Adaptive Filters

V. John Mathews
Scott C. Douglas

Contents

Chapter 3

Linear Estimation Using Lattice Filters

In Chapter 2, we discussed linear estimation when the system model was realized in direct form, and the parameters of the direct form model was estimated from the statistics or measurements of the input signals. This chapter derives an alternate approach to linear estimation using what are known as *lattice filters*. There are several reasons why lattice filters are important in the study of adaptive filters. We enumerate a few such reasons here:

1. In general, stochastic gradient adaptive filters, which are discussed in Chapters 4 - 9, exhibit superior convergence behavior to their direct form counterparts.

2. Adaptive lattice filters are known to have better numerical properties than direct form adaptive filters.

3. The ideas employed in deriving the lattice filters are key to the development of computationally efficient recursive least-squares adaptive filters described in Chapter 11.

4. Increasing the model order by adding more sections in a lattice filter does not change the parameters of the sections that existed prior to the addition. This property does not hold for direct form systems.

The fundamental idea behind the development of lattice filters is the creation of a set of input signals that are orthogonal to each other by a process known as *Gram-Schmidt orthogonalization*. We start our discussion with the derivation of the Gram-Schmidt orthogonalization procedure.

3.1 Orthogonal Basis Vectors

Equation (2.48) describes the solution to a variety of problems in linear estimation. In many situations including those in adaptive filtering, we also desire a solution that can be easily computed. Unfortunately, solving for the optimal coefficients as in (2.48) is somewhat tedious. Inverting an $L \times L$ element autocorrelation matrix requires $O(L^3)$ arithmetical operations in the most general case. There are a few special cases in which the inversion of this matrix is much simpler. For example, if the autocorrelation matrix is diagonal, its inverse is given by

$$\mathbf{R}_{\mathbf{XX}}^{-1} = \mathrm{diag}^{-1}\left[r_1, r_2, \cdots, r_L\right] = \mathrm{diag}\left[r_1^{-1}, r_2^{-1}, \cdots, r_L^{-1}\right]. \tag{3.1}$$

Thus, only L divisions are required to compute the optimal coefficients from the relevant inner products.

Let us now consider a case where the vectors $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_L$ are *mutually orthogonal* in the sense that

$$\langle \mathbf{X}_i, \mathbf{X}_j \rangle = \begin{cases} \parallel \mathbf{X}_i \parallel^2 \text{ ; if } i = j \\ 0 \text{ ; otherwise.} \end{cases} \tag{3.2}$$

The reader can verify that the optimal solution for the coefficients is given by

$$\begin{aligned} \mathbf{W}_{\mathbf{opt}} &= \mathbf{R}_{\mathbf{XX}}^{-1}\mathbf{P}_{\mathrm{DX}} \\ &= \mathrm{diag}^{-1}\left[\parallel \mathbf{X}_1 \parallel^2, \ \parallel \mathbf{X}_2 \parallel^2, \ \cdots, \ \parallel \mathbf{X}_L \parallel^2\right] \begin{bmatrix} \langle \mathbf{D}, \mathbf{X}_1 \rangle \\ \langle \mathbf{D}, \mathbf{X}_2 \rangle \\ \vdots \\ \langle \mathbf{D}, \mathbf{X}_L \rangle \end{bmatrix} \\ &= \left[\frac{\langle \mathbf{D}, \mathbf{X}_1 \rangle}{\parallel \mathbf{X}_1 \parallel^2} \ \frac{\langle \mathbf{D}, \mathbf{X}_2 \rangle}{\parallel \mathbf{X}_2 \parallel^2} \ \cdots \ \frac{\langle \mathbf{D}, \mathbf{X}_L \rangle}{\parallel \mathbf{X}_L \parallel^2} \right]^T. \end{aligned} \tag{3.3}$$

The ith element of the optimal coefficient vector is given by

$$w_i = \langle \mathbf{D}, \mathbf{X}_i \rangle / \parallel \mathbf{X}_i \parallel^2; \ i = 1, 2, \cdots, L. \tag{3.4}$$

Note that $w_i\mathbf{X}_i$ is also the optimal estimate of $\mathbf{D}$ based on the single vector $\mathbf{X}_i$.

Thus, there exists a simple solution to the estimation problem whenever the input vectors $\mathbf{X}_{1,}\mathbf{X}_{2,} \cdots, \mathbf{X}_L$ are mutually orthogonal. The optimal estimate $\hat{\mathbf{D}}$ using these L vectors can be obtained in the following two steps:

1. Find the optimal estimate $\hat{\mathbf{D}}(\mathbf{X}_i)$ of $\mathbf{D}$ using each of the individual input vectors $\mathbf{X}_i$ for $i = 1, 2, \cdots, L$, given by

$$\hat{\mathbf{D}}(\mathbf{X}_i) = w_i\mathbf{X}_i = \frac{\langle \mathbf{D}, \mathbf{X}_i \rangle}{\parallel \mathbf{X}_i \parallel^2} \cdot \mathbf{X}_i \tag{3.5}$$

2. Sum these individual estimates $\hat{\mathbf{D}}(\mathbf{X}_1)$, $\hat{\mathbf{D}}(\mathbf{X}_2), \cdots, \hat{\mathbf{D}}(\mathbf{X}_L)$ to get the optimal estimate of $\mathbf{D}$ using $\mathbf{X}_{1,}\mathbf{X}_{2,} \cdots, \mathbf{X}_L$.

For clarity, we introduce the following notation for the optimal estimates. Let S_K denote the linear span of $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K$. Then the optimal estimate of $\mathbf{D}$ in S_K is denoted interchangeably by the notation $\hat{\mathbf{D}}(\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K)$ and $\hat{\mathbf{D}}(S_K)$. In the same spirit, we use $\hat{\mathbf{D}}(S_{K-1}, \mathbf{X}_K)$ or any other similar notation to denote this estimate. Using this notation, we can write $\hat{\mathbf{D}}(S_L)$ as

$$\hat{\mathbf{D}}(S_L) = \sum_{i=1}^{L} \hat{\mathbf{D}}(\mathbf{X}_i) \tag{3.6}$$

whenever $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_L$ are mutually orthogonal.

This result naturally leads to the following question: *Is it possible to appropriately transform a linear estimation problem involving L arbitrary vectors into one involving L mutually orthogonal vectors and get identical results?* We now show that this is in fact possible. The method requires finding a set of *orthogonal basis vectors* that has the same linear span as the original vectors $\mathbf{X}_1,\ \mathbf{X}_2,\ \cdots\ \mathbf{X}_L$.

Definition: A set of vectors $\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_L$ that belong to a properly-defined inner product space is said to be an *orthogonal basis set* for a subset S_L of the inner product space if and only if

(i)

$$\langle \mathbf{V}_i, \mathbf{V}_j \rangle = 0 \ ;\ i \neq j \tag{3.7}$$

(ii) The linear span of $\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_L$ is S_L.

3.1.1 A Two-Vector Example

Consider the task of estimating a vector $\mathbf{D}$ using two vectors $\mathbf{X}_1$and $\mathbf{X}_2$. Suppose that we already have obtained an estimate $\hat{\mathbf{D}}(\mathbf{X}_1)$ of $\mathbf{D}$ using $\mathbf{X}_1$. We know from (3.4) that

$$\hat{\mathbf{D}}(\mathbf{X}_1) = \frac{\langle \mathbf{D}, \mathbf{X}_1 \rangle}{\| \mathbf{X}_1 \|^2} \cdot \mathbf{X}_1. \tag{3.8}$$

Now, how can we use $\mathbf{X}_2$ to improve upon the estimate $\hat{\mathbf{D}}(\mathbf{X}_1)$ so that the estimation error is reduced? To answer this question, examine Figure 3.1, where we have decomposed $\mathbf{X}_2$ into two orthogonal components $\hat{\mathbf{X}}_2(\mathbf{X}_1)$ and $\mathbf{V}_2$, such that

$$\hat{\mathbf{X}}_2(\mathbf{X}_1) = \frac{\langle \mathbf{X}_2, \mathbf{X}_1 \rangle}{\| \mathbf{X}_1 \|^2} \cdot \mathbf{X}_1 \tag{3.9}$$

is the minimum squared-norm estimate of $\mathbf{X}_2$ using $\mathbf{X}_1$ and

$$\mathbf{V}_2 = \mathbf{X}_2 - \hat{\mathbf{X}}_2(\mathbf{X}_1) \tag{3.10}$$

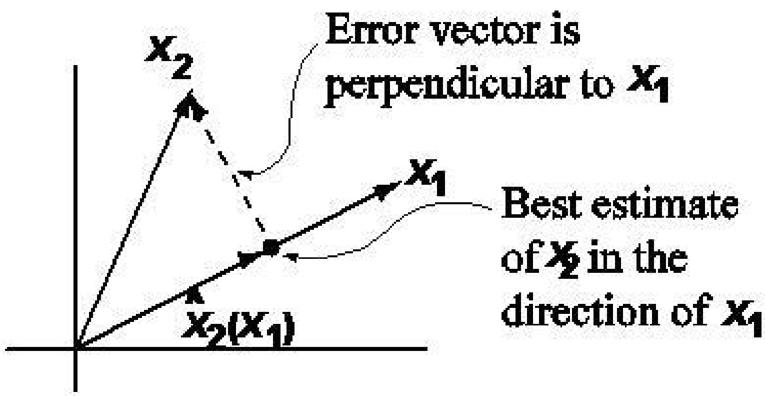

Figure 3.1: Decomposition of a vector into two orthogonal components.

is the error in this estimate. By virtue of the orthogonality principle, the vectors $\hat{\mathbf{X}}_2(\mathbf{X}_1)$ and $\mathbf{V}_2$ are orthogonal to each other. Furthermore, define $\mathbf{V}_1$ to be the same as $\mathbf{X}_1$. Since $\mathbf{X}_1$ and $\hat{\mathbf{X}}_2(\mathbf{X}_1)$ are aligned in the same direction, $\mathbf{V}_1$ and $\mathbf{V}_2$ are also mutually orthogonal.

Proposition: $(\mathbf{V}_1, \mathbf{V}_2)$ is an orthogonal basis set for S_2.

Proof: Recall that S_2 denotes the linear span of $\mathbf{X}_1$ and $\mathbf{X}_2$. We know from the statements above that $\mathbf{V}_1$ and $\mathbf{V}_2$ are orthogonal to each other. We only need to show that any linear combination of $\mathbf{X}_1$ and $\mathbf{X}_2$ can be expressed as a linear combination of $\mathbf{V}_1$ and $\mathbf{V}_2$ and vice versa. To prove this result, recall from (3.10) that

$$\mathbf{V}_2 = \mathbf{X}_2 - \alpha \mathbf{X}_1, \tag{3.11}$$

where α is the optimal coefficient for estimating $\mathbf{X}_2$ using $\mathbf{X}_1$. Let a and b be two arbitrary constants. We can write any linear combination $a\mathbf{V}_1 + b\mathbf{V}_2$ as

$$\begin{aligned} a\mathbf{V}_1 + b\mathbf{V}_2 &= a\mathbf{X}_1 + b(\mathbf{X}_2 - \alpha\mathbf{X}_1) \\ &= (a - b\alpha)\mathbf{X}_1 + b\mathbf{X}_2. \end{aligned} \tag{3.12}$$

Similarly, we can use (3.11) to express $\mathbf{X}_1$ and $\mathbf{X}_2$ as linear combinations of $\mathbf{V}_1$ and $\mathbf{V}_2$ as

$$\mathbf{X}_1 = \mathbf{V}_1 \tag{3.13}$$

and

$$\mathbf{X}_2 = \mathbf{V}_2 + \alpha\mathbf{V}_1, \tag{3.14}$$

respectively. Thus, any linear combination $\gamma\mathbf{X}_1 + \delta\mathbf{X}_2$ can be expressed as

$$\begin{aligned} \gamma\mathbf{X}_1 + \delta\mathbf{X}_2 &= \gamma\mathbf{V}_1 + \delta(\alpha\mathbf{V}_1 + \mathbf{V}_2) \\ &= (\gamma + \alpha\delta)\mathbf{V}_1 + \delta\mathbf{V}_2. \end{aligned} \tag{3.15}$$

Equations (3.12) and (3.15) together prove that the linear span of $\mathbf{X}_1$ and $\mathbf{X}_2$ is identical to the linear span of $\mathbf{V}_1$ and $\mathbf{V}_2$. For linear estimation problems, this result implies that the optimal linear estimate of $\mathbf{D}$ can be obtained using either $\mathbf{V}_1$ and $\mathbf{V}_2$ or $\mathbf{X}_1$ and $\mathbf{X}_2$. In the former case, the estimate will have the form

$$\begin{aligned}\hat{\mathbf{D}}(S_2) &= \hat{\mathbf{D}}(\mathbf{X}_1,\mathbf{X}_2) = \hat{\mathbf{D}}(\mathbf{V}_1,\mathbf{V}_2) \\ &= \frac{\langle \mathbf{D},\mathbf{V}_1\rangle}{\|\ \mathbf{V}_1\ \|^2}\mathbf{V}_1 + \frac{\langle \mathbf{D},\mathbf{V}_2\rangle}{\|\ \mathbf{V}_2\ \|^2}\mathbf{V}_2. \end{aligned} \tag{3.16}$$

Since the estimates are identical, the estimation errors due to $\hat{\mathbf{D}}(\mathbf{X}_1,\mathbf{X}_2)$ and $\hat{\mathbf{D}}(\mathbf{V}_1,\mathbf{V}_2)$ are also identical.

Example 3.1: Orthogonalization in a Five-Dimensional Euclidean Vector Space

Let $\mathbf{X}_1 = [1\ \ 2\ \ -1\ \ 2\ \ 0]^T, \mathbf{X}_2 = [2\ \ 2\ \ 1\ \ 1\ \ 1]^T$, and $\mathbf{D} = [-1\ \ 1\ \ 0\ \ 1\ \ -1]^T$ be three vectors in the five-dimensional Euclidean vector space with inner product given by

$$\langle \mathbf{X},\mathbf{Y}\rangle = \sum_{i=1}^{5} x_i y_i.$$

Find an orthogonal basis set for $\mathbf{X}_1$ and $\mathbf{X}_2$. Also, estimate $\mathbf{D}$ directly using $\mathbf{X}_1$ and $\mathbf{X}_2$ as well as with the orthogonal basis set so that the squared norm of the error vector is minimized.

Solution: The optimal coefficients can be found using (2.48) as

$$\begin{bmatrix} W_1 \\ W_2 \end{bmatrix} = \mathbf{R}_{\mathbf{XX}}^{-1}\mathbf{P}_{\mathbf{DX}} = \begin{bmatrix} 10 & 7 \\ 7 & 11 \end{bmatrix}^{-1} \begin{bmatrix} 3 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{33}{61} \\ \frac{-21}{61} \end{bmatrix}.$$

The optimal estimate is given by

$$\begin{aligned}\hat{\mathbf{D}}(S_2) &= \frac{33}{61}\mathbf{X}_1 - \frac{21}{61}\mathbf{X}_2 \\ &= \frac{1}{61}[-9\ \ 24\ \ -54\ \ 45\ \ -21] \end{aligned}.$$

An orthogonal basis set for S_2 is given by

$$\mathbf{V}_1 = \mathbf{X}_1 = [1\ \ 2\ \ -1\ \ 2\ \ 0]^T$$

and

$$\begin{aligned}\mathbf{V}_2 &= \mathbf{X}_2 - \frac{\langle \mathbf{X}_2,\mathbf{X}_1\rangle}{\|\ \mathbf{X}_1\ \|^2}\mathbf{X}_1 \\ &= \mathbf{X}_2 - \frac{7}{10}\mathbf{X}_1 \\ &= \frac{1}{10}[13\ \ 6\ \ 17\ \ -4\ \ 10]^T. \end{aligned}$$

The reader should verify that $\langle \mathbf{V}_1, \mathbf{V}_2 \rangle = 0$. Now,

$$\begin{aligned}\hat{\mathbf{D}}(\mathbf{V_1}) &= \frac{\langle \mathbf{D}, \mathbf{V}_1 \rangle}{\| \mathbf{V}_1 \|^2} \mathbf{V}_1 \\ &= \frac{3}{10} \mathbf{V}_1 \\ &= \frac{1}{10} [3 \;\; 6 \;\; -3 \;\; 6 \;\; 0]^T\end{aligned}$$

and

$$\begin{aligned}\hat{\mathbf{D}}(\mathbf{V}_2) &= \frac{\langle \mathbf{D}, \mathbf{V}_2 \rangle}{\| \mathbf{V}_2 \|^2} \mathbf{V}_2 \\ &= \frac{-21}{610} \mathbf{V}_2 \\ &= \frac{-21}{610} [13 \;\; 6 \;\; 17 \;\; -4 \;\; 10]^T\end{aligned}$$

We can easily verify that $\hat{\mathbf{D}}(\mathbf{V}_1) + \hat{\mathbf{D}}(\mathbf{V}_2)$ is identical to $\hat{\mathbf{D}}(\mathbf{X}_1, \mathbf{X}_2)$ computed earlier.

3.1.2 Extension to L Vectors

We can extend our previous results to the general L vector case. Let S_{L-1} denote the linear span of $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{L-1}$. Let $\hat{\mathbf{D}}(S_k)$ correspond to the vector closest to $\mathbf{D}$ in S_k for any positive integer k. Suppose that we have already computed $\hat{\mathbf{D}}(S_{L-1})$ and that we want to find the optimal estimate of $\mathbf{D}$ using $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{L-1}$ and $\mathbf{X}_L$. That is, we want to obtain a better estimate[1] $\hat{\mathbf{D}}(S_L)$ by employing an additional vector $\mathbf{X}_L$. How can we make use of the information we already have in S_{L-1} to compute $\hat{\mathbf{D}}(S_L)$?

The methods we use in this case are precisely the same as those employed in the two vector case. Let $\hat{\mathbf{X}}_L(S_{L-1})$ denote the optimal estimate of $\mathbf{X}_L$ using $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{L-1}$, and let the corresponding error vector be

$$\mathbf{V}_L = \mathbf{X}_L - \hat{\mathbf{X}}_L(S_{L-1}). \tag{3.17}$$

We can make the following observations about the properties of $\mathbf{V}_L$ at this time.

1. $$\mathbf{X}_L = \hat{\mathbf{X}}_L(S_{L-1}) + \mathbf{V}_L. \tag{3.18}$$

   This result follows from (3.17).

2. $\hat{\mathbf{X}}_L(S_{L-1})$ and $\mathbf{V}_L$ are orthogonal to each other. This is a consequence of the orthogonality principle. Moreover, $\mathbf{V}_L$ is orthogonal to every vector in S_{L-1}.

3. The linear span of $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{L-1}$ and $\mathbf{V}_L$ is S_L.

[1] $\hat{\mathbf{D}}(S_L)$ will be at least as good as $\hat{\mathbf{D}}(S_{L-1})$ since $\hat{\mathbf{D}}(S_{L-1}) + 0 \cdot \mathbf{X}_L$ is a well-defined linear combination of $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_L$.

We will prove the last result formally. Let $\{c_{i,L},\ i = 1, 2, \cdots, L-1\}$ represent the optimal coefficients for the estimate of $\mathbf{X}_L$ in S_{L-1}. Then,

$$\hat{\mathbf{X}}_L(S_{L-1}) = \sum_{i=1}^{L-1} c_{i,L}\,\mathbf{X}_i. \tag{3.19}$$

Substituting (3.19) in (3.17), we can express $\mathbf{V}_L$ as

$$\mathbf{V}_L = \mathbf{X}_L - \sum_{i=1}^{L-1} c_{i,L}\mathbf{X}_i. \tag{3.20}$$

Now, consider an arbitrary linear combination $\sum_{i=1}^{L} a_i\mathbf{X}_i$ of the L vectors $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_L$. This linear combination can be expressed using (3.20) as

$$\begin{aligned}\sum_{i=1}^{L} a_i\mathbf{X}_i &= \sum_{i=1}^{L-1} a_i\mathbf{X}_i + a_L\left(\mathbf{V}_L + \sum_{i=1}^{L-1} c_{i,L}\mathbf{X}_i\right)\\ &= \sum_{i=1}^{L-1}(a_i + a_L c_{i,L})\mathbf{X}_i + a_L\mathbf{V}_L,\end{aligned} \tag{3.21}$$

which is a linear combination of $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{L-1}$ and $\mathbf{V}_L$. Similarly, an arbitrary linear combination of $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{L-1}$ and $\mathbf{V}_L$ can be expressed as

$$\begin{aligned}\sum_{i=1}^{L-1} b_i\mathbf{X}_i + b_L\mathbf{V}_L &= \sum_{i=1}^{L-1} b_i\mathbf{X}_i + b_L\left(\mathbf{X}_L - \sum_{i=1}^{L-1} c_{i,_{L-1}}\mathbf{X}_i\right)\\ &= \sum_{i=1}^{L-1}\left(b_i - b_L c_{i,L-1}\right)\mathbf{X}_i + b_L\mathbf{X}_L.\end{aligned} \tag{3.22}$$

Equations (3.21) and (3.22) together prove the proposition.

Orthogonal Basis Vectors for S_L

We now have the tools necessary to compute an orthogonal basis vector set for S_L. Let

$$\mathbf{V}_1 = \mathbf{X}_1 \tag{3.23}$$

and

$$\begin{aligned}\mathbf{V}_i &= \mathbf{X}_i - \hat{\mathbf{X}}_i(S_{i-1})\\ &= \mathbf{X}_i - \sum_{j=1}^{i-1} c_{j,i}\mathbf{X}_j;\ i = 2, 3, \cdots, L.\end{aligned} \tag{3.24}$$

Note that the coefficient $c_{j,i}$ has two subscripts. The first subscript j indicates that the input vector $\mathbf{X}_j$ is being scaled by the coefficient. The second subscript i implies that the input vector $\mathbf{X}_i$ is being estimated.

Proposition. $\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_L$ is an orthogonal basis set for S_L.

Proof: We can apply the orthogonality principle to each vector $\mathbf{V}_i$ to show that $\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_L$ are mutually orthogonal. We can then use induction to prove that these vectors span S_L. We have already shown that $\mathbf{V}_1$ and $\mathbf{V}_2$ span S_2. Assume that $\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_{L-1}$ span S_{L-1}. We proved using (3.21) and (3.22) that the elements of S_{L-1} and $\mathbf{V}_L$ taken together span S_L. This completes the proof.

Example 3.2: Orthogonal Estimation in the Space of Stationary Random Processes

Consider the problem of linearly estimating a random process $d(n)$ using the most recent L samples of the process $x(n)$. Assume that $d(n)$ and $x(n)$ are jointly stationary and that the estimate we are seeking is optimal in the MMSE sense. Identify a set of orthogonal basis signals for this problem.

Solution. We shall use $x(n), x(n-1), \cdots, x(n-L+1)$ for estimating $d(n)$. According to (3.23) and (3.24), a set of orthogonal basis signals can be obtained by estimating $x(n-l)$ using $x(n), x(n-1), \cdots, x(n-l+1)$ and finding the estimation error for each l. The estimation process described above is known as the lth order *backward (linear) prediction*. The orthogonal signal set is given by

$$\begin{aligned} b_0(n) &= x(n) \\ b_l(n) &= x(n-l) - \sum_{k=0}^{l-1} b_{k,l}\, x(n-k); \;\; l = 1, \cdots, L-1. \end{aligned}$$

The optimal predictor coefficients for each order can be obtained by solving the appropriate set of Wiener-Hopf equations. However, this direct approach involves inverting several autocorrelation matrices. A much more efficient approach for evaluating the predictor coefficients will be discussed later in this chapter when we consider lattice filters.

REMARK 3.1: A problem closely related to backward linear prediction is that of *forward linear prediction*. The lth order forward prediction involves estimating $x(n)$ using $x(n-1), x(n-2), \cdots, x(n-l)$. Let $f_l(n)$ denote the lth order forward prediction error signal. We leave it to the reader to show that the members of the signal set $\{f_l(n); l = 1, 2, \ldots, L\}$ are not orthogonal to each other in general.

3.1.3 Gram-Schmidt Orthogonalization

The process of transforming a set of L arbitrary vectors into a set of L orthogonal basis vectors is known as *Gram-Schmidt orthogonalization*. A system that performs Gram-Schmidt

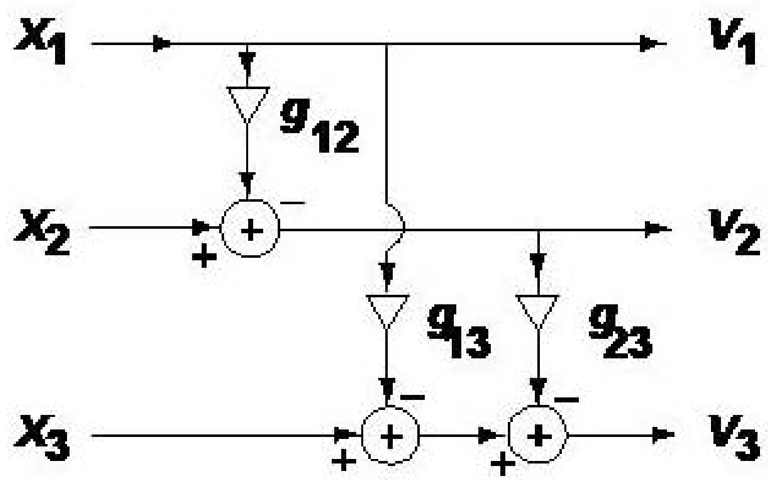

Figure 3.2: Gram-Schmidt processor for three input vectors.

orthogonalization is known as a *Gram-Schmidt processor.* Figure 3.2 shows a Gram-Schmidt processor for a set of three vectors.

The basic idea employed by the Gram-Schmidt orthogonalizer is that of estimating the ith vector $\mathbf{X}_i$ using the first $(i-1)$ vectors. If the estimation procedures are performed sequentially, and the estimation errors $\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_L$ are created in the indexed order, the vectors $\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_{i-1}$ are computed before $\mathbf{V}_i$ is computed. Consequently, we can use the set of orthogonal vectors $\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_{i-1}$ to estimate $\mathbf{X}_i$ and then define $\mathbf{V}_i$ as

$$\mathbf{V}_i = \mathbf{X}_i - \hat{\mathbf{X}}_i(S_{i-1}). \tag{3.25}$$

Let $g_{k,l}$ denote the coefficient for estimating $\mathbf{X}_l$ using $\mathbf{V}_k$. Then, we know from (3.4) that[2]

$$g_{k,l} = \frac{\langle \mathbf{X}_l, \mathbf{V}_k \rangle}{\| \mathbf{V}_k \|^2} \tag{3.26}$$

and that

$$\hat{\mathbf{X}}_l(S_{l-1}) = \sum_{k=1}^{l-1} g_{k,l} \, \mathbf{V}_k. \tag{3.27}$$

Calculation of the Inner Products

In order to perform the Gram-Schmidt orthogonalization, we need to calculate two types of quantities: a set of inner products of the form $\langle \mathbf{X}_l, \mathbf{V}_k \rangle$ and a set of squared norms of the form $\| \mathbf{V}_k \|^2$. We now derive an algorithm that calculates these quantities from known

[2]Note that $g_{k,l} \neq c_{k,l}$ in general since $g_{k,l}$ scales $\mathbf{V}_k$ and $\mathrm{c}_{k,l}$ scales $\mathbf{X}_k$ in the estimates.

values of $\langle \mathbf{X}_i, \mathbf{X}_j \rangle$ for all $1 \leq i, j \leq L$. Our derivation is given in the form of an inductive proof.

We begin the calculations by setting $\mathbf{V}_1$ to be identical to $\mathbf{X}_1$, *i.e.*,

$$\mathbf{V}_1 = \mathbf{X}_1. \tag{3.28}$$

Obviously

$$\| \mathbf{V}_1 \|^2 = \| \mathbf{X}_1 \|^2 \tag{3.29}$$

and

$$\langle \mathbf{X}_l, \mathbf{V}_1 \rangle = \langle \mathbf{X}_l, \mathbf{X}_1 \rangle;\ l = 1, 2, \ldots, L. \tag{3.30}$$

Consequently, we can evaluate the first set of coefficient values as

$$g_{1l} = \frac{\langle \mathbf{X}_l, \mathbf{V}_1 \rangle}{\| \mathbf{V}_1 \|^2};\ l = 2, 3, \ldots, L. \tag{3.31}$$

Now, suppose that we have completed the calculations up to $\mathbf{V}_{l-1}$. Our objective is to compute $\mathbf{V}_l$, given the orthogonal vectors $\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_{l-1}$, the coefficients $g_{i,j}$ for $i = 1$, 2, $\ldots, j-1$ and $j = i+1, i+2, \cdots, l-1$, the squared norm values $\| \mathbf{V}_i \|^2$ for $i = 1, 2, \cdots, l-1$ and the inner products $\langle \mathbf{X}_j, \mathbf{V}_i \rangle$ for $i = 1, 2, \cdots, l-1$ and $j = i+1, i+2, \ldots, L$. In order to continue the procedure for $\mathbf{V}_{l+1}, \mathbf{V}_{l+2}, \cdots, \mathbf{V}_L$, We also need to compute $\| \mathbf{V}_l \|^2$ and the inner products $\langle \mathbf{X}_j, \mathbf{V}_l \rangle$ for $j = l+1, \cdots, L$. Recall that

$$\mathbf{V}_k = \mathbf{X}_k - \sum_{j=1}^{k-1} g_{j,k} \mathbf{V}_j \tag{3.32}$$

for all k. Now,

$$\begin{aligned} \langle \mathbf{X}_k, \mathbf{V}_l \rangle &= \langle \mathbf{X}_k, \mathbf{X}_l - \sum_{j=1}^{l-1} g_{j,l} \mathbf{V}_j \rangle \\ &= \langle \mathbf{X}_k, \mathbf{X}_l \rangle - \sum_{j=1}^{l-1} g_{j,l} \langle \mathbf{X}_k, \mathbf{V}_j \rangle. \end{aligned} \tag{3.33}$$

Since all of the inner products $\langle \mathbf{X}_k, \mathbf{V}_l \rangle$ that appear on the right hand side of (3.33) have already been computed, the above quantity can be easily calculated. Now we can compute $g_{k,l}$ as

$$g_{k,l} = \frac{\langle \mathbf{X}_l, \mathbf{V}_k \rangle}{\| \mathbf{V}_k \|^2}. \tag{3.34}$$

It is left as an exercise to show that

$$\| \mathbf{V}_l \|^2 = \| \mathbf{X}_l \|^2 - \sum_{k=1}^{l-1} g_{k,l} \langle \mathbf{X}_l, \mathbf{V}_k \rangle. \tag{3.35}$$

Finally we can use (3.33) to compute $\langle \mathbf{X}_{l+1}, \mathbf{V}_k \rangle$. The complete algorithm for Gram-Schmidt orthogonalization is summarized in Table 3.1. A MATLAB file to implement the Gram-Schmidt processor is given in Figure 3.3.

Table 3.1: Gram-Schmidt Processor

Initialization

$$
\begin{aligned}
\mathbf{V}_1 &= \mathbf{X}_1 \\
\parallel \mathbf{V}_1 \parallel^2 &= \parallel \mathbf{X}_1 \parallel^2 \\
\langle \mathbf{X}_l, \mathbf{V}_1 \rangle &= \langle \mathbf{X}_l, \mathbf{X}_1 \rangle; \quad l = 2, 3, \cdots, L \\
g_{1,l} &= \frac{\langle \mathbf{X}_l, \mathbf{V}_1 \rangle}{\parallel \mathbf{V}_1 \parallel^2}; \quad l = 2, 3, \cdots, L
\end{aligned}
$$

Main Iteration

Do for $l = 2, 3, \ldots, L$

$$
\begin{aligned}
\mathbf{V}_l &= \mathbf{X}_l - \sum_{j=1}^{l-1} g_{j,l} \mathbf{V}_j \\
\parallel \mathbf{V}_l \parallel^2 &= \parallel \mathbf{X}_l \parallel^2 - \sum_{j=1}^{l-1} g_{j,l} \langle \mathbf{X}_l, \mathbf{V}_j \rangle
\end{aligned}
$$

Stop here if $l = L$

$$
\begin{aligned}
\langle \mathbf{X}_k, \mathbf{V}_l \rangle &= \langle \mathbf{X}_k, \mathbf{X}_l \rangle - \sum_{j=1}^{l-1} g_{j,l} \langle \mathbf{X}_k, \mathbf{V}_j \rangle; \quad k = l+1, l+2, \cdots, L \\
g_{l,k} &= \frac{\langle \mathbf{X}_k, \mathbf{V}_l \rangle}{\parallel \mathbf{V}_l \parallel^2}; \quad k = l+1, l+2, \cdots, L
\end{aligned}
$$

```
    function[g]=gs(rxx,L)
%
%
%   MATLAB script for determining the coefficients
%   of a Gram-Schmidt processor from knowledge of
%   the relevant inner products.
%
%
%   inputs: L -- number of vectors
%       rxx -- L x L matrix containing
%           the inner products of the
%           vectors.
%
%   output: g -- coefficients of the GS processor.
%           L x L triangular matrix
%

%
%   Initialization
%
    g=zeros(size(rxx));
    rvv(1)=rxx(1,1);
    g(1,[2:L]) = rxx([2:L])/rvv(1);
    rxv([2:L],1) = rxx([2:L],1);
    pause;
%
%   Main Iteration
%
    if L ~= 2
    for l=2:L-1,
        rvv(l) = rxx(l,l) - rxv(l,[1:l-1])*g([1:l-1],l);
        rxv([l+1:L],l) = rxx([l+1:L],l) - rxv([l+1:L],[1:l-1])*g([1:l-1],l);
        g(l,[l+1:L]) = rxv([l+1:L],l)'/rvv(l);
    end;
    end
    rvv(L) = rxx(L,L) - rxv(L,[1:L-1])*g([1:L-1],L);
%
%   We are done
%
```

Figure 3.3: MATLAB script for implementing a Gram-Schmidt processor.

Relationship Between $g_{i,j}$'s and $c_{i,j}$'s

From their definitions, we can write two block matrix equations relating the input vector set to the orthogonal basis vector set through either $g_{i,j}$'s or $c_{i,j}$'s. These relationships are

$$\mathbf{X}_i = \mathbf{G}^T \mathbf{V}_i \tag{3.36}$$

and

$$\mathbf{V}_i = \mathbf{C}^T \mathbf{X}_i, \tag{3.37}$$

where

$$\mathbf{G}^T = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ g_{1,2} & 1 & 0 & \cdots & 0 \\ g_{1,3} & g_{2,3} & 1 & \vdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ g_{1,L} & g_{2,L} & g_{3,L} & \cdots & 1 \end{bmatrix} \tag{3.38}$$

and

$$\mathbf{C}^T = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -c_{1,2} & 1 & 0 & \cdots & 0 \\ -c_{1,3} & -c_{2,3} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ -c_{1,L} & -c_{2,L} & -c_{3,L} & \cdots & 1 \end{bmatrix}. \tag{3.39}$$

It is obvious from (3.36) and (3.37) that

$$\mathbf{C} = \mathbf{G}^{-1}. \tag{3.40}$$

Since $\mathbf{G}^T$ is a lower triangular matrix, computation of the inverse can be accomplished using back substitution.

Example 3.3: Gram-Schmidt Orthogonalization for a Five-Dimensional Euclidean Space

Consider the five-dimensional vectors in the Euclidean space given by $\mathbf{X}_1 = [1 \;\; 2 \;\; -3 \;\; 4 \;\; 0]^T$, $\mathbf{X}_2 = [-1 \;\; 3 \;\; -4 \;\; 2 \;\; 1]^T$, $\mathbf{X}_3 = [2 \;\; 1 \;\; 3 \;\; 1 \;\; 5]$ and $\mathbf{X}_4 = [-2 \;\; 1 \;\; -3 \;\; 2 \;\; -4]^T$. Find an orthogonal basis set for these vectors. Assume that the inner product is defined as

$$\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i=1}^{5} x_i y_i.$$

Solution: The matrix $\mathbf{R_{XX}}$ for this case is given by

$$\mathbf{R_{XX}} = \begin{bmatrix} 30 & 25 & -1 & 17 \\ 25 & 31 & -4 & 17 \\ -1 & -4 & 40 & -30 \\ 17 & 17 & -30 & 34 \end{bmatrix}$$

Using the Gram-Schmidt processor of Table 3.1, we find that

$$\begin{aligned}
\mathbf{V}_1 &= [1 \;\; 2 \;\; -3 \;\; 4 \;\; 0]^T, \\
\mathbf{V}_2 &= [-1.83 \;\; 1.33 \;\; -1.50 \;\; -1.33 \;\; 1.00]^T, \\
\mathbf{V}_3 &= [1.46 \;\; 1.48 \;\; 2.43 \;\; 0.71 \;\; 5.31]^T
\end{aligned}$$

and

$$\mathbf{V}_4 = [-0.98 \;\; 0.58 \;\; 0.90 \;\; 0.63 \;\; -0.39]^T$$

is an orthogonal basis set for the span of $\mathbf{X}_1,\ \mathbf{X}_2,\ \mathbf{X}_3$ and $\mathbf{X}_4$. The coefficients of the Gram-Schmidt processor are given by

$$\begin{bmatrix} g_{1,2} & g_{1,3} & g_{1,4} \\ & g_{2,3} & g_{2,4} \\ & & g_{3,4} \end{bmatrix} = \begin{bmatrix} 0.83 & -0.03 & 0.57 \\ & -0.31 & 0.28 \\ & & -0.73 \end{bmatrix},$$

where the blank entries denote zero values. Finally, the squared norms of the orthogonal basis vectors are given by

$$\begin{aligned}
\| \mathbf{V}_1 \|^2 &= 30.00, \\
\| \mathbf{V}_2 \|^2 &= 10.17, \\
\| \mathbf{V}_3 \|^2 &= 38.98
\end{aligned}$$

and

$$\| \mathbf{V}_4 \|^2 = 2.67.$$

Example 3.4: Gram-Schmidt Orthogonalization in a Space of Stationary Random Processes

Let $x(n)$ belong to the space of jointly wide sense stationary random processes. Assume that this signal was generated as the output of an IIR filter with input-output relationship given by

$$x(n) = 0.6\xi(n) + 0.8x(n-1),$$

where the input signal $\xi(n)$ belonged to a Gaussian process with zero-mean and unit variance. Find an orthogonal basis set for the signal set given by $x(n),\ x(n-1),\ x(n-2)$ and $x(n-3)$.

Solution: The first task in solving this problem is to evaluate the 4×4 autocorrelation matrix $\mathbf{R_{xx}}$ of the signal $x(n)$. It is left as an exercise to show that $r_{xx}(0) = E\{x^2(n)\} = 1$, and that the autocorrelation function satisfies the relationship

$$r_{xx}(m) = 0.8r_{xx}(m-1); \quad m > 0.$$

The autocorrelation matrix is now given by

$$\mathbf{R_{xx}} = \begin{bmatrix} 1 & 0.8 & 0.64 & 0.512 \\ 0.8 & 1 & 0.8 & 0.64 \\ 0.64 & 0.8 & 1 & 0.8 \\ 0.512 & 0.64 & 0.8 & 1 \end{bmatrix}.$$

We have seen from Example 3.2 that the backward prediction error sequences $b_0(n)$, $b_1(n)$, $b_2(n)$ and $b_3(n)$ form an orthogonal basis set for the space spanned by $x(n), x(n-1), x(n-2)$ and $x(n-3)$. The coefficients that generate the backward prediction error signals can be calculated using the Gram-Schmidt processor described in Table 3.1. The coefficient matrix $\mathbf{G}$ is given by

$$\mathbf{G}^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ .8 & 1 & 0 & 0 \\ .64 & .8 & 1 & 0 \\ .512 & .64 & .8 & 1 \end{bmatrix}.$$

Furthermore, the mean-squared values of the backward prediction error sequences are given by

$$\begin{aligned} E\{b_0^2(n)\} &= 1.0, \\ E\{b_1^2(n)\} &= 0.36, \\ E\{b_2^2(n)\} &= 0.36 \end{aligned}$$

and

$$E\{b_3^2(n)\} = 0.36.$$

Joint Process Estimation Using the Gram-Schmidt Processor

The Gram-Schmidt processor can be used to create an orthogonal basis set for a set of vectors. Once the orthogonal basis set is created, how do we estimate an arbitrary vector $\mathbf{D}$ using this vector set? We now extend the ideas developed above to the joint process estimation problem.

Given a set of arbitrary vectors $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_L$, we assume that we have already calculated an orthogonal basis set $\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_L$, the coefficients $\{g_{i,j};\ i = 1,\ 2,\ \cdots,\ L;\ j = i+1\ ,\ i+2,\ \cdots,\ L\}$ of the Gram-Schmidt processor and the squared norms of the orthogonal basis vectors as in Table 3.1. Furthermore, we assume that the values of the inner products $\langle \mathbf{D}, \mathbf{X}_i \rangle$ are known for $i = 1,\ 2,\ \cdots,\ L$. Let h_i denote the optimal minimum squared-norm error coefficient for estimating $\mathbf{D}$ using $\mathbf{V}_i$. This coefficient is given by

$$h_i = \frac{\langle \mathbf{D}, \mathbf{V}_i \rangle}{\| \mathbf{V}_i \|^2}. \tag{3.41}$$

To evaluate h_i, we only need to compute the inner product in the numerator of the right hand side of the above equation. This quantity can be easily calculated as

$$\begin{aligned} \langle \mathbf{D}, \mathbf{V}_i \rangle &= \langle \mathbf{D}, \mathbf{X}_i - \sum_{j=1}^{i-1} g_{j,i} \mathbf{V}_j \rangle \\ &= \langle \mathbf{D}, \mathbf{X}_i \rangle - \sum_{j=1}^{i-1} g_{j,i} \langle \mathbf{D}, \mathbf{V}_j \rangle. \end{aligned} \tag{3.42}$$

Recall that all the inner products of the form $\langle \mathbf{D}, \mathbf{X}_k \rangle$ are known. Furthermore, we can calculate $\langle \mathbf{D}, \mathbf{V}_i \rangle$ in an iterative manner. Consequently, at the time $\langle \mathbf{D}, \mathbf{V}_i \rangle$ is computed, we would already have calculated $\langle \mathbf{D}, \mathbf{V}_j \rangle$ for all $j < i$. These iterations can be initialized with $\langle \mathbf{D}, \mathbf{V}_1 \rangle = \langle \mathbf{D}, \mathbf{X}_1 \rangle$ since $\mathbf{V}_1 = \mathbf{X}_1$. Since all the variables on the right-hand side of (3.42) is known, $\langle \mathbf{D}, \mathbf{V}_i \rangle$ can be calculated. The joint process estimate $\hat{\mathbf{D}}(\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_L)$ is now given by

$$\begin{aligned} \hat{\mathbf{D}}\left(\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_L\right) &= \hat{\mathbf{D}}\left(\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_L\right) \\ &= \sum_{i=1}^{L} \hat{\mathbf{D}}(\mathbf{V}_i) \qquad (3.43) \\ &= \sum_{i=1}^{L} h_i \mathbf{V}_i. \qquad (3.44) \end{aligned}$$

Finally, from (2.50) the squared norm of the estimation error vector is given by

$$\begin{aligned} \| \mathbf{D} - \hat{\mathbf{D}}\left(\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_L\right) \|^2 &= \| \mathbf{D} \|^2 - \sum_{i=1}^{L} h_i \langle \mathbf{D}, \mathbf{V}_i \rangle \\ &= \| \mathbf{D} \|^2 - \sum_{i=1}^{L} \frac{\langle \mathbf{V}_i, \mathbf{D} \rangle \langle \mathbf{D}, \mathbf{V}_i \rangle}{\| \mathbf{V}_i \|^2}. \qquad (3.45) \end{aligned}$$

Note that all the quantities on the right hand side of (3.45) are non-negative. Consequently, the use of additional components $\mathbf{V}_i$ will reduce the estimation error if $\| \mathbf{V}_i \|^2 > 0$ and $\langle \mathbf{D}, \mathbf{V}_i \rangle \neq 0$.

Example 3.4 (continued): Identification of an FIR System

Let us now consider the task of identifying the unknown system of Example 2.11 using the set of the four orthogonal basis signals evaluated earlier in this example. Direct calculations will show that

$$\begin{aligned} E\{d^2(n)\} &= 0.59 \\ \mathbf{P}_{d\mathbf{x}} &= [0.520 \;\; 0.200 \;\; 0.340 \;\; 0.272]^T. \end{aligned}$$

Using (2.50) and the coefficients of the Gram-Schmidt processor we find that

$$\begin{aligned} E\{d(n)b_0(n)\} &= 0.52, \\ E\{d(n)b_1(n)\} &= -0.216, \\ E\{d(n)b_2(n) &= 0.18 \end{aligned}$$

and

$$E\{d(n)b_3(n)\} = 0.0.$$

The joint process estimator coefficients can now be calculated from the coefficients of the Gram-Schmidt processor, the squared-norms of the orthogonal basis signals, and the iterations described by (3.41) and (3.42). They are given by

$$\begin{aligned} h_1 &= \frac{E\{d(n)b_0(n)\}}{E\{b_0^2(n)\}} = 0.52 \\ h_2 &= \frac{E\{d(n)b_1(n)\}}{E\{b_1^2(n)\}} = -0.6 \\ h_3 &= \frac{E\{d(n)b_2(n)\}}{E\{b_2^2(n)\}} = 0.5 \end{aligned}$$

and

$$h_4 = \frac{E\{d(n)b_3(n)\}}{E\{b_3^2(n)\}} = 0.0.$$

The MMSE error value is calculated using (2.50) and is given by

$$E\left\{\left(d(n) - \hat{d}(n)^2\right)\right\} = 0.1.$$

We leave it as an exercise to show that the MMSE due to direct estimation of $d(n)$ from $x(n)$ will also be 0.1.

Conversion of Coefficients of the Orthogonal Basis to Direct Form Coefficients

One disadvantage of using Gram-Schmidt orthogonalization for joint process estimation is that the coefficients of the orthogonal vectors are almost always different from those of the equivalent direct form estimates. We have already seen that both estimators provide identical estimates. We now derive a method for converting the coefficients of the orthogonal basis vectors $\{h_i;\ i = 1, 2, \cdots, L\}$ to $\{w_i;\ i = 1, 2, \cdots, L\}$, the coefficients of the direct form estimator.

$$\hat{\mathbf{D}}(\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_L) = \sum_{i=1}^{L} h_i \mathbf{V}_i. \tag{3.46}$$

Since the linear span of $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_i$ is identical to the linear span of $\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_i$, we can express the optimal estimate of a vector $\mathbf{D}$ in S_i as

$$\begin{aligned} \hat{\mathbf{D}}(S_i) &= \sum_{j=1}^{i} w_{j,i} \mathbf{X}_j \\ &= \sum_{j=1}^{i} h_j \mathbf{V}_j. \end{aligned} \tag{3.47}$$

In (3.47), we have used double subscripts for the coefficients of $\mathbf{X}_j$. The second subscript corresponds to the number of vectors used in the estimate while the first subscript denotes the particular vector that is scaled by the coefficient. In order to be consistent with our previous notation, we define

$$w_j = w_{j,L};\ j = 1,\ 2,\ \cdots,\ L. \tag{3.48}$$

In general, $w_{j,i} \neq w_{j,k}$ if $i \neq k$. However, if we use the orthogonal basis set, the optimal coefficient h_j associated with the vector $\mathbf{V}_j$ remains the same irrespective of the number of vectors used in the estimate.

As was the case for previous derivations, our approach is sequential, *i.e.*, we first transform the coefficients of the estimates in S_1, then transform the coefficients of the estimate in S_2, and so on until the coefficients of the estimates in S_L are transformed. Now,

$$\begin{aligned}\hat{\mathbf{D}}(S_1) &= \hat{\mathbf{D}}(\mathbf{X}_1) = w_{1,1}\mathbf{X}_1 \\ &= \hat{\mathbf{D}}(\mathbf{V}_1) = h_1\mathbf{V}_1. \end{aligned} \tag{3.49}$$

Since $\mathbf{V}_1 = \mathbf{X}_1$, it follows that

$$w_{1,1} = h_1. \tag{3.50}$$

Suppose now that we have performed the transformations for the estimates in S_i, *i.e.*, the precise mathematical relationship between the sets $\{h_1, h_2, \cdots, h_i\}$ and $\{w_{1,i}, w_{2,i} \cdots, w_{i,i}\}$ is known. We can use this information to convert the coefficients $h_1, h_2, \cdots, h_i, h_{i+1}$ to the direct form coefficients $w_{1,i+1},\ w_{2,i+1},\ \cdots,\ w_{i+1,i+1}$ for the optimal estimate in S_{i+1}. Let

$$\mathbf{V}_{i+1} = \mathbf{X}_{i+1} - \sum_{j=1}^{i} c_{j,i+1}\mathbf{X}_j \tag{3.51}$$

as in (3.24). Recall that

$$\begin{aligned}\hat{\mathbf{D}}(S_{i+1)} &= \sum_{j=1}^{i+1} w_{j,i+1}\mathbf{X}_j \\ &= \sum_{j=1}^{i+1} h_j\mathbf{V}_j \end{aligned} \tag{3.52}$$

and that we can also express $\hat{\mathbf{D}}(S_{i+1})$ as

$$\begin{aligned}\hat{\mathbf{D}}(S_{i+1}) &= \hat{\mathbf{D}}(S_i) + \hat{\mathbf{D}}(\mathbf{V}_{i+1}) \\ &= \sum_{j=1}^{i} w_{j,i}\mathbf{X}_j + h_{i+1}\left(\mathbf{X}_{i+1} - \sum_{j=1}^{i} c_{j,i+1}\mathbf{X}_j\right). \end{aligned} \tag{3.53}$$

We can find $w_{j,i+1}$ in terms of $w_{j,i}$ and h_{i+1} by equating the coefficients of $\mathbf{X}_j$ in (3.52) and (3.53). This results in

$$w_{i+1,i+1} = h_{i+1} \tag{3.54}$$

and

$$w_{j,i+1} = w_{j,i} - h_{i+1}c_{j,i+1};\ j = 1,\ 2,\ \cdots,\ i. \tag{3.55}$$

The complete set of iterative equations that define the transformation from $\{h_1, h_2, \cdots, h_L\}$ to $\{w_1, w_2, \cdots, w_L\}$ is given in Table 3.2. A MATLAB function that implements the transformation is provided in Figure 3.4. A similar set of recursions can be also derived to

Table 3.2: Transformation of the coefficients of the orthogonal basis set to those of the original input vectors.

Convert G to C

$$\mathbf{C} = \mathbf{G}^{-1}$$

Initialization

$$w_{1,1} = h_1$$

Main Iteration
Do for $i = 2,\ 3,\ \cdots,\ L$

$$w_{i,i} = h_i$$
$$w_{j,i} = w_{j,i-1} - h_i c_{j,i};\ \ j = 1,\ 2,\ \cdots,\ i-1$$

Final Stage

$$w_j = w_{j,L};\ j = 1,\ 2,\ \cdots,\ L.$$

```
    function [w] = gs2df(G,L,h)
%
%
%   Transformation of coefficients of the orthogonal basis set
%   obtained from a Gram-Schmidt orthogonalizer to coefficients
%   of the direct form estimator.
%
%
%   Input:
%       G - Coefficients of the Gram-Schmidt orthogonalizer
%           (L x L upper triangular matrix with ones in
%           the diagonal)
%       L - Number of coefficients
%       h - L-element coefficient vector of the orthogonal
%           basis set.
%
%   Output:
%       w - coefficients of the direct form estimator
%
%

    c=inv(G);
%
%   Initialization
%
    ww(1,1) = h(1);
%
%   Main Iteration
%
    for i=2:L,
        ww(i,i)=h(i);
        ww([1:i-1],i) = ww([1:i-1],i-1) +  h(i)*c([1:i-1],i);
%
%   Note that the definition of the coefficient matrix c
%   involves the negatives of the actual coefficients.
%
    end;
    w=ww([1:L],L);
%
%   We are done
%
```

Figure 3.4: A MATLAB script for transforming the coefficients of the orthogonal basis set to that of the direct form estimator.

transform $\{w_1, w_2, \cdots, w_L\}$ to $\{h_1, h_2, \cdots, h_L\}$. The derivation of this transformation is left as an exercise to the reader.

Example 3.4 (continued): Transformation of the Estimate to Direct Form

The **C** matrix corresponding to the coefficients of the Gram-Schmidt processor is given by

$$\mathbf{C}^T = \mathbf{G}^{-T} \quad = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -0.8 & 1 & 0 & 0 \\ 0 & -0.8 & 1 & 0 \\ 0 & 0 & -0.8 & 0 \end{bmatrix}$$

Transformation from $\{h_1, h_2, h_3, h_4\}$ to $\{w_1, w_2, w_3, w_4\}$ can be accomplished as follows:

$$\begin{aligned}
w_{1,1} &= h_1 = 0.52 \\
w_{2,2} &= h_2 = -0.6 \\
w_{1,2} &= w_{1,1} - h_2 \cdot c_{1,2} = 1.0 \\
w_{3,3} &= h_3 = 0.5 \\
w_{1,3} &= w_{1,2} - h_3 \cdot c_{1,3} = 1.0 \\
w_{2,3} &= w_{2,2} - h_3 \cdot c_{2,3} = -1.0 \\
w_4 &= w_{4,4} = h_4 = 0 \\
w_1 &= w_{1,4} = w_{1,3} - h_4 \cdot c_{1,4} = 1.0 \\
w_2 &= w_{2,4} = w_{2,3} - h_4 \cdot c_{2,4} = -1.0 \\
w_3 &= w_{3,4} = w_{3,3} - h_4 \cdot c_{3,4} = 0.5
\end{aligned}$$

The MMSE value using the direct-form estimation also can be calculated to be $E\{(d(n) - \hat{d}(n))^2\} = 0.1$, which is identical to that obtained using the Gram-Schmidt processor.

3.2 Lattice Filters

The Gram-Schmidt orthogonalization procedure we discussed in the previous section was motivated mainly by the fact that computation of the optimal coefficients is very easy when the input vectors are mutually orthogonal. However, the Gram-Schmidt processor itself requires

$$\sum_{i=1}^{L-1} i = \frac{L(L-1)}{2} \tag{3.56}$$

coefficients to orthogonalize L input vectors. This complexity is an order of magnitude larger than that of the joint-process estimator itself. While we can do no better than this for arbitrarily general cases, there are some very important practical cases for which much more efficient algorithms for orthogonalizing the input data are available. Perhaps the most important of such situations is the single channel estimation problem with finite memory. In this section, we consider minimum mean-squared error estimation for the single channel, finite memory case and derive the lattice structure for orthogonalizing the input signal.

3.2.1 Statement of the Problem

We consider two signals $d(n)$ and $x(n)$ that belong to the class of jointly wide-sense stationary random processes. We seek the MMSE estimate of $d(n)$ as a linear combination of the most recent L samples of $x(n)$, *i.e.*, $x(n), x(n-1), \cdots, x(n-L+1)$. The problem that we are interested in is three-fold:

1. Find an efficient linear transformation of the input signals $x(n), x(n-1), \cdots, x(n-L+1)$ such that it produces L mutually orthogonal signals at the output.

2. Find the coefficients of the joint process estimator based on the orthogonal signal set obtained above.

3. Derive a set of recursive equations to transform the coefficients of the orthogonal signal set to that of the original signals and vice versa.

The above statement of the problem is almost identical to that for the Gram-Schmidt orthogonalization procedure. The key difference is that the input signals are successive samples of a wide-sense stationary random process and this will facilitate a significant amount of simplification in the orthogonalization procedure. The simplifications will result in the derivation of the lattice filter as well as the development of an efficient algorithm to solve the Wiener-Hopf equations when the autocorrelation matrix is Toeplitz. The latter algorithm is known as the *Levinson-Durbin algorithm*.

3.2.2 The Levinson-Durbin Algorithm

We have seen in Example 3.2 that the backward prediction error signals of order $0, 1, \cdots, L-1$, denoted by $b_0(n), b_1(n), \cdots, b_{L-1}(n)$, respectively, and defined as

$$\begin{aligned} b_0(n) &= x(n) \\ b_i(n) &= x(n-i) - \sum_{l=0}^{i-1} g_{l,i} x(n-l); \quad i = 1, 2, \cdots, L-1 \end{aligned} \tag{3.57}$$

constitute an orthogonal basis signal set for the space spanned by $x(n)$, $x(n-1)$, $\cdots$, $x(n-l+1)$. In the above equation, the coefficient $g_{l,i}$ scales $x(n-l)$ and belongs to the optimal coefficient set for the ith order backward prediction problem. As usual, our approach for computing the backward prediction error signals is to proceed sequentially. It will become clear shortly that forward prediction error signals of order $0, 1, \cdots, L-1$, denoted by $f_0(n), f_1(n), \cdots, f_{L-1}(n)$, respectively, and defined as

$$\begin{aligned} f_0(n) &= x(n) \\ f_i(n) &= x(n) - \sum_{l=1}^{i} a_{l,i} x(n-l); \quad i = 1, 2, \cdots, L-1, \end{aligned} \tag{3.58}$$

where $a_{l,i}$ is the lth coefficient of the ith order forward predictor, are of fundamental importance in our derivations.

Suppose that we have computed $b_0(n), b_1(n), \cdots, b_k(n)$ and wish to determine $b_{k+1}(n)$, the $(k+1)$th order backward prediction error signal. We start with the following proposition.

Proposition. $b_0(n-1), b_1(n-1), \cdots, b_k(n-1)$ constitute an orthogonal basis set for $x(n-1), x(n-2), \cdots, x(n-k)$.

Proof: We know that $b_0(n), b_1(n), \cdots, b_k(n)$ form an orthogonal basis signal set for $x(n),\ x(n-1),\ \cdots,\ x(n-k+1)$. Recall also that $x(n)$ is a wide-sense, stationary random process. Let $v_0(n), v_1(n), \cdots, v_k(n)$ denote the orthogonal basis signal set for $x(n-1), x(n-2), \cdots, x(n-k)$. We have seen in our previous discussions that we can define the $v_i(n)$'s as

$$v_0(n) = x(n-1) \tag{3.59}$$

$$v_i(n) = x(n-1-i) - \sum_{l=0}^{i-1} g_{l,i}x(n-1-l); \quad i = 1, 2, \cdots, k, \tag{3.60}$$

where $g_{l,i}, l = 0, 1, \cdots, i-1$ denote the optimal MMSE coefficients for estimating $x(n-1-i)$ using $x(n-1), x(n-2), \cdots, x(n-i)$. These coefficients are identical to the coefficients in (3.57) needed to compute $b_i(n)$ because (i) the input signal is stationary, and (ii) all the signals involved in the computation $v_i(n)$ are the same as those required to compute $b_i(n)$, except that they are delayed by one sampling time. Now, comparing (3.57) with (3.60) we recognize that

$$v_i(n) = b_i(n-1); \quad i = 0, 1, \cdots, k-1, \tag{3.61}$$

which proves the proposition.

A Restatement of the Problem

The problem of finding $b_{k+1}(n)$ may now be restated as follows: Find the optimal minimum mean squared error signal for estimating $x(n-k-1)$ using $x(n),\ x(n-1),\ \cdots,\ x(n-k)$, given that $b_k(n-1)$ is the optimal error signal for estimating $x(n-k-1)$ using $x(n-1),\ x(n-2),\ \cdots,\ x(n-k)$. In other words, improve the estimate of $x(n-k-1)$ using one additional sample, namely, $x(n)$.

Calculation of $b_{k+1}(n)$.

The above problem is easy to solve, given our knowledge of Gram-Schmidt orthogonalization. The solution may be described in the following steps:

1. Find the component of $x(n)$ that is orthogonal to $x(n-1), x(n-2), \cdots, x(n-k)$. Since this component is the error in estimating $x(n)$ using $x(n-1), x(n-2), \cdots, x(n-k)$,

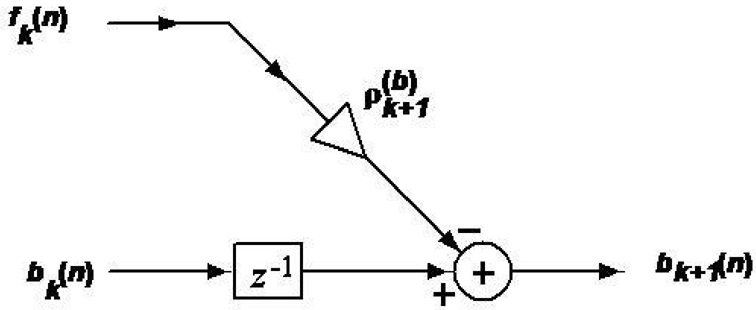

Figure 3.5: Computation of $b_{k+1}(n)$ from $b_k(n)$ and $f_k(n)$.

it is nothing but the kth order forward prediction error signal. Using (3.58) we can write this as

$$f_k(n) = x(n) - \sum_{l=1}^{k} a_{l,k} x(n-l). \tag{3.62}$$

2. Find the optimal estimate of $x(n-k-1)$ using $f_k(n)$. The optimal MMSE coefficient is given by

$$\rho^{(b)}_{k+1} = \frac{E\{x(n-k-1)f_k(n)\}}{E\{f_k^2(n)\}} \tag{3.63}$$

3. Assume that the required expected values have all been calculated. Then, $b_{k+1}(n)$ can be calculated as

$$\begin{aligned} b_{k+1}(n) &= x(n-k-1) - \sum_{l=0}^{k-1} g_{l,k} x(n-1-l) - \rho^{(b)}_{k+1} f_k(n) \\ &= b_k(n-1) - \rho^{(b)}_{k+1} f_k(n) \end{aligned} \tag{3.64}$$

Figure 3.5 demonstrates how the $(k+1)$th order backward prediction error can be computed from knowledge of the kth order forward and backward prediction error signals. It clearly shows that computation of $b_{k+1}(n)$ requires just one additional multiplication. It should also be obvious that we need to compute the $(k+1)$th order forward prediction error signal in order to be able to find the backward prediction error signals of order $(k+2)$ and more. We now discuss this.

Calculation of $f_{k+1}(n)$

The procedure to compute the $(k+1)$th order forward prediction error signal from $f_k(n)$ and $b_k(n)$ is very similar to that of computing $b_{k+1}(n)$. Recall that the $(k+1)$th order forward prediction involves estimating $x(n)$ using $x(n-1),\ x(n-2),\ \cdots,\ x(n-k-1)$ and that $f_k(n)$

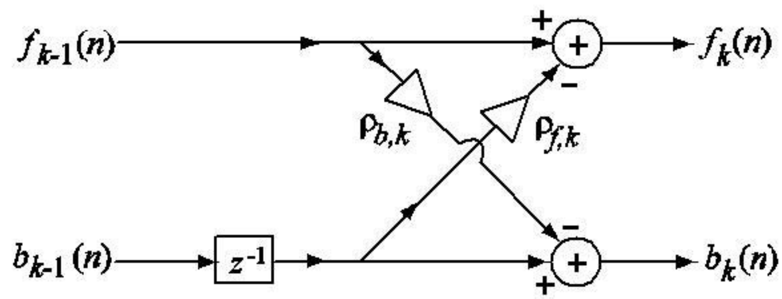

Figure 3.6: The (k)th stage of a lattice predictor.

is the error in optimally estimating $x(n)$ using $x(n-1), x(n-2), \cdots, x(n-k)$. Our objective for the $(k+1)$th order forward predictor is to improve upon the prediction of $x(n)$ using one additional sample, namely, $x(n-k-1)$. The portion of $x(n-k-1)$ that can provide an additional improvement in the estimate of $x(n)$ is orthogonal to $x(n-1), x(n-2), \cdots, x(n-k)$. As we saw in the earlier discussion, $b_k(n-1)$ is the error in estimating $x(n-k-1)$ using $x(n-1), x(n-2), \cdots, x(n-k)$, and is therefore the orthogonal component that we seek. Let $\rho^{(f)}_{k+1}$ denote the optimal coefficient for estimating $x(n)$ using $b_k(n-1)$. Then,

$$\begin{aligned} f_{k+1}(n) &= x(n) - \sum_{l=1}^{k} a_{l,k} x(n-l) - \rho^{(f)}_{k+1} b_k(n-1) \\ &= f_k(n) - \rho^{(f)}_{k+1} b_k(n-1). \end{aligned} \tag{3.65}$$

Once again, we see that the $(k+1)$th order forward prediction error can be evaluated from the kth order forward and backward prediction errors with only one additional multiplication. Figure 3.6 shows the update structure for the kth stage of both forward and backward prediction error sequences. A complete, three-stage predictor is shown in Figure 3.7. The structure in 3.7 is known as the *lattice predictor*. The coefficients of the lattice predictor are known as *reflection coefficients*. The negative values of the reflection coefficients are known as the *partial correlation coefficients* in statistical literature.

REMARK 3.2: The use of the term “reflection coefficient” arises from transmission line theory, where ρ_k can be considered as the reflection coefficient at the boundary between two sections with different impedances [Makhoul 1975]. Similar analogies can also be found in plane wave transmission and speech analysis.

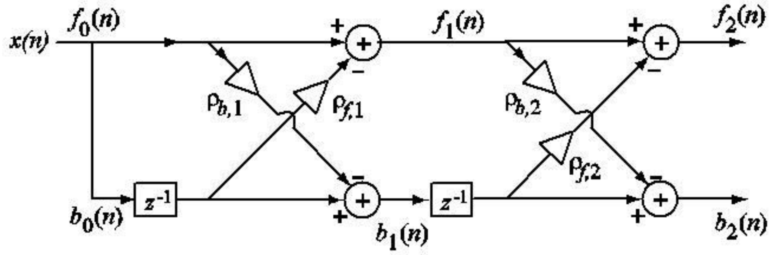

Figure 3.7: A three-stage lattice predictor.

Calculation of the Reflection Coefficients

We now consider the computation of the reflection coefficients $\rho_k^{(f)}$ and $\rho_k^{(b)}$ at the kth stage of the lattice predictor. From (3.63) we can compute the backward reflection coefficient as

$$\rho_k^{(b)} = \frac{E\{x(n-k)f_{k-1}(n)\}}{E\{f_{k-1}^2(n)\}}. \tag{3.66}$$

Similarly, we can determine the forward reflection coefficient as

$$\rho_k^{(f)} = \frac{E\{x(n)b_{k-1}(n-1)\}}{E\{b_{k-1}^2(n-1)\}}. \tag{3.67}$$

The above parameters can be evaluated if the statistical expectations that appear on the right-hand-sides of the above equations are known. Assuming that all the expectations involving $f_{k-1}(n)$ and $b_{k-1}(n-1)$ have been calculated, we can compute $E\{f_k^2(n)\}$, $E\{b_k^2(n-1)\}$, $E\{x(n)b_k(n-1)\}$ and $E\{x(n-k-1)f_k(n)\}$ that are required for evaluating the reflection coefficients at the $(k+1)$th stage. The following two results can be easily obtained by using the basic definition of the kth order forward and backward prediction error signals.

$$\begin{aligned} E\{x(n)b_k(n-1)\} &= E\left\{x(n)\left(x(n-k-1) - \sum_{l=0}^{k-1} g_{l,k}x(n-1-l)\right)\right\} \\ &= r_{xx}(k+1) - \sum_{l=0}^{k-1} g_{l,k}r_{xx}(l+1). \end{aligned} \tag{3.68}$$

Similarly,

$$\begin{aligned} E\{x(n-k-1)f_k(n)\} &= E\left\{x(n-k-1)\left(x(n) - \sum_{l=1}^{k} a_{l,k}x(n-l)\right)\right\} \\ &= r_{xx}(k+1) - \sum_{l=1}^{k} a_{l,k}r_{xx}(l-k-1). \end{aligned} \tag{3.69}$$

Before developing explicit expressions for the reflection coefficients, we derive some useful properties of the reflection coefficients.

Proposition: The optimal MMSE coefficient for estimating $x(n-k)$ using $f_{k-1}(n)$, given by $\rho_k^{(b)}$, is also the optimal MMSE coefficient for estimating $b_{k-1}(n-1)$ using $f_{k-1}(n)$. Similarly, $\rho_k^{(f)}$ is the optimal MMSE coefficient for estimating $x(n)$ using $b_{k-1}(n-1)$ and for estimating $f_{k-1}(n)$ using $b_{k-1}(n-1)$.

Proof: The proof for the first part of the proposition involves decomposing $x(n-k)$ into two orthogonal components and recognizing that $f_{k-1}(n)$ is orthogonal to one of the components. Recall from (3.57) that

$$x(n-k) = \sum_{l=0}^{k-2} g_{l,k-1} x(n-l-1) + b_{k-1}(n-1). \tag{3.70}$$

Since $f_{k-1}(n)$ is orthogonal to all linear combinations of $x(n-1),\ x(n-2),\ \cdots,\ x(n-l+1)$, we see that

$$\begin{aligned} E\{x(n-k)f_{k-1}(n)\} &= E\left\{\left(\sum_{l=0}^{k-2} g_{l,k-1} x(n-l-1) + b_{k-1}(n-1)\right) f_{k-1}(n)\right\} \\ &= E\{b_{k-1}(n-1)f_{k-1}(n)\}. \end{aligned} \tag{3.71}$$

Substituting (3.71) in (3.66) we get

$$\begin{aligned} \rho_k^{(b)} &= \frac{E\{x(n-k)f_{k-1}(n)\}}{E\{f_{k-1}^2(n)\}} \\ &= \frac{E\{b_{k-1}(n-1)f_{k-1}(n)\}}{E\{f_{k-1}^2(n)\}}. \end{aligned} \tag{3.72}$$

The right-hand side of the above equation is, in fact, the optimal coefficient for estimating $b_{k-1}(n-1)$ using $f_{k-1}(n)$, and this proves the first part of the proposition.

The proof of the second part of the proposition involves decomposing $x(n)$ as

$$x(n) = \sum_{l=1}^{k-1} a_{l,k-1} x(n-l) + f_{k-1}(n). \tag{3.73}$$

The proof is similar to that given above and is left as an exercise for the reader.

Symmetry of Forward and Backward Predictors

In order to complete our derivation of the lattice predictor, we need to derive a method to compute $a_{l,k}$ and $b_{l,k}$, the direct form coefficients of the forward and backward predictors for each prediction order. Before we derive a recursive update strategy for computing these coefficients, we prove the following three symmetry relationships among the parameters of the two predictors:

i. The coefficients of the kth order forward and backward predictors are mirror images of each other for all k, *i.e.*,

$$a_{l,k} = g_{k-l,k}; \quad l = 1, 2, \cdots, k. \tag{3.74}$$

ii. The reflection coefficients $\rho_k^{(f)}$ and $\rho_k^{(b)}$ are identical to each other for all k. We will denote both coefficients by ρ_k from now on, *i.e.*,

$$\rho_k = \rho_k^{(f)} = \rho_k^{(b)}. \tag{3.75}$$

iii. The kth order mean-squared forward prediction and backward prediction error values are identical for all k. We will, from now on, denote the k-th order mean-squared prediction error value by σ_k^2, *i.e.*,

$$\sigma_k^2 = E\{f_k^2(n)\} = E\{b_k^2(n)\}. \tag{3.76}$$

We prove all three symmetries now.

Symmetry of Forward and Backward Predictor Coefficients: Let $\mathbf{A}_k$ denote the vector formed by the kth order forward prediction error coefficients. Then,

$$\mathbf{A}_k = [a_{1,k} \; a_{2,k} \; \cdots \; a_{k,k}]^T. \tag{3.77}$$

Let us also define the input vector $\mathbf{X}_k(n-1)$ as formed by the k samples of $x(n-1)$ used in the forward prediction process, *i.e.*,

$$\mathbf{X}_k(n-1) = [x(n-1) \; x(n-2) \; \cdots \; x(n-k)]^T. \tag{3.78}$$

Finally, let $\hat{x}_k^{(f)}(n)$ denote the kth order forward prediction of $x(n)$. Using the definitions in (3.77) and (3.78), we see that

$$\hat{x}_k^{(f)}(n) = \mathbf{A}_k^T \mathbf{X}_k(n-1). \tag{3.79}$$

The optimal solution for the coefficient vector $\mathbf{A}_k$ that minimizes the mean-squared prediction error is given by

$$\mathbf{A}_k = \left(E\{\mathbf{X}_k(n-1)\mathbf{X}_k^T(n-1)\}\right)^{-1} E\{x(n)\mathbf{X}_k(n-1)\}. \tag{3.80}$$

For the kth order backward prediction problem, define the coefficient vector as

$$\mathbf{G}_k^{'} = [g_{k-1,k} \; g_{k-2,k} \; \cdots g_{0,k}]^T \tag{3.81}$$

Note that the coefficients in $\mathbf{G}_k^{'}$ are arranged in the reverse order of the normal arrangement such that the input vector corresponding to this coefficient vector is

$$\mathbf{X}_k^{'}(n) = [x(n-k+1) \; x(n-k+2) \; \cdots \; x(n)]^T. \tag{3.82}$$

The k-th order backward prediction of $x(n-k)$ can be expressed using the above definitions as

$$\hat{x}_k^{(b)}(n-k) = \left(\mathbf{G}_k'\right)^T \mathbf{X}_k'(n). \tag{3.83}$$

The MMSE solution for the $\mathbf{G}_k'$ is

$$\mathbf{G}_k' = \left(E\left\{\mathbf{X}_k'(n)\left(\mathbf{X}_k'(n)\right)^T\right\}\right)^{-1} E\left\{x(n-k)\mathbf{X}_k'(n)\right\}. \tag{3.84}$$

Since $x(n)$ is a stationary signal, one can show by direct evaluation of the expectations that

$$E\left\{\mathbf{X}_k'(n)\left(\mathbf{X}_k'(n)\right)^T\right\} = E\left\{\mathbf{X}_k(n-1)\mathbf{X}_k^T(n-1)\right\} \tag{3.85}$$

and

$$E\left\{x(n-k)\mathbf{X}_k'(n)\right\} = E\left\{x(n)\ \mathbf{X}_k(n-1)\right\}. \tag{3.86}$$

Substituting the above results in (3.84) we can see that

$$\mathbf{A}_k = \mathbf{G}_k'. \tag{3.87}$$

Equating the corresponding elements of $\mathbf{A}_k$ and $\mathbf{G}_k'$ results in the symmetry relationship of (3.74).

Symmetry of Prediction Error Powers: The MMSE value of the kth order forward prediction error can be evaluated using (2.50) as

$$E\{f_k^2(n)\} = E\{x^2(n)\} - \mathbf{A}_k^T E\{x(n)\ \mathbf{X}_k(n-1)\}. \tag{3.88}$$

Similarly, we can evaluate the MMSE value of the backward prediction error as

$$E\{b_k^2(n)\} = E\{x^2(n-k)\} - \left(\mathbf{G}_k'\right)^T E\{x(n-k)\mathbf{X}_k'(n)\}. \tag{3.89}$$

Substituting (3.86) and (3.87) in (3.89) and using the stationarity of $x(n)$, we see that the right-hand-sides of (3.88) and (3.89) are identical. Thus, (3.76) follows.

Equality of the Forward and Backward Reflection Coefficients: Recall from (3.72) that $\rho_k^{(b)}$ is given by

$$\rho_k^{(b)} = \frac{E\{b_{k-1}(n-1)f_{k-1}(n)\}}{E\{f_{k-1}^2(n)\}}. \tag{3.90}$$

Similarly,

$$\rho_k^{(f)} = \frac{E\{f_{k-1}(n)b_{k-1}(n-1)\}}{E\{b_{k-1}^2(n-1)\}}, \tag{3.91}$$

respectively. Since the numerators of the two equations are identical and the denominators are also identical because of (3.76), we see that the two reflection coefficients are also identical.

Order-Recursive Computation of the Mean-Squared Prediction Error Values

The above results provide us with an easy way for recursively computing the mean-squared values of the prediction error signals. Using the results in (2.50) with that in (3.65) and making use of the symmetries derived above, we get

$$E\{f_k^2(n)\} = E\{f_{k-1}^2(n-1)\} - \rho_k E\{b_{k-1}(n-1) f_{k-1}(n)\}. \tag{3.92}$$

Now, from (3.72), we note that

$$E\{b_{k-1}(n-1) f_{k-1}(n)\} = \rho_k E\{f_{k-1}^2(n-1)\}. \tag{3.93}$$

Substituting (3.93) in (3.92), we get the following simplified recursion for $E\{f_k^2(n)\}$:

$$E\{f_k^2(n)\} = \left(1 - \rho_k^2\right) E\{f_{k-1}^2(n)\}. \tag{3.94}$$

Since $E\{f_0^2(n)\} = E\{x^2(n)\} = r_{xx}(0)$, we can evaluate the mean-square prediction error value at any stage of the predictor if all the reflection coefficients up to that stage are known.

Order Update for the Forward Predictor Coefficients

Assume that we have already computed the predictor coefficients and mean-squared prediction error values for the $(k-1)$th order forward predictor. We can evaluate the reflection coefficient ρ_k using (3.72) and (3.67), given $a_{l,k-1}$ for $l = 1, 2, \cdots, k-1$. For this calculation, we proceed as follows: The kth order forward prediction of $x(n)$ is given by

$$\hat{x}_k^{(f)}(n) = \sum_{l=1}^{k} a_{l,k} x(n-l). \tag{3.95}$$

We can also express this quantity as the sum of the $(k-1)$th order forward prediction of $x(n)$ and the estimate of $x(n)$ using $b_{k-1}(n-1)$ as can be seen from the discussion surrounding (3.65). Consequently, we can represent the predicted value of the input signal using its most recent k samples as

$$\hat{x}_k^{(f)}(n) = \sum_{l=1}^{k-1} a_{l,k-1} x(n-l) + \rho_k \left(x(n-k) - \sum_{l=0}^{k-2} g_{l,k-1} x(n-l-1) \right). \tag{3.96}$$

Finally, substituting $a_{k-1-l,k-1}$ for $g_{l,k-1}$ in the above equation coupled with some minor simplifications in the indexing of the summations gives

$$\hat{x}_k^{(f)}(n) = \rho_k x(n-k) + \sum_{l=1}^{k-1} (a_{l,k-1} - \rho_k a_{k-l,k-1}) x(n-l). \tag{3.97}$$

Table 3.3: The Levinson-Durbin algorithm

Given:
$\mathbf{R_{xx}}$ – $(L+1) \times (L+1)$ autocorrelation matrix (Toeplitz)
L – Order of prediction

Initialization

$$\begin{aligned} \rho_1 &= \frac{r_{xx}(1)}{r_{xx}(0)} \\ a_{1,1} &= \rho_1 \\ \sigma_1^2 &= r_{xx}(0)(1-\rho_1^2) \end{aligned}$$

Main Iteration
Do for $k = 2, 3, \cdots, L$

$$\begin{aligned} \rho_k &= \frac{r_{xx}(k) - \sum_{l=1}^{k-1} a_{l,k-1} r_{xx}(k-l)}{\sigma_{k-1}^2} \\ a_{k,k} &= \rho_k \\ a_{l,k} &= a_{l,k-1} - \rho_k a_{k-l,k-1}; \quad l = 1, 2, \cdots, k-1 \\ \sigma_k^2 &= \sigma_{k-1}^2(1-\rho_k^2) \end{aligned}$$

Equating the coefficients of $x(n-l)$ on the right-hand-sides of equations (3.95) and (3.97) results in the desired update strategy:

$$a_{k,k} = \rho_k \tag{3.98}$$

and

$$a_{l,k} = a_{l,k-1} - \rho_k a_{k-l,k-1}; \quad l = 1, 2, \cdots, k-1. \tag{3.99}$$

The Complete Algorithm

We now have all the necessary information for computing the coefficients of the lattice structure for orthogonalizing the input signals. The iterative equations for updating the coefficients from order $k-1$ to order k is given in Table 3.3. These equations constitute the Levinson-Durbin algorithm. A MATLAB program that calculates the lattice coefficients from knowledge of the input signal's autocorrelation matrix is given in Figure 3.8. Another MATLAB script that implements the lattice filter from a given set of coefficients is given in Figure 3.9.

```
    function [rho,sig,a]= ld(rxx,L)
%
%
%   The Levinson-Durbin Algorithm
%
%
%   Input:
%       L - Order of the predictor
%       rxx - Autocorrelation of input for lags 0 to L
%           (L+1)-element row vector
%
%   Output:
%       rho - reflection coefficients (L-element vector)
%       sig - prediction error powers for orders 1 to L
%           (L-element vector)
%               a - matrix containing coefficients of direct form
%                       predictors of order 1 to L
%
%

%
%   Initialization
%
    Rxx=toeplitz(rxx);
    rho(1)=rxx(2)/rxx(1);
    a(1,1)=rho(1);
    sig(1)=rxx(1)*(1-rho(1)*rho(1));
%
%   Main Iteration
%
    for k=2:L,
        rho(k) = (rxx(k+1) - Rxx(k+1,[2:k])*a([1:k-1],k-1))/sig(k-1);
        a(k,k)=rho(k);
        a([1:k-1],k)=a([1:k-1],k-1)-rho(k)*a([k-1:-1:1],k-1);
        sig(k) = sig(k-1)*(1-rho(k)*rho(k));
    end;
%
%   We are done.
%
```

Figure 3.8: MATLAB function for evaluating the reflection coefficients from the autocorrelation values of the input signal.

```
    function [f] = lat_pred(x,P,rho,L)
%
%
%   Lattice Predictor
%
%
%   Input:
%       L - Order of prediction
%       rho - vector of L reflection coefficients
%       P - Number of input samples
%       x - input signal of length P samples (row vector)
%
%   Output:
%       f - (L)th order forward prediction error
%           signal of length P samples
%

%
%   Initialization
%
    f=x;
    b=x;
%
%   Main Iteration
%
    for k=1:L,
        b1=[0,b([1:P-1])];
        temp=f-rho(k)*b1;
        b=b1-rho(k)*f;
        f=temp;
    end;
```

Figure 3.9: MATLAB function for implementing a lattice predictor.

Example 3.5: Derivation of a Three-Stage Lattice Predictor

Let the 4×4-element autocorrelation matrix of a wide sense stationary signal $x(n)$ be given by

$$\mathbf{R_{xx}} = \begin{bmatrix} 1.0 & 0.8 & 0.4 & 0.1 \\ 0.8 & 1.0 & 0.8 & 0.4 \\ 0.4 & 0.8 & 1.0 & 0.8 \\ 0.1 & 0.4 & 0.8 & 1.0 \end{bmatrix}$$

Derive a three-stage lattice predictor for this signal using the MMSE criterion.

Solution: We can directly apply the Levinson-Durbin Algorithm of Table 3.3 to solve this problem. The steps are as follows:

$$\begin{aligned}
\rho_1 &= \frac{r_{xx}(1)}{r_{xx}(0)} && = 0.8 \\
a_{1,1} &= \rho_1 && = 0.8 \\
\sigma_1^2 &= r_{xx}(0)(1-\rho_1^2) && = 0.36 \\
\rho_2 &= \frac{r_{xx}(2) - a_{1,1} r_{xx}(1)}{\sigma_1^2} && = -0.67 \\
a_{2,2} &= \rho_2 && = -0.67 \\
a_{1,2} &= a_{1,1} - \rho_2 a_{1,1} && = 1.33 \\
\sigma_2^2 &= \sigma_1^2(1-\rho_2^2) && = 0.20 \\
\rho_3 &= \frac{r_{xx}(3) - a_{1,2} r_{xx}(2) - a_{2,2} r_{xx}(1)}{\sigma_2^2} && = 0.50 \\
a_{3,3} &= \rho_3 && = 0.50 \\
a_{1,3} &= a_{1,2} - \rho_3 a_{2,2} && = 1.67 \\
a_{2,3} &= a_{2,2} - \rho_3 a_{1,2} && = -1.33
\end{aligned}$$

Transformation of Reflection Coefficients to Direct Form Predictor Coefficients

We can use equations (3.98) and (3.99) to determine a set of reflection coefficients from a set of equivalent direct form predictor coefficients and vice versa. To transform the reflection coefficients to the direct form coefficients, the algorithm can be initialized with

$$a_{1,1} = \rho_1 \tag{3.100}$$

and then the iterations can be carried out on equations (3.98) and (3.99) for $k = 2, 3, \cdots, L$. The relevant equations that perform this transformation are summarized in Table 3.4.

Example 3.6: Transformation of Reflection Coefficients to Direct Form Predictor Coefficients

Table 3.4: Transformation of reflection coefficients to direct form predictor coefficients.

Given: ρ_l; $l = 1, 2, \cdots, L$; – Reflection coefficients upto stage L

Initialization

$$a_{1,1} = \rho_1$$

Main Iteration
Do for $k = 2, 3, \cdots, L$

$$a_{k,k} = \rho_k$$
$$a_{l,k} = a_{l,k-1} - \rho_k a_{k-l,k-1}; \quad l = 1, 2, \cdots, k-1.$$

Obtain the direct form representation of a lattice predictor whose reflection coefficients are given by $\rho_1 = 0.8$, $\rho_2 = -0.67$ and $\rho_3 = 0.50$.

Solution: Note that the lattice predictor is identical to that in Example 3.5. Consequently, we can verify the results of our calculations with those in that example. The transformation to direct form can be achieved using the equations in Table 3.4. The relevant steps are as follows:

$$\begin{array}{lcl} a_{1,1} &=& \rho_1 &=& 0.80 \\ a_{2,2} &=& \rho_2 &=& -0.67 \\ a_{1,2} &=& a_{1,1} - \rho_2 a_{1,1} &=& 1.33 \\ a_{3,3} &=& \rho_3 &=& 0.5 \\ a_{1,3} &=& a_{1,2} - \rho_3 a_{2,2} &=& 1.67 \\ a_{2,3} &=& a_{2,2} - \rho_3 a_{1,2} &=& -1.33 \end{array}$$

A block diagram of the direct form realization of the predictor is shown in Figure 3.10

Transformation of Direct Form Coefficients to Reflection Coefficients

To perform this transformation using equations (3.98) and (3.99), we initialize our iterations with

$$\rho_L = a_{L,L} \tag{3.101}$$

and then solve for $\rho_{L-1}, \rho_{L-2}, \rho_{L-3}$, etc. in the descending order of the coefficient index. To solve for ρ_{k-1}, let us assume that we have already computed $a_{l,k}$ for $l = 1,\ 2,\ \cdots,\ k$. Recall also that

$$\rho_k = a_{k,k}. \tag{3.102}$$

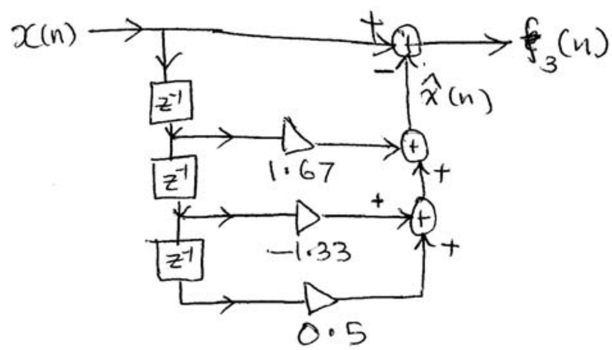

Figure 3.10: Direct form realization of the predictor in Example 3.6.

Now, from (3.99), we see that

$$a_{k-l,k} = a_{k-l,k-1} - \rho_k \, a_{l,k-1}; l = 1, \; 2, \; \cdots, \; k-1. \tag{3.103}$$

In equations (3.99) and (3.103), we have two simultaneous equations in two unknowns, namely $a_{l,k-1}$ and $a_{k-l,k-1}$. We can eliminate $a_{k-l,k-1}$ from these two equations by multiplying (3.103) with ρ_k and adding the result to (3.99). This operation results in

$$a_{l,k} + \rho_k \, a_{k-l,k} = (1 - \rho_k^2) \, a_{l,k-1}; l = 1, \; 2, \; \cdots, \; k-1. \tag{3.104}$$

Assuming that $\rho_k^2 \neq 1$, we can solve for $a_{l,k-1}$to get

$$a_{l,k-1} = \frac{a_{l,k} + \rho_k \, a_{k-l,k}}{(1 - \rho_k^2)} \tag{3.105}$$

Once the direct form coefficients for the $(k-1)$th order forward predictor have been calculated, we can obtain ρ_{k-1} as

$$\rho_{k-1} = a_{k-1,k-1}. \tag{3.106}$$

The complete iterations are summarized in Table 3.5.

Example 3.7: Conversion of the Direct Form Predictor to the Lattice Form

Find the lattice realization of the three-stage forward predictor given in Figure 3.10 using the results in Table 3.5.

Table 3.5: Transformation of direct form predictor coefficients to reflection coefficients.

Given: $a_{l,L}; l = 1, 2, \cdots, L;$ Lth order predictor coefficients

Initialization

$$\rho_L = a_{L,L}$$

Main Iteration
Do for $k = L, L-1, \cdots, 2$

$$a_{l,k-1} = \frac{a_{l,k} + \rho_k a_{k-l,k}}{(1-\rho_k^2)}; l = 1, 2, \cdots, k-1$$

$$\rho_{k-1} = a_{k-1,k-1}$$

Solution: The steps involved in the transformation are as follows:

$$\begin{aligned}
\rho_3 &= a_{3,3} &&= 0.50 \\
a_{1,2} &= \frac{a_{1,3} + \rho_3 a_{2,3}}{(1-\rho_3^2)} &&= 1.33 \\
a_{2,2} &= \frac{a_{2,3} + \rho_3 a_{1,3}}{(1-\rho_3^2)} &&= -0.67 \\
\rho_2 &= a_{2,2} &&= -0.67 \\
a_{1,1} &= \frac{a_{1,2} + \rho_2 a_{1,2}}{(1-\rho_2^2)} &&= 0.80 \\
\rho_1 &= a_{1,1} = 0.80
\end{aligned}$$

It should not surprise the reader to realize that the transformed predictor is identical to that shown in Example 3.5.

3.2.3 Lattice Joint Process Estimator

Since the backward prediction error signals $b_0(n), b_1(n), \cdots, b_{l-1}(n)$ form an orthogonal basis set for $x(n), x(n-1), \cdots, x(n-l+1)$, the lattice joint processor estimator estimates a desired response signal $d(n)$ as a linear combination of the backward prediction error signals. Figure 3.11 displays a lattice joint process estimator. The coefficients of the joint process estimator can be found using (3.41) as

$$h_k = \frac{E\{d(n)b_k(n)\}}{E\{b_k^2(n)\}}; k = 0, 1, \cdots, L-1. \tag{3.107}$$

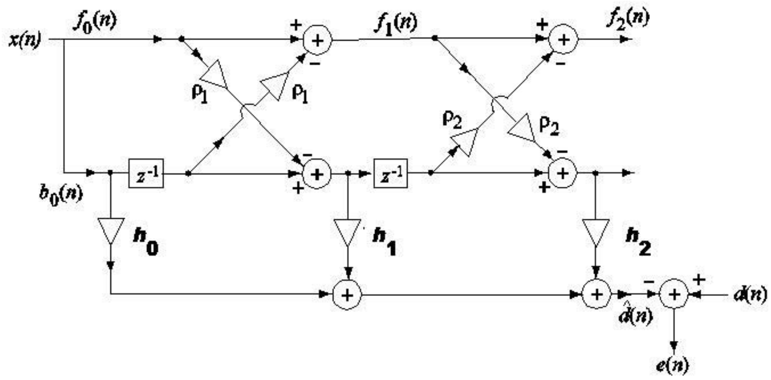

Figure 3.11: A lattice joint process estimator.

It is clear from (3.107) that one can calculate the coefficient of $b_k(n)$ in the joint process estimator for all k if the cross-correlation function $E\{d(n)b_k(n)\}$ can be evaluated for various values of k. The calculation of the cross-correlation functions can be performed as follows:

$$\begin{aligned} E\{d(n)b_k(n)\} &= E\left\{d(n)\left(x(n-k) - \sum_{l=0}^{k-1} g_{l,k}x(n-l)\right)\right\} \\ &= E\left\{d(n)\left(x(n-k) - \sum_{l=0}^{k-1} a_{k-l,k}x(n-l)\right)\right\} \\ &= r_{dx}(k) - \sum_{l=0}^{k-1} a_{k-l,k} r_{dx}(l). \end{aligned} \tag{3.108}$$

The complete algorithm for joint process estimation employing lattice orthogonalization is summarized in Table 3.6. A MATLAB program that implements this algorithm is shown in Figure 3.12.

3.2.4 Lattice Orthogonalization for Arbitrary Vector Spaces

Even though the derivations in this section were specifically done for MMSE estimation problems, they apply to problems formulated in a larger class of vector spaces. It is left as an exercise for the reader to show that the algorithms in Tables 3.3, 3.4, 3.5 and 3.6 are valid for any vector space and signal set in which the autocorrelation matrix is Toeplitz. One case of particular interest to us for which this is indeed the case is the least-squares formulation using the autocorrelation method. Other types of least-squares formulations do not result in Toeplitz autocorrelation matrices. Consequently, appropriate modifications must be made during the derivation of the lattice filter structures for such formulations. One class of such algorithms is discussed in Chapter 12, where we discuss exponentially weighted recursive least-squares adaptive filters.

Table 3.6: Algorithm for finding the coefficients of the lattice joint process estimator.

Given : $\mathbf{R_{xx}}$ – $L \times L$ – element autocorrelation matrix (Toeplitz)
$\mathbf{P}_{d\mathbf{x}}$ – $L \times 1$ – cross-correlation vector
L – Order of the joint process estimator

Initialization

$$\begin{aligned}
h_0 &= \frac{r_{dx}(0)}{r_{xx}(0)} \\
\rho_1 &= \frac{r_{xx}(1)}{r_{xx}(0)} \\
a_{1,1} &= \rho_1 \\
\sigma_1^2 &= r_{xx}(0)(1-\rho_1^2) \\
h_1 &= \frac{r_{dx}(1) - a_{1,1} r_{dx}(0)}{\sigma_1^2}
\end{aligned}$$

Main Iteration
Do for $k = 2, \cdots, L-1$

$$\begin{aligned}
\rho_k &= \frac{r_{xx}(k) - \sum_{l=1}^{k-1} a_{l,k-1} r_{xx}(k-l)}{\sigma_{k-1}^2} \\
a_{k,k} &= \rho_k \\
a_{l,k} &= a_{l,k-1} - \rho_k a_{k-l,k-1}; \quad l = 1, 2, \cdots, k-1 \\
\sigma_k^2 &= \sigma_{k-1}^2 (1-\rho_k^2) \\
h_k &= \frac{r_{dx}(k) - \sum_{l=0}^{k-1} a_{k-l,k} r_{dx}(l)}{\sigma_k^2}
\end{aligned}$$

```
    function [rho,h]= ld_jp(rxx,Pdx,L)
%
%
%   Algorithm for evaluating the coefficients of the lattice
%   joint process estimator.
%
%
%   Input:
%       L - Order of the estimator
%       rxx - Autocorrelation of input for lags 0 to L-1
%           L-element row vector
%       Pdx - L-element row vector of the cross-correlation
%           function
%
%   Output:
%       rho - reflection coefficients ((L-1)-element vector)
%       sig - prediction error powers for orders 1 to L
%           (L-element vector)
%       a - matrix containg coefficients of direct form
%           predictors of order 1 to L
%
%


%
%   Initialization
%
    Rxx=toeplitz(rxx);
    rho(1)=rxx(2)/rxx(1);
    a(1,1)=rho(1);
    sig(1)=rxx(1)*(1-rho(1)*rho(1));
    h(1)=Pdx(1)/rxx(1);
    h(2)=(Pdx(2)-a(1,1)*Pdx(1))/sig(1);
%
%   Main Iteration
%
    for k=2:L-1,
        rho(k) = (rxx(k+1) - Rxx(k+1,[2:k])*a([1:k-1],k-1))/sig(k-1);
        a(k,k)=rho(k);
        a([1:k-1],k)=a([1:k-1],k-1)-rho(k)*a([k-1:-1:1],k-1);
        sig(k) = sig(k-1)*(1-rho(k)*rho(k));
        h(k+1)=(Pdx(k+1) - Pdx([1:k])*a([k:-1:1],k))/sig(k);
    end;
```

Figure 3.12: A MATLAB program for evaluating the lattice joint process estimator coefficients.

Table 3.7: Statistics of the estimates in Example 3.8.

	True value	Mean	MSD
ρ_1	0.7	0.6985	0.1731×10^{-3}
ρ_2	-0.7	-0.6983	0.1825×10^{-3}
ρ_3	0.5	0.4933	0.6699×10^{-3}
ρ_4	-0.5	-0.4895	0.9584×10^{-3}
σ_ξ^2	0.1600	0.1636	0.1012×10^{-4}

Example 3.8: Autoregressive Spectrum Estimation Using Lattice Orthogonalization

In this example, we revisit the autoregressive spectrum estimation problem considered in Example 2.15. Table 3.7 displays the statistics of the estimates of the reflection coefficients of the predictor obtained using least-squares estimation techniques. The results presented here are averages of one hundred independent estimates obtained using the same set of signals as in Example 2.15. The reflection coefficients were estimated by first estimating the autocorrelation function of the input signal using (2.74) and then substituting the estimated autocorrelation values in the Levinson-Durbin algorithm of Table 3.3. The results demonstrate the viability of using lattice orthogonalization techniques in least-squares estimation problems.

3.2.5 Additional Advantages of Lattice Filters

Our study of the lattice filter structure was motivated by its efficiency in orthogonalizing stationary input signals. However, the lattice filter structure possesses several other advantages over direct form structures, making it even attractive in many practical applications. We list several such advantages below. The discussion of these properties is brief since rigorous analyses of many of the properties are beyond the scope of this chapter.

Modularity. The lattice structure is modular in the sense that all the stages have identical structures. Consequently, lattice filters are amenable to efficient VLSI implementations. Furthermore, the complete lattice filter may be implemented by sharing a single section implemented in hardware using time division multiplexing techniques.

Independence of Lattice Sections. Since the input signals for the estimators in a lattice filter are mutually orthogonal, the coefficients for any one stage depend only on the statistics of the input signals to that stage. Furthermore, if the Lth order predictor or estimator coefficients of the lattice structure are known, so are all of the coefficients of the lower order predictors and estimators. To increase the order of the estimator or predictor, we simply cascade the additional number of sections to the already-existing estimator structure. By contrast, the direct form structures possess none of these useful properties.

Superior Numerical Properties. Lattice filters have better numerical properties in general than their direct form counterparts. This property is important since discrete-time filters are implemented with finite-precision hardware. In our discussion of recursive least-squares (RLS) adaptive filters in Chapter 10, lattice filters will play a key role in the development of numerically-stable and computationally-efficient RLS adaptive filters.

Better Convergence Properties of Adaptive Realizations. Since the lattice filter structure orthogonalizes the input signals, it turns out that stochastic gradient adaptive filters, which are discussed in Chapters 3 and 4, often exhibit faster convergence behaviors when implemented using lattice structures. This property will be discussed in more detail in Chapter 7.

Stability Check for Recursive Filters. It can be shown that a transfer function of the type

$$A_M(z) = 1 - \sum_{i=1}^{M} a_m(i) z^{-i} \tag{3.109}$$

is minimum phase[3] if and only if the corresponding reflection coefficients are less than one in magnitude. There are several applications in which we need to obtain an inverse filter for the transfer function in (3.109). In such cases, we can guarantee the stability of the inverse filter whenever all the reflection coefficients are less than one in magnitude. This kind of stability check is not easily performed in most direct form realizations.

3.3 Main Points of This Chapter

- Gram-Schmidt orthogonalization performs a linear transformation of a set of input vectors that results in a set of orthogonal basis vectors that span the same space spanned by the input vectors.

- The coefficients of the estimates using the orthogonal basis set are, in general, different from the coefficients of the input vectors themselves. However, the estimates using both sets of vectors are identical in the sense that the estimates as well as the estimation errors are identical for both cases. Moreover, each optimal coefficient set can be obtained from a linear transformation of the other optimal coefficient set.

- The lattice structure provides an efficient method for orthogonalizing the input signals for single-channel, finite-memory, linear estimation problems. More general lattice structures for broader classes of estimation problems are possible but are beyond the scope of this chapter.

[3] A minimum phase FIR filter has all its zeroes inside the unit circle.

- The lattice filter structure has several additional advantages over direct form filter structures, such as modularity, better finite precision properties, and a simple stability check for inverse filters. Consequently, lattice filters are very attractive in many applications.

3.4 Bibliographical Notes

As was evident from this chapter, linear prediction plays a key role in linear estimation theory as well as its applications. A tutorial review on linear prediction can be found in [Makhoul 1975]. An efficient algorithm for evaluating the coefficients of the estimator when $\mathbf{R_{xx}}$ is Toeplitz was developed by Levinson [Levinson 1947]. This algorithm was later rediscovered in the context of linear prediction by Durbin [Durbin 1960]. Variations of the Levinson-Durbin algorithm that are more efficient than the one discussed in this chapter are derived in [Delsarte 1986, Krishna 1988].

A related algorithm for solving the normal equations is the Schür algorithm [Schür 1917, Gohberg 1986]. This method requires more calculations than the Levinson-Durbin algorithm, but has the advantage of better parallellizability.

Wold's work [Wold 1938] appears to be the first to suggest replacing correlated input signals by "equivalent" uncorrelated signals in estimation problems. A modified Gram-Schmidt orthogonalization technique employing pivoting was developed by Rice [Rice 1963]. Gram-Schmidt pre-processors are routinely used in adaptive antenna arrays [Compton 1993].

This chapter did not discuss in detail many properties of lattice filters, or alternate lattice structures. Good resources for more information in this area are [Makhoul 1978, Friedlander 1982, Markel 1976]. Lattice predictors appear to have been first used in applications in geophysics [Burg 1965, Robinson 1967, Robinson 1980]. They have also found use in speech analysis [Itakura 1971, Markel 1973], communication channel equalization [Satorious 1979], and spectrum estimation [Friedlander 1982], among others.

3.5 Exercises

3.1. *A Formula for Computing the Lengths of the Orthogonal Basis Vectors:* Derive (3.35) for the squared norms of the orthogonal basis vectors.

3.2. *Gram-Schmidt Orthogonalization for a Nonlinear Estimation Problem:* We are interested in estimating a random signal $d(n)$ as a memoryless, polynomial function of $x(n)$ as

$$\hat{d}(n) = a_0 + a_1 x(n) + a_2 x^2(n) + a_3 x^3(n).$$

It is known that $x(n)$ is a Gaussian signal with zero mean value and unit variance. Derive a Gram-Schmidt orthogonalizer for the signal set consisting of 1, $x(n)$, $x^2(n)$ and $x^3(n)$ for this task.

3.3. *Transformation of the Direct Form Coefficients to the Coefficients of the Orthogonal Basis Vectors:* Suppose that we know the parameters of the Gram-Schmidt orthogonalizer as well as the coefficients of the direct form estimator for an arbitrary number of coefficients. Derive a set of recursive equations for the coefficients of the orthogonal basis vectors for the same estimation problem.

3.4. *Two Interpretations of the Forward Reflection Coefficients:* Show that the forward reflection coefficient $\rho_k^{(f)}$ is the optimal MMSE coefficient for estimating $x(n)$ using $b_{k-1}(n-1)$ and for estimating $f_{k-1}(n)$ using $b_{k-1}(n-1)$.

3.5. *Levinson-Durbin Algorithm For the Least-Squares Estimation Problem:* Show that the Levinson-Durbin algorithm is also valid when the estimation problem is formulated to minimize the following least-squares cost function:

$$J = \sum_{n=0}^{P-1} (d(n) - \hat{d}(n))^2,$$

where $\hat{d}(n) = \sum_{i=0}^{L-1} w_i x(n-i)$.

Hint: Note that the least-squares autocorrelation matrix for this formulation is Toeplitz.

3.6. *Bound on Reflection Coefficients:* Show that the optimum MMSE reflection coefficients of a lattice predictor are bounded by one when the input signal is stationary and has non-zero, but finite variance.

3.7. *The Minimum Mean-Square Prediction Error is a Non-Increasing Function of the Prediction Order:* Show, using the expression for updating the mean-square prediction error, that the minimum mean-square prediction error is a monotone non-increasing function of the prediction order.

3.8. *A Normalized Lattice Predictor:* Consider the predictor structure shown in Figure 3.13. The parameters ρ_1 and ρ_2 are the optimal MMSE reflection coefficients for the input signal $x(n)$. Show that the signal set $b_0(n), b_1(n)$ and $b_2(n)$ have unit variance and are mutually uncorrelated. Show also that $b_0(n), b_1(n)$ and $b_2(n)$ form an orthogonal basis set for $x(n), x(n-1)$ and $x(n-2)$. The structure shown is known as the normalized lattice predictor.

3.9. *Inverse Lattice Predictor:* In many applications, it is required to implement the inverse of a lattice predictor given by the transfer function

$$H^{-1}(z) = \frac{1}{1 - A(z)}.$$

An example of such a situation is linear predictive coding. Consider a second-order lattice predictor with reflection coefficients ρ_1 and ρ_2. Show that the inverse system

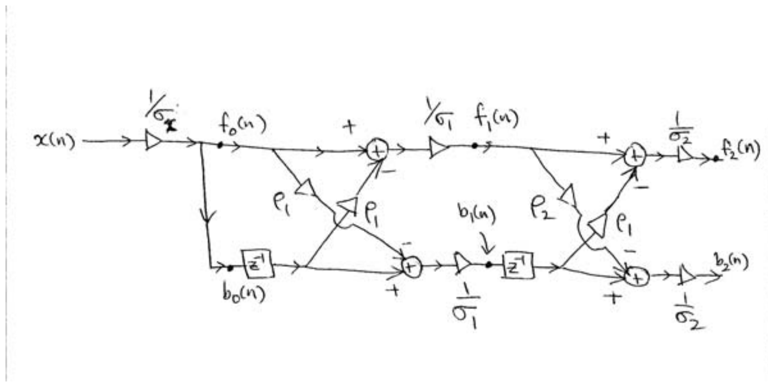

Figure 3.13: The normalized lattice predictor.

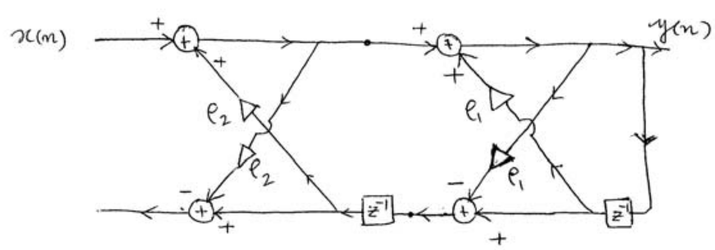

Figure 3.14: The all-pole lattice filter.

for this predictor is given by the system shown in Figure 3.14 Generalize the above result to arbitrary order of prediction.

3.10. *Transformation of Lattice Joint Process Estimator Coefficients to the Direct Form:* Derive a set of recursive equations to transform the set of lattice joint process estimator coefficients and the reflection coefficients to the direct form joint process estimator coefficients.

3.11. Let the input signal $x(n)$ be given by

$$x(n) = \sin(0.10n + \theta_1) + \sin(0.25n + \theta_2),$$

where θ_1 and θ_2 are uniformly distributed random variables in the range $[-\pi, \pi)$, and are uncorrelated with each other. Find the optimum fourth-order lattice MMSE predictor for this signal in the following two different ways:

a. Find the autocorrelation function of $x(n)$ and solve for the predictor coefficients using the Levinson-Durbin algorithm.

b. Find the direct form predictor coefficients using the results of Exercise 3.10 and transform them into the lattice form.

3.12. *Multichannel Levinson-Durbin Algorithm:* Consider a stationary two-channel signal $\mathbf{X}(n) = [x_1(n)\ x_2(n)]^T$. We desire to develop a lattice filter to orthogonalize the most recent L samples of this signal such that the "backward prediction error vectors" $\mathbf{b}_0(n), \mathbf{b}_1(n), \ldots, \mathbf{b}_{L-1}(n)$ is an orthogonal basis set for the vectors $\mathbf{X}(n), \mathbf{X}(n-1), \ldots, \mathbf{X}(n-L+1)$. The orthogonality is in the mean-square sense such that

$$E\left\{\mathbf{b}_i(n)\mathbf{b}_j^T(n)\right\} = \mathbf{R}_{\mathbf{bb}}(i)\delta(i-j),$$

where $\mathbf{R}_{\mathbf{bb}}(i)$ is the 2×2-element autocorrelation matrix of the two-element vector $\mathbf{b}_i(n)$.

a. Develop an extended version of the Levinson-Durbin algorithm to accomplish the orthogonalization of the two-channel signal.

b. Suppose we also desire that the elements of each backward prediction error vector be mutually uncorrelated, *i.e.*, we desire that $\mathbf{R}_{\mathbf{bb}}(i)$ be a diagonal matrix. Combine the procedure in part (a) with a Gram-Schmidt orthogonalizer for each backward prediction error vector to develop a completely orthogonal basis signal set.

3.13. *Computing Assignment: System Identification Using Lattice Orthogonalization.* Repeat Exercise 2.7 using lattice parameterization. Convert the lattice coefficients to direct form coefficients before evaluating the performance measures. Explain any differences or lack of difference in the performance measure calculated using the two approaches.