfer function $H(z)$ as in Eq. (8.1)) we get the relation

$$h(n) = \sum_{k=1}^{p} \alpha_k h(n-k) + G\delta(n) \tag{8.34}$$

It is readily shown [Problem 8.1] that the autocorrelation function of $h(n)$, defined as

$$\tilde{R}(m) = \sum_{n=0}^{\infty} h(n)h(m+n) \tag{8.35}$$

satisfies the relations

$$\tilde{R}(m) = \sum_{k=1}^{p} \alpha_k \tilde{R}(|m-k|) \qquad m=1,2,\ldots,p \tag{8.36a}$$

and

$$\tilde{R}(0) = \sum_{k=1}^{p} \alpha_k \tilde{R}(k) + G^2 \tag{8.36b}$$

Since Eqs. (8.36) are identical to Eqs. (8.23) it follows that

$$\tilde{R}(m) = R_n(m) \qquad 1 \leqslant m \leqslant p \tag{8.37}$$

Since the total energies in the signal ($R(0)$) and the impulse response ($\tilde{R}(0)$) must be equal we can use Eqs. (8.24) , (8.33) and (8.36b) to obtain

$$G^2 = R_n(0) - \sum_{k=1}^{p} \alpha_k R_n(k) = E_n \tag{8.38}$$

It is interesting to note that Eq. (8.37) and the requirement that the energy of the impulse response be equal to the energy of the signal together require that the first $p+1$ coefficients of the autocorrelation function of the impulse response of the model are identical to the first $p+1$ coefficients of the autocorrelation function of the speech signal.

For the case of unvoiced speech, the correlations are defined as statistical averages. It is assumed that the input is white noise with zero mean and unity variance; i.e.,

$$E[u(n)u(n-m)] = \delta(m) \tag{8.39}$$

If we excite the system with the random input $Gu(n)$ and call the output $g(n)$ then

$$g(n) = \sum_{k=1}^{p} \alpha_k g(n-k) + Gu(n) \tag{8.40}$$

If we now let $\tilde{R}(m)$ denote the autocorrelation function of $g(n)$, then

$$\tilde{R}(m) = E[g(n)g(n-m)] = \sum_{k=1}^{p} \alpha_k E[g(n-k)g(n-m)] + E[Gu(n)g(n-m)]$$

$$= \sum_{k=1}^{p} \alpha_k \tilde{R}(m-k) \qquad m \neq 0 \tag{8.41}$$

since $E[u(n)g(n-m)]=0$ for $m > 0$ because $u(n)$ is uncorrelated with any signal prior to $u(n)$. For $m = 0$ we get

$$\tilde{R}(0) = \sum_{k=1}^{p} \alpha_k \tilde{R}(k) + GE[u(n)g(n)]$$

$$= \sum_{k=1}^{p} \alpha_k \tilde{R}(k) + G^2 \tag{8.42}$$

since $E[u(n)g(n)] = E[u(n)(Gu(n) + \textit{terms prior to } n)] = G$. Since the energy in the response to $Gu(n)$ must equal the energy in the signal, we get

$$\tilde{R}(m) = R_n(m) \quad 0 \leqslant m \leqslant p \tag{8.43}$$

or

$$G^2 = R_n(0) - \sum_{k=1}^{p} \alpha_k R_n(k) \tag{8.44}$$

as was the case for the impulse excitation for voiced speech.

8.3 Solution of the LPC Equations

In order to effectively implement a linear predictive analysis system, it is necessary to solve the linear equations in an efficient manner. Although a variety of techniques can be applied to solve a system of p linear equations in p unknowns, these techniques are not equally efficient. Because of the special properties of the coefficient matrices it is possible to solve the equations much more efficiently than is possible in general. In this section we will discuss in detail two methods for obtaining the predictor coefficients, and then we will compare and contrast several properties of these solutions.

8.3.1 Cholesky decomposition solution for the covariance method [3]

For the covariance method, the set of equations which must be solved is of the form:

$$\sum_{k=1}^{p} \alpha_k \phi_n(i,k) = \phi_n(i,0) \qquad i=1,2,\ldots,p \tag{8.45}$$

or in matrix notation

$$\boldsymbol{\Phi}\boldsymbol{\alpha} = \boldsymbol{\psi} \tag{8.46}$$

where $\boldsymbol{\Phi}$ is a positive definite symmetric matrix with $(i,j)^{th}$ element $\phi_n(i,j)$, and $\boldsymbol{\alpha}$ and $\boldsymbol{\psi}$ are column vectors with elements α_i, and $\phi_n(i,0)$ respectively. The system of equations given by Eq. (8.45) can be solved in an efficient manner since the matrix $\boldsymbol{\Phi}$ is a symmetric, positive definite matrix. The resulting method of solution is called the Cholesky decomposition (or sometimes it is

called the square root method) [3]. For this method the matrix $\mathbf{\Phi}$ is expressed in the form

$$\mathbf{\Phi} = \mathbf{V}\mathbf{D}\mathbf{V}^t \tag{8.47}$$

where $\mathbf{V}$ is a lower triangular matrix (whose main diagonal elements are all 1's), and $\mathbf{D}$ is a diagonal matrix. The superscript t denotes matrix transpose. The elements of the matrices $\mathbf{V}$ and $\mathbf{D}$ are readily determined from Eq. (8.47) by solving for the $(i,j)^{th}$ element of both sides of Eq. (8.47) giving

$$\phi_n(i,j) = \sum_{k=1}^{j} V_{ik} d_k V_{jk} \qquad 1 \leqslant j \leqslant i-1 \tag{8.48}$$

or

$$V_{ij} d_j = \phi_n(i,j) - \sum_{k=1}^{j-1} V_{ik} d_k V_{jk} \qquad 1 \leqslant j \leqslant i-1 \tag{8.49}$$

and, for the diagonal elements

$$\phi_n(i,i) = \sum_{k=1}^{i} V_{ik} d_k V_{ik} \tag{8.50}$$

or

$$d_i = \phi_n(i,i) - \sum_{k=1}^{i-1} V_{ik}^2 d_k \qquad i \geqslant 2 \tag{8.51}$$

with

$$d_1 = \phi_n(1,1) \tag{8.52}$$

To illustrate the use of Eqs. (8.47)-(8.52) consider an example with $p = 4$, and matrix elements $\phi_n(i,j) = \phi_{ij}$. Equation (8.47) is thus of the form

$$\begin{bmatrix} \phi_{11} & \phi_{21} & \phi_{31} & \phi_{41} \\ \phi_{21} & \phi_{22} & \phi_{32} & \phi_{42} \\ \phi_{31} & \phi_{32} & \phi_{33} & \phi_{43} \\ \phi_{41} & \phi_{42} & \phi_{43} & \phi_{44} \end{bmatrix} =$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ V_{21} & 1 & 0 & 0 \\ V_{31} & V_{32} & 1 & 0 \\ V_{41} & V_{42} & V_{43} & 1 \end{bmatrix} \begin{bmatrix} d_1 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 \\ 0 & 0 & d_3 & 0 \\ 0 & 0 & 0 & d_4 \end{bmatrix} \begin{bmatrix} 1 & V_{21} & V_{31} & V_{41} \\ 0 & 1 & V_{32} & V_{42} \\ 0 & 0 & 1 & V_{43} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To solve for d_1 to d_4, and the V_{ij}'s we begin with Eq. (8.52) for $i = 1$ giving

$$d_1 = \phi_{11}$$

Using Eq. (8.49) for $i = 2,3,4$ we solve for V_{21}, V_{31}, and V_{41} as

$$V_{21} d_1 = \phi_{21} \quad , \quad V_{31} d_1 = \phi_{31} \quad , \quad V_{41} d_1 = \phi_{41}$$

$$V_{21} = \phi_{21}/d_1 \, , \quad V_{31} = \phi_{31}/d_1 \, , \quad V_{41} = \phi_{41}/d_1$$

Using Eq. (8.51) for $i = 2$ gives

$$d_2 = \phi_{22} - V_{21}^2 d_1$$

Using Eq. (8.49) for $i = 3$ and 4 gives

$$V_{32} d_2 = \phi_{32} - V_{31} d_1 V_{21}$$
$$V_{42} d_2 = \phi_{42} - V_{41} d_1 V_{21}$$

or

$$V_{32} = (\phi_{32} - V_{31} d_1 V_{21})/d_2$$
$$V_{42} = (\phi_{42} - V_{41} d_1 V_{21})/d_2$$

Equation (8.51) is now used for $i = 3$ to solve for d_3, then Eq. (8.49) is used for $i = 4$ to solve for V_{43}, and finally Eq. (8.51) is used for $i = 4$ to solve for d_4.

Once the matrices **V** and **D** have been determined, it is relatively simple to solve for the column vector $\boldsymbol{\alpha}$ in a two-step procedure. From Eqs. (8.46) and (8.47) we get

$$\mathbf{V}\mathbf{D}\mathbf{V}^t\boldsymbol{\alpha} = \boldsymbol{\psi} \tag{8.53}$$

which can be written as

$$\mathbf{V}\mathbf{Y} = \boldsymbol{\psi} \tag{8.54}$$

and

$$\mathbf{D}\mathbf{V}^t\boldsymbol{\alpha} = \mathbf{Y} \tag{8.55}$$

or

$$\mathbf{V}^t\boldsymbol{\alpha} = \mathbf{D}^{-1}\mathbf{Y} \tag{8.56}$$

Thus from the matrix **V**, Eq. (8.54) can be solved for the column vector **Y** using a simple recursion of the form

$$Y_i = \psi_i - \sum_{j=1}^{i-1} V_{ij} Y_j, \qquad p \geqslant i \geqslant 2 \tag{8.57}$$

with initial condition

$$Y_1 = \psi_1 \tag{8.58}$$

Similarly having solved for **Y**, Eq. (8.56) can be solved recursively for $\boldsymbol{\alpha}$ using the relation

$$\alpha_i = Y_i/d_i - \sum_{j=i+1}^{p} V_{ji}\alpha_j \quad 1 \leqslant i \leqslant p-1 \tag{8.59}$$

with initial condition

$$\alpha_p = Y_p/d_p \tag{8.60}$$

It should be noted that the index i in Eq. (8.59) proceeds backwards from $i = p - 1$ down to $i = 1$.

To illustrate the use of Eqs. (8.57)-(8.60) we continue our previous example and first solve for the $Y_i's$ assuming **V** and **D** are now known. In matrix form we have the equation

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ V_{21} & 1 & 0 & 0 \\ V_{31} & V_{32} & 1 & 0 \\ V_{41} & V_{42} & V_{43} & 1 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} = \begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \end{bmatrix}$$

From Eqs. (8.57) and (8.58) we get

$$\begin{aligned} Y_1 &= \psi_1 \\ Y_2 &= \psi_2 - V_{21}Y_1 \\ Y_3 &= \psi_3 - V_{31}Y_1 - V_{32}Y_2 \\ Y_4 &= \psi_4 - V_{41}Y_1 - V_{42}Y_2 - V_{43}Y_3 \end{aligned}$$

From the $Y_i's$ we solve Eq. (8.56) which is of the form

$$\begin{bmatrix} 1 & V_{21} & V_{31} & V_{41} \\ 0 & 1 & V_{32} & V_{42} \\ 0 & 0 & 1 & V_{43} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} =$$

$$\begin{bmatrix} 1/d_1 & 0 & 0 & 0 \\ 0 & 1/d_2 & 0 & 0 \\ 0 & 0 & 1/d_3 & 0 \\ 0 & 0 & 0 & 1/d_4 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} = \begin{bmatrix} Y_1/d_1 \\ Y_2/d_2 \\ Y_3/d_3 \\ Y_4/d_4 \end{bmatrix}$$

From Eqs. (8.59) and (8.60) we get

$$\begin{aligned} \alpha_4 &= Y_4/d_4 \\ \alpha_3 &= Y_3/d_3 - V_{43}\alpha_4 \\ \alpha_2 &= Y_2/d_2 - V_{32}\alpha_3 - V_{42}\alpha_4 \\ \alpha_1 &= Y_1/d_1 - V_{21}\alpha_2 - V_{31}\alpha_3 - V_{41}\alpha_4 \end{aligned}$$

thus completing the solution to the covariance equations.

The use of the Cholesky decomposition procedure leads to a very simple expression for the minimum error of the covariance method in terms of the column vector **Y** and the matrix **D**. We recall that for the covariance method, the prediction error E_n was of the form

$$E_n = \phi_n(0,0) - \sum_{k=1}^{p} \alpha_k \phi_n(0,k) \tag{8.61}$$

or in matrix notation

$$E_n = \phi_n(0,0) - \boldsymbol{\alpha}^t \boldsymbol{\psi} \tag{8.62}$$

From Eq. (8.56) we can substitute for α^t the expression $\mathbf{Y}^t\mathbf{D}^{-1}\mathbf{V}^{-1}$ giving

$$E_n = \phi_n(0,0) - \mathbf{Y}^t\mathbf{D}^{-1}\mathbf{V}^{-1}\psi \tag{8.63}$$

Using Eq. (8.54) we get

$$E_n = \phi_n(0,0) - \mathbf{Y}^t\mathbf{D}^{-1}\mathbf{Y} \tag{8.64}$$

or

$$E_n = \phi_n(0,0) - \sum_{k=1}^{p} Y_k^2/d_k \tag{8.65}$$

Thus the mean-squared prediction error E_n can be determined directly from the column vector $\mathbf{Y}$ and the matrix $\mathbf{D}$. Furthermore Eq. (8.65) can be used to give the value of E_n for any value of p up to the value of p used in solving the matrix equations. Thus one can get an idea as to how the mean-squared prediction error varies with the number of predictor coefficients used in the solution.

8.3.2 Durbin's recursive solution for the autocorrelation equations [2]

For the autocorrelation method the matrix equation for solving for the predictor coefficients is of the form

$$\sum_{k=1}^{p} \alpha_k R_n(|i-k|) = R_n(i) \qquad 1 \leqslant i \leqslant p \tag{8.66}$$

By exploiting the Toeplitz nature of the matrix of coefficients, several efficient recursive procedures have been devised for solving this system of equations. Although the most popular and well known of these methods are the Levinson and Robinson algorithms [1], the most efficient method known for solving this particular system of equations is Durbin's recursive procedure [2] which can be stated as follows (for convenience of notation we shall omit the subscript on the autocorrelation function):

$$E^{(0)} = R(0) \tag{8.67}$$

$$k_i = \left[R(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R(i-j)\right] / E^{(i-1)} \qquad 1 \leqslant i \leqslant p \tag{8.68}$$

$$\alpha_i^{(i)} = k_i \tag{8.69}$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \qquad 1 \leqslant j \leqslant i-1 \tag{8.70}$$

$$E^{(i)} = (1-k_i^2) E^{(i-1)} \tag{8.71}$$

Equations (8.68)-(8.71) are solved recursively for $i = 1, 2, \ldots, p$ and the final solution is given as

$$\alpha_j = \alpha_j^{(p)} \qquad 1 \leqslant j \leqslant p \tag{8.72}$$

Note that in the process of solving for the predictor coefficients for a predictor of order p, the solutions for the predictor coefficients of all orders less than p

have also been obtained — i.e., $\alpha_j^{(i)}$ is the j^{th} predictor coefficient for a predictor of order i.

To illustrate the above procedure, consider an example of obtaining the predictor coefficients for a predictor of order 2. The original matrix equation is of the form

$$\begin{bmatrix} R(0) & R(1) \\ R(1) & R(0) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \end{bmatrix}$$

Using Eqs. (8.67)-(8.72), we get

$$E^{(0)} = R(0)$$

$$k_1 = R(1)/R(0)$$

$$\alpha_1^{(1)} = R(1)/R(0)$$

$$E^{(1)} = \frac{R^2(0)-R^2(1)}{R(0)}$$

$$k_2 = \frac{R(2)R(0)-R^2(1)}{R^2(0)-R^2(1)}$$

$$\alpha_2^{(2)} = \frac{R(2)R(0)-R^2(1)}{R^2(0)-R^2(1)}$$

$$\alpha_1^{(2)} = \frac{R(1)R(0)-R(1)R(2)}{R^2(0)-R^2(1)}$$

$$\alpha_1 = \alpha_1^{(2)}$$

$$\alpha_2 = \alpha_2^{(2)}$$

It should be noted that the quantity $E^{(i)}$ in Eq. (8.71) is the prediction error for a predictor of order i. Thus at each stage of the computation the prediction error for a predictor of order i can be monitored. Also, if the autocorrelation coefficients $R(i)$ are replaced by a set of normalized autocorrelation coefficients, i.e., $r(k) = R(k)/R(0)$, then the solution to the matrix equation remains unchanged. However, the error $E^{(i)}$ is now interpreted as a normalized error. If we call this normalized error $V^{(i)}$, then

$$V^{(i)} = \frac{E^{(i)}}{R(0)} = 1 - \sum_{k=1}^{i} \alpha_k r(k) \tag{8.73}$$

with

$$0 < V^{(i)} \leqslant 1 \qquad i \geqslant 0 \tag{8.74}$$

It can be shown that the normalized error for $i = p$ (i.e., $V^{(p)}$) can be written in the form

$$V^{(p)} = \prod_{i=1}^{p} (1 - k_i^2) \tag{8.75}$$

where the quantities k_i are in the range

$$-1 \leqslant k_i \leqslant 1 \tag{8.76}$$

This condition on the parameters k_i is important since it can be shown [1,18] that it is a necessary and sufficient condition for all of the roots of the polynomial $A(z)$ to be inside the unit circle, thereby guaranteeing the stability of the system $H(z)$. Unfortunately a proof of this result would take us too far afield; however, the fact that we do not give a proof does not diminish the importance of this result. Furthermore, it is possible to show that no such guarantee of stability is available in the covariance method.

8.3.3 Lattice formulations and solutions [11]

As we have seen, both the covariance and the autocorrelation methods consist of two steps:

1. Computation of a matrix of correlation values.
2. Solution of a set of linear equations.

These methods have been widely used with great success in speech processing applications. However, another class of methods, called *lattice methods,* has evolved in which the above two steps have in a sense been combined into a recursive algorithm for determining the linear predictor parameters. To see how these methods are related, it is helpful to begin with the Durbin algorithm. First, let us recall that at the i^{th} stage of this procedure, the set of coefficients $\{\alpha_j^{(i)}\ j=1,2,\ \ldots,\ i\}$ are the coefficients of the i^{th} order optimum linear predictor. Using these coefficients we can define

$$A^{(i)}(z) = 1 - \sum_{k=1}^{i} \alpha_k^{(i)} z^{-k} \tag{8.77}$$

to be the system function of the i^{th}-order inverse filter (or prediction error filter). If the input to this filter is the segment of the signal, $s_n(m) = s(n+m)\,w(m)$, then the output would be the prediction error, $e_n^{(i)}(m) = e^{(i)}(n+m)$, where

$$e^{(i)}(m) = s(m) - \sum_{k=1}^{i} \alpha_k^{(i)} s(m-k) \tag{8.78}$$

Note that for the sake of simplicity we shall henceforth drop the subscript n which denotes the fact that we are considering a segment of the signal located at sample n. In terms of z-transforms Eq. (8.78) is

$$E^{(i)}(z) = A^{(i)}(z) S(z) \tag{8.79}$$

By substituting Eq. (8.70) into Eq. (8.77) we obtain a recurrence formula for $A^{(i)}(z)$ in terms of $A^{(i-1)}(z)$; i.e.,

$$A^{(i)}(z) = A^{(i-1)}(z) - k_i z^{-i} A^{(i-1)}(z^{-1}) \tag{8.80}$$

(See Problem 8.5.) Substituting Eq. (8.80) into Eq. (8.79) we obtain

$$E^{(i)}(z) = A^{(i-1)}(z) S(z) - k_i z^{-i} A^{(i-1)}(z^{-1}) S(z) \tag{8.81}$$

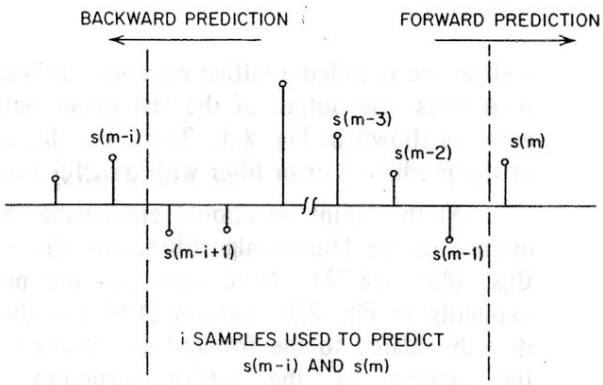


Fig. 8.2 Illustration of forward and backward prediction using an i^{th} order predictor.

The first term in Eq. (8.81) is obviously the z-transform of the prediction error for an $(i-1)^{th}$ order predictor. The second term can be given a similar interpretation if we define

$$B^{(i)}(z) = z^{-i}A^{(i)}(z^{-1})S(z) \tag{8.82}$$

It is easily shown that the inverse transform of $B^{(i)}(z)$ is

$$b^{(i)}(m) = s(m-i) - \sum_{k=1}^{i} \alpha_k^{(i)} s(m+k-i) \tag{8.83}$$

This equation suggests that we are attempting to predict $s(m-i)$ from the i samples of the input $\{s(m-i+k),\ k=1,2,\ldots,i\}$ that follow $s(m-i)$. Thus $b^{(i)}(m)$ is called the backward prediction error sequence. In Fig. 8.2 it is shown that the i samples involved in the prediction are the same ones used to predict $s(m)$ in terms of i *past* samples in Eq. (8.78). Now returning to Eq. (8.81) we see that the prediction error sequence $e^{(i)}(m)$ can be expressed as

$$e^{(i)}(m) = e^{(i-1)}(m) - k_i b^{(i-1)}(m-1) \tag{8.84}$$

By substituting Eq. (8.80) into Eq. (8.82) we obtain

$$B^{(i)}(z) = z^{-i}A^{(i-1)}(z^{-1})S(z) - k_i A^{(i-1)}(z)S(z) \tag{8.85}$$

or

$$B^{(i)}(z) = z^{-1}B^{(i-1)}(z) - k_i E^{(i-1)}(z) \tag{8.86}$$

Thus the i^{th} stage backward prediction error is

$$b^{(i)}(m) = b^{(i-1)}(m-1) - k_i e^{(i-1)}(m) \tag{8.87}$$

Now Eqs. (8.84) and (8.87) define the forward and backward prediction error sequences for an i^{th} order predictor in terms of the corresponding prediction errors of an $(i-1)^{th}$ order predictor. Using a zeroth order predictor is equivalent to using no predictor at all so that

$$e^{(0)}(m) = b^{(0)}(m) = s(m) \tag{8.88}$$

Thus we can depict Eqs. (8.84) and (8.87) by the flow graph of Fig. 8.3. Such

a structure is called a lattice network. It is clear that if we extend the lattice to p sections, the output of the last upper branch will be the forward prediction error as shown in Fig. 8.3. Thus, Fig. 8.3 is a digital network implementation of the prediction error filter with transfer function $A(z)$.

At this point we should emphasize that this structure is a direct consequence of the Durbin algorithm, and the parameters k_i can be obtained as in Eqs. (8.67)-(8.72). Note also that the predictor coefficients do not appear explicitly in Fig. 8.3. Itakura [4,6] has shown that the k_i parameters can be directly related to the forward and backward prediction errors and because of the nature of the lattice structure the entire set of coefficients k_i, $i=1,2,\ldots,p\}$ can be computed without computing the predictor coefficients. The relationship is [11]

$$k_i = \frac{\sum_{m=0}^{N-1} e^{(i-1)}(m)\, b^{(i-1)}(m-1)}{\left\{\sum_{m=0}^{N-1} (e^{(i-1)}(m))^2 \sum_{m=0}^{N-1} (b^{(i-1)}(m-1))^2\right\}^{1/2}} \tag{8.89}$$

This expression is in the form of a normalized cross-correlation function; i.e., it is indicative of the degree of correlation between the forward and backward prediction error. For this reason the parameters k_i are called the partial correlation coefficients or PARCOR coefficients [4,6]. It is relatively straightforward to verify that Eq. (8.89) is identical to Eq. (8.68) by substituting Eqs. (8.78) and (8.83) into Eq. (8.89).

It can be seen that if Eq. (8.89) replaces Eq. (8.68) in the Durbin algorithm, the predictor coefficients can be computed recursively as before. Thus the PARCOR analysis leads to an alternative to the inversion of a matrix and

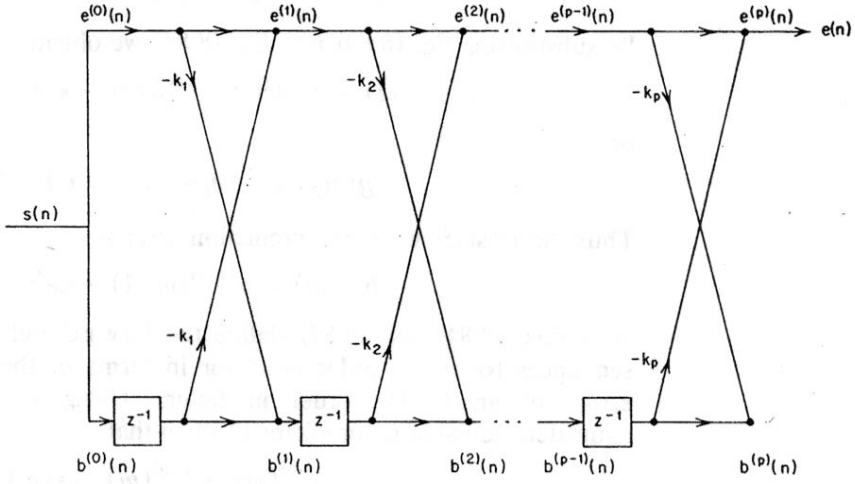


Fig. 8.3 Block diagram of a realizable implementation of the lattice method.

gives results identical to the autocorrelation method; i.e., the set of PARCOR coefficients is equivalent to a set of predictor coefficients that minimize the mean-squared forward prediction error. More importantly, this approach opens up a whole new class of procedures based upon the lattice configuration of Fig. 8.3 [11].

In particular, Burg [12] has developed a procedure based upon minimizing the *sum* of the mean-squared forward and backward prediction errors in Fig. 8.3; i.e.,

$$\tilde{E}^{(i)} = \sum_{m=0}^{N-1} \left[(e^{(i)}(m))^2 + (b^{(i)}(m))^2 \right] \tag{8.90}$$

Substituting Eqs. (8.84) and (8.87) into Eq. (8.90) and differentiating $\tilde{E}^{(i)}$ with respect to k_i, we obtain

$$\frac{\partial \tilde{E}^{(i)}}{\partial k_i} = -2 \sum_{m=0}^{N-1} \left[e^{(i-1)}(m) - k_i b^{(i-1)}(m-1) \right] b^{(i-1)}(m-1)$$
$$-2 \sum_{m=0}^{N-1} \left[b^{(i-1)}(m-1) - k_i e^{(i-1)}(m) \right] e^{(i-1)}(m) \tag{8.91}$$

Setting the derivative equal to zero and solving for k_i gives

$$k_i = \frac{2 \sum_{m=0}^{N-1} \left[e^{(i-1)}(m) b^{(i-1)}(m-1) \right]}{\sum_{m=0}^{N-1} \left[e^{(i-1)}(m) \right]^2 + \sum_{m=0}^{N-1} \left[b^{(i-1)}(m-1) \right]^2} \tag{8.92}$$

It can be shown [1] that if k_i is estimated using Eq. (8.92) then

$$-1 \leqslant k_i \leqslant 1 \tag{8.93}$$

However, it should be clear that the k_i's estimated using Eq. (8.92) will in general differ from those estimated using Eq. (8.89), or equivalently, the autocorrelation method.

In summary, the steps involved in determining the predictor coefficients and the k parameters are as follows:

1. Initially set $e^{(0)}(m) = s(m) = b^{(0)}(m)$.
2. Compute $k_1 = \alpha_1^{(1)}$ from Eq. (8.92).
3. Determine forward and backward prediction errors $e^{(1)}(m)$ and $b^{(1)}(m)$ from Eqs. (8.84) and (8.87).
4. Set $i = 2$.
5. Determine $k_i = \alpha_i^{(i)}$ from Eq. (8.92).
6. Determine $\alpha_j^{(i)}$ for $j = 1, 2, \ldots, i-1$ from Eq. (8.70).
7. Determine $e^{(i)}(m)$ and $b^{(i)}(m)$ from Eq. (8.84) and (8.87).
8. Set $i = i + 1$.
9. If i is less than or equal to p, go to step 5.
10. Procedure is terminated.

There are clearly several differences in implementation between the lattice method and the covariance and autocorrelation implementations discussed earlier. One major difference is that in the lattice method the predictor coefficients are obtained directly from the speech samples without an intermediate calculation of an autocorrelation function. At the same time the method is guaranteed to yield a stable filter without requiring the use of a window. For these reasons the lattice formulation has become an important and viable approach to the implementation of linear predictive analysis.

8.4 Comparisons Between the Methods of Solution of the LPC Analysis Equations

We have already discussed the differences in the theoretical formulations of the covariance, autocorrelation, and lattice formulations of the linear predictive analysis equations. In this section we discuss the issues involved in practical implementations of the analysis equations. Included among these issues are computational considerations, numerical and physical stability of the solutions, and the question of how to choose the number of poles and section length used in the analysis. We begin first with the computational considerations involved in obtaining the predictor coefficients from the speech waveform.

The two major issues in the computation of the predictor coefficients are the amount of storage, and the number of multiplications. Table 8.1 (due to Portnoff et al. [13] and Makhoul [11]) shows the required computation for the covariance, the autocorrelation and the lattice methods. In terms of storage, for the covariance method, the requirements are essentially N_1 locations for the data, and on the order of $p^2/2$ locations for the correlation matrix, where N_1 is the number of points in the analysis. For the autocorrelation method the requirements are N_2 locations for both the data and the window, and a number of locations proportional to p for the autocorrelation matrix. For the lattice method the requirements are $3N_3$ locations for the data and the forward and backward prediction errors. For emphasis we have assumed that the N_1 for the covariance method, the N_2 for the autocorrelation method, and the N_3 for the lattice method need not be the same. We will discuss this question later in this section. Thus in terms of storage (assuming N_1, N_2, and N_3 are comparable) the covariance and autocorrelation methods require somewhat less storage than the lattice method.

The computational requirements for the three methods, in terms of multiplications, are shown at the bottom of Table 8.1. For the covariance method, the computation of the correlation matrix requires about N_1p multiplications, whereas the solution to the matrix equation (using the Cholesky decomposition procedure) requires a number of multiplications proportional to p^3. (Portnoff et al. give an exact figure of $(p^3+9p^2+2p)/6$ multiplications, p divides, and p square roots.) For the autocorrelation method, the computation of the autocorrelation matrix requires about N_2p multiplications, whereas the solution to the matrix equations requires about p^2 multiplications. Thus if N_1 and N_2 are

Table 8.1 Computational Considerations in the LPC Solutions

	Covariance Method	*Autocorrelation Method*	*Lattice Method*
	(Cholesky Decomposition)	(Durbin Method)	(Burg Method)
Storage			
Data	N_1	N_2	$3N_3$
Matrix	proportional to $p^2/2$	proportional to p	—
Window	0	N_2	—
Computation (Multiplications)			
Windowing	0	N_2	—
Correlation	proportional to $N_1 p$	proportional to $N_2 p$	—
Matrix Solution	proportional to p^3	proportional to p^2	$5N_3 p$

approximately equal, and with $N_1 >> p$, $N_2 >> p$, then the autocorrelation method will require somewhat less computation than the covariance method. However, since in most speech problems the number of multiplications required to compute the correlation function far exceeds the number of multiplications to solve the matrix equations, the computation times for both these formulations are quite comparable. For the lattice method a total of $5N_3 p$ multiplications are needed to compute the set of partial correlation coefficients.[5] Thus the lattice method is the least computationally efficient method for solving the LPC equations. However, the other advantages of the lattice method must be kept in mind when considering the use of this method.

Another consideration in comparing these three formulations is the stability of the resulting system

$$H(z) = \frac{G}{A(z)} \tag{8.94}$$

This system is stable if all its poles lie strictly inside the unit circle in the z-plane. The poles of the system, $H(z)$, are the zeros of denominator polynomial $A(z)$, where

$$A(z) = 1 - \sum_{k=1}^{p} \alpha_k z^{-k} \tag{8.95}$$

As we have asserted, for the autocorrelation method all the roots of $A(z)$ lie inside the unit circle — i.e., $H(z)$ is guaranteed to be stable. It should be noted that this theoretical guarantee of stability for the autocorrelation method may not hold in practice if the autocorrelation function is computed without sufficient accuracy. In such cases the roundoff encountered in computing the autocorrelation can cause the matrix to become ill conditioned. Markel and Gray have shown that these undesirable effects can be minimized by pre-emphasizing the speech to make its spectrum as flat as possible [1]. With the use of a pre-emphasizing filter, smaller wordlengths can be used in practice and

[5]Makhoul has discussed a modified lattice method for obtaining the partial correlation coefficients with the same efficiency as the normal covariance method [11].

the resulting predictor polynomials will generally remain stable. The Durbin algorithm provides a convenient test for stability since it is necessary and sufficient that the parameters k_i (PARCOR's) must satisfy the condition

$$-1 \leqslant k_i \leqslant 1 \tag{8.96}$$

Thus if, in the process of determining the predictor coefficients $\{\alpha_i\}$, any of the quantities k_i violate Eq. (8.96) then it is known that there are roots of $A(z)$ outside the unit circle.

For the covariance method, the stability of the predictor polynomial cannot be guaranteed. However, in practice, if the number of samples in the frame is sufficiently large, then the resulting predictor polynomials will almost always be stable. This is due to the fact that for a large number of samples in the analysis frame, the covariance and autocorrelation methods yield almost identical results.

For the lattice method the predictor polynomial is guaranteed to be stable since the predictor coefficients are obtained from the partial correlation coefficients which, by definition, satisfy Eq. (8.96). In addition, the stability is preserved even when the computation is performed using finite word length computations [1].

In the case when the predictor polynomial stability is uncertain, it is generally required that the roots of the predictor polynomial be determined and tested for stability. If a root is found to be outside the unit circle, a simple correction procedure is to reflect the root inside the unit circle, thereby ensuring a stable predictor polynomial with the same frequency response as the unstable polynomial.

Two other considerations in comparing and contrasting the three formulations of the LPC equations are the choice of number of predictor parameters, p, and the choice of the frame length N. The choice of p depends primarily on the sampling rate and is essentially independent of the LPC method being used. Since the speech spectrum being analyzed can generally be represented as having an average density of 2 poles (i.e., one complex pole) per kiloHertz due to the vocal tract contribution, then a total of F_s poles are required to represent this contribution to the speech spectrum, where F_s is the sampling rate in kiloHertz. Thus for a 10 kHz sampling rate, a total of 10 poles is required to represent the vocal tract. In addition, a total of 3-4 poles is required to adequately represent the source excitation spectrum and the radiation load. Thus for a 10 kHz simulation, a value of p of about 13 or 14 is required. To verify this conclusion, Figure 8.4 shows a plot (due to Atal and Hanauer [3]) of the normalized rms prediction error versus the predictor order p for sections of voiced and unvoiced speech for a 10 kHz simulation. Although the prediction error steadily decreases as p increases, for p on the order of 13-14 the error has essentially flattened off showing only small decreases as p is increased further. It is interesting to note from this figure that the normalized rms prediction error for unvoiced speech is significantly higher than for voiced speech. This is of course as expected since the model for unvoiced speech is nowhere near as

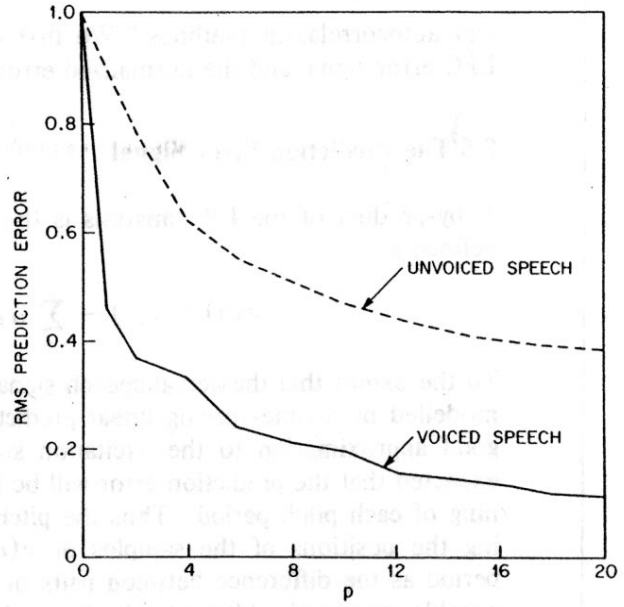


Fig. 8.4 Variation of the RMS prediction error with the number of predictor coefficients, p. (After Atal and Hanauer [3].)

accurate as it is for voiced speech. Additional experimental evidence of the behavior of the prediction error as a function of p is given in the next section.

The choice of section length N is a very important consideration in implementation of most LPC analysis systems. Clearly, it is advantageous to keep N as small as possible since the total computation load, for all three methods, is essentially proportional to N. For the autocorrelation method it has been shown that N must be on the order of several pitch periods to ensure reliable results [1,2]. Since a window is used to weight the speech in the autocorrelation method, the section duration must be sufficiently long so that the tapering effects of the window do not seriously affect the results. Thus analysis durations from $N = 100$ to $N = 400$ samples (at a 10 kHz rate) have been used in LPC implementations of the autocorrelation method, with most systems leaning toward the larger values of N. For both the covariance and lattice methods, the choice of section length is governed by several considerations. Since no windowing is required, there are no real limitations on how small the section size can be. If the analysis can be restricted to regions within each pitch period (i.e., a pitch synchronous analysis is performed) then values of N on the order of $2p$ have been used successfully. However if such small values of N are used and if a pitch pulse occurs within the analysis interval, unsatisfactory results are obtained. Thus in most practical systems in which it is not possible to perform a pitch synchronous analysis, values of N for the covariance and lattice methods are comparable to those for the autocorrelation method. In the next few sections we show results from experimental evaluations of the effects of section length, and section position on the prediction error for the covariance

and autocorrelation methods.[6] We first digress into a brief discussion of the LPC error signal and the normalized error derived from it.

8.5 The Prediction Error Signal

A by-product of the LPC analysis is the generation of the error signal, $e(n)$, defined as

$$e(n) = s(n) - \sum_{k=1}^{p} \alpha_k s(n-k) = Gu(n) \tag{8.97}$$

To the extent that the actual speech signal is generated by a system that is well modelled by a time-varying linear predictor of order p, then $e(n)$ is equally a good approximation to the excitation source. Based on this reasoning, it is expected that the prediction error will be large (for voiced speech) at the beginning of each pitch period. Thus the pitch period can be determined by detecting the positions of the samples of $e(n)$ which are large, and defining the period as the difference between pairs of samples of $e(n)$ which exceed a reasonable threshold. Alternatively the pitch period can be estimated by performing an autocorrelation analysis on $e(n)$ and detecting the largest peak in the

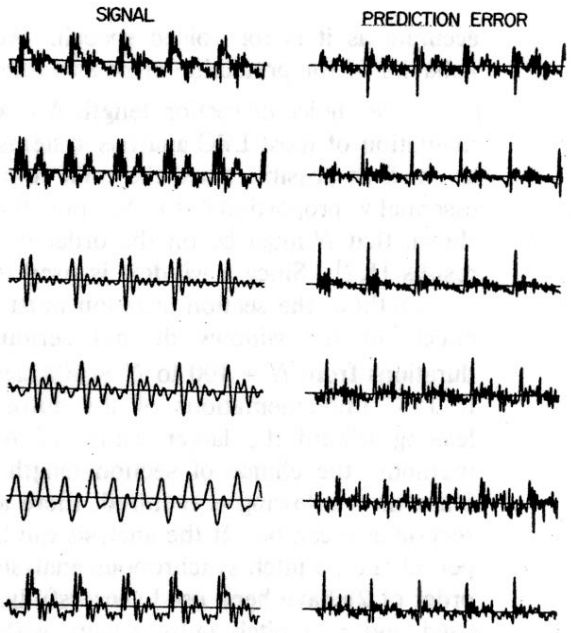


Fig. 8.5 Examples of signal (differentiated) and prediction error for vowels (*i, e, a, o, u, y*). (After Strube [14].)

[6]Investigations by Rabiner et al. [16] have found that a good choice of parameters for the lattice method are essentially those used for the covariance method. Thus we do not differentiate between these methods in the following sections.

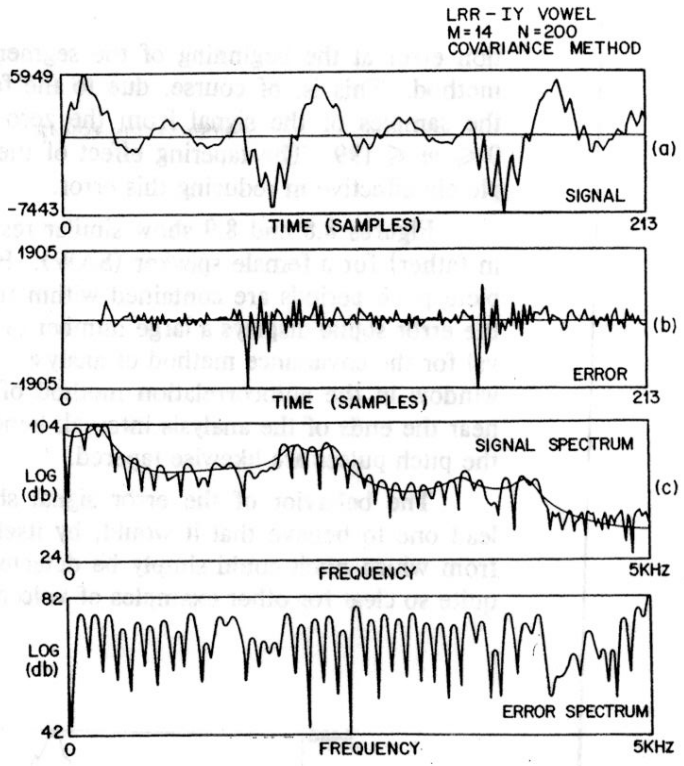


Fig. 8.6 Typical signals and spectra for LPC covariance method for a male speaker. (After Rabiner et al. [16].)

appropriate range. Another way of interpreting why the error signal is valuable for pitch detection is the observation that the spectrum of the error signal is approximately flat; thus the effects of the formants have been eliminated in the error signal.

To illustrate the nature of the error signal Figure 8.5 (due to Strube [14]) shows a series of sections of the waveforms for several vowels, and the corresponding prediction error signals. For all these simple vowel sounds the error signal exhibits sharp pulses at intervals corresponding to the pitch periods of these vowels.

Some further examples of LPC error signals are given in Figures 8.6-8.9. In each of these figures part (a) shows the section of speech being analyzed, part (b) shows the resulting prediction error signal, part (c) shows the log magnitude of the DFT of the signal in part (a) (obtained via FFT computation) with the log magnitude of $H(e^{j\omega T})$ superimposed, and part (d) shows the log magnitude spectrum of the error signal (obtained via FFT computation). Figures 8.6 and 8.7 are for 20 msec of an /*i*/ vowel (as in we) spoken by a male speaker (LRR) using the covariance and autocorrelation methods (with a Hamming window) respectively. The error signal is seen to be sharply peaked at the beginning of each pitch period, and the error spectrum is fairly flat, showing a comb effect due to the effects of the pitch period. Note the rather large predic-

tion error at the beginning of the segment in Fig. 8.7 for the autocorrelation method. This is, of course, due to the fact that we are attempting to predict the samples of the signal from the zero valued samples outside the interval $0 \leqslant m \leqslant 199$. The tapering effect of the Hamming window is thus not completely effective in reducing this error.

Figures 8.8 and 8.9 show similar results for 20 msec of an /a/ vowel (as in father) for a female speaker (SAW). For this speaker approximately 5 complete pitch periods are contained within the analysis interval. Thus in Fig. 8.8 the error signal displays a large number of sharp peaks during the analysis interval for the covariance method of analysis. However, the effect of the Hamming window in the autocorrelation method of Fig. 8.9 is to taper the pitch pulses near the ends of the analysis interval; hence the peaks in the error signal due to the pitch pulses are likewise tapered.

The behavior of the error signal shown in the preceding figures would lead one to believe that it would, by itself, be a natural candidate for a signal from which pitch could simply be detected. Unfortunately the situation is not quite so clear for other examples of voiced speech. Makhoul and Wolf [5] have

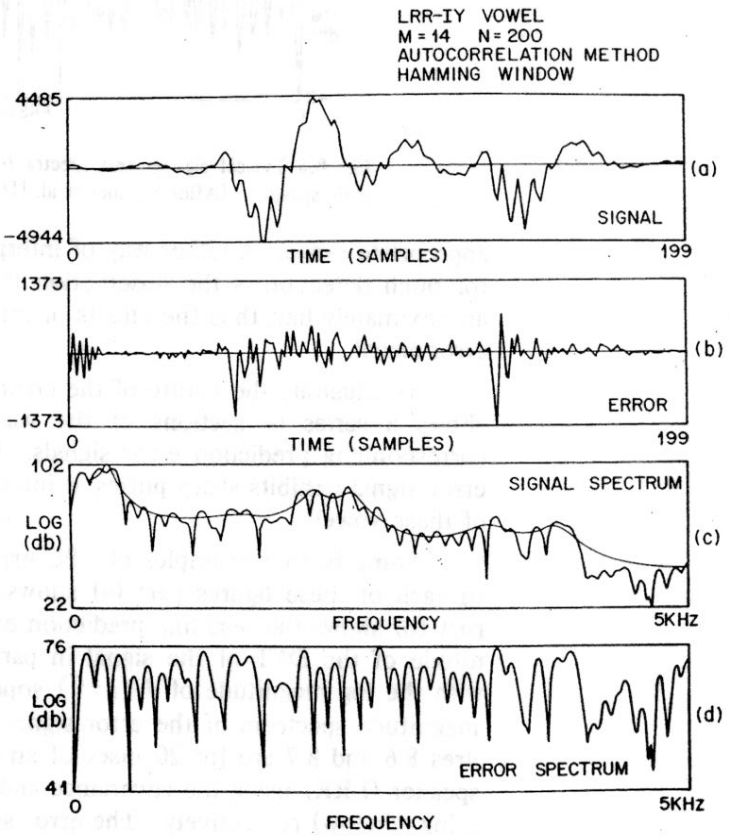


Fig. 8.7 Typical signals and spectra for LPC autocorrelation method for a male speaker. (After Rabiner et al. [16].)

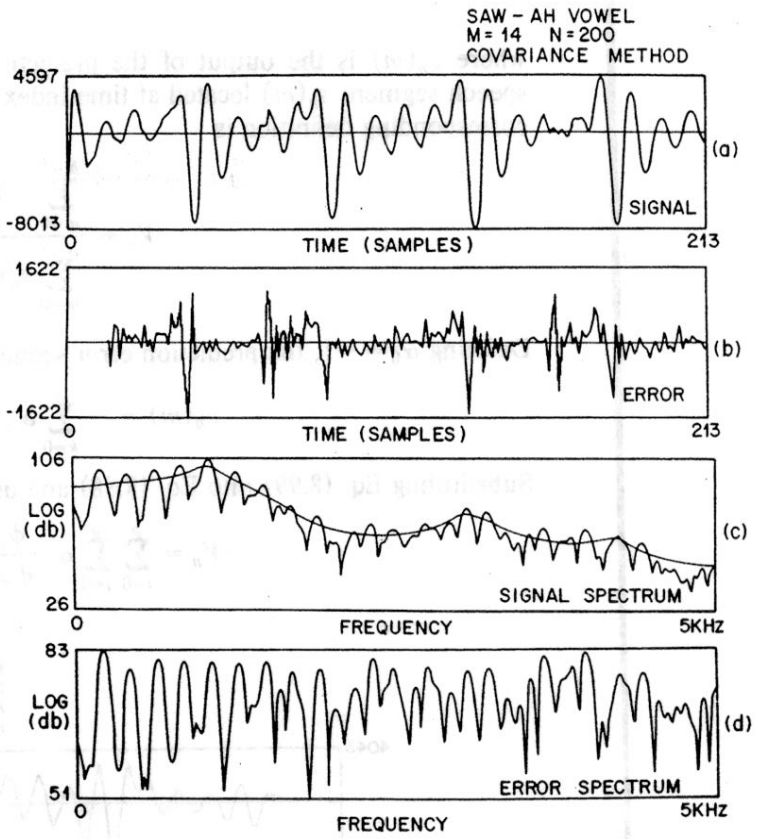


Fig. 8.8 Typical signals and spectra for LPC covariance method for a female speaker. (After Rabiner et al. [16].)

shown that for sounds which are not rich in harmonic structure, e.g., liquids like *r*, *l*, or nasals such as *m*, *n*, the peaks in the error signal are not always very sharp or distinct. Additionally at the junctions between voiced and unvoiced sounds, the pitch markers in the error signal often essentially disappear.

In summary, although the error signal $e(n)$ appears to be an ideal candidate for a pitch detector, it has its own difficulties in locating pitch markers for a wide variety of voiced sounds, and thus cannot be relied on exclusively for this purpose. In Section 8.10.1 we shall discuss one pitch detection scheme based upon the prediction error signal.

8.5.1 Alternative expressions for the normalized mean-squared error

The normalized mean squared prediction error for the autocorrelation method is defined as

$$V_n = \frac{\sum_{m=0}^{N+p-1} e_n^2(m)}{\sum_{m=0}^{N-1} s_n^2(m)} \tag{8.98a}$$

where $e_n(m)$ is the output of the prediction error filter corresponding to the speech segment $s_n(m)$ located at time index n. For the covariance method, the corresponding definition is

$$V_n = \frac{\sum_{m=0}^{N-1} e_n^2(m)}{\sum_{m=0}^{N-1} s_n^2(m)} \tag{8.98b}$$

Defining $\alpha_0 = -1$, the prediction error sequence can be expressed as

$$e_n(m) = -\sum_{k=0}^{p} \alpha_k s_n(m-k) \tag{8.99}$$

Substituting Eq. (8.99) into Eq. (8.98) and using Eq. (8.13) it follows that

$$V_n = \sum_{i=0}^{p} \sum_{j=0}^{p} \alpha_i \frac{\phi_n(i,j)}{\phi_n(0,0)} \alpha_j \tag{8.100a}$$

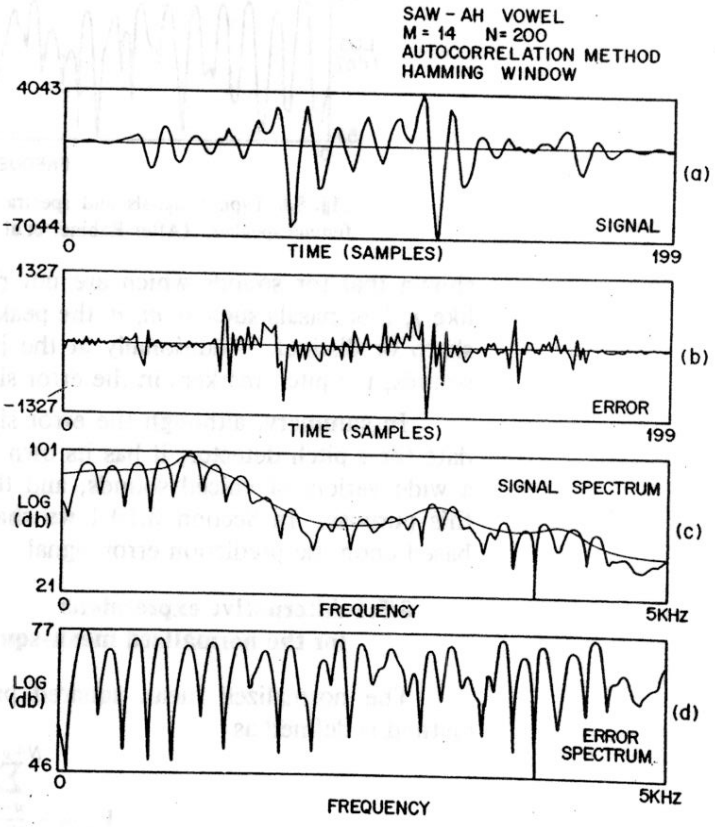


Fig. 8.9 Typical signals and spectra for LPC autocorrelation method for a female speaker. (After Rabiner et al. [16].)

and substituting Eq. (8.14) into (8.100) gives

$$V_n = -\sum_{i=0}^{p} \alpha_i \frac{\phi_n(0,i)}{\phi_n(0,0)} \tag{8.100b}$$

Still another expression for V_n was obtained in the Durbin algorithm; i.e.,

$$V_n = \prod_{i=1}^{p} (1-k_i^2) \tag{8.101}$$

The above expressions are not all equivalent and are subject to interpretation in terms of the details of a given linear predictive method. For example, Eq. (8.101), being based upon the Durbin algorithm is valid only for the autocorrelation and lattice methods. Also, since the lattice method does not explicitly require the computation of the correlation functions Eqs. (8.100a) and (8.100b) do not apply directly to the lattice method. Table 8.2 summarizes the above expressions for normalized mean-squared error and indicates the scope of validity of each expression. (Note that the subscript n and the superscript p have been eliminated in the table for simplicity.)

Table 8.2 Expressions for the Normalized Error

	Covariance Method	*Autocorrelation Method*	*Lattice Method*
$V = \dfrac{\sum_m e^2(m)}{\sum_m s^2(m)}$	Valid	Valid*	Valid
$V = \sum_i \sum_j \alpha_i \dfrac{\phi(i,j)}{\phi(0,0)} \alpha_j$	Valid	Valid**	Not Valid
$V = \sum_i \alpha_i \dfrac{\phi(i,j)}{\phi(0,0)}$	Valid	Valid**	Not Valid
$V = \prod_i (1-k_i^2)$	Not Valid	Valid	Valid

*This expression is computed using the windowed signal and upper limit is $N - 1 + p$.
**In these cases $\phi(i,j) = R(i-j)$.

8.5.2 Experimental evaluation of values for the LPC parameters

To provide guidelines to aid in the choice of the LPC parameters p and N for practical implementations, Chandra and Lin [15] performed a series of investigations in which they plotted the normalized mean-squared prediction error, for a p^{th} order predictor versus the relevant parameter for the following conditions:

1. The covariance method and the autocorrelation method
2. Synthetic vowel and natural speech
3. Pitch synchronous and pitch asynchronous analysis

where V is defined as in Table 8.2. Figures 8.10-8.15 show the results obtained by Chandra and Lin for the above conditions [15].

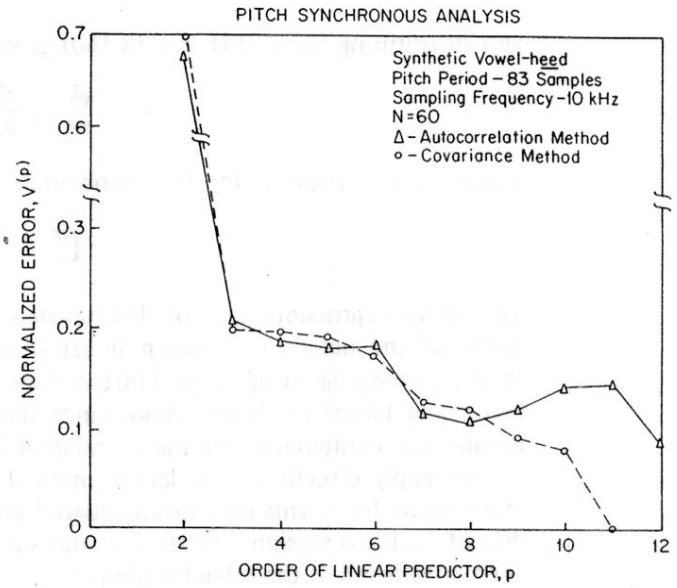


Fig. 8.10 Variation of prediction error with predictor order, p, for voiced section of a synthetic vowel—pitch synchronous analysis. (After Chandra and Lin [15].)

Figure 8.10 shows the variation of V with the order of the linear predictor, p, for a section of a synthetic vowel (/*i*/ in heed) whose pitch period was 83 samples. The analysis section length N was 60 samples beginning at the beginning of a pitch period — i.e., these results are for a pitch synchronous analysis. For the covariance method the prediction error decreases monotonically to 0 at $p = 11$ which was the order of the system used to create the synthetic speech. For the autocorrelation method the prediction remains at a value of about 0.1 for values of p greater than about 7. This behavior is due to the fact that for the autocorrelation method with short windows ($N = 60$) the prediction error at the beginning of the segment is an appreciable part of the total mean-squared error. This is, of course, not the case with the covariance method, where speech samples from outside the averaging interval are available for prediction.

Figure 8.11 shows the variation of V with the order of the linear predictor for a pitch asynchronous analysis for the same section of speech as used in Fig. 8.10. This time, however, the section length was $N = 120$ samples. For this case the covariance and autocorrelation methods yielded nearly identical values of V for different values of p. Further the values of V decreased monotonically to a value of about 0.1 near $p = 11$. Thus in the case of an asynchronous LPC analysis, at least for the example of a synthetic vowel, both analysis methods appear to yield similar results.

Figure 8.12 shows the variation of V with N (section length) for a linear predictor of order 12 for the synthetic speech section. As anticipated, for values of N below the pitch period (83 samples) the covariance method gives significantly smaller values of V than the autocorrelation method. For values of

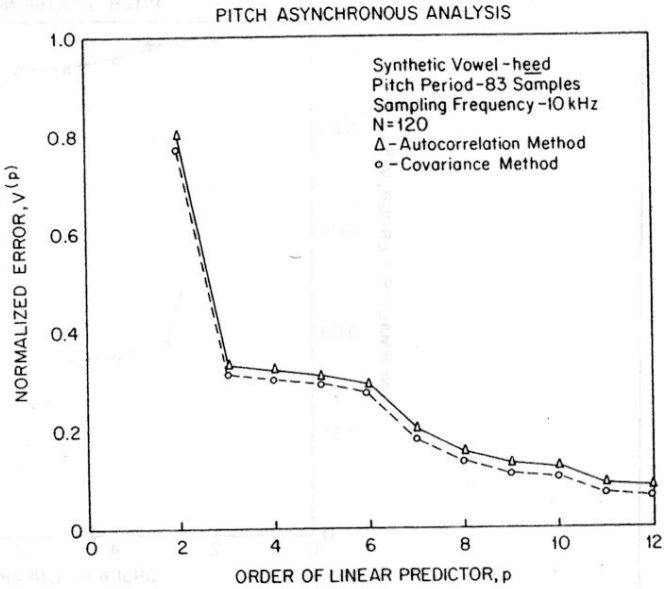


Fig. 8.11 Variation of prediction error with predictor order, p, for voiced section of a synthetic vowel—pitch asynchronous analysis. (After Chandra and Lin [15].)

V at or near multiples of the pitch period, the values of V show fairly large jumps due to the large prediction error when a pitch pulse is used to excite the system. However, for most values of N on the order of 2 or more pitch periods, both analysis methods yield comparable values of V.

Figures 8.13-8.15 show a similar set of figures for the case of a section of natural voiced speech. Figure 8.13 shows that the normalized error for the

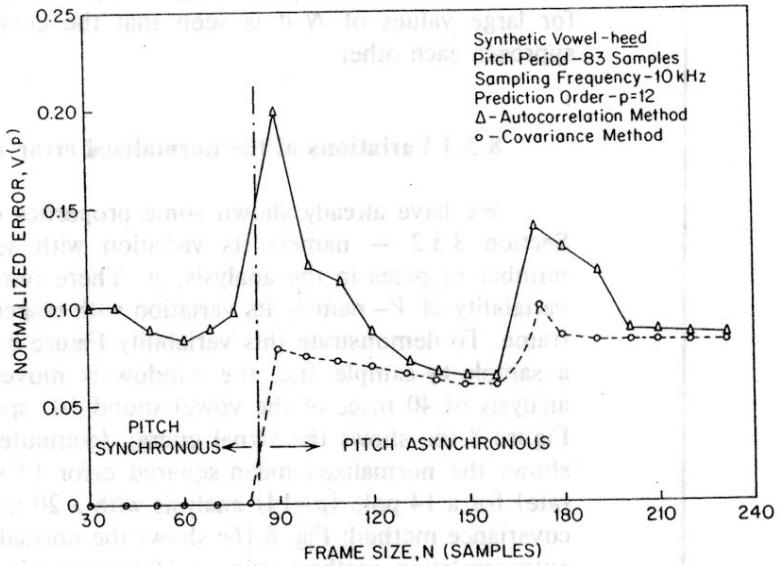


Fig. 8.12 Variation of prediction error with section length, N, for a voiced section of synthetic speech. (After Chandra and Lin [15].)

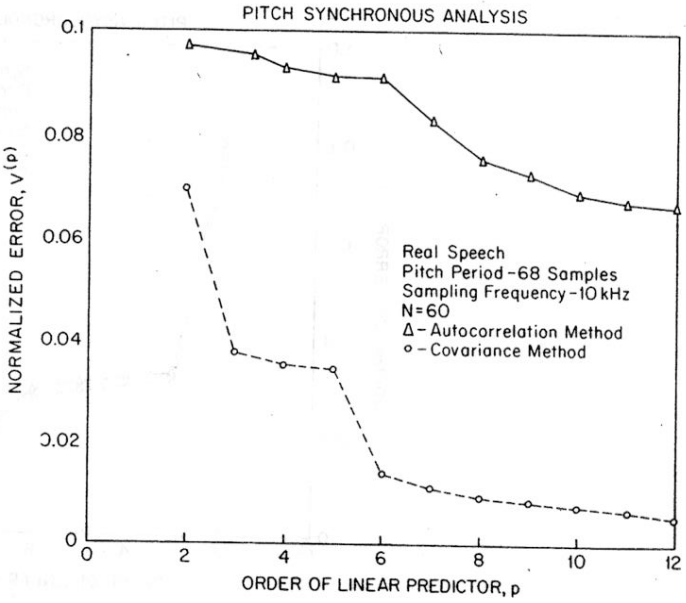


Fig. 8.13 Variation of prediction error with predictor order, p, for a voiced section of a natural vowel—pitch synchronous analysis. (After Chandra and Lin [15].)

covariance method is significantly lower than the normalized error for the autocorrelation method for a pitch synchronous analysis, whereas Figure 8.14 shows that for a pitch asynchronous analysis, the values of V are comparable. Finally Figure 8.15 shows how the values of V vary as N varies for an analysis with $p = 12$. It can be seen that in the region of pitch pulse occurrences, the value of V for the autocorrelation analysis jumps significantly whereas the value of V for the covariance analysis changes only a small amount at these points. Also for large values of N it is seen that the curves of V for the two methods approach each other.

8.5.3 Variations of the normalized error with frame position

We have already shown some properties of the LPC normalized error in Section 8.5.2 — namely its variation with section length N, and with the number of poles in the analysis, p. There remains one other major source of variability of V—namely its variation with respect to the position of the analysis frame. To demonstrate this variability Figure 8.16 shows plots of the results of a sample-by-sample (i.e. the window is moved one sample at a time) LPC analysis of 40 msec of the vowel sound /*i*/, spoken by a male speaker (LRR). Figure 8.16a shows the signal energy (computed at a 10 kHz rate); Fig. 8.16b shows the normalized mean-squared error (V) (again computed at a 10 kHz rate) for a 14 pole (p=14) analysis with a 20 msec (N=200) frame size for the covariance method; Fig. 8.16c shows the normalized mean-squared error for the autocorrelation method using a Hamming window; and Fig. 8.16d shows the normalized mean-squared error for the autocorrelation method using a rec-

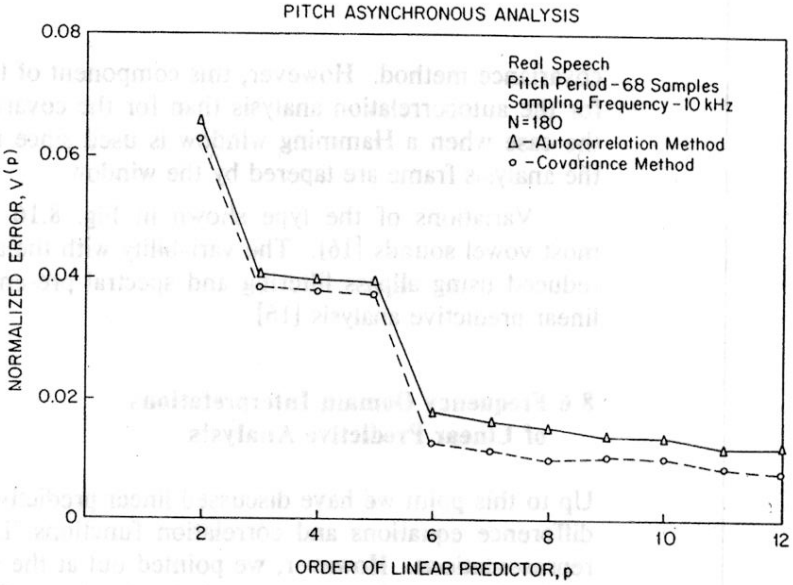


Fig. 8.14 Variation of prediction error with predictor order for a voiced section of a natural vowel—pitch asynchronous analysis. (After Chandra and Lin [15].)

tangular window. The average pitch period for this speaker was 84 samples (8.4 msec); thus about 2.5 pitch periods were contained within the 20 msec frame. For the covariance method the normalized error shows a substantial variation with the position of the analysis frame (i.e., the error is not a smooth function of time). This effect is essentially due to the large peaks in the error signal, $e(n)$, at the beginning of each pitch period as discussed previously. Thus, in this example, when the analysis frame is positioned to encompass 3 sets of error peaks, the normalized error is much larger than when only 2 sets of error peaks are included in the analysis interval. This accounts for the normalized error showing a fairly large discrete jump in level as each new error peak is included in the analysis frame. Each discrete jump of the normalized error is followed by a gradual tapering off and flattening of the normalized error. The exact detailed behavior of the normalized error between discrete jumps depends on details of the signal and the analysis method.

Figures 8.16c and 8.16d show somewhat different behavior of the LPC normalized error for the autocorrelation analysis method using a Hamming window, and a rectangular window respectively. As seen in this figure the normalized mean-squared error shows a substantial amount of high frequency variation, as well as a small amount of low frequency and pitch synchronous variation. The high frequency variation is due primarily to the error signal for the first p samples in which the signal is not linearly predictable. The magnitude of this variation is considerably smaller for the analysis using the Hamming window than for the analysis with the rectangular window due to the tapering of the Hamming window at the ends of the analysis window. Another component of the high frequency variation of the normalized error is related to the position of the analysis frame with respect to pitch pulses as discussed previously for the

covariance method. However, this component of the error is much less a factor for the autocorrelation analysis than for the covariance method — especially in the case when a Hamming window is used since new pitch pulses which enter the analysis frame are tapered by the window.

Variations of the type shown in Fig. 8.16 have been found typical for most vowel sounds [16]. The variability with the analysis frame position can be reduced using allpass filtering and spectral pre-emphasis of the signal prior to linear predictive analysis [16].

8.6 Frequency Domain Interpretations of Linear Predictive Analysis

Up to this point we have discussed linear predictive methods mainly in terms of difference equations and correlation functions; i.e., in terms of time domain representations. However, we pointed out at the beginning that the coefficients of the linear predictor are *assumed* to be the coefficients of the denominator of the system function that models the combined effects of vocal tract response, glottal wave shape, and radiation. Thus, given the set of predictor coefficients we can find the frequency response of the model for speech production simply

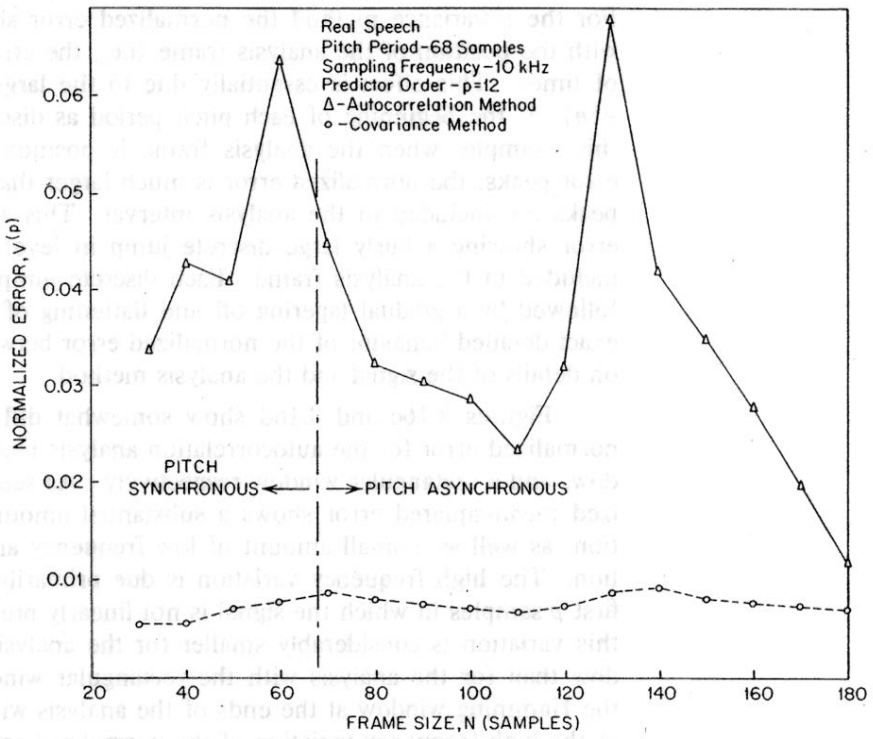


Fig. 8.15 Variation of prediction error with section length for a voiced section of natural speech. (After Chandra and Lin [15].)

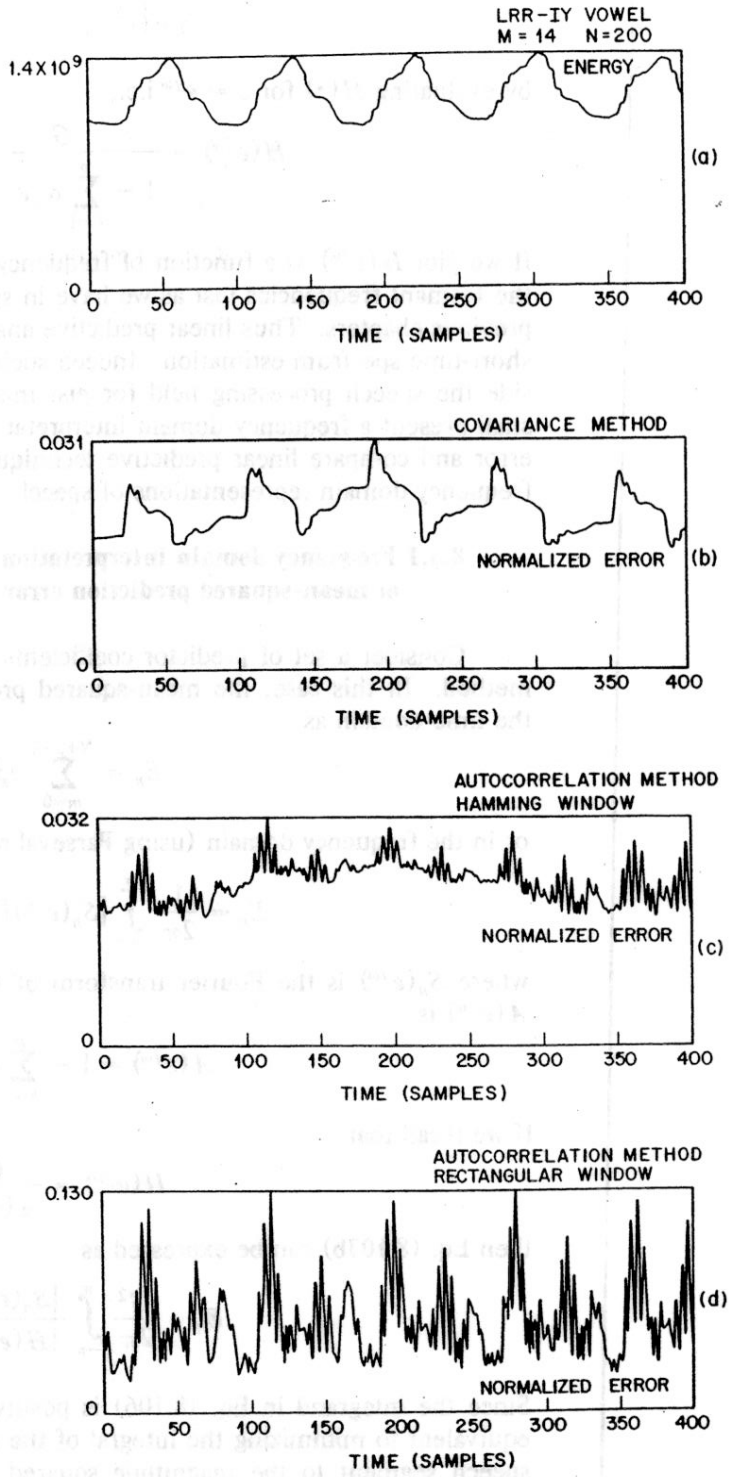


Fig. 8.16 Prediction error sequences for 200 samples of speech for three LPC systems. (After Rabiner et al. [16].)

by evaluating $H(z)$ for $z = e^{j\omega}$ i.e.,

$$H(e^{j\omega}) = \frac{G}{1 - \sum_{k=1}^{p} \alpha_k e^{-j\omega k}} = \frac{G}{A(e^{j\omega})} \tag{8.102}$$

If we plot $H(e^{j\omega})$ as a function of frequency[7] we should expect to see peaks at the formant frequencies just as we have in spectral representations discussed in previous chapters. Thus linear predictive analysis can be viewed as a method of short-time spectrum estimation. Indeed such techniques are widely applied outside the speech processing field for just this purpose [12]. In this section we shall present a frequency domain interpretation of the mean-squared prediction error and compare linear predictive techniques to other methods of estimating frequency domain representations of speech.

8.6.1 Frequency domain interpretation of mean-squared prediction error

Consider a set of predictor coefficients obtained using the autocorrelation method. In this case, the mean-squared prediction error can be expressed in the time-domain as

$$E_n = \sum_{m=0}^{N+p-1} e_n^2(m) \tag{8.103a}$$

or in the frequency domain (using Parseval's Theorem) as

$$E_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} |S_n(e^{j\omega})|^2 |A(e^{j\omega})|^2 d\omega \tag{8.103b}$$

where $S_n(e^{j\omega})$ is the Fourier transform of the segment of speech $s_n(m)$, and $A(e^{j\omega})$ is

$$A(e^{j\omega}) = 1 - \sum_{k=1}^{p} \alpha_k e^{-j\omega k} \tag{8.104}$$

If we recall that

$$H(e^{j\omega}) = \frac{G}{A(e^{j\omega})} \tag{8.105}$$

then Eq. (8.103b) can be expressed as

$$E_n = \frac{G^2}{2\pi} \int_{-\pi}^{\pi} \frac{|S_n(e^{j\omega})|^2}{|H(e^{j\omega})|^2} d\omega \tag{8.106}$$

Since the integrand in Eq. (8.106) is positive it follows that minimizing E_n is equivalent to minimizing the integral of the ratio of the energy spectrum of the speech segment to the magnitude squared of the frequency response of the linear system in the model for speech production.

[7]See Problem 8.2 for a consideration of how to evaluate $H(e^{j\omega})$ using the FFT.

In Section 8.2 it was shown that the autocorrelation function, $R_n(m)$, of the segment of speech, $s_n(m)$, and the autocorrelation function, $\tilde{R}(m)$, of the impulse response, $h(m)$, corresponding to the system function, $H(z)$, are equal for the first $(p+1)$ values. Thus, as $p \rightarrow \infty$ the respective autocorrelation functions are equal for all values and therefore

$$\lim_{p\to\infty} |H(e^{j\omega})|^2 = |S_n(e^{j\omega})|^2 \tag{8.107}$$

This implies that if p is large enough we can approximate the signal spectrum with arbitrarily small error with the all-pole model, $H(z)$.

It is interesting to note that even though Eq. (8.107) says that as $p \rightarrow \infty$, $|H(e^{j\omega})|^2 = |S_n(e^{j\omega})|^2$, it is not necessarily (or generally) true that $H(e^{j\omega}) = S_n(e^{j\omega})$ — i.e., the frequency response of the model need not equal the Fourier transform of the signal. This is so because $S_n(e^{j\omega})$ need not be minimum phase, whereas $H(e^{j\omega})$ is required to be minimum phase since it is the transfer function of an all-pole filter with poles inside the unit circle.

To illustrate the nature of the spectral modelling capability of linear predictive spectra, Fig. 8.17 (due to Makhoul [7]) shows a comparison between $20 \log_{10}|H(e^{j\omega})|$ and $20 \log_{10}|S_n(e^{j\omega})|$. The signal spectrum was obtained by an FFT analysis of a 20 msec section of speech (sampled at 20 kHz), weighted by a Hamming window as discussed in Chapter 6. The speech sound was the vowel /*ae*/. The LPC spectrum was that of a 28-pole predictor (p=28) obtained by the autocorrelation method [2]. The harmonic structure of the signal spectrum is clearly seen in this figure. A significant feature of the LPC spectral modelling can also be seen in this figure. This is the fact that the LPC spectrum matches the signal spectrum much more closely in the regions of large signal energy (i.e., near the spectrum peaks) than near the regions of low signal energy (i.e., near the spectral valleys). This is to be expected in view of Eq. (8.106) since regions where $|S_n(e^{j\omega})| > |H(e^{j\omega})|$ contribute more to the total error than regions where $|S_n(e^{j\omega})| < |H(e^{j\omega})|$. Thus the LPC spectral error criterion favors a good fit near the spectral peaks, whereas the fit near the spectral valleys is nowhere near as good.

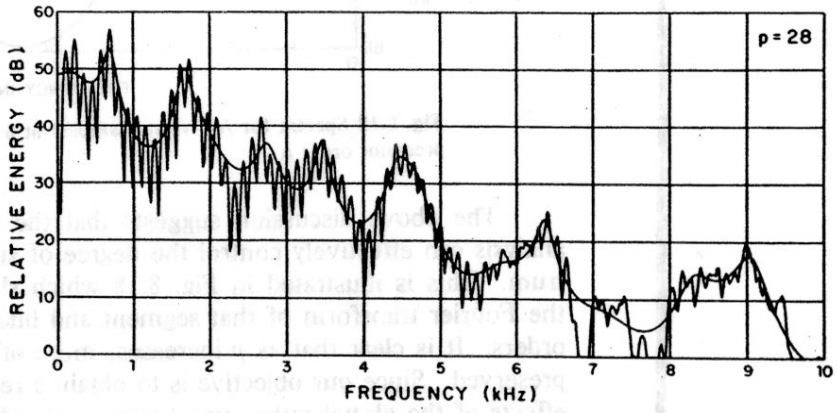


Fig. 8.17 28 pole fit to an FFT signal spectrum. (After Makhoul [17].)

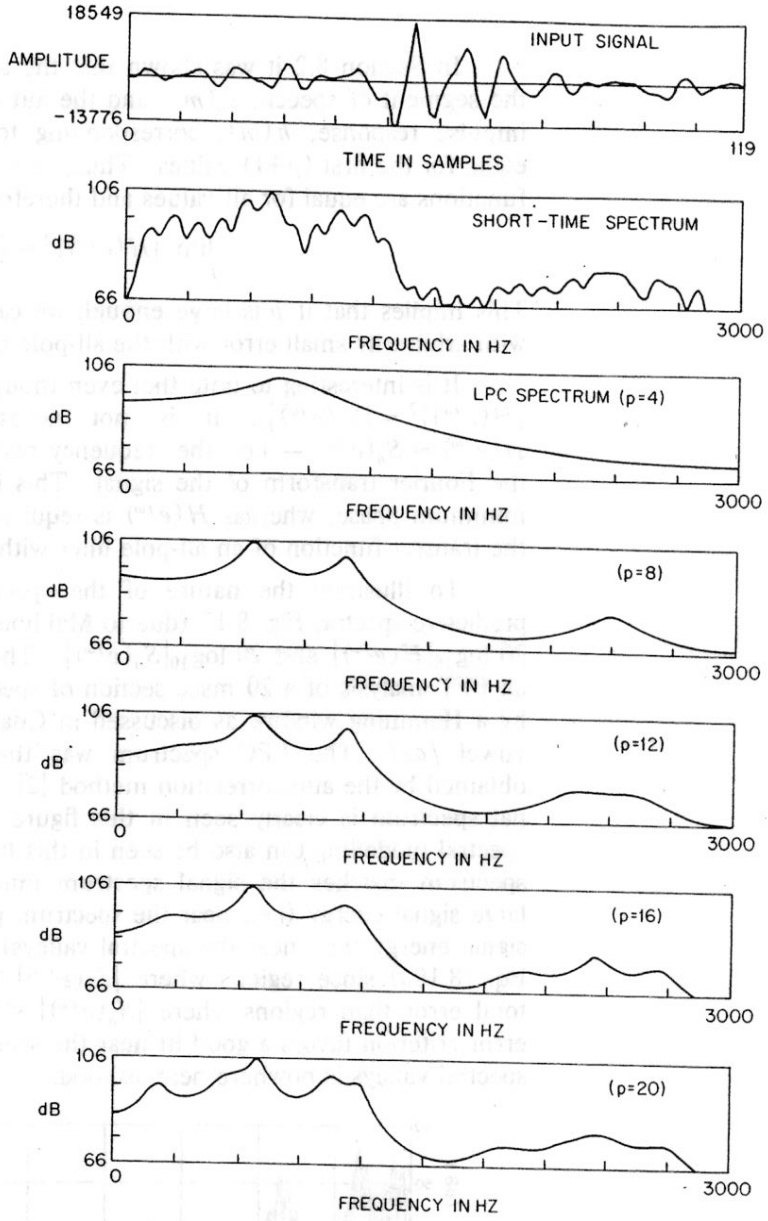


Fig. 8.18 Spectra for /*a*/ vowel sampled at 6 kHz for several values of predictor order *p*.

The above discussion suggests that the order p of the linear predictive analysis can effectively control the degree of smoothness of the resulting spectrum. This is illustrated in Fig. 8.18 which shows the input speech segment, the Fourier transform of that segment and linear predictive spectra for various orders. It is clear that as p increases, more of the details of the spectrum are preserved. Since our objective is to obtain a representation of only the spectral effects of the glottal pulse, vocal tract, and radiation, it is clear that we should

choose p as discussed before so that the formant resonances and the general spectrum shape are preserved.

It should be pointed out that we have assumed in this discussion that the predictor parameters were computed using the autocorrelation method. This was necessary because only in this case is the Fourier transform of the short-time autocorrelation function equal to the magnitude squared of the short-time Fourier transform of the signal. However this does not preclude the use of $H(e^{j\omega})$ as a spectrum estimate even if the predictor coefficients are estimated by the covariance method.

8.6.2 Comparison to other spectrum analysis methods

We have already discussed methods of obtaining the short-time spectrum of speech in Chapters 6 and 7. It is instructive to compare these methods with the spectrum obtained by linear predictive analysis.

As an example, Fig. 8.19 (due to Zue [10]) shows four log spectra of a section of the synthetic vowel /a/. The first two spectra were obtained using the short-time spectrum method discussed in Chapter 6. For the first spectrum, a section of 512 samples (51.2 msec) was windowed, and then transformed (using a 512 point FFT) to give the relatively narrow band spectral analysis shown at the top of Fig. 8.19. In this spectrum the individual harmonics of the excitation are clearly in evidence due to the relatively long duration of the window. For the second spectrum the analysis duration was decreased to 128 samples (12.8 msec) leading to a wideband spectral analysis. Now the excitation harmonics are not resolved; instead the overall spectral envelope can be seen. Although the formant frequencies are in evidence in this spectrum, it is not a simple matter to reliably locate or identify them. The third spectrum was obtained by homomorphic smoothing as discussed in Chapter 7. The unsmoothed spectrum was obtained from a 300 sample (30 msec) section using the FFT method described above. The smoothed spectrum shown in this figure was obtained by linear smoothing of the log spectrum. For this example the individual formants are well resolved and are easily measured from the smoothed spectrum using a simple peak picker. However, the bandwidths of the formants are not easily obtained from the homomorphically smoothed spectrum due to all the smoothing processes which have been used in obtaining the final spectrum. Finally the bottom spectrum is the result of a linear predictive analysis using $p = 12$ and a section of $N = 128$ samples (12.8 msec). A comparison of the linear prediction spectrum to the other spectra shows that the parametric representation appears to represent the formant structure very well with no extraneous peaks or ripples. This is due to the fact that the linear predictive model is very good for vowel sounds if the correct order, p, is used. Since the correct order can be determined knowing the speech bandwidth, the linear prediction method leads to very good estimates of the spectral properties due to the glottal pulse, vocal tract and radiation.

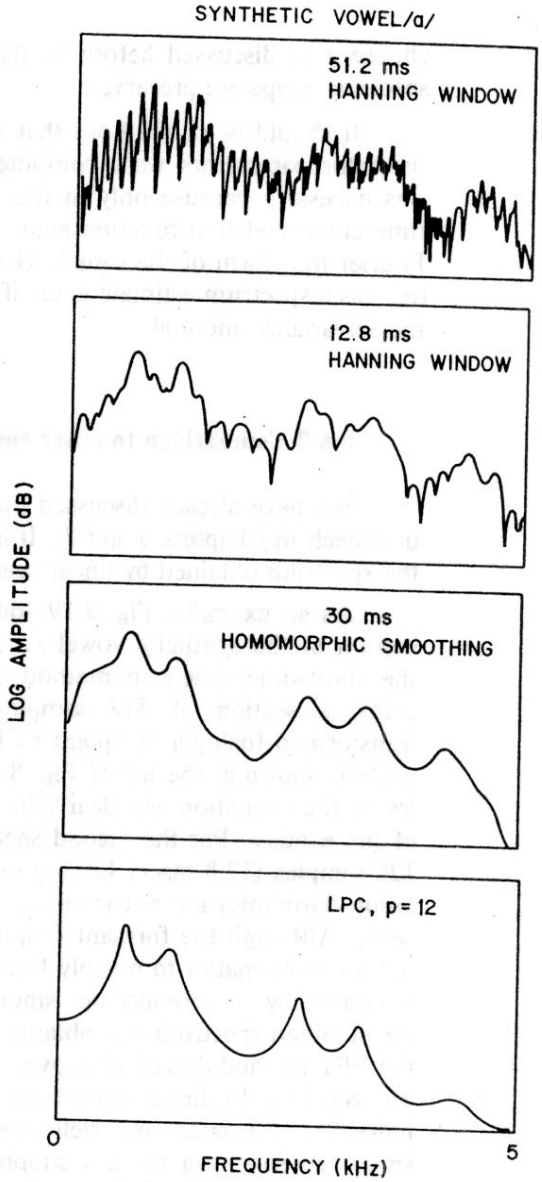


Fig. 8.19 Spectra of synthetic vowel /*a*/. (Afer Zue [10].)

Figure 8.20 shows a direct comparison of the spectra of a voiced section from natural speech obtained by both homomorphic smoothing and linear prediction. Although the formant frequencies are clearly in evidence in both plots, it can be seen that the LPC spectrum has fewer extraneous peaks than the homomorphic spectrum. This is because the LPC analysis assumed a value of $p = 12$ so that at most 6 resonance peaks could occur. For the homomorphic spectrum no such restriction existed. As noted above, the spectrum peaks from the LPC analysis are much narrower than the spectrum peaks from the homomorphic analysis due to the smoothing of the short-time log spectrum.

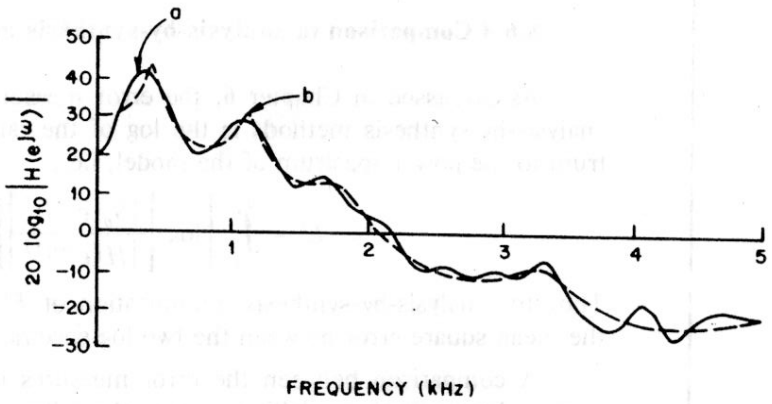


Fig. 8.20 Comparison of speech spectra obtained by (a) cepstrum smoothing; and (b) linear prediction.

8.6.3 Selective linear prediction

It is possible to apply the above ideas to a selected portion of the spectrum, rather than uniformly over the entire spectral range. This idea has been called selective linear prediction by Makhoul [8]. The reason this method is of potential value is that one can model only those regions of the spectrum which are important to the intended application. For example, a sampling rate of 20 kHz is required in many speech recognition applications to adequately represent the spectrum of fricatives. For voiced sounds one is generally interested in the region from 0 to about 4 kHz. For unvoiced sounds the region from 4 kHz to 8 kHz is generally of most importance. Using selective linear prediction the signal spectrum from 0 to 4 kHz can be modelled by a predictor of order p_1; whereas the region from 4 kHz to 8 kHz can be modelled by a different predictor of order p_2.

The way in which selective linear prediction is implemented is relatively straightforward. To model only the frequency region from $f = f_A$ to $f = f_B$, all that is required is a simple linear mapping of the frequency scale such that $f = f_A$ is mapped to $f' = 0$ and $f = f_B$ is mapped to $f' = \omega'/2\pi = 0.5$ (i.e., half the sampling frequency). The predictor parameters are computed by solving the predictor equations where the autocorrelation coefficients are obtained from

$$R'(i) = \frac{1}{2\pi} \int_{-\pi}^{\pi} |S_n(e^{j\omega'})|^2 \, e^{j\omega' i} d\omega' \tag{8.108}$$

Figure 8.21 (due to Makhoul [8]) illustrates the method of selective linear prediction. The signal spectrum is identical to the one of Fig. 8.17. The region from 0 to 5 kHz is modelled by a 14-pole predictor (p_1=14), whereas the region from 5-10 kHz is modelled independently by a 5-pole predictor (p_2=5). It can be seen that at 5 kHz, the model spectra show a discontinuity since there is no constraint that they agree at any frequency.

8.6.4 Comparison to analysis-by-synthesis methods

As discussed in Chapter 6, the error measure which is normally used in analysis-by-synthesis methods is the log of the ratio of the signal power spectrum to the power spectrum of the model, i.e.,

$$E' = \int_{-\pi}^{\pi} \left\{ \log \left[\frac{|S_n(e^{j\omega})|^2}{|H(e^{j\omega})|^2} \right] \right\}^2 d\omega \tag{8.109}$$

Thus for analysis-by-synthesis minimization of E' is equivalent to minimizing the mean square error between the two log spectra.

A comparison between the error measures used for LPC modelling and analysis-by-synthesis modelling leads to the following observations:

1. Both error measures are related to the ratio of the signal to model spectra.
2. Both error measures tend to perform uniformly over the whole frequency range.
3. Both error measures are suitable to selective error minimization over specified frequency regions.
4. The error criterion for linear predictive modelling places higher weight on frequency regions where $|S_n(e^{j\omega})|^2 > |H(e^{j\omega})|^2$ than when $|S_n(e^{j\omega})|^2 < |H(e^{j\omega})|^2$, whereas the error criterion for analysis-by-synthesis places equal weight on both these regions.

The conclusion which is drawn from these observations is that when dealing with signal spectra which are unsmoothed (as in Figure 8.17) the linear predictive error criterion yields better spectral matches than the analysis-by-synthesis method [7]. Furthermore the required computation for the linear predictive modelling is significantly less than for the analysis-by-synthesis method. If one is modelling smooth signal spectra (as might be obtained at the

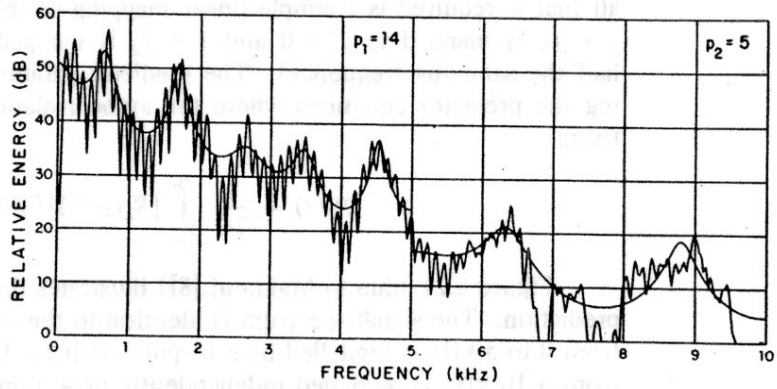


Fig. 8.21 Application of selective linear prediction to the signal spectrum of Fig. 8.17 with a 14-pole fit to the 0-5 kHz region and a 5-pole fit to the 5-10 kHz region. (After Makhoul [2].)

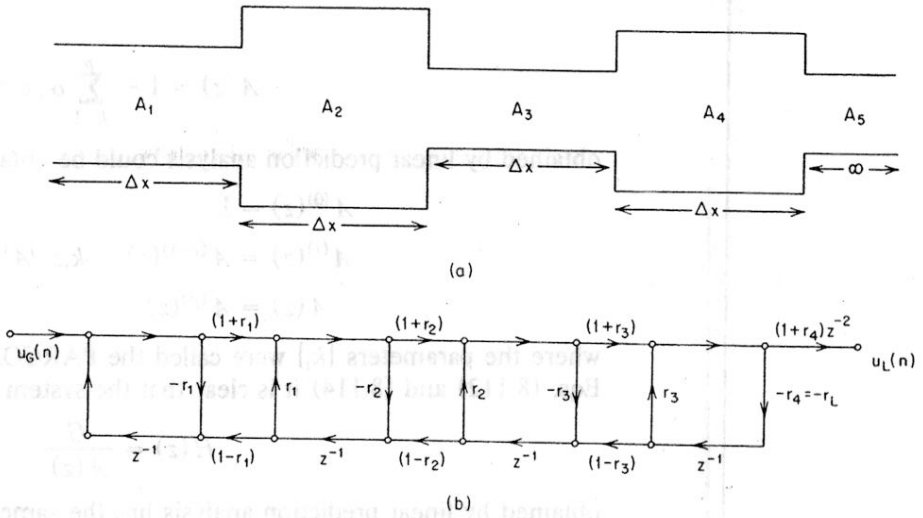


Fig. 8.22 (a) Lossless tube model terminated in infinitely long tube; (b) corresponding signal flow graph for infinite glottal impedance.

output of a filter bank) then both the LPC and analysis-by-synthesis methods give reasonably good fits to the spectra. In practice the analysis-by-synthesis method is applied almost always to this type of signal spectrum.

8.7 Relation of Linear Predictive Analysis to Lossless Tube Models

In Chapter 3 we discussed a model for speech production that consisted of a concatenation of N lossless acoustic tubes as shown in Fig. 8.22. The reflection coefficients r_k in Fig. 8.22b are related to the areas of the lossless tubes by

$$r_k = \frac{A_{k+1} - A_k}{A_{k+1} + A_k} \tag{8.110}$$

In Section 3.3.4, the transfer function of such a system was derived subject to the condition that the reflection coefficient at the glottis was $r_G = 1$, i.e., the glottal impedance was assumed to be infinite. In Section 3.3.4, the system function of a system such as shown in Fig. 8.22 was shown to be

$$V(z) = \frac{\prod_{k=1}^{N} (1+r_k) z^{-N/2}}{D(z)} \tag{8.111}$$

where $D(z)$ satisfies the polynomial recursion

$$D_0(z) = 1 \tag{8.112a}$$

$$D_k(z) = D_{k-1}(z) + r_k z^{-k} D_{k-1}(z^{-1}) \tag{8.112b}$$

$$D(z) = D_N(z) \tag{8.112c}$$

All of this is very reminiscent of the discussion of the lattice formulation in Section 8.3.3. Indeed, there it was shown that the polynomial

$$A(z) = 1 - \sum_{k=1}^{p} \alpha_k z^{-k} \tag{8.113}$$

obtained by linear prediction analysis could be obtained by the recursion

$$A^{(0)}(z) = 1 \tag{8.114a}$$

$$A^{(i)}(z) = A^{(i-1)}(z) - k_i z^{-i} A^{(i-1)}(z^{-1}) \tag{8.114b}$$

$$A(z) = A^{(p)}(z) \tag{8.114c}$$

where the parameters $\{k_i\}$ were called the PARCOR coefficients. By comparing Eqs. (8.112) and (8.114) it is clear that the system function

$$H(z) = \frac{G}{A(z)} \tag{8.115}$$

obtained by linear prediction analysis has the same form as the system function of a lossless tube model consisting of p sections. If

$$r_i = -k_i \tag{8.116}$$

then it is clear that

$$D(z) = A(z) \tag{8.117}$$

Using Eq. (8.110) and Eq. (8.116) it is easy to show that the areas of the equivalent tube model are related to the PARCOR coefficients by

$$A_{i+1} = \left(\frac{1-k_i}{1+k_i}\right) A_i \tag{8.118}$$

Note that the PARCOR coefficient gives us a ratio between areas of adjacent sections. Thus the areas of the equivalent tube model are not absolutely determined and any convenient normalization will produce a tube model with the same transfer function.

It should be pointed out that the "area function" obtained using Eq. (8.118) cannot be said to be the area function of the human vocal tract. However, Wakita [17] has shown that if pre-emphasis is used prior to linear predictive analysis to remove the effects due to the glottal pulse and radiation, then the resulting area functions are often very similar to vocal tract configurations that would be used in human speech.

8.8 Relations Between the Various Speech Parameters

Although the set of predictor coefficients, α_k, $1 \leqslant k \leqslant p$, is often thought of as the basic parameter set of the linear predictive analysis, it is straightforward to transform this set of coefficients to a number of other parameter sets, to obtain alternative representations of speech. Such alternative representations often are more convenient for applications of linear predictive analysis. In this section we discuss how other useful parameter sets can be obtained directly from LPC coefficients [1,2].

8.8.1 Roots of the predictor polynomial

Perhaps the simplest alternative to the predictor parameters is the set of roots of the polynomial

$$A(z) = 1 - \sum_{k=1}^{p} \alpha_k z^{-k} = \prod_{k=1}^{p} (1 - z_k z^{-1}) \tag{8.119}$$

That is, the roots $\{z_i,\ i=1,2,...,p\}$ are an equivalent representation of $A(z)$. If conversion of the z-plane roots to the s-plane is desired, this can be achieved by setting

$$z_i = e^{s_i T} \tag{8.120}$$

where $s_i = \sigma_i + j\Omega_i$ is the s-plane root corresponding to z_i in the z-plane. If $z_i = z_{ir} + jz_{ii}$ then

$$\Omega_i = \frac{1}{T} \tan^{-1}\left(\frac{z_{ii}}{z_{ir}}\right) \tag{8.121}$$

and

$$\sigma_i = \frac{1}{2T} \log(z_{ir}^2 + z_{ii}^2) \tag{8.122}$$

Equations (8.121) and (8.122) are useful for formant analysis applications of LPC analysis systems.

8.8.2 Cepstrum

Another alternative to the LPC coefficients is the cepstrum of the impulse response of the overall LPC system. If the overall LPC system has transfer function $H(z)$ with impulse response $h(n)$ and complex cepstrum $\hat{h}(n)$ then it can be shown that $\hat{h}(n)$ can be obtained from the recursion

$$\hat{h}(n) = \alpha_n + \sum_{k=1}^{n-1} \left(\frac{k}{n}\right) \hat{h}(k) \alpha_{n-k} \quad 1 \leqslant n \tag{8.123}$$

where

$$H(z) = \sum_{n=0}^{\infty} h(n) z^{-n} = \frac{G}{1 - \sum_{k=1}^{p} \alpha_k z^{-k}} \tag{8.124}$$

8.8.3 Impulse response of the all-pole system

The impulse response, $h(n)$, of the all-pole system with the transfer function of Eq. (8.124) can be solved for recursively from the LPC coefficients as

$$h(n) = \sum_{k=1}^{p} \alpha_k h(n-k) + G\delta(n) \qquad 0 \leqslant n \tag{8.125}$$

where $h(n)$ is assumed (by definition) to be 0 for $n < 0$, and G is the amplitude of the excitation.

8.8.4 Autocorrelation of the impulse response

As discussed in Section 8.2, it is easily shown (see Problem 8.1) that the autocorrelation function of impulse response of the filter defined as

$$\tilde{R}(i) = \sum_{n=0}^{\infty} h(n)h(n-i) = \tilde{R}(-i) \tag{8.126}$$

satisfies the relations

$$\tilde{R}(i) = \sum_{k=1}^{p} \alpha_k \tilde{R}(|i-k|) \quad 1 \leqslant i \tag{8.127}$$

and

$$\tilde{R}(0) = \sum_{k=1}^{p} \alpha_k \tilde{R}(k) + G^2 \tag{8.128}$$

Equations (8.127) and (8.128) can be used to determine $\tilde{R}(i)$ from the predictor coefficients and vice versa.

8.8.5 Autocorrelation coefficients of the predictor polynomial

Corresponding to the predictor polynomial, or inverse filter,

$$A(z) = 1 - \sum_{k=1}^{p} \alpha_k z^{-k} \tag{8.129}$$

is the impulse response of the inverse filter

$$a(n) = \delta(n) - \sum_{k=1}^{p} \alpha_k \delta(n-k)$$

The autocorrelation function of the inverse filter impulse response is

$$R_a(i) = \sum_{k=0}^{p-i} a(k)a(k+i) \qquad 0 \leqslant i \leqslant p \tag{8.130}$$

8.8.6 PARCOR coefficients

For the autocorrelation method the predictor coefficients may be obtained from the PARCOR coefficients using the recursion

$$a_i^{(i)} = k_i \tag{8.131a}$$

$$a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)} \qquad 1 \leqslant j \leqslant i-1 \tag{8.131b}$$

with Eqs. (8.131a) and (8.131b) being solved for $i = 1, 2, \ldots, p$ and with the

final set being defined as

$$\alpha_j = a_j^{(p)} \qquad 1 \leqslant j \leqslant p \tag{8.131c}$$

Similarly the set of PARCORS may be obtained from the set of LPC coefficients using a backward recursion of the form

$$k_i = a_i^{(i)} \tag{8.132a}$$

$$a_j^{(i-1)} = \frac{a_j^{(i)} + a_i^{(i)} a_{i-j}^{(i)}}{1 - k_i^2} \qquad 1 \leqslant j \leqslant i-1 \tag{8.132b}$$

where i goes from p, to $p-1$, down to 1 and initially we set

$$a_j^{(p)} = \alpha_j \qquad 1 \leqslant j \leqslant p \tag{8.132c}$$

8.8.7 Log area ratio coefficients

An important set of equivalent parameters which can be derived from the PARCOR parameters is the log area ratio parameters defined as

$$g_i = \log\left[\frac{A_{i+1}}{A_i}\right] = \log\left(\frac{1-k_i}{1+k_i}\right) \qquad 1 \leqslant i \leqslant p \tag{8.133}$$

The g_i parameters are equal to the log of the ratio of the areas of adjacent sections of a lossless tube equivalent of the vocal tract having the same transfer function as the linear predictive model as discussed in Section 8.7. The g_i parameters have also been found to be especially appropriate for quantization by Makhoul [2] and others [1] because of the relatively flat spectral sensitivity of the $g_i's$.

The k_i parameters may be directly obtained from the g_i by the inverse transformation

$$k_i = \frac{1 - e^{g_i}}{1 + e^{g_i}}, \qquad 1 \leqslant i \leqslant p \tag{8.134}$$

8.9 Synthesis of Speech from Linear Predictive Parameters

Speech can be synthesized from the linear predictive analysis parameters in several different ways. The simplest way is to use a system which is the same parametric representation as was used in the analysis. Figure 8.23 shows a block diagram of such a speech synthesizer. The time varying control parameters needed by the synthesizer are the pitch period, a voiced/unvoiced switch, the gain or rms speech value, and the p predictor coefficients. The impulse generator acts as the excitation source for voiced sounds producing a pulse of unit amplitude at the beginning of each pitch period. The white noise generator acts as the excitation source for unvoiced sounds producing uncorrelated, uniformly distributed random samples with unity standard deviation, and zero mean. The selection between the two sources is made by the voiced/unvoiced

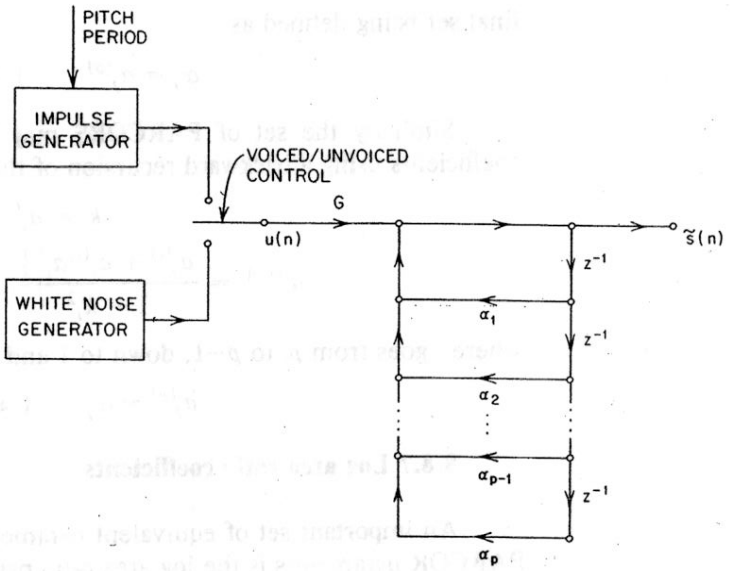


Fig. 8.23 Block diagram of linear predictive synthesizer.

control. The gain control G determines the overall amplitude of the excitation. The synthetic speech samples are determined by

$$\tilde{s}(n) = \sum_{k=1}^{p} \alpha_k \tilde{s}(n-k) + Gu(n) \tag{8.135}$$

A network which realizes Eq. (8.135) is shown in Fig. 8.23. This direct form network is the most simple and straightforward method for synthesizing speech from the predictor parameters. A total of p multiplies and p adds are required to generate each output sample.

In the synthesis model of Fig. 8.23 the synthesis parameters must be changed with time. Although the parameters are usually estimated at regular intervals during regions of voiced speech, the control parameters are changed at the beginning of each period. For unvoiced speech they are simply changed once per frame (i.e., every 10 msec for a 100 frame/sec rate). The updating of control parameters at the beginning of each pitch period (called pitch synchronous synthesis) has been found to be a much more effective synthesis strategy than the process of updating the parameters once each frame (called asynchronous synthesis). This requires that the control parameters be interpolated to obtain the values at the beginning of each pitch period. Atal has found that the pitch and gain parameters should be interpolated geometrically [3] (i.e., linearly on a log scale); however, due to stability constraints, the predictor parameters themselves cannot be interpolated. This is due to the fact that interpolation between two sets of stable predictor coefficients can lead to an unstable interpolated result. One way around this difficulty, according to Atal, is to interpolate the first p samples of the autocorrelation function of the impulse response of the filter of Fig. 8.21. Using the relations of Section 8.4, the predictor coefficients can be obtained from the first p samples of the autocorrelation of

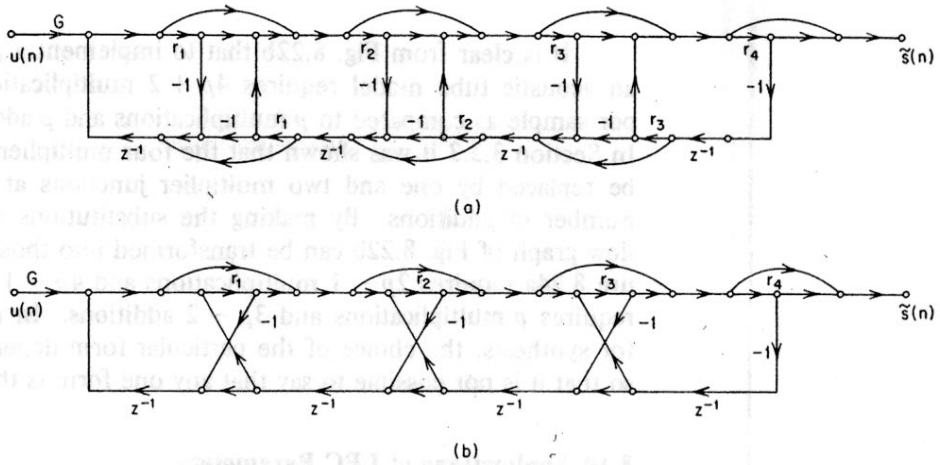


Fig. 8.24 Equivalent lossless tube models using (a) two multiplier junctions; and (b) one multiplier junction.

the impulse response, and vice versa. Furthermore, the interpolated autocorrelation coefficients always lead to a stable filter.[8]

The synthesizer of Fig. 8.23 has been used in a wide variety of simulations of LPC systems. Its main advantage is its simplicity and ease of implementation. Its main drawback is that it requires considerable computational accuracy to synthesize the speech because the structure is basically a direct form recursive structure which tends to be quite sensitive to changes in the coefficients. Perhaps the most attractive alternative to synthesis based on the predictor parameters is the use of the reflection coefficients or the PARCOR coefficients in a lossless tube equivalent. In other words, this direct form network in Fig. 8.23 can be replaced by a structure such as Fig. 8.22. The advantage of such a structure is that the multipliers are the reflection coefficients, $r_i = -k_i$, which have the property that they are bounded ($|k_i| < 1$), and also that they can be interpolated directly while maintaining a stable filter. Such structures are also less sensitive to quantization effects in finite word length implementations of the synthesizer than the direct form implementation of Fig. 8.23.

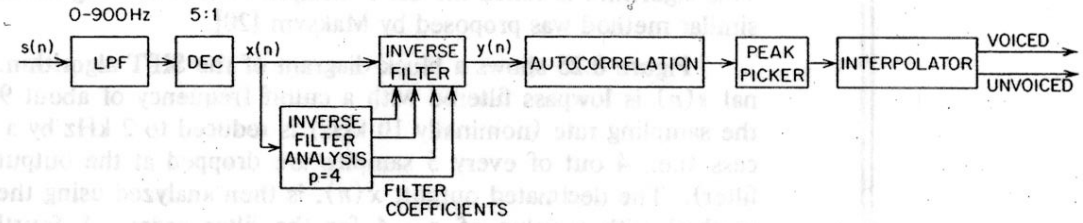


Fig. 8.25 Block diagram of the SIFT algorithm for pitch detection.

[8]Similarly the PARCOR coefficients or the log area ratio coefficients can be interpolated and the resulting system is guaranteed to be stable if the parameter sets which are being interpolated are stable.

It is clear from Fig. 8.22b that to implement a p^{th} order synthesis filter as an acoustic tube model requires $4p + 2$ multiplications and $2(p-1)$ additions per sample as compared to p multiplications and p additions for the direct form. In Section 3.3.3 it was shown that the four multiplier junctions in Fig. 8.22 can be replaced by one and two multiplier junctions at the expense of increased number of additions. By making the substitutions indicated in Fig. 3.41, the flow graph of Fig. 8.22b can be transformed into those shown in Fig. 8.24. Figure 8.24a requires $2p - 1$ multiplications and $4p - 1$ additions while Fig. 8.24b requires p multiplications and $3p - 2$ additions. In using lossless tube models for synthesis, the choice of the particular form depends on a variety of factors so that it is not possible to say that any one form is the most efficient.

8.10 Applications of LPC Parameters

As evidenced in the preceding sections of this chapter, the theory of linear prediction is highly developed. Based on this theory, and its implications, a large variety and range of applications of linear predictive analysis to speech processing has evolved. Schemes have been devised for estimating all the basic speech parameters from linear predictive analyses. Based on such analyses, vocoders have been studied extensively, leading to an understanding of the quantization properties of the various LPC representations. Finally these techniques have been used in many speech processing systems for speaker verification and identification, speech recognition, speech classification, speech dereverberation, etc. In the following sections and in Chapter 9 we present outlines of several representative methods for estimating speech parameters using linear predictive analysis.

8.10.1 Pitch detection using LPC parameters

We have already discussed how the error signal $e(n)$ from the LPC analysis can, in theory, be used to estimate the pitch period directly. Although this method will generally be capable of finding the correct period, a somewhat more sophisticated method of pitch detection was proposed by Markel [19]. This algorithm is called the SIFT (simple inverse filtering tracking) method. A similar method was proposed by Maksym [20].

Figure 8.25 shows a block diagram of the SIFT algorithm. The input signal $s(n)$ is lowpass filtered with a cutoff frequency of about 900Hz, and then the sampling rate (nominally 10 kHz) is reduced to 2 kHz by a decimation process (i.e., 4 out of every 5 samples are dropped at the output of the lowpass filter). The decimated output, $x(n)$, is then analyzed using the autocorrelation method with a value of $p = 4$ for the filter order. A fourth order filter is sufficient to model the signal spectrum in the frequency range 0-1 kHz because there will generally be only 1-2 formants in this range. The signal $x(n)$ is then inverse filtered to give $y(n)$, a signal with an approximately flat spectrum.[9]

[9]The output $y(n)$ is simply the prediction error for the fourth order predictor.

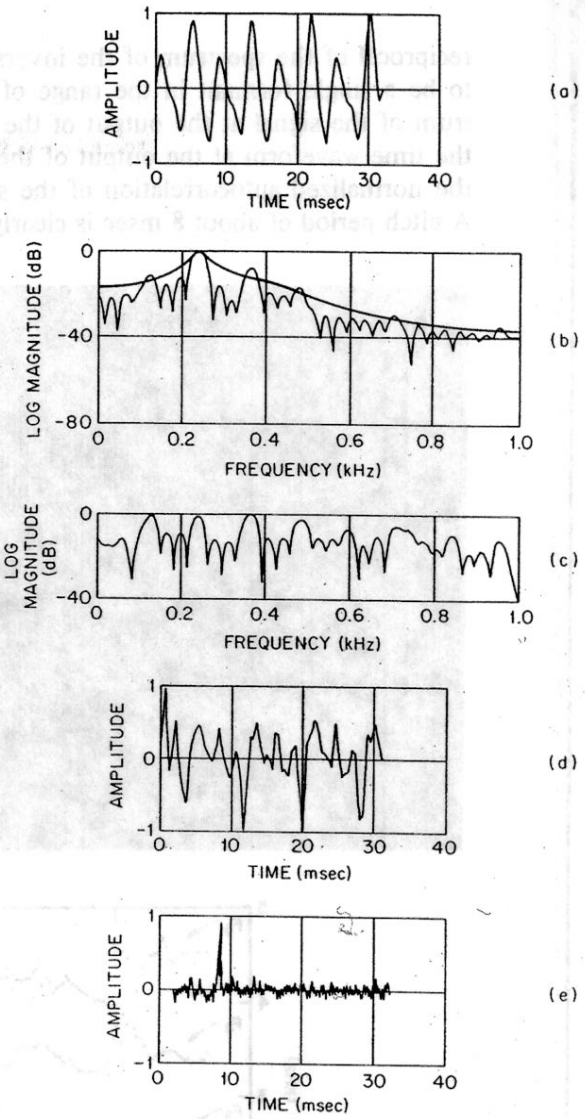
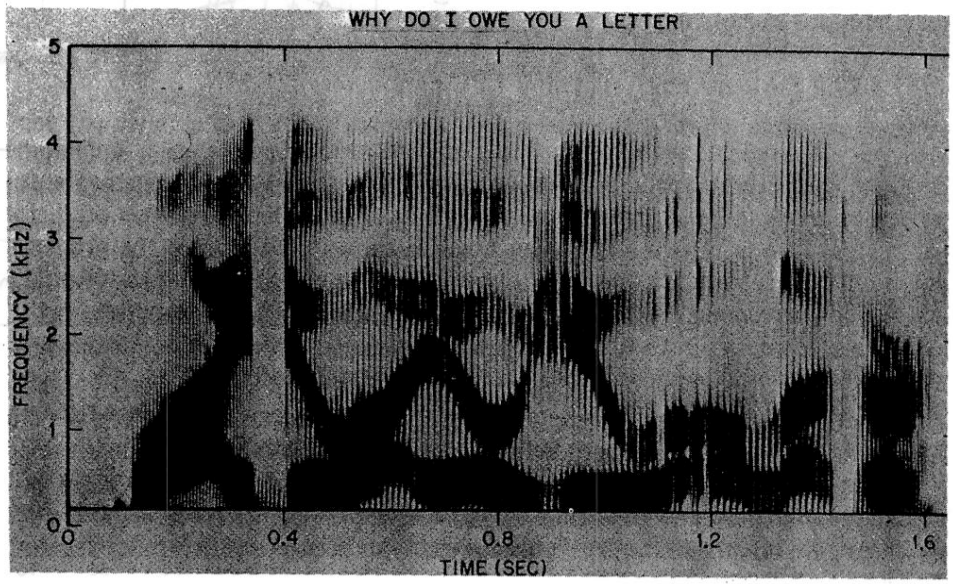


Fig. 8.26 Typical signals from the SIFT algorithm. (After Markel [19].)

Thus the purpose of the linear predictive analysis is to spectrally flatten the input signal, similar to the clipping methods discussed in Chapter 4. The short-time autocorrelation of the inverse filtered signal is computed and the largest peak in the appropriate range is chosen as the pitch period. To obtain additional resolution in the value of the pitch period, the autocorrelation function is interpolated in the region of the maximum value. An unvoiced classification is chosen when the level of the autocorrelation peak (suitably normalized) falls below a given threshold.

Figure 8.26 (due to Markel [19]) illustrates some typical waveforms obtained at several points in the analysis. Figure 8.26a shows a section of the input waveform being analyzed; Fig. 8.26b shows the input spectrum, and the

reciprocal of the spectrum of the inverse filter. For this example there appears to be a single formant in the range of 250 Hz. Figure 8.26c shows the spectrum of the signal at the output of the inverse filter, whereas Fig. 8.26d shows the time waveform at the output of the inverse filter. Finally Fig. 8.26e shows the normalized autocorrelation of the signal at the output of the inverse filter. A pitch period of about 8 msec is clearly in evidence.



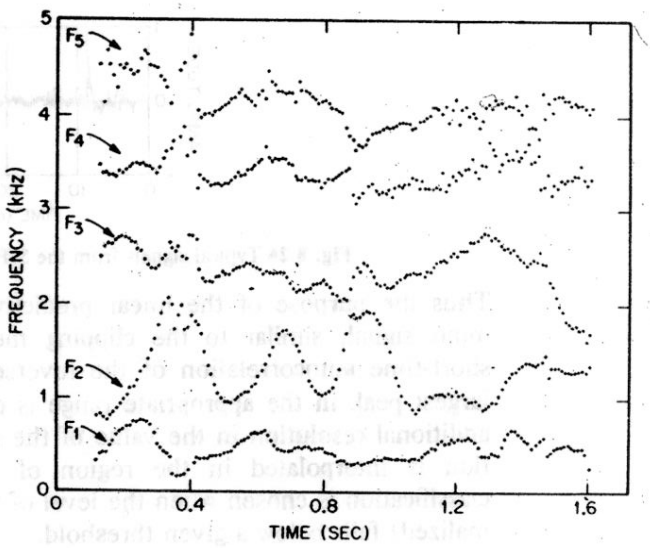


Fig. 8.27 (a) Spectrogram of original speech; (b) center frequencies of complex pole locations for 12^{th} order linear predictive analysis. (After Atal and Hanauer [3].)

The SIFT algorithm uses the linear predictive analysis to provide a spectrally flattened signal to facilitate pitch detection. To the extent that this spectral flattening is successful, the method appears to be a reasonably good one for pitch analysis. However, for high pitched speakers (such as children) the spectral flattening is generally unsuccessful due to the lack of more than one pitch harmonic in the band from 0 to 900 Hz (especially for telephone line inputs). For such speakers and transmission conditions, other pitch detection methods may be more successful.

8.10.2 Formant analysis using LPC parameters [21-23]

Linear predictive analysis of speech has several advantages, and some disadvantages when applied to the problem of estimating the formants for voiced sections of speech. Formants can be estimated from the predictor parameters in one of two ways. The most direct way is to factor the predictor polynomial and, based on the roots obtained, try to decide which are formants, and which correspond to spectral shaping poles [21,22]. The alternative way of estimating formants is to obtain the spectrum, and choose the formants by a peak picking method similar to the one discussed in Chapter 7 [23].

A distinct advantage inherent in the linear predictive method of formant analysis is that the formant center frequency and bandwidth can be determined accurately by factoring the predictor polynomial. Since the predictor order p is chosen a priori, the maximum possible number of complex conjugate poles which can be obtained is $p/2$. Thus the labelling problem inherent in deciding which poles correspond to which formants is less complicated for the LPC method since there are generally fewer poles to choose from than for comparable methods of obtaining the spectrum such as cepstral smoothing. Finally extraneous poles are generally easily isolated in the LPC analysis since their bandwidths are often very large, compared to what one would expect for bandwidths typical of speech formants. Figure 8.27 shows an example that illustrates that the pole locations do indeed give a good representation of the formant frequencies [3].

The disadvantage inherent in the LPC method is that an all-pole model is used to model the speech spectrum. For sounds such as nasals and nasalized vowels, although the analysis is adequate in terms of its spectral matching capabilities, the physical significance of the roots of the predictor polynomial is unclear. It is not clear if the roots correspond to the nasal zeros or the additional nasal poles, or if they are at all related to the expected resonances of the vocal tract. Another difficulty with the analysis is that although the bandwidth of the root is readily determined, it is generally not clear how it is related to the actual formant bandwidth. This is because the bandwidth of the root has been shown to be sensitive to the frame duration, frame position, and method of analysis.

With these advantages and disadvantages in mind, several methods have been proposed for estimating formants from LPC derived spectra using peak picking methods, and from the predictor polynomial by factoring methods.

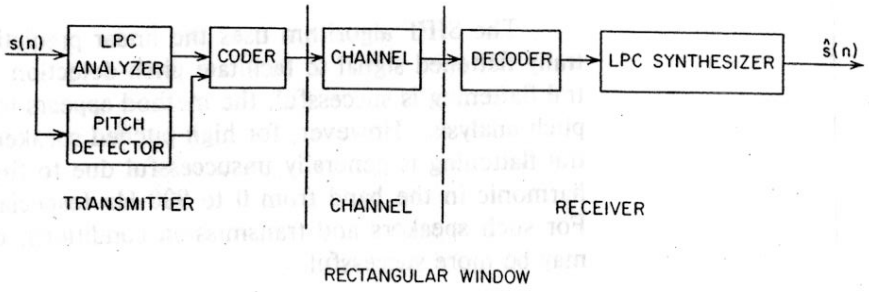


Fig. 8.28 Block diagram of LPC vocoder.

Once the candidates for the formants have been chosen, the techniques used to label these candidates — i.e., the assigning of a candidate to a particular formant, are similar to those used for any other analysis method. These include reliance on formant continuity, a need for spectral pre-emphasis to minimize the possibility of close formants merging, and the use of an off the unit circle contour for evaluating the LPC spectrum thereby sharpening the spectral peaks. Discussion of the various methods is given by Markel [21,22], Atal [3], Makhoul and Wolf [5], and McCandless [23].

8.10.3 An LPC vocoder — quantization considerations [24-25]

One of the most important applications of linear predictive analysis has been the area of low bit rate encoding of speech for transmission (the LPC vocoder) and storage (for computer voiced response systems). Figure 8.28 shows a block diagram of an LPC vocoder. The vocoder consists of a transmitter which performs the LPC analysis and pitch detection, and then codes the parameters for transmission, a channel over which the parameters are sent, and a receiver which decodes the parameters and synthesizes the output speech from them. We have already discussed both the analyzer and the synthesizer. We assume, for simplicity, that the channel is an error free transmission medium. Thus in this section we look at the coder and decoder to see which set of parameters is most appropriate for encoding at a given bit rate.

The basic LPC analysis parameters are the set of p predictor coefficients, the pitch period, a voiced/unvoiced parameter, and the gain parameter. Techniques for properly coding pitch, voiced/unvoiced switch, and the gain are fairly well understood. For the pitch period 6 bit quantization is adequate; for the voiced/unvoiced switch, 1 bit is required; and for the gain a total of about 5 bits distributed on a logarithmic scale are sufficient [3].

Although one could consider direct quantization of the predictor coefficients, this approach is not recommended because, to ensure stability of the predictor polynomial, a relatively high accuracy (8-10 bits per coefficient) is required. The reason for this is that small changes in the predictor coefficients can lead to relatively large changes in the pole positions. Thus direct quantization of the predictor coefficients is generally avoided.

This leaves open the question as to an appropriate parameter set for coding and transmission. Among the proposed parameter sets the most reasonable

candidates are the predictor polynomial roots, and the set of reflection coefficients. The predictor polynomial roots can readily be quantized in a manner which guarantees that the resulting polynomial is stable. This is because roots inside the unit circle guarantee stability of the predictor polynomial. Using this approach Atal [3] has found 5 bits per root (i.e., 5 bits for the center frequency and 5 bits for the bandwidth) are adequate to preserve the quality of the synthesized speech so as to make it essentially indistinguishable from speech synthesized from the unquantized parameters.

Using such a coding scheme, the overall bit rate for transmission or storage is $72 \cdot F_s$ bits per second where F_s is the number of frames per second which are stored or transmitted. Typical values for F_s are 100, 67, and 33 giving bit rates of 7200, 4800, and 2400 bits per second respectively.

Another interesting parameter set which can be easily quantized and for which stability can be guaranteed is the set of PARCOR coefficients, k_i. The stability condition on the $k_i's$ is $|k_i| < 1$ which is simple to preserve under quantization. Makhoul and Viswanathan [25] have found that the distribution of the reflection coefficients is highly skewed; thus a transformation of these parameters is required to optimally allocate the fixed number of bits in a reasonable manner. Using a spectral sensitivity measure, Makhoul and Viswanathan [25] found the optimal transformation to be of the form

$$g_i = f(k_i) = \log\left(\frac{1-k_i}{1+k_i}\right) = \log\left(\frac{A_{i+1}}{A_i}\right) \qquad 1 \leqslant i \leqslant p \tag{8.136}$$

where A_i is the area function of a lossless tube representation of the vocal tract. Thus the optimal parameter for linear encoding is the logarithm of the ratio of areas of a lossless tube representation of the vocal tract. It is easily seen that Eq. (8.136) maps the region $-1 \leqslant k_i \leqslant 1$ to $-\infty \leqslant g_i \leqslant \infty$. Using this transformation Atal [27] found that the coefficients g_i had a fairly uniform amplitude distribution, and low inter-parameter correlations; therefore these parameters were quite good for digital transmission. With this parameter set a total of about 5-6 bits per log area ratio is necessary to achieve the same quality synthetic speech as obtained from the uncoded parameters.

In all the above coding schemes it was assumed the parameters were being encoded using some type of PCM representation. It has recently been demonstrated by Sambur [26] that the coding techniques discussed in Chapter 5 can be applied directly to the various LPC parameter sets leading to further decreases in the required bit rates for transmission and storage. Using ADPCM coding of the predictor parameters, Sambur claims good quality speech with bit rates on the order of 1000-2000 bits per second.

8.10.4 Voice-excited LPC vocoders [27,28]

We have already shown that the weakest link in most vocoders is accurate estimation and representation of the excitation function. In Chapter 6 we discussed some vocoder systems which did not require direct estimation of pitch

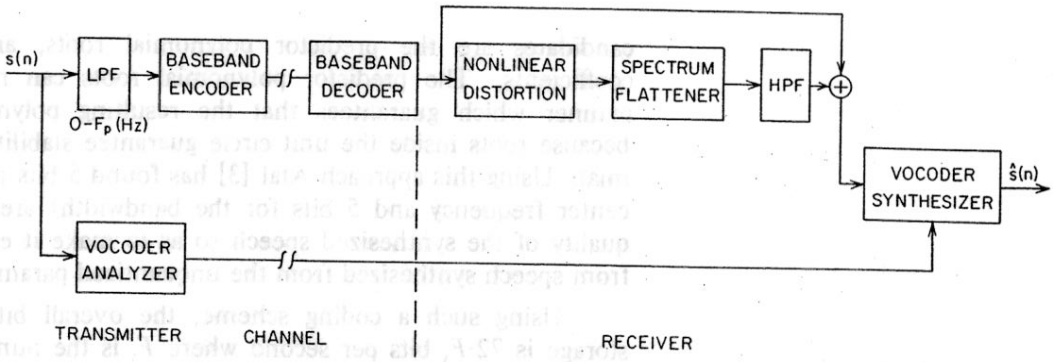


Fig. 8.29 Block diagram of a voice-excited LPC vocoder.

and voiced/unvoiced classification, but instead represented the excitation in terms of the phase (or phase derivative) of the signal. Another approach to avoiding direct estimation of excitation parameters for a vocoder is the voice-excited vocoder. Systems of this type have been studied by Atal et al. [27] and Weinstein [28]. Figure 8.29 shows a block diagram of a voice-excited LPC vocoder. There are two distinct transmission paths in this system; one producing a low frequency band of the direct signal, one producing the normal vocoder parameters (e.g., LPC coefficients, spectral magnitudes, etc.). The low frequency band, which can be coded using any of the methods described in Chapter 5, is used to generate the excitation signal for the synthesizer by an appropriate combination of nonlinear distortion and spectral flattening. The reason this procedure is effective is that the low frequency band contains all the necessary information about the excitation — i.e., it is periodic with the correct period for voiced speech, and it is noise-like for unvoiced speech. Thus, using such a scheme to generate the excitation eliminates the need for methods for estimating pitch, and voiced/unvoiced classification. However, this method has the disadvantage that additional information must be transmitted over the channel to accurately describe the low frequency band of the signal; thus voice-excited vocoders generally require somewhat higher bit rates than conventional vocoders. For example a voice-excited LPC vocoder requires on the order of 3000-4000 bps or about 1000-2000 bps more than the conventional LPC vocoder described in the previous section. The benefit obtained from the higher bit rates is an increased uniformity in the speech quality for different speakers and transmission conditions, due to the elimination of the pitch and voiced/unvoiced detector. The details of implementation of voice excited LPC vocoders are given by Atal et al. [27] and Weinstein [28].

8.11 Summary

In this chapter we have studied the technique of linear prediction of speech. We have primarily focused on the formulations which provide the most insight into the modeling of the process of speech production. We have discussed the issues involved with implementing these systems and have tried to compare the similarities and differences between the basic methods whenever possible.

REFERENCES

1. J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech,* Springer-Verlag, New York, 1976.
2. J. Makhoul, "Linear Prediction: A Tutorial Review," *Proc. IEEE,* Vol. 63, pp. 561-580, 1975.
3. B. S. Atal and S. L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," *J. Acoust. Soc. Am.,* Vol. 50, pp. 637-655, 1971.
4. F. I. Itakura and S. Saito, "Analysis-Synthesis Telephony Based Upon the Maximum Likelihood Method," *Proc. 6th Int. Congress on Acoustics,* pp. C17-20, Tokyo, 1968.
5. J. Makhoul, and J. Wolf, "Linear Prediction and the Spectral Analysis of Speech," *BBN Report No. 2304,* August 1972.
6. F. I. Itakura and S. Saito, "A Statistical Method for Estimation of Speech Spectral Density and Formant Frequencies," *Elec. and Comm. in Japan,* Vol. 53-A, No. 1, pp. 36-43, 1970.
7. J. Makhoul, "Spectral Linear Prediction: Properties and Applications," *IEEE Trans. on Acoustics, Speech, and Signal Proc.,* Vol. ASSP-23, No. 3, pp. 283-296, June 1975.
8. J. Makhoul, "Spectral Analysis of Speech by Linear Prediction," *IEEE Trans. on Audio and Electroacoustics,* Vol. AU-21, No. 3, pp. 140-148, June 1973.
9. J. D. Markel and A. H. Gray Jr., "On Autocorrelation Equations as Applied to Speech Analysis," *IEEE Trans. on Audio and Electroacoustics,* Vol. AU-21, pp. 69-79, April 1973.
10. V. Zue, "Speech Analysis by Linear Prediction," *MIT QPR No. 105, Research Lab of Electronics,* April 1972.
11. J. Makhoul, "Stable and Efficient Lattice Methods for Linear Prediction," *IEEE Trans. Acoustics, Speech, and Signal Proc.,* Vol. ASSP-25, No. 5, pp. 423-428, October 1977.
12. J. Burg, "A New Analysis Technique for Time Series Data," *Proc. NATO Advanced Study Institute on Signal Proc.,* Enschede Netherlands, 1968.
13. M. R. Portnoff, V. W. Zue, and A. V. Oppenheim, "Some Considerations in the Use of Linear Prediction for Speech Analysis," *MIT QPR No. 106., Research Lab of Electronics,* July 1972.
14. H. Strube, "Determination of the Instant of Glottal Closure from the Speech Wave," *J. Acoust. Soc. Am.,* Vol. 56, No. 5, pp. 1625-1629, November 1974.
15. S. Chandra and W. C. Lin, "Experimental Comparison Between Stationary and Non-stationary Formulations of Linear Prediction Applied to Speech," *IEEE Trans. Acoustics, Speech, and Signal Proc.,* Vol. ASSP-22, pp. 403-415, 1974.
16. L. R. Rabiner, B. S. Atal, and M. R. Sambur, "LPC Prediction Error-Analysis of Its Variation with the Position of the Analysis Frame," *IEEE Trans. Acoustics, Speech, and Signal Proc.,* Vol. ASSP-25, No. 5, pp. 434-442, October 1977.

17. H. Wakita, "Direct Estimation of the Vocal Tract Shape by Inverse Filtering of Acoustic Speech Waveforms," *IEEE Trans. on Audio and Electroacoustics,* Vol. AU-21, No. 5, pp. 417-427, October 1973.
18. E. M. Hofstetter, "An Introduction to the Mathematics of Linear Predictive Filtering as Applied to Speech Analysis and Synthesis," *Tech. Note 1973-36, MIT Lincoln Labs,* July 1973.
19. J. D. Markel, "The SIFT Algorithm for Fundamental Frequency Estimation," *IEEE Trans. on Audio and Electroacoustics,* Vol. AU-20, No. 5, pp. 367-377, December 1972.
20. J. N. Maksym, "Real-Time Pitch Extraction by Adaptive Prediction of the Speech Waveform," *IEEE Trans. on Audio and Electroacoustics,* Vol. AU-21, No. 3, pp. 149-153, June 1973.
21. J. D. Markel, "Application of a Digital Inverse Filter for Automatic Formant and F_o Analysis," *IEEE Trans. on Audio and Electroacoustics,* Vol. AU-21, No. 3, pp. 149-153, June 1973.
22. J. D. Markel, "Digital Inverse Filtering — A New Tool for Formant Trajectory Estimation," *IEEE Trans. on Audio and Electroacoustics,* Vol. AU-20, No. 2, pp. 129-137, June 1972.
23. S. S. McCandless, "An Algorithm for Automatic Formant Extraction Using Linear Prediction Spectra," *IEEE Trans. on Acoustics, Speech, and Signal Proc.,* Vol. ASSP-22, No. 2, pp. 135-141, April 1974.
24. J. D. Markel and A. H. Gray Jr., "A Linear Prediction Vocoder Simulation Based Upon the Autocorrelation Method," *IEEE Trans. on Acoustics, Speech, and Signal Proc.,* Vol. ASSP-22, No. 2, pp. 124-134, April 1974.
25. R. Viswanathan and J. Makhoul, "Quantization Properties of Transmission Parameters in Linear Predictive Systems," *IEEE Trans. on Acoustics, Speech, and Signal Proc.,* Vol. ASSP-23, No. 3, pp. 309-321, June 1975.
26. M. R. Sambur, "An Efficient Linear Prediction Vocoder," *Bell Syst. Tech. J.,* Vol. 54, No. 10, pp. 1693-1723, December 1975.
27. B. S. Atal, M. R. Schroeder, and V. Stover, "Voice-Excited Predictive Coding System for Low Bit-Rate Transmission of Speech," *Proc. ICC,* pp. 30-37 to 30-40, 1975.
28. C. J. Weinstein, "A Linear Predictive Vocoder with Voice Excitation," *Proc. Eascon,* September 1975.