Carnegie Mellon DSP-I Electrical & Computer ENGINEERING

Digital Signal Processing I (18-791)
Fall Semester, 2005

NOTES ON LINEAR PREDICTION AND LATTICE FILTERS

1. Introduction

Up to now we have discussed various approaches to discrete-time filter design that are all based on approximating the response of ideal lowpass, highpass, or bandpass filters, etc., with the designs involving various deterministic satisfaction of constraints such as passband and stopband ripple, and passband and stopband edge frequencies. In many actual situations we seek to design a filter that produces a particular deterministic frequency response, or perhaps an empirically-measured power spectral density function of a random process.

In this discussion we consider various types of *filter design by modeling*, where we try to come up with the best parametric approximation to an arbitrary frequency response.

In general, there are three types of models that can be considered. The *moving average* (MA) model has zeros but not poles:

$$H(z) = B(z)$$

The *autoregressive* (AR) model has poles but not zeros:

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

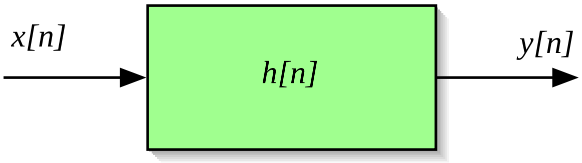
The third type of model has both poles and zeros and is called (unsurprisingly) the *autoregressive moving-average* (ARMA) model:

$$H(z) = \frac{B(z)}{A(z)}$$

Of the three types of filter design by modelling, the all-pole AR model is the most commonly used, largely because the design equations used to obtain the best-fit AR model are simpler than those used for MA or ARMA modelling. Serendipitiously, the all-pole model also has the ability to describe most types of speech sounds quite well, and for that reason it has become widely used in speech processing.

In these notes, we will begin by discussing a deterministic formulation of the all-pole model by matching unit sample responses. We will then reformulate the modelling problem in terms of stochastic signals, arriving at basically the same result. Finally, we will describe the simplest solution procedure to the all-pole model, Levinson-Durbin recursion.

2. Deterministic formulation of the all-pole model



Consider an arbitrary filter with the all-pole transfer function

$$H(z) = \frac{Y(z)}{X(z)} = \frac{G}{1 - \sum_{k=1}^{P} \alpha_k^{(P)} z^{-k}} \tag{1}$$

Note that the polynomial coefficients in the denominator α have the superscript P to indicate the order of the all-pole model. Because the representation is not orthogonal, *all* of the coefficients change when the order of the model changes.

The inverse *z*-transform of Eq. (1*)* is

$$y[n] = Gx[n] + \sum_{k=1}^{P} \alpha_k^{(P)} y[n-k] \tag{2}$$

Now we will consider the problem of obtaining a filter transfer function of the form in Eq. (1) to an arbitrary desired filter transfer function, $H_d(z)$. A reasonable objective is to minimize the average squared error between the magnitude of the frequency response of the desired filter desired $H_d(e^{j\omega})$ and the all-pole filter that is obtained $H(e^{j\omega})$

$$\xi^2 \equiv \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| H(e^{j\omega}) - H_d(e^{j\omega}) \right|^2 d\omega \tag{3}$$

Applying Parseval’s theorem, we obtain from Eq. (3)

$$\xi^2 = \sum_{n=0}^{N-1} (h[n] - h_d[n])^2 \tag{4}$$

Since $h[n]$ is the system’s response to the unit sample function $\delta[n]$, we obtain from Eq. (2)

$$h[n] = G\delta[n] + \sum_{k=1}^{P} \alpha_k^{(P)} h[n-k] \tag{5}$$

and

$$\xi^2 = \sum_{n=0}^{N-1}\left(\left(G\delta[n] + \sum_{k=1}^{P}\alpha_k^{(P)}h[n-k]\right) - h_d[n]\right)^2 \qquad (6)$$

For a particular model order *P* we solve for each α_k by rewriting Eq. (6) with a different internal dummy variable, obtaining the derivative of ξ^2 with respect to α_k and setting the result to zero.

$$\frac{d\xi^2}{d\alpha_k} = \sum_{n=0}^{N-1}\left(\left(G\delta[n] + \sum_{l=1}^{P}\alpha_l^{(P)}h[n-l]\right) - h_d[n]\right)h[n-k] = 0 \qquad (7)$$

$$\sum_{n=0}^{N-1}\left(G\delta[n]h[n-k] + \sum_{l=1}^{P}\alpha_l^{(P)}h[n-l]h[n-k]\right) = \sum_{n=0}^{N-1}h_d[n]h[n-k] \qquad (8)$$

Because we will be solving this equation for values of *k* ranging from 1 to $N-1$ and because $h[n]$ is causal, the term with *G* will not enter into the solution for the $\{\alpha_k\}$. Hence our final form is

$$\sum_{l=1}^{P}\alpha_l^{(P)}\sum_{n=0}^{N-1}h[n-l]h[n-k] = \sum_{n=0}^{N-1}h_d[n]h[n-k] \qquad (9)$$

This equation is known as the *Yule-Walker equation.* The inner sum of the first term can be considered to be proportional to the time-averaged autocorrelation function of the unit sample response $h[n]$. The second term can be considered to be the time-averaged cross-correlation function of $h[n]$ with the ideal sample response $h_d[n]$.

We will defer the solution to this equation until after the next section in which we derive a similar equation but for random signals.

3. Stochastic Formulation of the all-pole model: linear prediction

3.1. A quick review of random processes

The basic theory of random processes is outlined briefly in Appendix A of Oppenheim, Schafer, and Buck (1998), specifically Sections A.1 through A.4.

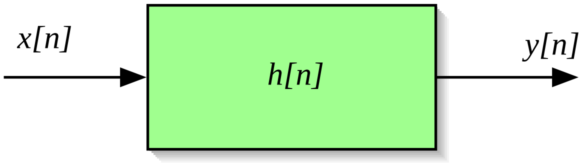
As Appendix A describes, a real zero-mean wide-sense stationary random processes $x[n]$, has the *autocorrelation function* $E[x[n+m]x[n]] \equiv \phi_{xx}[m]$. Note that $\phi_{xx}[m] = \phi_{xx}[-m]$. The *power spectral density function,* $\Phi_{xx}(e^{j\omega})$ describes the distribution of frequency components of the process. $\Phi_{xx}(e^{j\omega})$ is actually the discrete-time Fourier transform of $\phi_{xx}[m]$, as was proved by Wiener and Khintchine. A random process $x[n]$ is said to be *white* if $\Phi_{xx}(e^{j\omega})$ is constant over all values of ω. Hence, for a white random process $x[n]$, $\phi_{xx}[m]$, as the inverse DTFT of $\Phi_{xx}(e^{j\omega})$, will be of the form

$$\phi_{xx}[m] = \frac{N_0}{2}\delta[m] \tag{10}$$

This means that if $x[n]$ is a white a random process with zero mean, successive samples of $x[n]$ will be statistically independent.

3.2. Stochastic formulation of all-pole modelling

Now let us consider the problem of an all-pole LSI system once again, but this time with a random process as input. Specifically, let $x[n]$, the input to the filter, be a wide-sense stationary white random process.



Because the input to the linear filter is random, its output will be random as well. We assume that a random process $y[n]$ autocorrelation function $\phi_{yy}[m]$ and power spectral density function $\Phi_{yy}(e^{j\omega})$ is observed. The ensemble-average autocorrelation function $E[y[n+m]y[n]] \equiv \phi_{yy}[m]$ is typically estimated by the corresponding time-averaged autocorrelation function

$$\langle x[n+m]x[n]\rangle \equiv \lim_{N\to\infty}\frac{1}{N+1}\sum_{n=-N/2}^{N/2} x[n+m]x[n] \tag{11}$$

The time-averaged autocorrelation function $\langle x[n+m]x[n]\rangle$ is equal to the ensemble-averaged autocorrelation function $E[y[n+m]y[n]] = \phi_{yy}[m]$ if the random process is *ergodic*. Ergodicity is normally tacitly assumed for most random processes that you will encounter in the real world.

In the stochastic formulation of the LPC problem, we assume that a random process $y[n]$ is observed, and we attempt to develop the all-pole filter

$$H(z) = \frac{G}{1-\sum_{k=1}^{P}\alpha_k^{(P)}z^{-k}}$$

that would take a white process $x[n]$ as input and produce as output a random process with the power spectrum and autocorrelation function that most closely resembles $\Phi_{yy}(e^{j\omega})$ and $\phi_{yy}[m]$, respectively. As before, the difference equation that characterizes this system is

$$y[n] = \sum_{k=1}^{P}\alpha_k^{(P)}y[n-k] + Gx[n] \tag{12}$$

Note that this equation expresses the current value of the output $y[n]$ as a linear combination of the previous P and a second term representing perturbations due to the contribution of the random input process $x[n]$. It is common to think of the sum that constitutes the first term of Eq. (12) as a *linear estimate* $\hat{y}[n]$

of the current sample $y[n]$ based on the previous P estimates:

$$\hat{y}[n] = \sum_{k=1}^{P} \alpha_k^{(P)} y[n-k] \tag{13}$$

As it turns out, it can be shown that the filter coefficients α_k that provides the best all-pole model to the power spectral density function of a particular random process $y[n]$ are also the coefficients that minimize the mean squared error between the current value of $y[n]$ and its linear estimate $\hat{y}[n]$. In other words, we will let

$$e[n] = \hat{y}[n] - y[n] \tag{14}$$

and find the $\{\alpha_k\}$ that minimize

$$\xi^2 = E[(e^2)] = E[(\hat{y}[n] - y[n])^2] \tag{15}$$

This produces the expression

$$\xi^2 = E\left[\left(\sum_{k=1}^{P} \alpha_k^{(P)} y[n-k] - y[n]\right)^2\right] \tag{16}$$

which of course is rather similar to Eq. (6). The solution proceeds in a similar fashion.... we obtain the derivative of ξ^2 with respect to α_k and set the result to zero.

$$\frac{d\xi^2}{d\alpha_k} = E\left[\left(\sum_{l=1}^{P} \alpha_l^{(P)} y[n-l] - y[n]\right) y[n-k]\right] = 0 \tag{17}$$

$$E\left[\sum_{l=1}^{P} \alpha_l^{(P)} y[n-l] y[n-k]\right] = E[y[n]y[n-k]] \tag{18}$$

Since linear operations may be interchanged, we bring the expectation operator inside the sum on the left side of the equation. Using the definition above of the autocorrelation function Eq. (18) can be rewritten as

$$\sum_{l=1}^{P} \alpha_l^{(P)} \phi_{yy}[k-l] = \phi_{yy}[k] \tag{19}$$

This equation is very similar to Eq. (9). In fact, if we divide Eq. (9) by N, we can rewrite it as

$$\sum_{l=1}^{P} \alpha_l^{(P)} \langle h[n-l]h[n-k] \rangle = \langle h_d[n]h[n-k] \rangle \tag{20}$$

Note that both Eqs. (19) and (20) express the optimal LPC coefficients implicitly, and in terms of autocorrelation functions (of the observed random data in Eq. (19) and of deterministic unit sample response in Eq. (20)).

4. Solution of the LPC equations

4.1. General solution of the LPC equation

Let us assume for the sake of example that $P = 4$. Using the simplified notational conventions

$$\phi_{yy}[m] = \phi[m] \quad \text{and}$$

$$\alpha_k^{(P)} = \alpha_k$$

The system of equations in Eq. (19) can be written for $1 \le k \le P$ as

$$\alpha_1\phi[0] + \alpha_2\phi[1] + \alpha_3\phi[2] + \alpha_4\phi[3] = \phi[1] \tag{21}$$

$$\alpha_1\phi[1] + \alpha_2\phi[0] + \alpha_3\phi[1] + \alpha_4\phi[2] = \phi[2]$$

$$\alpha_1\phi[2] + \alpha_2\phi[1] + \alpha_3\phi[0] + \alpha_4\phi[1] = \phi[3]$$

$$\alpha_1\phi[3] + \alpha_2\phi[2] + \alpha_3\phi[1] + \alpha_4\phi[0] = \phi[4]$$

These equations can be written in matrix-vector form as

$$\begin{bmatrix} \phi[0] & \phi[1] & \phi[2] & \phi[3] \\ \phi[1] & \phi[0] & \phi[1] & \phi[2] \\ \phi[2] & \phi[1] & \phi[0] & \phi[1] \\ \phi[3] & \phi[2] & \phi[1] & \phi[0] \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} \phi[1] \\ \phi[2] \\ \phi[3] \\ \phi[4] \end{bmatrix} \tag{22}$$

Note that Eq. (22) is of the form

$$\underline{\boldsymbol{R}}\underline{\alpha} = \underline{\boldsymbol{P}} \tag{23}$$

where $\underline{\boldsymbol{R}}$ is a $P \times P$ matrix of autocorrelation coefficients, $\underline{\alpha}$ is a $P \times 1$ vector of the $\{\alpha_k\}$, and $\underline{\boldsymbol{P}}$ is a $P \times 1$ vector of autocorrelation coefficients. This equation is known as the *Wiener-Hopf equation*, which is encountered frequently in optimal signal processing.

A direct solution to the Wiener Hopf equation can be obtained by pre-multiplying both sides of Eq. (23) by the inverse of $\underline{\boldsymbol{R}}$:

$$\underline{\alpha} = \underline{\boldsymbol{R}}^{-1}\underline{\boldsymbol{P}} \tag{24}$$

The inversion of the $\underline{\boldsymbol{R}}$ matrix can be accomplished by Gaussian elimination and other similar techniques, which are $O(N^3)$ in computational complexity. In this case, however, a simpler solution known as Levinson-Durbin recursion is possible because the correlation matrix $\underline{\boldsymbol{R}}$ is *Toeplitz*; all the matrix elements of each diagonal, major and minor, are identical. As we will see, Levinson-Durbin recursion is $O(N^2)$ in complexity.

4.2. Levinson-Durbin recursion

Levinson-Durbin recursion provides for a faster solution for $\{\alpha_k\}$ in the system of equations

$$\begin{bmatrix} \phi[0] & \phi[1] & \phi[2] & \phi[3] \\ \phi[1] & \phi[0] & \phi[1] & \phi[2] \\ \phi[2] & \phi[1] & \phi[0] & \phi[1] \\ \phi[3] & \phi[2] & \phi[1] & \phi[0] \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} \phi[1] \\ \phi[2] \\ \phi[3] \\ \phi[4] \end{bmatrix}$$

for situations in which the matrix on the left side of the equation is Toeplitz. In our application, the $\{\phi[k]\}$ represent the autocorrelation coefficients of the random process $y[n]$. The solution $\{\alpha_k\}$ are the P^{th}-order predictor coefficients for the best-fit linear predictive model that transforms a white random process $x[n]$ into a random process that has autocorrelation coefficients $\{\phi_{yy}[k]\}$ according to the equation

$$y[n] = \sum_{k=1}^{P} \alpha_k^{(P)} y[n-k] + Gx[n]$$

In the equations below, $\phi[m]$ can represent either $\phi_{yy}[m]$ in the stochastic formulation or $\phi_h[m] = \langle h[n+m]h[n]\rangle$ in the deterministic formulations of linear prediction as outlined in the previous sections.

The equations of the Levinson-Durbin recursion, which are used to compute the corresponding reflection coefficients and LPC parameters are

$$E^{(0)} = \phi[0] \tag{25}$$

$$k_i = \frac{\left\{\phi[i] - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} \phi[i-j]\right\}}{E^{(i-1)}}\text{, calculated for } 1 \le i \le P \tag{26}$$

$$\alpha_i^{(i)} = k_i \tag{27}$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)}\text{, for } 1 \le j \le i-1 \tag{28}$$

$$E^{(i)} = (1-k_i^2)E^{(i-1)} \tag{29}$$

Equations (26) through (29) are solved recursively for $i = 1, 2, \ldots, P$ and the final solution is given by

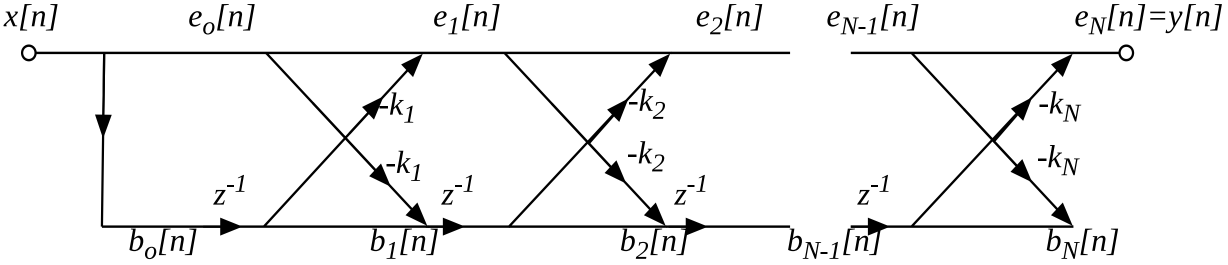
$$\alpha_j = \alpha_j^{(P)} \text{ for } 1 \le j \le P \tag{30}$$

The coefficients $\{k_i\}$ for $1 \le i \le P$ are referred to as the *reflection coefficients.* They constitute an alternate specification of the random process $y[n]$ that is as unique and complete as the LPC predictor coefficients $\left\{\alpha_k^{(P)}\right\}$. The reflection coefficients are actually far more robust to coefficient quantization than the predictor coefficients, so they are frequently the representation of choice in applications such as speech coding or speech compression.

If the magnitude of the reflection coefficients $|k_i|$ is less than 1 for $1 \le i \le P$, all of the roots of the polynomial $A(z) = 1 - \sum_{k=1}^{P} \alpha_k^{(P)} z^{-k}$ will lie inside the unit circle. This means that if $|k_i| < 1$, the resulting filter $H(z)$ will be stable. It can be shown that deriving the $\{k_i\}$ in the fashion described above using Levinson-Durbin recursion guarantees that $|k_i| < 1$.

We will make extensive use of the reflection coefficients $\{k_i\}$ in our discussion of lattice filters.

5. The FIR lattice filter



Consider the basic lattice filter structure in the figure above. It should be obvious that this is an FIR filter structure, as it contains no feedback loops. In addition, if we let the input $x[n]$ be equal to $\delta[n]$, we can observe easily by inspection that $h[0] = 1$ and $h[N] = -k_N$. The value of $h[n]$ for other values of *n* is obtained by observing all the different ways of passing a signal through the lattice while incurring exactly *n* delays, and adding all of the corresponding branch transmittances. It can be seen that the sample response will be a linear combination of the k_i.

5.1. Time-domain and frequency-domain characterization of the lattice filter

Although it may not be totally obvious from the figure above, the FIR lattice filter is defined by the following recursive relations:

$$x[n] = e_0[n] = b_0[n] \tag{31}$$

$$e_i[n] = e_{i-1}[n] - k_i b_{i-1}[n-1] \tag{32}$$

$$b_i[n] = -k_i e_{i-1}[n] + b_{i-1}[n-1] \tag{33}$$

$$y[n] = e_N[n] \tag{34}$$

Because the structure is FIR, we can make use of the following general characterization of its transfer function for the entire filter:

$$\frac{Y(z)}{X(z)} = 1 - \sum_{l=1}^{N} \alpha_l^{(N)} z^{-l} \equiv A(z) \tag{35}$$

We will also make use of the transfer function from the input to the $e_i[n]$ at a given stage of the lattice. For this, let

$$A_i(z) \equiv \frac{E_i(z)}{E_0(z)} = 1 - \sum_{l=1}^{i} \alpha_l^{(i)} z^{-l} \tag{36}$$

The corresponding transfer function from the input to the $b_i[n]$ at a given stage of the lattice is similarly

$$\tilde{A}_i(z) \equiv \frac{B_i(z)}{B_0(z)} \tag{37}$$

We note that $A_0(z) = \tilde{A}_0(z) = 1$ and $A_N(z) = (Y(z))/(X(z))$.

Using this notation, we can write the *z*-transforms of the equations that define the lattice as

$$E_0(z) = B_0(z) = X(z) \tag{38}$$

$$E_i(z) = E_{i-1}(z) - k_i z^{-1} B_{i-1}(z) \tag{39}$$

$$B_i(z) = -k_i E_{i-1}(z) + z^{-1} B_{i-1}(z) \tag{40}$$

$$Y(z) = E_N(z) \tag{41}$$

It can be shown (I will add the proof as an appendix to this later) that if the $\{\alpha_l\}$ and $\{k_i\}$ are related by the Levinson-Durbin equation (and specifically Eq. (28) above), then

$$A_i(z) = A_{i-1}(z) - k_i z^{-i} A_{i-1}(z^{-1}) \tag{42}$$

and

$$\tilde{A}_i(z) = z^{-i} A_i(z^{-1}) \tag{43}$$

These equations are important because they enable us to develop a recursive characterization of the transfer function of the lattice filter stage by stage. Substituting Eq. (36) into Eq. (42) we obtain:

$$1 - \sum_{l=1}^{i} \alpha_l^{(i)} z^{-l} = 1 - \sum_{l=1}^{i-1} \alpha_l^{(i-1)} z^{-l} - k_i z^{-i} \left(1 - \sum_{l=1}^{i-1} \alpha_l^{(i-1)} z^{l}\right) \tag{44}$$

Matching the coefficients of an arbitrary term of power z^{-r} we obtain

$$-\alpha_r^{(i)} z^{-r} = -\alpha_r^{(i-1)} z^{-r} + k_i \alpha_{i-r}^{(i-1)} z^{-r} \text{ or, of course}$$

$$\alpha_r^{(i)} = \alpha_r^{(i-1)} - k_i \alpha_{i-r}^{(i-1)} \tag{45}$$

as specified by the Levinson-Durbin recursion. Hence, just as the standard FIR filter implements the unit sample response of a system, with the sample response values as the coefficients or parameters of the filter, the lattice filter implements the Levinson-Durbin recursion, with its reflection coefficients k_i as the parameters of the filter! The all-zero transfer function of this lattice filter is the reciprocal of the all-pole model used to describe the original random process, or in other words the filter $A(z)$ is the *inverse* of $H(z)$ in Eq. (1) if we set the gain parameter G equal to 1.

5.2. Recursive relationships between the LPC coefficients and reflection coefficients

In Sec. 3 above, we discussed how the LPC coefficients can be obtained from the autocorrelation coefficients of an observed random process. We also noted in that section that the reflection coefficients k_i completely specify the LPC characterization of a random process just as the LPC coefficients α_i do. In fact, given either set of coefficients, we can always obtain the other by a simple linear recursion.

Specifically, to convert from the reflection coefficients k_i to the LPC coefficients α_i, we use the recursion

Let $\alpha_i^{(i)} = k_i$ starting at $i = 1$

$$\alpha_l^{(i)} = \alpha_l^{(i-1)} - k_i \alpha_{i-l}^{(i-1)} \text{ for } 1 \le l \le i-1 \tag{46}$$

Repeat for $i = 1, 2, \ldots, N$

Similarly, we can convert from the LPC coefficients α_i to the reflection coefficients k_i if we have all of the $\alpha_l^{(i)}$ for $i = 1, 2, 3, \ldots, N$:

Let $k_N = \alpha_N^{(N)}$

Starting with $i = N$, let $k_i = \alpha_i^{(i)}$

$$\alpha_l^{(i-1)} = \frac{\alpha_l^{(i)} + k_i \alpha_{i-l}^{(i)}}{1 - k_i^2} \text{ for } i = N, N-1, \ldots, 2, 1 \tag{47}$$

5.3. Physical interpretation of the functions $e_i[n]$ and $b_i[n]$

Up until now we have been thinking of the functions $e_i[n]$ and $b_i[n]$ as arbitrary internal functions. Nevertheless, they each do have a physical meaning relating to linear prediction error. Consider first the transfer function to the functions in the upper "rail" of the lattice:

$$A_i(z) = \frac{E_i(z)}{E_0(z)} = \frac{E_i(z)}{X(z)} = 1 - \sum_{l=1}^{i} \alpha_l^{(i)} z^{-l} \tag{48}$$

Taking the inverse z-transform we obtain

$$e_i[n] = x[n] - \sum_{l=1}^{i} \alpha_l^{(i)} x[n-l] = x[n] - \hat{x}[n] \tag{49}$$

which is identical (to within a sign) to the linear prediction error defined in Eq. (14). Again, this expression describes the difference between the current sample $x[n]$ and the "best" linear prediction of $x[n]$ using the previous i samples. Hence the expression $e_i[n]$ is referred to as the i^{th}*-order forward prediction error.*

Let us now consider the functions $b_i[n]$ in the lower "rail" of the lattice. Combining Eqs. (37) and (43) we obtain

$$\tilde{A}_i(z) = z^{-i} A_i(z^{-1}) = \frac{B_i(z)}{B_0(z)} = \frac{B_i(z)}{X(z)} \text{ and} \tag{50}$$

$$\frac{B_i(z)}{X(z)} = z^{-i}\left(1 - \sum_{l=1}^{i} \alpha_l^{(i)} z^l\right) = z^{-i} - \sum_{l=1}^{i} \alpha_l^{(i)} z^{l-i} \tag{51}$$

Again, taking the inverse z-transform we obtain

$$b_i[n] = x[n-i] - \sum_{l=1}^{i} \alpha_l^{(i)} x[n+(l-i)] \tag{52}$$

Comparing Eqs. (49) and (52) we observe that $b_i[n]$ represents the difference between the $x[n-i]$, the value of the input function i samples ago, and some linear combination of the following i samples of the input, running from $x[n-(i-1)]$ right up to $x[n]$.. In fact, the same linear prediction coefficients are used, but they are applied backward. One way of thinking about this is that $b_i[n]$ is what we would have obtained if we calculated $e_i[n]$ but with the input function presented in time-reversed order. Because of all this, $b_i[n]$ is referred to as the i^{th}*-order backward prediction error.*

5.4. Deriving the reflection coefficients from the forward and backward prediction error

In Secs. 3 and 4 we derived the LPC coefficients α_i and the reflection coefficients k_i of the best-fit all-pole model to the samples of a random process by implementing the Levinson-Durbin recursion to solve the autocorrelation equations. As you will recall, equations were developed by starting with the difference equation relating the input and output,

$$y[n] = \sum_{m=1}^{P} \alpha_m^{(P)} y[n-m] + Gx[n]$$

and finding the values of the $\alpha_m^{(P)}$ that minimize the expected value of the square of the forward error.

Specifically, starting with

$$\xi^2 = E\left[\left(y[n] - \sum_{m=1}^{P} \alpha_m^{(P)} y[n-m]\right)^2\right]$$

we computed the partial derivative of ξ^2 with respect to each of the α_m we obtained the equations

$$\sum_{m=1}^{P} \alpha_m^{(P)} \phi_y[|i-m|] = \phi_y[i]$$

where

$$\phi_y[i] = E(y[n]y[n+i])$$

With knowledge of the values of the autocorrelation coefficients $\phi_y[i]$ for $i = 0, 1, 2, \ldots, P$ we can use the Levinson-Durbin recursion to obtain all the LPC coefficients $\alpha_m^{(i)}$ for model orders 1 through *P* and the reflection coefficients k_i.

We can also obtain estimates of the reflection coefficients k_i (and subsequently the LPC coefficients $\alpha_m^{(i)}$) using expressions for forward and backward error developed in the previous section. Specifically, if we let

$$\xi^2 = E[(e_i[n])^2]$$

we can compute the derivative of ξ^2 with respect to k_i using the expression for $e_i[n]$ in Eq. (49). Setting that derivative equal to zero provides a value of the reflection coefficient k_i of a given stage of the lattice filter

$$k_i^f = \frac{E(e_{i-1}[n]b_{i-1}[n-1])}{E(b_{i-1}^2[n-1])} \qquad (53)$$

where the superscript *f* in the symbol k_i^f reminds us that this version of the reflection coefficient was derived using the forward error $e_i[n]$

Note that this estimate for the reflection coefficient is expressed in terms of the expected values of the products of the forward and backward errors of the previous stage in the numerator, and the expected value of the square of the backward prediction error in the denominator. The expression in the numerator is actually the *cross-correlation* of the forward and backward error functions of the previous stage, and the expression in the denominator is the *energy* of the backward error of the previous stage. Because of these physical interpretations, this method of obtaining the estimate of the reflection coefficients is referred to as the *partial correlation* or *PARCOR* method. In the approach of Sec. 3 we began by calculating the autocorrelation functions of the input directly; in this approach the input autocorrelation is done indirectly, through a recursive computation of cross-correlation of prediction error functions. This approach has some very attractive statistical properties and is widely used.

Of course, there is nothing magic about the forward prediction error. We can just as easily perform a similar calculation with the backward prediction error $b_i[n]$. Performing a similar set of operations on the backward prediction error produces the very similar estimate for the reflection coefficient

$$k_i^b = \frac{E(e_{i-1}[n]b_{i-1}[n-1])}{E(e_{i-1}^2[n])} \tag{54}$$

Various methods have been proposed for combining the two estimates of the reflection coefficients obtained using the PARCOR method, k_i^f and k_i^b. For example, the *Itakura estimate* of the reflection coefficients is obtained by combining these two results according to the equation

$$k_l^I = \sqrt{k_l^f k_l^b} \tag{55}$$

The *Burg estimate* of the reflection coefficients produced by combining these two results according to the equation

$$k_l^B = \frac{2k_l^f k_l^b}{k_l^f + k_l^{fb}} \tag{56}$$

Although we derived the expressions for the reflection coefficients using functions derived from the *ensemble averaged* forward and backward error, $E(e_i[n])$ and $E(b_i[n])$, in practice we normally use the corresponding *time averages* of these functions such as

$$E[e_i[n]b_i[n-1]] \approx \langle e_i[n]b_i[n-1] \rangle = \frac{1}{N}\sum_{n=0}^{N-1} e_i[n]b_i[n-1] \tag{57}$$

Time averages are equal to the corresponding ensemble averages if the random processes concerned are all stationary and ergodic.

6. IIR lattice filters

As noted above, we have developed an all-zero lattice filter in the previous section with the transfer function

$$A(z) = 1 - \sum_{l=1}^{N} \alpha_l^{(N)} z^{-l} = \frac{E_N(z)}{E_0(z)}$$

Referring to the figure at the begining of Sec. 5, we note that the input is $x[n] = e_0[n]$ and the output is $y[n] = e_N[n]$. If we could maintain the same filter structure but interchange the input and output, we would obtain the transfer function

$$\frac{E_0(z)}{E_N(z)} = \frac{1}{1 - \sum_{l=1}^{N} \alpha_l^{(N)} z^{-l}} \tag{58}$$

which clearly is an all-pole transfer function, and in fact is exactly the transfer function of the original filter considered, $H(z)$, with the gain factor G set equal to 1.

Recall that the original definitions of the stages of the FIR lattice filter were

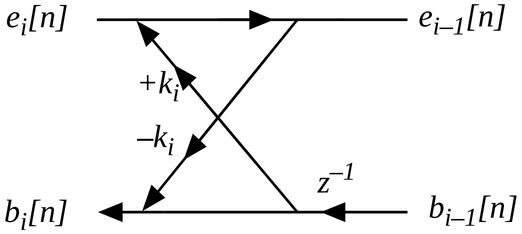
$$e_i[n] = e_{i-1}[n] - k_i b_{i-1}[n-1] \tag{59}$$

$$b_i[n] = -k_i e_{i-1}[n] + b_{i-1}[n-1] \tag{60}$$

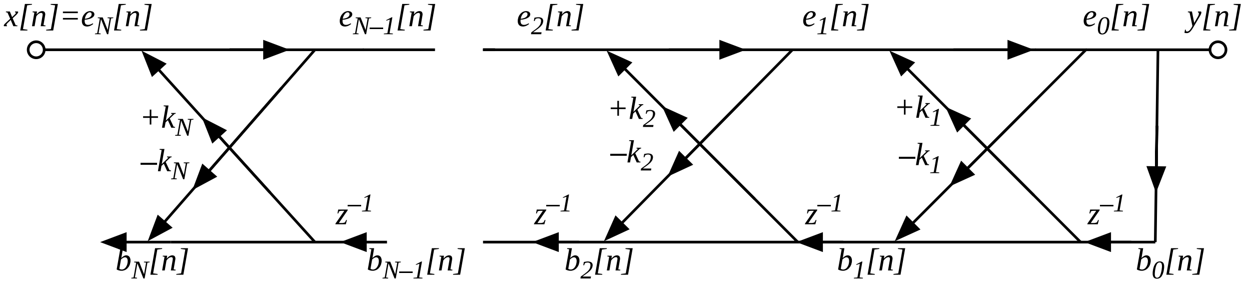
With a trivial amount of algebra, Eq. (59) can be rewritten as

$$e_{i-1}[n] = e_i[n] + k_i b_{i-1}[n-1] \tag{61}$$

Eqs. (60) and (61) suggest the following lattice stucture for a single stage:



Combining into multiple stages, we obtain the following IIR lattice structure:



Note that $e_N[n]$ is now the input and that $e_0[n]$ is the output. This filter will have the transfer function

$$H(z) = \frac{1}{1 - \sum_{l=1}^{N} \alpha_l^{(N)} z^{-l}} \tag{62}$$

where the LPC parameters are related to the reflection coefficients according to the usual Levinson-Durbin relationship. Since the filter is IIR with feedback loops, it does have the potential to be unstable. However, it is guaranteed to remain stable if

$$|k_i| < 1 \text{ for all } i. \tag{63}$$

7. Additional reading

The discussions on linear prediction were based on my class notes, which in turn are largely derived from the text *Digital Processing of Speech Signals* by L. R. Rabiner and R. W. Schafer (Prentice-Hall, 1978).

Additional material on lattice filters was derived from Section 6.6 of *Discrete-Time Signal Processing* by A. V. Oppenheim and R. W. Schafer (Prentice-Hall, 1988). This material, which was passed out in class, was omitted from the current second edition of that text. The newer text *Discrete-Time Speech Signal Processing* by T. F. Quatieri (Prentice-Hall, 2002) is also highly recommended and goes into deeper detail than Rabiner and Schafer in some aspects of the topics considered.