11

ADAPTIVE EQUALIZATION

In Chapter 10, we introduced both optimum and suboptimum receivers that compensate for ISI in the transmission of digital information through band-limited, nonideal channels. The optimum receiver employed maximum-likelihood sequence estimation for detecting the information sequence from the samples of the demodulation filter. The suboptimum receivers employed either a linear equalizer or a decision-feedback equalizer.

In the development of the three equalization methods, we implicitly assumed that the channel characteristics, either the impulse response or the frequency response, were known at the receiver. However, in most communication systems that employ equalizers, the channel characteristics are unknown a priori and, in many cases, the channel response is time-variant. In such a case, the equalizers are designed to be adjustable to the channel response and, for time-variant channels, to be adaptive to the time variations in the channel response.

In this chapter, we present algorithms for automatically adjusting the equalizer coefficients to optimize a specified performance index and to adaptively compensate for time variations in the channel characteristics. We also analyze the performance characteristics of the algorithm, including their rate of convergence and their computational complexity.

11-1 ADAPTIVE LINEAR EQUALIZER

In the case of the linear equalizer, recall that we considered two different criteria for determining the values of the equalizer coefficients $\{c_k\}$. One criterion was based on the minimization of the peak distortion at the output of the equalizer, which is defined by (10-2-4). The other criterion was based on the minimization of the mean-square error at the output of the equalizer, which is defined by (10-2-25). Below, we describe two algorithms for performing the optimization automatically and adaptively.

11-1-1 The Zero-Forcing Algorithm

In the peak-distortion criterion, the peak distortion $\mathcal{D}(\mathbf{c})$, given by (10-2-22), is minimized by selecting the equalizer coefficients $\{c_k\}$. In general, there is no simple computational algorithm for performing this optimization, except in the special case where the peak distortion at the input to the equalizer, defined as $\mathcal{D}_0$ in (10-2-23), is less than unity. When $\mathcal{D}_0 < 1$, the distortion $\mathcal{D}(\mathbf{c})$ at the output of the equalizer is minimized by forcing the equalizer response $q_n = 0$, for $1 \leq |n| \leq K$, and $q_0 = 1$. In this case, there is a simple computational algorithm, called the zero-forcing algorithm, that achieves these conditions.

The zero-forcing solution is achieved by forcing the cross-correlation between the error sequence $\varepsilon_k = I_k - \hat{I}_k$ and the desired information sequence $\{I_k\}$ to be zero for shifts in the range $0 \leq |n| \leq K$. The demonstration that this leads to the desired solution is quite simple. We have

$$\begin{aligned} E(\varepsilon_k I^*_{k-j}) &= E[(I_k - \hat{I}_k)I^*_{k-j}] \\ &= E(I_k I^*_{k-j}) - E(\hat{I}_k I^*_{k-j}), \quad j = -K, \ldots, K \end{aligned} \tag{11-1-1}$$

We assume that the information symbols are uncorrelated, i.e., $E(I_k I_j^*) = \delta_{kj}$, and that the information sequence $\{I_k\}$ is uncorrelated with the additive noise sequence $\{\eta_k\}$. For $\hat{I}_k$, we use the expression given in (10-2-41). Then, after taking the expected values in (11-1-1), we obtain

$$E(\varepsilon_k I^*_{k-j}) = \delta_{j0} - q_j, \quad j = -K, \ldots, K \tag{11-1-2}$$

Therefore, the conditions

$$E(\varepsilon_k I^*_{k-j}) = 0, \quad j = -K, \ldots, K \tag{11-1-3}$$

are fulfilled when $q_0 = 1$ and $q_n = 0$, $1 \leq |n| \leq K$.

When the channel response is unknown, the cross-correlations given by (11-1-1) are also unknown. This difficulty can be circumvented by transmitting a known training sequence $\{I_k\}$ to the receiver, which can be used to estimate the cross-correlation by substituting time averages for the ensemble averages given in (11-1-1). After the initial training, which will require the transmission of a training sequence of some predetermined length that equals or exceeds the equalizer length, the equalizer coefficients that satisfy (11-1-3) can be determined.

A simple recursive algorithm for adjusting the equalizer coefficients is

$$c_j^{(k+1)} = c_j^{(k)} + \Delta \varepsilon_k I_{k-j}^*, \qquad j = -K, \ldots, -1, 0, 1, \ldots, K \tag{11-1-4}$$

where $c_j^{(k)}$ is the value of the jth coefficient at time $t = kT$, $\varepsilon_k = I_k - \hat{I}_k$ is the error signal at time $t = kT$, and Δ is a scale factor that controls the rate of adjustment, as will be explained later in this section. This is the *zero-forcing algorithm*. The term $\varepsilon_k I_{k-j}^*$ is an estimate of the cross-correlation (ensemble average) $E(\varepsilon_k I_{k-j}^*)$. The averaging operation of the cross-correlation is accomplished by means of the recursive first-order difference equation algorithm in (11-1-4), which represents a simple discrete-time integrator.

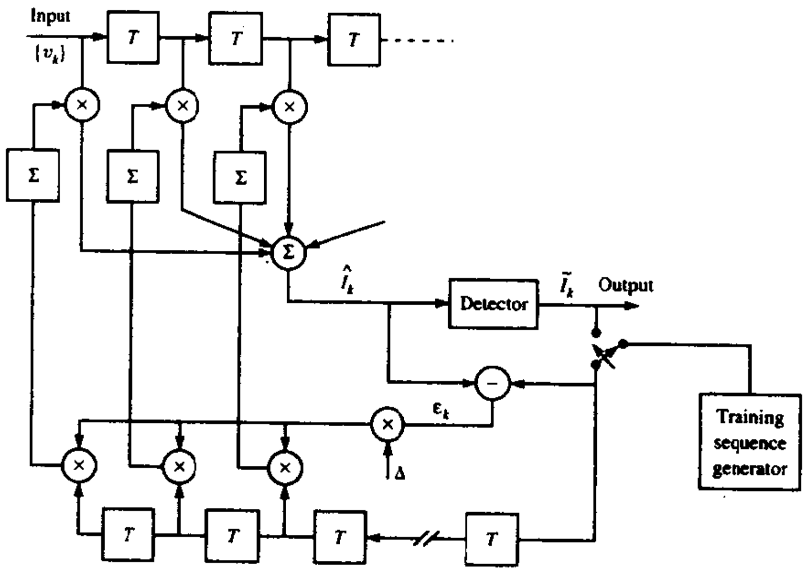
Following the training period, after which the equalizer coefficients have converged to their optimum values, the decisions at the output of the detector are generally sufficiently reliable so that they may be used to continue the coefficient adaptation process. This is called a *decision-directed mode* of adaptation. In such a case, the cross-correlations in (11-1-4) involve the error signal $\tilde{\varepsilon}_k = \tilde{I}_k - \hat{I}_k$ and the detected output sequence $\tilde{I}_{k-j}$, $j = -K, \ldots, K$. Thus, in the adaptive mode, (11-1-4) becomes

$$c_j^{(k+j)} = c_j^{(k)} + \Delta \tilde{\varepsilon}_k \tilde{I}_{k-j}^* \tag{11-1-5}$$

Figure 11-1-1 illustrates the zero-forcing equalizer in the training mode and the adaptive mode of operation.

The characteristics of the zero-forcing algorithm are similar to those of the LMS algorithm, which minimizes the MSE and which is described in detail in the following section.

FIGURE 11-1-1 An adaptive zero-forcing equalizer.



11-1-2 The LMS Algorithm

In the minimization of the MSE, treated in Section 10-2-2, we found that the optimum equalizer coefficients are determined from the solution of the set of linear equations, expressed in matrix form as

$$\mathbf{\Gamma C} = \boldsymbol{\xi} \tag{11-1-6}$$

where $\mathbf{\Gamma}$ is the $(2K+1)\times(2K+1)$ covariance matrix of the signal samples $\{v_k\}$, $\mathbf{C}$ is the column vector of $(2K+1)$ equalizer coefficients, and $\boldsymbol{\xi}$ is a $(2K+1)$-dimensional column vector of channel filter coefficients. The solution for the optimum equalizer coefficients vector $\mathbf{C}_{\text{opt}}$ can be determined by inverting the covariance matrix $\mathbf{\Gamma}$, which can be efficiently performed by use of the Levinson–Durbin algorithm described in Appendix A.

Alternatively, an iterative procedure that avoids the direct matrix inversion may be used to compute $\mathbf{C}_{\text{opt}}$. Probably the simplest iterative procedure is the method of steepest descent, in which one begins by arbitrarily choosing the vector $\mathbf{C}$, say as $\mathbf{C}_0$. This initial choice of coefficients corresponds to some point on the quadratic MSE surface in the $(2K+1)$-dimensional space of coefficients. The gradient vector $\mathbf{G}_0$, having the $2K+1$ gradient components $\frac{1}{2}\,\partial J/\partial c_{0k}$, $k=-K,\ldots,-1,0,1,\ldots,K$, is then computed at this point on the MSE surface, and each tap weight is changed in the direction opposite to its corresponding gradient component. The change in the jth tap weight is proportional to the size of the jth gradient component. Thus, succeeding values of the coefficient vector $\mathbf{C}$ are obtained according to the relation

$$\mathbf{C}_{k+1} = \mathbf{C}_k - \Delta \mathbf{G}_k, \qquad k = 0, 1, 2, \ldots \tag{11-1-7}$$

where the gradient vector $\mathbf{G}_k$ is

$$\mathbf{G}_k = \frac{1}{2}\frac{dJ}{d\mathbf{C}_k} = \mathbf{\Gamma C}_k - \boldsymbol{\xi} = -E(\varepsilon_k \mathbf{V}_k^*) \tag{11-1-8}$$

The vector $\mathbf{C}_k$ represents the set of coefficients at the kth iteration, $\varepsilon_k = I_k - \hat{I}_k$ is the error signal at the kth iteration, $\mathbf{V}_k$ is the vector of received signal samples that make up the estimate $\hat{I}_k$, i.e., $\mathbf{V}_k = [v_{k+K} \;\ldots\; v_k \;\ldots\; v_{k-K}]'$, and Δ is a positive number chosen small enough to ensure convergence of the iterative procedure. If the minimum MSE is reached for some $k = k_0$ then $\mathbf{G}_k = \mathbf{0}$, so that no further change occurs in the tap weights. In general, $J_{\min}(K)$ cannot be attained for a finite value of k_0 with the steepest-descent method. It can, however, be approached as closely as desired for some finite value of k_0.

The basic difficulty with the method of steepest descent for determining the optimum tap weights is the lack of knowledge of the gradient vector $\mathbf{G}_k$, which depends on both the covariance matrix $\mathbf{\Gamma}$ and the vector $\boldsymbol{\xi}$ of cross-correlations. In turn, these quantities depend on the coefficients $\{f_k\}$ of the equivalent discrete-time channel model and on the covariance of the information sequence and the additive noise, all of which may be unknown at the receiver in general. To overcome the difficulty, estimates of the gradient vector may be used. That is, the algorithm for adjusting the tap weight coefficients may be expressed in the form

$$\hat{\mathbf{C}}_{k+1} = \hat{\mathbf{C}}_k - \Delta \hat{\mathbf{G}}_k \tag{11-1-9}$$

where $\hat{\mathbf{G}}_k$ denotes an estimate of the gradient vector $\mathbf{G}_k$ and $\hat{\mathbf{C}}_k$ denotes the estimate of the vector of coefficients.

From (11-1-8) we note that $\mathbf{G}_k$ is the negative of the expected value of the $\varepsilon_k \mathbf{V}_k^*$. Consequently, an estimate of $\mathbf{G}_k$ is

$$\hat{\mathbf{G}}_k = -\varepsilon_k \mathbf{V}_k^* \tag{11-1-10}$$

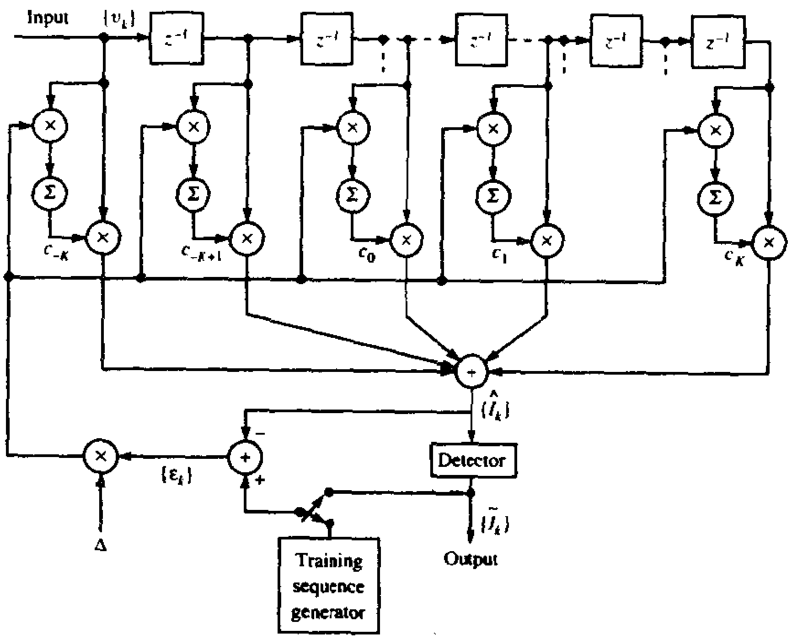
Since $E(\hat{\mathbf{G}}_k) = \mathbf{G}_k$, the estimate $\hat{\mathbf{G}}_k$ is an unbiased estimate of the true gradient vector $\mathbf{G}_k$. Incorporation of (11-1-10) into (11-1-9) yields the algorithm

$$\hat{\mathbf{C}}_{k+1} = \hat{\mathbf{C}}_k + \Delta \varepsilon_k \mathbf{V}_k^* \tag{11-1-11}$$

This is the basic LMS (least-mean-square) algorithm for recursively adjusting the tap weight coefficients of the equalizer first proposed by Widrow and Hoff (1960). It is illustrated in the equalizer shown in Fig. 11-1-2.

The basic algorithm given by (11-1-11) and some of its possible variations have been incorporated into manv commercial adaptive equalizers that are

FIGURE 11-1-2 Linear adaptive equalizer based on MSE criterion.



used in high-speed modems. Three variations of the basic algorithm are obtained by using only sign information contained in the error signal ε_k and/or in the components of $\mathbf{V}_k$. Hence, the three possible variations are

$$c_{(k+1)j} = c_{kj} + \Delta \operatorname{csgn}(\varepsilon_k) v_{k-j}^*, \quad j = -K, \ldots, -1, 0, 1, \ldots, K \tag{11-1-12}$$

$$c_{(k+1)j} = c_{kj} + \Delta \varepsilon_k \operatorname{csgn}(v_{k-j}^*), \quad j = -K, \ldots, -1, 0, 1, \ldots, K \tag{11-1-13}$$

$$c_{(k+1)j} = c_{kj} + \Delta \operatorname{csgn}(\varepsilon_k) \operatorname{csgn}(v_{k-j}^*), \quad j = -K, \ldots, -1, 0, 1, \ldots, K \tag{11-1-14}$$

where csgn (x) is defined as

$$\operatorname{csgn}(x) = \begin{cases} 1+j & (\operatorname{Re}(x) > 0, \operatorname{Im}(x) > 0) \\ 1-j & (\operatorname{Re}(x) > 0, \operatorname{Im}(x) < 0) \\ -1+j & (\operatorname{Re}(x) < 0, \operatorname{Im}(x) > 0) \\ -1-j & (\operatorname{Re}(x) < 0, \operatorname{Im}(x) < 0) \end{cases} \tag{11-1-15}$$

(Note that in (11-1-15), $j \equiv \sqrt{-1}$, as distinct from the index j in (11-1-12)–(11-1-14).) Clearly, the algorithm in (11-1-14) is the most easily implemented, but it gives the slowest rate of convergence to the others.

Several other variations of the LMS algorithm are obtained by averaging or filtering the gradient vectors over several iterations prior to making adjustments of the equalizer coefficients. For example, the average over N gradient vectors is

$$\bar{\mathbf{G}}_{mN} = -\frac{1}{N} \sum_{n=0}^{N-1} \varepsilon_{mN+n} \mathbf{V}_{mN+n}^* \tag{11-1-16}$$

and the corresponding recursive equation for updating the equalizer coefficients once very N iterations is

$$\hat{\mathbf{C}}_{(k+1)N} = \hat{\mathbf{C}}_{kN} - \Delta \bar{\mathbf{G}}_{kN} \tag{11-1-17}$$

In effect, the averaging operation performed in (11-1-16) reduces the noise in the estimate of the gradient vector, as shown by Gardner (1984).

An alternative approach is to filter the noisy gradient vectors by a lowpass filter and use the output of the filter as an estimate of the gradient vector. For example, a simple lowpass filter for the noisy gradients yields as an output

$$\bar{\mathbf{G}}_k = w \bar{\mathbf{G}}_{k-1} + (1-w) \hat{\mathbf{G}}_k, \quad \bar{\mathbf{G}}(0) = \hat{\mathbf{G}}(0) \tag{11-1-18}$$

where the choice of $0 \leq w < 1$ determines the bandwidth of the lowpass filter. When w is close to unity, the filter bandwidth is small and the effective averaging is performed over many gradient vectors. On the other hand, when w is small, the lowpass filter has a large bandwidth and, hence, it provides little averaging of the gradient vectors. With the filtered gradient vectors given by

(11-1-18) in place of $\mathbf{G}_k$, we obtain the filtered gradient LMS algorithm given by

$$\hat{\mathbf{C}}_{k+1} = \hat{\mathbf{C}}_k - \Delta\bar{\hat{\mathbf{G}}}_k \tag{11-1-19}$$

In the above discussion, it has been assumed that the receiver has knowledge of the transmitted information sequence in forming the error signal between the desired symbol and its estimate. Such knowledge can be made available during a short training period in which a signal with a known information sequence is transmitted to the receiver for initially adjusting the tap weights. The length of this sequence must be at least as long as the length of the equalizer so that the spectrum of the transmitted signal adequately covers the bandwidth of the channel being equalized.

In practice, the training sequence is often selected to be a periodic pseudo-random sequence, such as a maximum length shift-register sequence whose period N is equal to the length of the equalizer ($N = 2K + 1$). In this case, the gradient is usually averaged over the length of the sequence as indicated in (11-1-16) and the equalizer is adjusted once a period according to (11-1-17). A practical scheme for continuous adjustment of the tap weights may be either a decision-directed mode of operation in which decisions on the information symbols are assumed to be correct and used in place of I_k in forming the error signal ε_k, or one in which a known pseudo-random-probe sequence is inserted in the information-bearing signal either additively or by interleaving in time and the tap weights adjusted by comparing the received probe symbols with the known transmitted probe symbols. In the decision-directed mode of operation, the error signal becomes $\tilde{\varepsilon}_k = \tilde{I}_k - \hat{I}_k$, where $\tilde{I}_k$ is the decision of the receiver based on the estimate $\hat{I}_k$. As long as the receiver is operating at low error rates, an occasional error will have a negligible effect on the convergence of the algorithm.

If the channel response changes, this change is reflected in the coefficients $\{f_k\}$ of the equivalent discrete-time channel model. It is also reflected in the error signal ε_k, since it depends on $\{f_k\}$. Hence, the tap weights will be changed according to (11-1-11) to reflect the change in the channel. A similar change in the tap weights occurs if the statistics of the noise or the information sequence change. Thus, the equalizer is adaptive.

11-1-3 Convergence Properties of the LMS Algorithm

The convergence properties of the LMS algorithm given by (11-1-11) are governed by the step-size parameter Δ. We shall now consider the choice of the parameter Δ to ensure convergence of the steepest-descent algorithm in (11-1-7), which employs the exact value of the gradient.

From (11-1-7) and (11-1-8), we have

$$\begin{aligned}\mathbf{C}_{k+1} &= \mathbf{C}_k - \Delta\mathbf{G}_k \\ &= (\mathbf{I} - \Delta\mathbf{\Gamma})\mathbf{C}_k + \Delta\boldsymbol{\xi}\end{aligned} \tag{11-1-20}$$

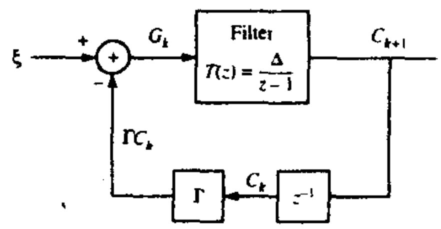


FIGURE 11-1-3 Closed-loop control system representation of recursive equation in (11-1-20).

where $\mathbf{I}$ is the identity matrix, $\mathbf{\Gamma}$ is the autocorrelation matrix of the received signal, $\mathbf{C}_k$ is the $(2K+1)$-dimensional vector of equalizer tap gains, and $\boldsymbol{\xi}$ is the vector of cross-correlations given by (10-2-45). The recursive relation in (11-1-20) can be represented as a closed-loop control system as shown in Fig. 11-1-3. Unfortunately, the set of $2K+1$ first-order difference equations in (11-1-20) are coupled through the autocorrelation matrix $\mathbf{\Gamma}$. In order to solve these equations and, thus, establish the convergence properties of the recursive algorithm, it is mathematically convenient to decouple the equations by performing a linear transformation. The appropriate transformation is obtained by noting that the matrix $\mathbf{\Gamma}$ is Hermitian and, hence, can be represented as

$$\mathbf{\Gamma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}'^* \tag{11-1-21}$$

where $\mathbf{U}$ is the normalized modal matrix of $\mathbf{\Gamma}$ and $\mathbf{\Lambda}$ is a diagonal matrix with diagonal elements equal to the eigenvalues of $\mathbf{\Gamma}$.

When (11-1-21) is substituted into (11-1-20) and if we define the transformed (orthogonalized) vectors $\mathbf{C}_k^o = \mathbf{U}'^*\mathbf{C}_k$ and $\boldsymbol{\xi}^o = \mathbf{U}'^*\boldsymbol{\xi}$, we obtain

$$\mathbf{C}_{k+1}^o = (\mathbf{I} - \Delta\mathbf{\Lambda})\mathbf{C}_k^o + \Delta\boldsymbol{\xi}^o \tag{11-1-22}$$

This set of first order difference equations is now decoupled. Their convergence is determined from the homogeneous equation

$$\mathbf{C}_{k+1}^o = (\mathbf{I} - \Delta\mathbf{\Lambda})\mathbf{C}_k^o \tag{11-1-23}$$

We see that the recursive relation will converge provided that all the poles lie inside the unit circle, i.e.,

$$|1 - \Delta\lambda_k| < 1, \qquad k = -K, \ldots, -1, 0, 1, \ldots, K \tag{11-1-24}$$

where $\{\lambda_k\}$ is the set of $2K+1$ (possibly nondistinct) eigenvalues of $\mathbf{\Gamma}$. Since $\mathbf{\Gamma}$ is an autocorrelation matrix, it is positive-definite and, hence, $\lambda_k > 0$ for all k. Consequently convergence of the recursive relation in (11-1-22) is ensured if Δ satisfies the inequality

$$0 < \Delta < \frac{2}{\lambda_{\max}} \tag{11-1-25}$$

where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{\Gamma}$.

Since the largest eigenvalue of a positive-definite matrix is less than the sum of all the eigenvalues of the matrix and, furthermore, since the sum of the eigenvalues of a matrix is equal to its trace, we have the following simple upper bound on λ_{max}:

$$\lambda_{max} < \sum_{k=-K}^{K} \lambda_k = \text{tr}\,\Gamma = (2K+1)\Gamma_{kk}$$
$$= (2K+1)(x_0 + N_0) \tag{11-1-26}$$

From (11-1-23) and (11-1-24) we observe that rapid convergence occurs when $|1 - \Delta\lambda_k|$ is small, i.e., when the pole positions are far from the unit circle. But we cannot achieve this desirable condition and still satisfy (11-1-25) if there is a large difference between the largest and smallest eigenvalues of Γ. In other words, even if we select Δ to be near the upper bound given in (11-1-25), the convergence rate of the recursive MSE algorithm is determined by the smallest eigenvalue λ_{min}. Consequently, the ratio $\lambda_{max}/\lambda_{min}$ ultimately determines the convergence rate. If $\lambda_{max}/\lambda_{min}$ is small, Δ can be selected so as to achieve rapid convergence. However, if the ratio $\lambda_{max}/\lambda_{min}$ is large, as is the case when the channel frequency response has deep spectral nulls, the convergence rate of the algorithm will be slow.

11-1-4 Excess MSE Due to Noisy Gradient Estimates

The recursive algorithm in (11-1-11) for adjusting the coefficients of the linear equalizer employs unbiased noisy estimates of the gradient vector. The noise in these estimates causes random fluctuations in the coefficients about their optimal values and, thus, leads to an increase in the MSE at the output of the equalizer. That is, the final MSE is $J_{min} + J_\Delta$, where J_Δ is the variance of the measurement noise. The term J_Δ due to the estimation noise has been termed *excess means-square error* by Widrow (1966).

The total MSE at the output of the equalizer for any set of coefficients $\mathbf{C}$ can be expressed as

$$J = J_{min} + (\mathbf{C} - \mathbf{C}_{opt})'^{*}\Gamma(\mathbf{C} - \mathbf{C}_{opt}) \tag{11-1-27}$$

where $\mathbf{C}_{opt}$ represents the optimum coefficients, which satisfy (11-1-6). This expression for the MSE can be simplified by performing the linear orthogonal transformation used above to establish convergence. The result of this transformation applied to (11-1-27) is

$$J = J_{min} + \sum_{k=-K}^{K} \lambda_k E\,|c_k^o - c_{k\,opt}^o|^2 \tag{11-1-28}$$

where the $\{c_k^o\}$ are the set of transformed equalizer coefficients. The excess MSE is the expected value of the second term in (11-1-28), i.e.,

$$J_\Delta = \sum_{k=-K}^{K} \lambda_k E\,|c_k^o - c_{k\,opt}^o|^2 \tag{11-1-29}$$

It has been shown by Widrow (1970, 1975) that the excess MSE is

$$J_\Delta = \Delta^2 J_{\min} \sum_{k=-K}^{K} \frac{\lambda_k^2}{1-(1-\Delta\lambda_k)^2} \tag{11-1-30}$$

The expression in (11-1-30) can be simplified when Δ is selected such that $\Delta\lambda_k \ll 1$ for all k. Then

$$\begin{aligned} J_\Delta &\approx \tfrac{1}{2}\Delta J_{\min} \sum_{k=-K}^{K} \lambda_k \\ &\approx \tfrac{1}{2}\Delta J_{\min} \operatorname{tr} \mathbf{\Gamma} \\ &\approx \tfrac{1}{2}\Delta(2K+1)J_{\min}(x_0+N_0) \end{aligned} \tag{11-1-31}$$

Note that $x_0 + N_0$ represents the received signal plus noise power.

It is desirable to have $J_\Delta < J_{\min}$. That is, Δ should be selected such that

$$\frac{J_\Delta}{J_{\min}} \approx \tfrac{1}{2}\Delta(2K+1)(x_0+N_0) < 1$$

or, equivalently,

$$\Delta < \frac{2}{(2K+1)(x_0+N_0)} \tag{11-1-32}$$

For example, if Δ is selected as

$$\Delta = \frac{0.2}{(2K+1)(x_0+N_0)} \tag{11-1-33}$$

the degradation in the output SNR of the equalizer due to the excess MSE is less than 1 dB.

The analysis given above on the excess mean square error is based on the assumption that the mean value of the equalizer coefficients has converged to the optimum value $\mathbf{C}_{\text{opt}}$. Under this condition, the step size Δ should satisfy the bound in (11-1-32). On the other hand, we have determined that convergence of the mean coefficient vector requires that $\Delta < 2/\lambda_{\max}$. While a choice of Δ near the upper bound $2/\lambda_{\max}$ may lead to initial convergence of the deterministic (known) steepest-descent gradient algorithm, such a large value of Δ will usually result in instability of the LMS stochastic gradient algorithm.

The initial convergence or transient behavior of the LMS algorithm has been investigated by several researchers. Their results clearly indicate that the step size must be reduced in direct proportion to the length of the equalizer as specified by (11-1-32). Hence, the upper bound given by (11-1-32) is also necessary to ensure the initial convergence of the LMS algorithm. The papers by Gitlin and Weinstein (1979) and Ungerboeck (1972) contain analyses of the transient behavior and the convergence properties of the LMS algorithm.

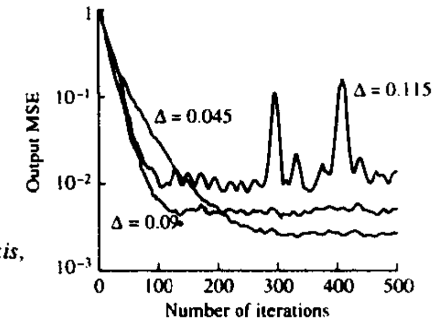


FIGURE 11-1-4 Initial convergence characteristics of the LMS algorithm with different step sizes. [*From* Digital Signal Processing, *by J. G. Proakis and D. G. Manolakis, 1988, Macmillan Publishing Company. Reprinted with permission of the publisher.*]

The following example serves to reinforce the important points made above regarding the initial convergence of the LMS algorithm.

Example 11-1-1

The LMS algorithm was used to adaptively equalize a communication channel for which the autocorrelation matrix $\mathbf{\Gamma}$ has an eigenvalue spread of $\lambda_{max}/\lambda_{min} = 11$. The number of taps selected for the equalizer was $2K + 1 = 11$. The input signal plus noise power $x_0 + N_0$ was normalized to unity. Hence, the upper bound on Δ given by (11-1-32) is 0.18. Figure 11-1-4 illustrates the initial convergence characteristics of the LMS algorithm for $\Delta = 0.045$, 0.09, and 0.115, by averaging the (estimated) MSE in 200 simulations. We observe that by selecting $\Delta = 0.09$ (one-half of the upper bound) we obtain relatively fast initial convergence. If we divide Δ by a factor of 2 to $\Delta = 0.045$, the convergence rate is reduced but the excess mean square error is also reduced, so that the LMS algorithm performs better in steady state (in a time-invariant signal environment). Finally, we note that a choice of $\Delta = 0.115$, which is still far below the upper bound, causes large undesirable fluctuations in the output MSE of the algorithm.

In a digital implementation of the LMS algorithm, the choice of the step-size parameter becomes even more critical. In an attempt to reduce the excess mean square error, it is possible to reduce the step-size parameter to the point where the total mean square error actually increases. This condition occurs when the estimated gradient components of the vector $\varepsilon_k \mathbf{V}_k^*$ after multiplication by the small step-size parameter Δ are smaller than one-half of the least significant bit in the fixed-point representation of the equalizer coefficients. In such a case, adaptation ceases. Consequently, it is important for the step size to be large enough to bring the equalizer coefficients in the vicinity of $\mathbf{C}_{opt}$. If it is desired to decrease the step size significantly, it is necessary to increase the precision in the equalizer coefficients. Typically, 16 bits of precision may be used for the coefficients, with about 10–12 of the most significant bits used for arithmetic operations in the equalization of the data. The remaining least significant bits are required to provide the necessary precision for the adaptation process. Thus, the scaled, estimated gradient components $\Delta \varepsilon \mathbf{V}_k^*$ usually affect only the least-significant bits in any one iteration. In effect, the added precision also allows for the noise to be averaged out, since many incremental changes in the least-significant bits are required before any change occurs in the upper more significant bits used in arithmetic operations for equalizing the data. For an analysis of roundoff errors in a digital implementation of the LMS algorithm, the reader is referred to the papers by Gitlin and Weinstein (1979), Gitlin *et al.* (1982), and Caraiscos and Liu (1984).

As a final point, we should indicate that the LMS algorithm is appropriate for tracking slowly time-invariant signal statistics. In such a case, the minimum MSE and the optimum coefficient vector will be time-variant. In other words, $J_{\min}(n)$ is a function of time and the $(2K+1)$-dimensional error surface is moving with the time index n. The LMS algorithm attempts to follow the moving minimum $J_{\min}(n)$ in the $(2K+1)$-dimensional space, but it is always lagging behind due to its use of (estimated) gradient vectors. As a consequence, the LMS algorithm incurs another form of error, called the *lag error*, whose mean square value decreases with an increase in the step size Δ. The total MSE error can now be expressed as

$$J_{\text{total}} = J_{\min}(n) + J_\Delta + J_l$$

where J_l denotes the mean square error due to the lag.

In any given nonstationary adaptive equalization problem, if we plot the errors J_Δ and J_l as a function of Δ, we expect these errors to behave as illustrated in Fig. 11-1-5. We observe that J_Δ increases with an increase in Δ while J_l decreases with an increase in Δ. The total error will exhibit a minimum, which will determine the optimum choice of the step-size parameter.

When the statistical time variations of the signal occur rapidly, the lag error

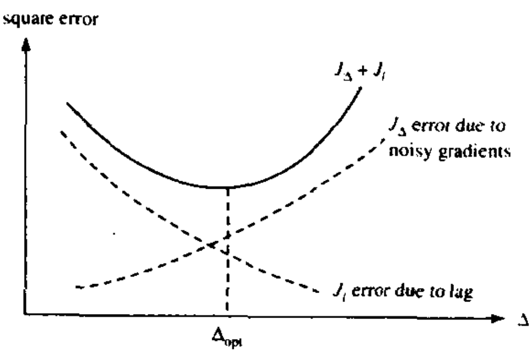


FIGURE 11-1-5 Excess mean square error J_Δ and lag error J_l as a function of the step size. [*From* Digital Signal Processing, *by J. G. Proakis and D. G. Manolakis, 1988, Macmillan Publishing Company. Reprinted with permission of the publisher.*]

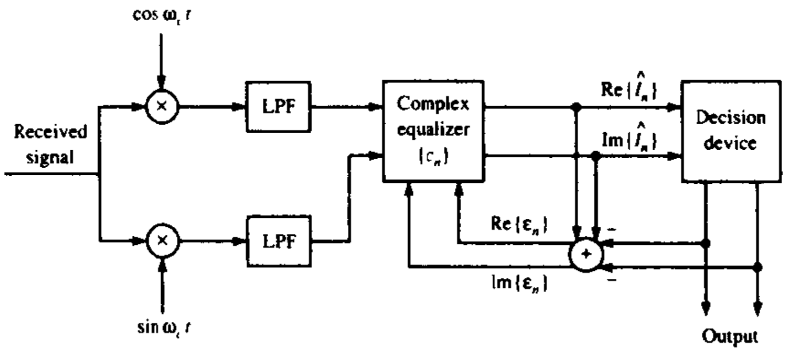


FIGURE 11-1-6 QAM signal demodulation.

will dominate the performance of the adaptive equalizer. In such a case, $J_l \gg J_{\min} + J_\Delta$, even when the largest possible value of Δ is used. When this condition occurs, the LMS algorithm is inappropriate for the application and one must rely on the more complex recursive least-squares algorithms described in Section 11-4 to obtain faster convergence and tracking.

11-1-5 Baseband and Passband Linear Equalizers

Our treatment of adaptive linear equalizers has been in terms of equivalent lowpass signals. However, in a practical implementation, the linear adaptive equalizer shown in Fig. 11-1-2 can be realized either at baseband or at bandpass. For example Fig. 11-1-6 illustrates the demodulation of QAM (or multiphase PSK) by first translating the signal to baseband and equalizing the baseband signal with an equalizer having complex-valued coefficients. In effect, the complex equalizer with complex-valued (in-phase and quadrature components) input is equivalent to four parallel equalizers with real-valued tap coefficients as shown in Fig. 11-1-7.

As an alternative, we may equalize the signal at passband. This is

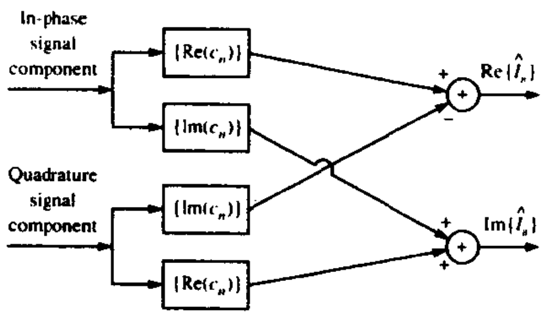


FIGURE 11-1-7 Complex-valued baseband equalizer for QAM signals.

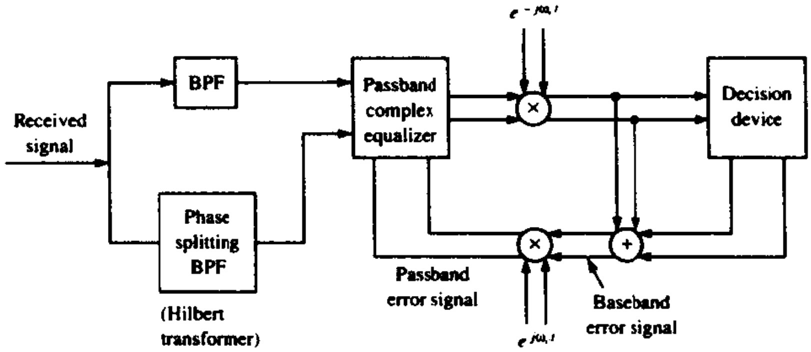


FIGURE 11-1-8 QAM or PSK signal equalization at passband.

accomplished as shown in Fig. 11-1-8 for a two-dimensional signal constellation such as QAM and PSK. The received signal is filtered and, in parallel, it is passed through a Hilbert transformer, called a *phase-splitting filter*. Thus, we have the equivalent of in-phase and quadrature components at passband, which are fed to a passband complex equalizer. Following the equalization, the signal is down-converted to a baseband and detected. The error signal generated for the purpose of adjusting the equalizer coefficients is formed at baseband and frequency-translated to passband as illustrated in Fig. 11-1-8.

11-2 ADAPTIVE DECISION-FEEDBACK EQUALIZER

As in the case of the linear adaptive equalizer, the coefficients of the feedforward filter and the feedback filter in a decision-feedback equalizer may be adjusted recursively, instead of inverting a matrix as implied by (10-3-3). Based on the minimization of the MSE at the output of the DFE, the steepest-descent algorithm takes the form

$$\mathbf{C}_{k+1} = \mathbf{C}_k + \Delta E(\varepsilon_k \mathbf{V}_k^*) \tag{11-2-1}$$

where $\mathbf{C}_k$ is the vector of equalizer coefficients in the kth signal interval, $E(\varepsilon_k \mathbf{V}_k^*)$ is the cross-correlation of the error signal $\varepsilon_k = I_k - \hat{I}_k$ with $\mathbf{V}_k$ and $\mathbf{V}_k = [v_{k+K_1} \; \dots \; v_k \; I_{k-1} \; \dots \; I_{k-K_2}]'$, representing the signal values in the feedforward and feedback filters at time $t = kT$. The MSE is minimized when the cross-correlation vector $E(\varepsilon_k \mathbf{V}_k^*) = \mathbf{0}$ as $k \to \infty$.

Since the exact cross-correlation vector is unknown at any time instant, we use as an estimate the vector $\varepsilon_k \mathbf{V}_k^*$ and average out the noise in the estimate through the recursive equation

$$\hat{\mathbf{C}}_{k+1} = \hat{\mathbf{C}}_k + \Delta \varepsilon_k \mathbf{V}_k^* \tag{11-2-2}$$

This is the LMS algorithm for the DFE.

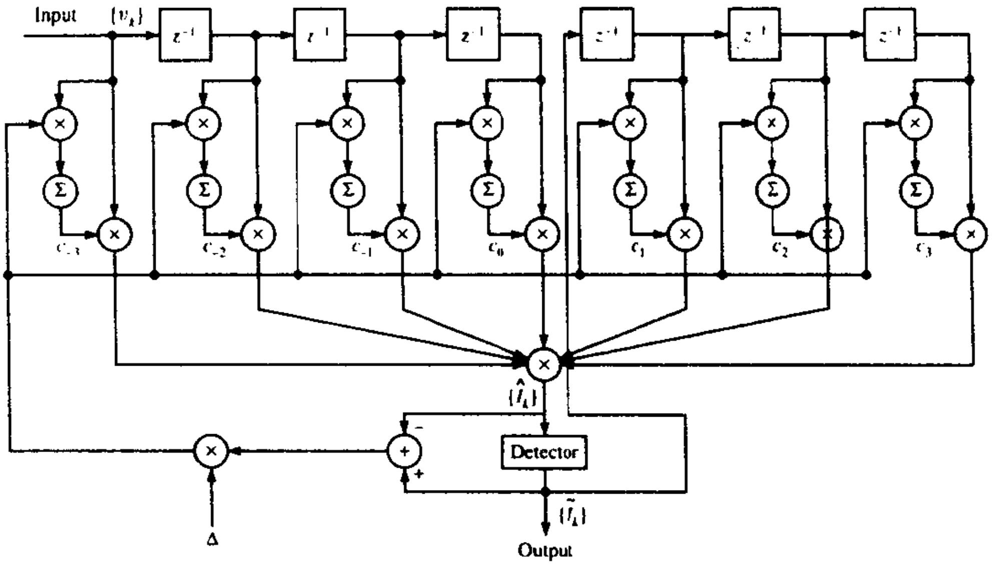


FIGURE 11-2-1 Decision-feedback equalizer.

As in the case of a linear equalizer, we may use a training sequence to adjust the coefficients of the DFE initially. Upon convergence to the (near-) optimum coefficients (minimum MSE), we may switch to a decision-directed mode where the decisions at the output of the detector are used in forming the error signal ε_k and fed to the feedback filter. This is the adaptive mode of the DFE, which is illustrated in Fig 11-2-1. In this case, the recursive equation for adjusting the equalizer coefficient is

$$\tilde{\mathbf{C}}_{k+1} = \tilde{\mathbf{C}}_k + \Delta \tilde{\varepsilon}_k \mathbf{V}_k^* \tag{11-2-3}$$

where $\tilde{\varepsilon}_k = \tilde{I}_k - \hat{I}_k$ and $\mathbf{V}_k = [v_{k+K_1} \ \ldots \ v_k \ \tilde{I}_{k-1} \ \ldots \ \tilde{I}_{k-K_2}]'$.

The performance characteristics of the LMS algorithm for the DFE are basically the same as the development given in Sections 11-1-3 and 11-1-4 for the linear adaptive equalizer.

11-2-1 Adaptive Equalization of Trellis-Coded Signals

Bandwidth efficient trellis-coded modulation that was described in Section 8-3 is frequently used in digital communications over telephone channels to reduce the required SNR per bit for achieving a specified error rate. Channel distortion of the trellis-coded signal forces us to use adaptive equalization in order to reduce the intersymbol interference. The output of the equalizer is then fed to the Viterbi decoder, which performs soft-decision decoding of the trellis-coded signal.

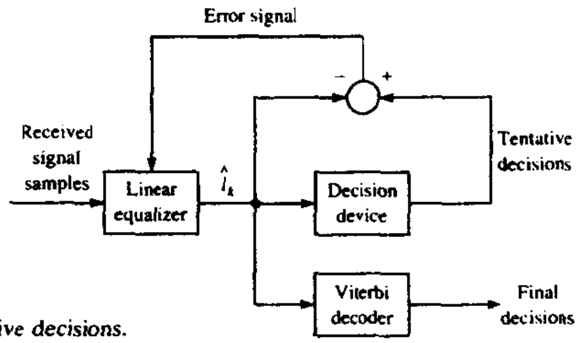
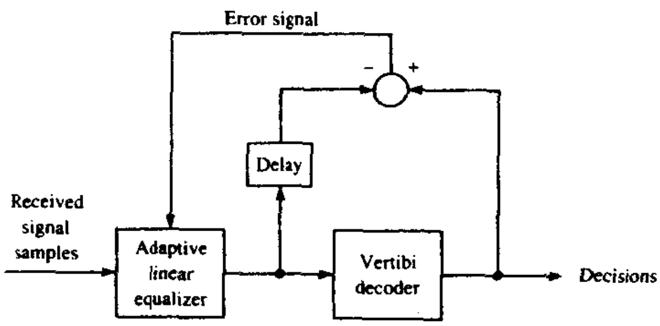


FIGURE 11-2-2 *Adjustment of equalizer based on tentative decisions.*

The question that arises regarding such a receiver is how do we adapt the equalizer in a data transmission mode? One possibility is to have the equalizer make its own decisions at its output solely for the purpose of generating an error signal for adjusting its tap coefficients, as shown in the block diagram in Fig. 11-2-2. The problem with this approach is that such decisions are generally unreliable, since the pre-decoding coded symbol SNR is relatively low. A high error rate would cause a significant degradation in the operation of the equalizer, which would ultimately affect the reliability of the decisions at the output of the decoder. The more desirable alternative is to use the post-decoding decisions from the Viterbi decoder, which are much more reliable, to continuously adapt the equalizer. This approach is certainly preferable and viable when a linear equalizer is used prior to the Viterbi decoder. The decoding delay inherent in the Viterbi decoder can be overcome by introducing an identical delay in the tap weight adjustment of the equalizer coefficients as shown in Fig. 11-2-3. The major price that must be paid for the added delay is that the step-size parameter in the LMS algorithm must be reduced, as described by Long *et al.* (1987, 1989), in order to achieve stability in the algorithm.

In channels with one or more in-band spectral nulls, the linear equalizer is

FIGURE 11-2-3 Adjustment of equalizer based on decisions from the Viterbi decoder.



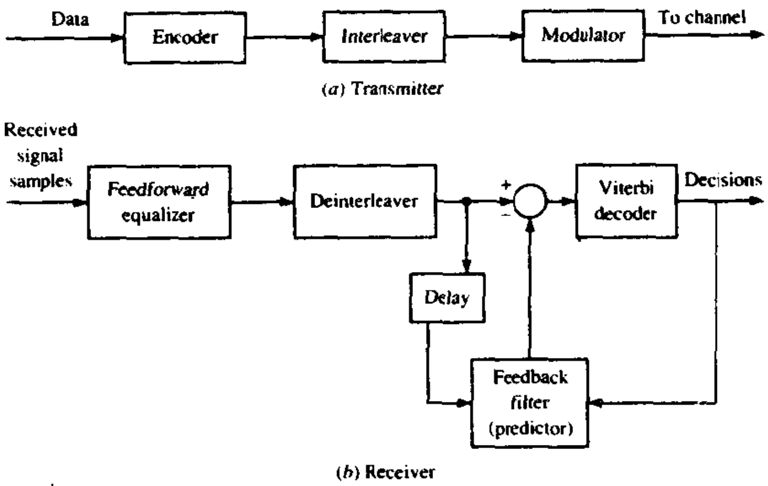


FIGURE 11-2-4 Use of predictive DFE with interleaving and trellis-coded modulation.

no longer adequate for compensating the channel intersymbol interference. Instead, we should like to use a DFE. But the DFE requires reliable decisions in its feedback filter in order to cancel out the intersymbol interference from previously detected symbols. Tentative decisions prior to decoding would be highly unreliable and, hence, inappropriate. Unfortunately, the conventional DFE cannot be cascaded with the Viterbi algorithm in which post-decoding decisions from the decoder are fed back to the DFE.

One alternative is to use the predictive DFE described in Section 10-3-3. In order to accommodate for the decoding delay as it affects the linear predictor, we introduce a periodic interleaver/deinterleaver pair that has the same delay as the Viterbi decoder and, thus, makes it possible to generate the appropriate error signal to the predictor as illustrated in the block diagram of Fig. 11-2-4. The novel way in which a predictive DFE can be combined with Viterbi decoding to equalize trellis-coded signals is described and analyzed by Eyuboglu (1988). This same idea has been carried over to the equalization of fading multipath channels by Zhou *et al.* (1988, 1990), but the structure of the DFE was modified to use recursive least-squares lattice-type filters, which provide faster adaptation to the time variations encountered in the channel.

11-3 AN ADAPTIVE CHANNEL ESTIMATOR FOR ML SEQUENCE DETECTION

The ML sequence detection criterion implemented via the Viterbi algorithm as embodied in the metric computation given by (10-1-23) and the probabilistic symbol-by-symbol detection algorithm described in Section 5-1-5 require knowledge of the equivalent discrete-time channel coefficients $\{f_k\}$. To accommodate a channel that is unknown or slowly time-varying, one may include a

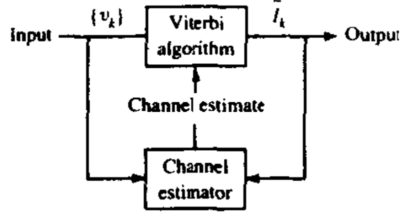


FIGURE 11-3-1 Block diagram of method for estimating the channel characteristics for the Viterbi algorithm.

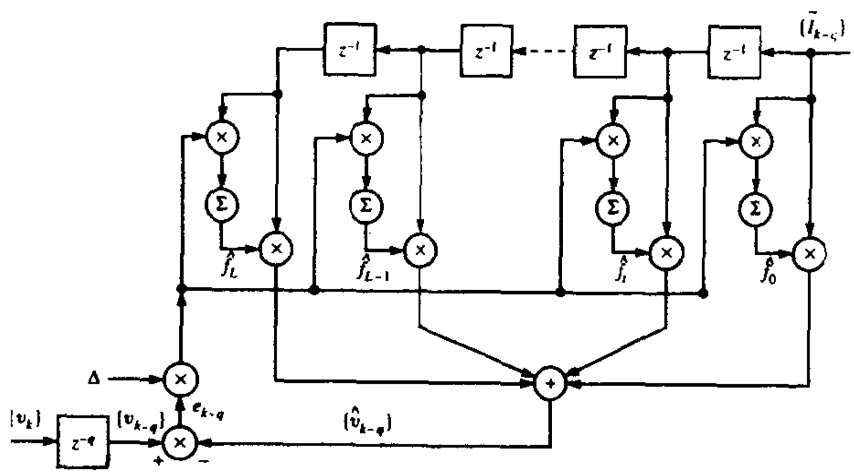
channel estimator connected in parallel with the detection algorithm, as shown in Fig. 11-3-1. The channel estimator, which is shown in Fig. 11-3-2 is identical in structure to the linear transversal equalizer discussed previously in Section 11-1. In fact, the channel estimator is a replica of the equivalent discrete-time channel filter that models the intersymbol interference. The estimated tap coefficients, denoted by $\{\hat{f}_k\}$, are adjusted recursively to minimize the MSE between the actual received sequence and the output of the estimator. For example, the steepest-descent algorithm in a decision-directed mode of operation is

$$\hat{\mathbf{f}}_{k+1} = \hat{\mathbf{f}}_k + \Delta \varepsilon_k \tilde{\mathbf{I}}_k^* \tag{11-3-1}$$

where $\hat{\mathbf{f}}_k$ is the vector of tap gain coefficients at the kth iteration, Δ is the step size, $\varepsilon_k = v_k - \hat{v}_k$ is the error signal, and $\tilde{\mathbf{I}}_k$ denotes the vector of detected information symbols in the channel estimator at the kth iteration.

We now show that when the MSE between v_k and $\hat{v}_k$ is minimized, the resulting values of the tap gain coefficients of the channel estimator are the values of the discrete-time channel model. For mathematical tractability, we assume that the detected information sequence $\{\tilde{I}_k\}$ is correct, i.e., $\{\tilde{I}_k\}$ is

FIGURE 11-3-2 Adaptive transversal filter for estimating the channel dispersion.



identical to the transmitted sequence $\{I_k\}$. This is a reasonable assumption when the system is operating at a low probability of error. Thus, the MSE between the received signal v_k and the estimate $\hat{v}_k$ is

$$J(\hat{\mathbf{f}}) = E\left(\left|v_k - \sum_{j=0}^{N-1} \hat{f}_j I_{k-j}\right|^2\right) \tag{11-3-2}$$

The tap coefficients $\{\hat{f}_k\}$ that minimize $J(\hat{\mathbf{f}})$ in (11-3-2) satisfy the set of N linear equations

$$\sum_{j=0}^{N-1} \hat{f}_j \phi_{kj} = d_k, \qquad k = 0, 1, \ldots, N-1 \tag{11-3-3}$$

where

$$\phi_{kj} = E(I_k I_j^*), \qquad d_k = \sum_{j=0}^{N-1} f_j \phi_j \tag{11-3-4}$$

From (11-3-3) and (11-3-4), we conclude that, as long as the information sequence $\{I_k\}$ is uncorrelated, the optimum coefficients are exactly equal to the respective values of the equivalent discrete-time channel. It is also apparent that when the number of taps N in the channel estimator is greater than or equal to $L+1$, the optimum tap gain coefficients $\{\hat{f}_k\}$ are equal to the respective values of the $\{f_k\}$, even when the information sequence is correlated. Subject to the above conditions, the minimum MSE is simply equal to the noise variance N_0.

In the above discussion, the estimated information sequence at the output of the Viterbi algorithm or the probabilistic symbol-by-symbol algorithm was used in making adjustments of the channel estimator. For startup operation, one may send a short training sequence to perform the initial adjustment of the tap coefficients, as is usually done in the case of the linear transversal equalizer. In an adaptive mode of operation, the receiver simply uses its own decisions to form an error signal.

11-4 RECURSIVE LEAST-SQUARES ALGORITHMS FOR ADAPTIVE EQUALIZATION

The LMS algorithm that we described in Sections 11-1 and 11-2 for adaptively adjusting the tap coefficients of a linear equalizer or a DFE is basically a (stochastic) steepest-descent algorithm in which the true gradient vector is approximated by an estimate obtained directly from the data.

The major advantage of the steepest-descent algorithm lies in its computational simplicity. However, the price paid for the simplicity is slow convergence, especially when the channel characteristics result in an autocorrelation matrix Γ whose eigenvalues have a large spread, i.e., $\lambda_{max}/\lambda_{min} \gg 1$. Viewed in another way, the gradient algorithm has only a single adjustable parameter for controlling the convergence rate, namely, the parameter Δ. Consequently the slow convergence is due to this fundamental limitation.

In order to obtain faster convergence, it is necessary to devise more complex algorithms involving additional parameters. In particular, if the matrix $\mathbf{\Gamma}$ is $N \times N$ and has eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_N$, we may use an algorithm that contains N parameters—one for each of the eigenvalues. The optimum selection of these parameters to achieve rapid convergence is a topic of this section.

In deriving faster converging algorithms, we shall adopt a least-squares approach. Thus, we shall deal directly with the received data in minimizing the quadratic performance index, whereas previously we minimized the expected value of the squared error. Put simply, this means that the performance index is expressed in terms of a time average instead of a statistical average.

It is convenient to express the recursive least-squares algorithms in matrix form. Hence, we shall define a number of vectors and matrices that are needed in this development. In so doing, we shall change the notation slightly. Specifically, the estimate of the information symbol at time t, where t is an integer, from a linear equalizer is now expressed as

$$\tilde{I}(t) = \sum_{j=-K}^{K} c_j(t-1)v_{t-j}$$

By changing the index j on $c_j(t-1)$ to run from $j=0$ to $j=N-1$ and simultaneously defining

$$y(t) = v_{t+K}$$

the estimate $\tilde{I}(t)$ becomes

$$\begin{aligned}\tilde{I}(t) &= \sum_{j=0}^{N-1} c_j(t-1)y(t-j) \\ &= \mathbf{C}'_N(t-1)\mathbf{Y}_N(t) \qquad (11\text{-}4\text{-}1)\end{aligned}$$

where $\mathbf{C}_N(t-1)$ and $\mathbf{Y}_N(t)$ are, respectively, the column vectors of the equalizer coefficients $c_j(t-1)$, $j=0, 1, \ldots, N-1$, and the input signals $y(t-j)$, $j=0, 1, 2, \ldots, N-1$.

Similarly, in the decision-feedback equalizer, we have tap coefficients $c_j(t)$, $j=0, 1, \ldots, N-1$, where the first K_1+1 are the coefficients of the feedforward filter and the remaining $K_2 = N - K_1 - 1$ are the coefficients of the feedback filter. The data in the estimate $\hat{I}(t)$ is $v_{t+K_1}, \ldots, v_{t+1}, \tilde{I}_{t-1}, \ldots, \tilde{I}_{t-K_2}$, where $\tilde{I}_{t-j}$, $1 \le j \le K_2$, denote the decisions on previously detected symbols. In this development, we neglect the effect of decision errors in the algorithms. Hence, we assume that $\tilde{I}_{t-j} = I_{t-j}$, $1 \le j \le K_2$. For notational convenience, we also define

$$y(t-j) = \begin{cases} v_{t+K_1-j} & (0 \le j \le K_1) \\ I_{t+K_1-j} & (K_1 < j \le N-1) \end{cases} \qquad (11\text{-}4\text{-}2)$$

Thus,

$$\mathbf{Y}_N(t) = [y(t) \quad y(t-1) \quad \dots \quad y(t-N+1)]'$$
$$= [v_{t+K_1} \quad \dots \quad v_{t+1} \quad v_t \quad I_{t-1} \quad \dots \quad I_{t-K_2}]' \tag{11-4-3}$$

11-4-1 Recursive Least-Squares (Kalman) Algorithm

The recursive least-squares (RLS) estimation of $\hat{I}(t)$ may be formulated as follows. Suppose we have observed the vectors $\mathbf{Y}_N(n)$, $n = 0, 1, \dots, t$, and we wish to determine the coefficient vector $\mathbf{C}_N(t)$ of the equalizer (linear or decision-feedback) that minimizes the time-average weighted squared error

$$\mathscr{E}_N^{LS} = \sum_{n=0}^{t} w^{t-n} |e_N(n, t)|^2 \tag{11-4-4}$$

where the error is defined as

$$e_N(n, t) = I(n) - \mathbf{C}'_N(t)\mathbf{Y}_N(n) \tag{11-4-5}$$

and w represents a weighting factor $0 < w < 1$. Thus we introduce exponential weighting into past data, which is appropriate when the channel characteristics are time-variant. Minimization of $\mathscr{E}_N^{LS}$ with respect to the coefficient vector $\mathbf{C}_N(t)$ yields the set of linear equations

$$\mathbf{R}_N(t)\mathbf{C}_N(t) = \mathbf{D}_N(t) \tag{11-4-6}$$

where $\mathbf{R}_N(t)$ is the signal correlation matrix defined as

$$\mathbf{R}_N(t) = \sum_{n=0}^{t} w^{t-n}\mathbf{Y}_N^*(n)\mathbf{Y}'_N(n) \tag{11-4-7}$$

and $\mathbf{D}_N(t)$ is the cross-correlation vector

$$\mathbf{D}_N(t) = \sum_{n=0}^{t} w^{t-n} I(n)\mathbf{Y}_N^*(n) \tag{11-4-8}$$

The solution of (11-4-6) is

$$\mathbf{C}_N(t) = \mathbf{R}_N^{-1}(t)\mathbf{D}_N(t) \tag{11-4-9}$$

The matrix $\mathbf{R}_N(t)$ is akin to the statistical autocorrelation matrix $\mathbf{\Gamma}_N$, while the vector $\mathbf{D}_N(t)$ is akin to the cross-correlation vector $\boldsymbol{\xi}_N$, defined previously. We emphasize, however, that $\mathbf{R}_N(t)$ is not a Toeplitz matrix. We also should mention that, for small values of t, $\mathbf{R}_N(t)$ may be ill conditioned; hence, it is customary to initially add the matrix $\delta\mathbf{I}_N$ to $\mathbf{R}_N(t)$, where δ is a small positive constant and $\mathbf{I}_N$ is the identity matrix. With exponential weighting into the past, the effect of adding $\delta \mathbf{I}_N$ dissipates with time.

Now suppose we have the solution (11-4-9) for time $t-1$, i.e., $\mathbf{C}_N(t-1)$, and we wish to compute $\mathbf{C}_N(t)$. It is inefficient and, hence, impractical to solve the set of N linear equations for each new signal component that is received. To avoid this, we proceed as follows. First, $\mathbf{R}_N(t)$ may be computed recursively as

$$\mathbf{R}_N(t) = w\mathbf{R}_N(t-1) + \mathbf{Y}_N^*(t)\mathbf{Y}_N'(t) \tag{11-4-10}$$

We call (11-4-10) the *time-update equation* for $\mathbf{R}_N(t)$.

Since the inverse of $\mathbf{R}_N(t)$ is needed in (11-4-9), we use the matrix-inverse identity

$$\mathbf{R}_N^{-1}(t) = \frac{1}{w}\left[\mathbf{R}_N^{-1}(t-1) - \frac{\mathbf{R}_N^{-1}(t-1)\mathbf{Y}_N^*(t)\mathbf{Y}_N'(t)\mathbf{R}_N^{-1}(t-1)}{w + \mathbf{Y}_N'(t)\mathbf{R}_N^{-1}(t-1)\mathbf{Y}_N^*(t)}\right] \tag{11-4-11}$$

Thus $\mathbf{R}_N^{-1}(t)$ may be computed recursively according to (11-4-11).

For convenience, we define $\mathbf{P}_N(t) = \mathbf{R}_N^{-1}(t)$. It is also convenient to define an N-dimensional vector, called the *Kalman gain vector*, as

$$\mathbf{K}_N(t) = \frac{1}{w + \mu_N(t)}\mathbf{P}_N(t-1)\mathbf{Y}_N^*(t) \tag{11-4-12}$$

where $\mu_N(t)$ is a scalar defined as

$$\mu_N(t) = \mathbf{Y}_N'(t)\mathbf{P}_N(t-1)\mathbf{Y}_N^*(t) \tag{11-4-13}$$

With these definitions, (11-4-11) becomes

$$\mathbf{P}_N(t) = \frac{1}{w}[\mathbf{P}_N(t-1) - \mathbf{K}_N(t)\mathbf{Y}_N'(t)\mathbf{P}_N(t-1)] \tag{11-4-14}$$

Suppose we postmultiply both sides of (11-4-14) by $\mathbf{Y}_N^*(t)$. Then

$$\begin{aligned}\mathbf{P}_N(t)\mathbf{Y}_N^*(t) &= \frac{1}{w}[\mathbf{P}_N(t-1)\mathbf{Y}_N^*(t) - \mathbf{K}_N(t)\mathbf{Y}_N'(t)\mathbf{P}_N(t-1)\mathbf{Y}_N^*(t)] \\ &= \frac{1}{w}\{[w + \mu_N(t)]\mathbf{K}_N(t) - \mathbf{K}_N(t)\mu_N(t)\} \\ &= \mathbf{K}_N(t)\end{aligned} \tag{11-4-15}$$

Therefore, the Kalman gain vector may also be defined as $\mathbf{P}_N(t)\mathbf{Y}_N(t)$.

Now we use the matrix inversion identity to derive an equation for obtaining $\mathbf{C}_N(t)$ from $\mathbf{C}_N(t-1)$. Since

$$\mathbf{C}_N(t) = \mathbf{P}_N(t)\mathbf{D}_N(t)$$

and

$$\mathbf{D}_N(t) = w\mathbf{D}_N(t-1) + I(t)\mathbf{Y}_N^*(t) \tag{11-4-16}$$

we have

$$\begin{aligned}\mathbf{C}_N(t) &= \frac{1}{w}[\mathbf{P}_N(t-1) - \mathbf{K}_N(t)\mathbf{Y}'_N(t)\mathbf{P}_N(t-1)][w\mathbf{D}_N(t-1) + I(t)\mathbf{Y}^*_N(t)] \\ &= \mathbf{P}_N(t-1)\mathbf{D}_N(t-1) + \frac{1}{w}I(t)\mathbf{P}_N(t-1)\mathbf{Y}^*_N(t) \\ &\quad - \mathbf{K}_N(t)\mathbf{Y}'_N(t)\mathbf{P}_N(t-1)\mathbf{D}_N(t-1) \\ &\quad - \frac{1}{w}I(t)\mathbf{K}_N(t)\mathbf{Y}'_N(t)\mathbf{P}_N(t-1)\mathbf{Y}^*_N(t) \\ &= \mathbf{C}_N(t-1) + \mathbf{K}_N(t)[I(t) - \mathbf{Y}'_N(t)\mathbf{C}_N(t-1)] \end{aligned} \tag{11-4-17}$$

Note that $\mathbf{Y}'_N(t)\mathbf{C}_N(t-1)$ is the output of the equalizer at time t, i.e.,

$$\hat{I}(t) = \mathbf{Y}'_N(t)\mathbf{C}_N(t-1) \tag{11-4-18}$$

and

$$e_N(t, t-1) = I(t) - \hat{I}(t) \equiv e_N(t) \tag{11-4-19}$$

is the error between the desired symbol and the estimate. Hence, $\mathbf{C}_N(t)$ is updated recursively according to the relation

$$\mathbf{C}_N(t) = \mathbf{C}_N(t-1) + \mathbf{K}_N(t)e_N(t) \tag{11-4-20}$$

The residual MSE resulting from this optimization is

$$\mathscr{E}^{LS}_{N\,\min} = \sum_{n=0}^{t} w^{t-n}|I(n)|^2 - \mathbf{C}'_N(t)\mathbf{D}^*_N(t) \tag{11-4-21}$$

To summarize, suppose we have $\mathbf{C}_N(t-1)$ and $\mathbf{P}_N(t-1)$. When a new signal component is received, we have $\mathbf{Y}_N(t)$. Then the recursive computation for the time update of $\mathbf{C}_N(t)$ and $\mathbf{P}_N(t)$ proceeds as follows:

- compute output:

$$\hat{I}(t) = \mathbf{Y}'_N(t)\mathbf{C}_N(t-1)$$

- compute error:

$$e_N(t) = I(t) - \hat{I}(t)$$

- compute Kalman gain vector:

$$\mathbf{K}_N(t) = \frac{\mathbf{P}_N(t-1)\mathbf{Y}'_N(t)}{w + \mathbf{Y}'_N(t)\mathbf{P}_N(t-1)\mathbf{Y}^*_N(t)}$$

- update inverse of the correlation matrix:

$$\mathbf{P}_N(t) = \frac{1}{w}[\mathbf{P}_N(t-1) - \mathbf{K}_N(t)\mathbf{Y}'_N(t)\mathbf{P}_N(t-1)]$$

- update coefficients:

$$\begin{aligned}\mathbf{C}_N(t) &= \mathbf{C}_N(t-1) + \mathbf{K}_N(t)e_N(t) \\ &= \mathbf{C}_N(t-1) + \mathbf{P}_N(t)\mathbf{Y}^*(t)e_N(t)\end{aligned} \tag{11-4-22}$$

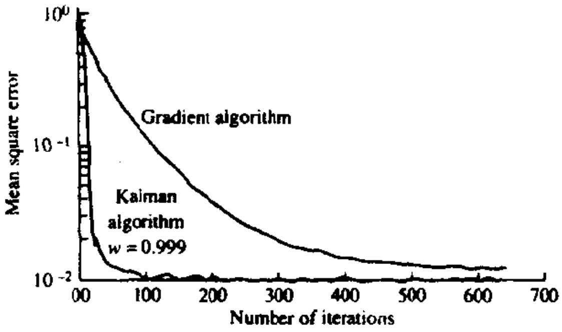


FIGURE 11-4-1 Comparison of convergence rate for the Kalman and gradient algorithms.

The algorithm described by (11-4-22) is called the *RLS direct form* or *Kalman algorithm*. It is appropriate when the equalizer has a transversal (direct-form) structure.

Note that the equalizer coefficients change with time by an amount equal to the error $e_N(t)$ multiplied by the Kalman gain vector $\mathbf{K}_N(t)$. Since $\mathbf{K}_N(t)$ is N-dimensional, each tap coefficient in effect is controlled by one of the elements of $\mathbf{K}_N(t)$. Consequently rapid convergence is obtained. In contrast, the steepest-descent algorithm, expressed in our present notation, is

$$\mathbf{C}_N(t) = \mathbf{C}_N(t-1) + \Delta \mathbf{Y}_N^*(t) e_N(t) \tag{11-4-23}$$

and the only variable parameter is the step size Δ.

Figure 11-4-1 illustrates the initial convergence rate of these two algorithms for a channel with fixed parameters $f_0 = 0.26$, $f_1 = 0.93$, $f_2 = 0.26$, and a linear equalizer with 11 taps. The eigenvalue ratio for this channel is $\lambda_{max}/\lambda_{min} = 11$. All the equalizer coefficients were initialized to zero. The steepest-descent algorithm was implemented with $\Delta = 0.020$. The superiority of the Kalman algorithm is clearly evident. This is especially important in tracking a time-variant channel. For example, the time variations in the characteristics of an (ionospheric) high-frequency (HF) radio channel are too rapid to be equalized by the gradient algorithm, but the Kalman algorithm adapts sufficiently rapidly to track such variations.

In spite of its superior tracking performance, the Kalman algorithm described above have two disadvantages. One is its complexity. The second is its sensitivity to roundoff noise that accumulates due to the recursive computations. The latter may cause instabilities in the algorithm.

The number of computations or operations (multiplications, divisions, and subtractions) in computing the variables in (11-4-22) is proportional to N^2. Most of these operations are involved in the updating of $\mathbf{P}_N(t)$. This part of the computation is also susceptible to roundoff noise. To remedy that problem, algorithms have been developed that avoid the computation of $\mathbf{P}_N(t)$ according to (11-4-14). The basis of these algorithms lies in the decomposition of $\mathbf{P}_N(t)$ in the form

$$\mathbf{P}_N(t) = \mathbf{S}_N(t) \mathbf{\Lambda}_N(t) \mathbf{S}_N'(t) \tag{11-4-24}$$

where $\mathbf{S}_N(t)$ is a lower-triangular matrix whose diagonal elements are unity, and $\Lambda_N(t)$ is a diagonal matrix. Such a decomposition is called a *square-root factorization* (see Bierman, 1977). This factorization is described in Appendix D. In a square-root algorithm, $\mathbf{P}_N(t)$ is not updated as in (11-4-14) nor is it computed. Instead, the time updating is performed on $\mathbf{S}_N(t)$ and $\Lambda_N(t)$.

Square-root algorithms are frequently used in control systems applications in which Kalman filtering is involved. In digital communications, the square-root Kalman algorithm has been implemented in a decision-feedback-equalized PSK modem designed to transmit at high speed over HF radio channels with a nominal 3 kHz bandwidth. This algorithm is described in the paper by Hsu (1982). It has a computational complexity of $1.5N^2 + 6.5N$ (complex-valued multiplications and divisions per output symbol). It is also numerically stable and exhibits good numerical properties. For a detailed discussion of square-root algorithms in sequential estimation, the reader is referred to the book by Bierman (1977).

It is also possible to derive RLS algorithms with computational complexities that grow linearly with the number N of equalizer coefficients. Such algorithms are generally called *fast RLS algorithms* and have been described in the papers by Carayannis *et al.* (1983), Cioffi and Kailath (1984), and Slock and Kailath (1988).

11-4-2 Linear Prediction and the Lattice Filter

In Chapter 3, we considered the linear prediction of a signal, in the context of speech encoding. In this section, we shall establish the connection between linear prediction and a lattice filter.

The linear prediction problem may be stated as follows: given a set of data $y(t-1), y(t-2), \ldots, y(t-p)$, predict the value of the next data point $y(t)$. The predictor of order p is

$$\hat{y}(t) = \sum_{k=1}^{p} a_{pk}\, y(t-k) \tag{11-4-25}$$

Minimization of the MSE, defined as

$$\begin{aligned} \mathscr{E}_p &= E[y(t) - \hat{y}(t)]^2 \\ &= E\left[y(t) - \sum_{k=1}^{p} a_{pk}\, y(t-k)\right]^2 \end{aligned} \tag{11-4-26}$$

with respect to the predictor coefficients $\{a_{pk}\}$ yields the set of linear equations

$$\sum_{k=1}^{p} a_{pk}\,\phi(k-l) = \phi(l), \qquad l = 1, 2, \ldots, p \tag{11-4-27}$$

where

$$\phi(l) = E[y(t)y(t+l)]$$

These are called the *normal equations* or the *Yule–Walker equations*.

The matrix $\mathbf{\Phi}$ with elements $\phi(k-l)$ is a Toeplitz matrix, and, hence, the Levinson–Durbin algorithm described in Appendix A provides an efficient means for solving the linear equations recursively, starting with a first-order predictor and proceeding recursively to the solution of the coefficients for the predictor of order p. The recursive relations for the Levinson–Durbin algorithm are

$$\begin{aligned} a_{11} &= \frac{\phi(1)}{\phi(0)}, \qquad \mathscr{E}_0 = \phi(0) \\ a_{mm} &= \frac{\phi(m) - \mathbf{A}'_m \mathbf{\Phi}^r_{m-1}}{\mathscr{E}_{m-1}} \\ a_{mk} &= a_{m-1\,k} - a_{mm} a_{m-1\,m-k} \\ \mathscr{E}_m &= \mathscr{E}_{m-1}(1 - a_{mm}^2) \end{aligned} \tag{11-4-28}$$

for $m = 1, 2, \ldots, p$, where the vectors $\mathbf{A}_{m-1}$ and $\mathbf{\Phi}^r_{m-1}$ are defined as

$$\begin{aligned} \mathbf{A}_{m-1} &= [a_{m-1\,1} \quad a_{m-1\,2} \quad \cdots \quad a_{m-1\,m-1}]' \\ \mathbf{\Phi}^r_{m-1} &= [\phi(m-1) \quad \phi(m-2) \quad \ldots \quad \phi(1)]' \end{aligned}$$

The linear prediction filter of order m may be realized as a transversal filter with transfer function

$$A_m(z) = 1 - \sum_{k=1}^{m} a_m z^{-k} \tag{11-4-29}$$

Its input is the data $\{y(t)\}$ and its output is the error $e(t) = y(t) - \hat{y}(t)$. The prediction filter can also be realized in the form of a lattice, as we now demonstrate.

Our starting point is the use of the Levinson–Durbin algorithm for the predictor coefficients a_{mk} in (11-4-29). This substitution yields

$$\begin{aligned} A_m(z) &= 1 - \sum_{k=1}^{m-1} (a_{m-1\,k} - a_{mm} a_{m-1\,m-k}) z^{-k} - a_{mm} z^{-m} \\ &= 1 - \sum_{k=1}^{m-1} a_{m-1\,k} z^{-k} - a_{mm} z^{-m} \left(1 - \sum_{k=1}^{m-1} a_{m-1\,k} z^{k}\right) \\ &= A_{m-1}(z) - a_{mm} z^{-m} A_{m-1}(z^{-1}) \end{aligned} \tag{11-4-30}$$

Thus we have the transfer function of the mth-order predictor in terms of the transfer function of the $(m-1)$th-order predictor.

Now suppose we define a filter with transfer function $G_m(z)$ as

$$G_m(z) = z^{-m} A_m(z^{-1}) \tag{11-4-31}$$

Then (11-4-30) may be expressed as

$$A_m(z) = A_{m-1}(z) - a_{mm} z^{-1} G_{m-1}(z) \tag{11-4-32}$$

Note that $G_{m-1}(z)$ represents a transversal filter with tap coefficients $(-a_{m-1\,m-1}, -a_{m-1\,m-2}, \ldots, -a_{m-1\,1}, 1)$, while the coefficients of $A_{m-1}(z)$ are exactly the same except that they are given in reverse order.

More insight into the relationship between $A_m(z)$ and $G_m(z)$ can be obtained by computing the output of these two filters to an input sequence $y(t)$. Using z-transform relations, we have

$$A_m(z)Y(z) = A_{m-1}(z)Y(z) - a_{mm}z^{-1}G_{m-1}(z)Y(z) \qquad (11\text{-}4\text{-}33)$$

We define the outputs of the filters as

$$\begin{aligned} F_m(z) &= A_m(z)Y(z) \\ B_m(z) &= G_m(z)Y(z) \end{aligned} \qquad (11\text{-}4\text{-}34)$$

Then (11-4-33) becomes

$$F_m(z) = F_{m-1}(z) - a_{mm}z^{-1}B_{m-1}(z) \qquad (11\text{-}4\text{-}35)$$

In the time domain, the relation in (11-4-35) becomes

$$f_m(t) = f_{m-1}(t) - a_{mm}b_{m-1}(t-1), \qquad m \geqslant 1 \qquad (11\text{-}4\text{-}36)$$

where

$$f_m(t) = y(t) - \sum_{k=1}^{m-1} a_{mk}\,y(t-k) \qquad (11\text{-}4\text{-}37)$$

$$b_m(t) = y(t-m) - \sum_{k=1}^{m-1} a_{mk}\,y(t-m+k) \qquad (11\text{-}4\text{-}38)$$

To elaborate, $f_m(t)$ in (11-4-37) represents the error of an mth-order forward predictor, while $b_m(t)$ represents the error of an mth-order backward predictor.

The relation in (11-4-36) is one of two that specifies a lattice filter. The second relation is obtained from $G_m(z)$ as follows:

$$\begin{aligned} G_m(z) &= z^{-m}A_m(z^{-1}) \\ &= z^{-m}[A_{m-1}(z^{-1}) - a_{mm}z^m A_{m-1}(z)] \\ &= z^{-1}G_{m-1}(z) - a_{mm}A_{m-1}(z) \end{aligned} \qquad (11\text{-}4\text{-}39)$$

Now, if we multiply both sides of (11-4-39) by $Y(z)$ and express the result in terms of $F_m(z)$ and $B_m(z)$ using the definitions in (11-4-34), we obtain

$$B_m(z) = z^{-1}B_{m-1}(z) - a_{mm}F_{m-1}(z) \qquad (11\text{-}4\text{-}40)$$

By transforming (11-4-40) into the time domain, we obtain the second relation that corresponds to the lattice filter, namely,

$$b_m(t) = b_{m-1}(t-1) - a_{mm}f_{m-1}(t), \qquad m \geqslant 1 \qquad (11\text{-}4\text{-}41)$$

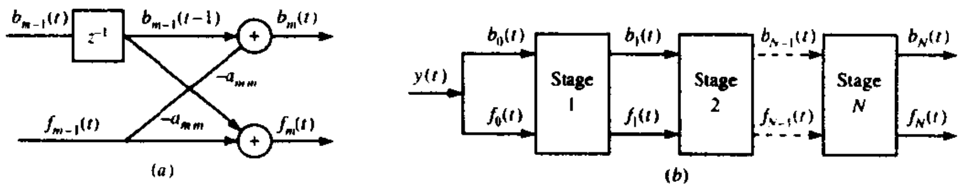


FIGURE 11-4-2 A lattice filter.

The initial condition is

$$f_0(t) = b_0(t) = y(t) \tag{11-4-42}$$

The lattice filter described by the recursive relations in (11-4-36) and (11-4-41) is illustrated in Fig. 11-4-2. Each stage is characterized by its own multiplication factor $\{a_{ii}\}$, $i = 1, 2, \ldots, m$, which is defined in the Levinson–Durbin algorithm. The forward and backward errors $f_m(t)$ and $b_m(t)$ are usually called the *residuals*. The mean square value of these residuals is

$$\mathscr{E}_m = E[f_m^2(t)] = E[b_m^2(t)] \tag{11-4-43}$$

$\mathscr{E}_m$ is given recursively, as indicated in the Levinson–Durbin algorithm, by

$$\begin{aligned} \mathscr{E}_m &= \mathscr{E}_{m-1}(1 - a_{mm}^2) \\ &= \mathscr{E}_0 \prod_{i=1}^{m} (1 - a_{ii}^2) \end{aligned} \tag{11-4-44}$$

where $\mathscr{E}_0 = \phi(0)$.

The residuals $\{f_m(t)\}$ and $\{b_m(t)\}$ satisfy a number of interesting properties, as described by Makhoul (1978). Most important of these are the orthogonality properties

$$\begin{aligned} E[b_m(t)b_n(t)] &= \mathscr{E}_m \delta_{mn} \\ E[f_m(t+m)f_n(t+n)] &= \mathscr{E}_m \delta_{mn} \end{aligned} \tag{11-4-45}$$

Furthermore, the cross-correlation between $f_m(t)$ and $b_n(t)$ is

$$E[f_m(t)b_n(t)] = \begin{cases} a_{nn}\mathscr{E}_m & (m \geqslant n) \\ 0 & (m < n) \end{cases} \qquad m, n \geqslant 0 \tag{11-4-46}$$

As a consequence of the orthogonality properties of the residuals, the different sections of the lattice exhibit a form of independence that allows us to add or delete one or more of the last stages without affecting the parameters of the remaining stages. Since the residual mean square error $\mathscr{E}_m$ decreases monotonically with the number of sections, $\mathscr{E}_m$ can be used as a performance index in determining where the lattice should be terminated.

From the above discussion, we observe that a linear prediction filter can be implemented either as a linear transversal filter or as a lattice filter. The lattice filter is order-recursive, and, as a consequence, the number of sections it contains can be easily increased or decreased without affecting the parameters of the remaining sections. In contrast, the coefficients of a transversal filter obtained on the basis of the RLS criterion are interdependent. This means that an increase or a decrease in the size of the filter results in a change in all coefficients. Consequently, the Kalman algorithm described in Section 11-4-1 is recursive in time but not in order.

Based on least-squares optimization, RLS lattice algorithms have been developed whose computational complexity grow linearly with the number N of filter coefficients (lattice stages). Hence, the lattice equalizer structure is computationally competitive with the direct-form fast RLS equalizer algorithms. RLS lattice algorithms are described in the papers by Morf *et al.* (1973), Satorius and Alexander (1979), Satorius and Pack (1981), Ling and Proakis (1984), and Ling *et al.* (1986).

RLS lattice algorithms have the distinct feature of being numerically robust to round-off error inherent in digital implementations of the algorithm. A treatment of their numerical properties may be found in the papers by Ling *et al.* (1984, 1986).

11-5 SELF-RECOVERING (BLIND) EQUALIZATION

In the conventional zero-forcing or minimum MSE equalizers, we assumed that a known training sequence is transmitted to the receiver for the purpose of initially adjusting the equalizer coefficients. However, there are some applications, such as multipoint communication networks, where it is desirable for the receiver to synchronize to the received signal and to adjust the equalizer without having a known training sequence available. Equalization techniques based on initial adjustment of the coefficients without the benefit of a training sequence are said to be *self-recovering* or *blind*.

Beginning with the paper by Sato (1975), three different classes of adaptive blind equalization algorithms have been developed over the past two decades. One class of algorithms is based on steepest descent for adaptation of the equalizer. A second class of algorithms is based on the use of second- and higher-order (generally, fourth-order) statistics of the received signal to estimate the channel characteristics and to design the equalizer. More recently, a third class of blind equalization algorithms based on the maximum-likelihood criterion have been investigated. In this section, we briefly describe these approaches and give several relevant references to the literature.

11-5-1 Blind Equalization Based on Maximum-Likelihood Criterion

It is convenient to use the equivalent, discrete-time channel model described in Section 10-1-2. Recall that the output of this channel model with ISI is

$$v_n = \sum_{k=0}^{L} f_k I_{n-k} + \eta_n \tag{11-5-1}$$

where $\{f_k\}$ are the equivalent discrete-time channel coefficients, $\{I_n\}$ represents the information sequence, and $\{\eta_n\}$ is a white gaussian noise sequence.

For a block of N received data points, the (joint) probability density function of the received data vector $\mathbf{v} = [v_1 \ v_2 \ \dots \ v_N]'$ conditioned on knowing the impulse response vector $\mathbf{f} = [f_0 \ f_1 \ \dots \ f_L]'$ and the data vector $\mathbf{I} = [I_1 \ I_2 \ \dots \ I_N]'$ is

$$p(\mathbf{v} \mid \mathbf{f}, \mathbf{I}) = \frac{1}{(2\pi\sigma^2)^N} \exp\left(-\frac{1}{2\sigma^2} \sum_{n=1}^{N} \left| v_n - \sum_{k=0}^{L} f_k I_{n-k} \right|^2\right) \tag{11-5-2}$$

The joint maximum-likelihood estimates of $\mathbf{f}$ and $\mathbf{I}$ are the values of these vectors that maximize the joint probability density function $p(\mathbf{v} \mid \mathbf{f}, \mathbf{I})$ or, equivalently, the values of $\mathbf{f}$ and $\mathbf{I}$ that minimize the term in the exponent. Hence, the ML solution is simply the minimum over $\mathbf{f}$ and $\mathbf{I}$ of the metric

$$\begin{aligned} DM(\mathbf{I}, \mathbf{f}) &= \sum_{n=1}^{N} \left| v_n - \sum_{k=0}^{L} f_k I_{n-k} \right|^2 \\ &= \|\mathbf{v} - \mathbf{A}\mathbf{f}\|^2 \end{aligned} \tag{11-5-3}$$

where the matrix $\mathbf{A}$ is called the *data matrix* and is defined as

$$\mathbf{A} = \begin{bmatrix} I_1 & 0 & 0 & \dots & 0 \\ I_2 & I_1 & 0 & \dots & 0 \\ I_3 & I_2 & I_1 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ I_N & I_{N-1} & I_{N-2} & \dots & I_{N-L} \end{bmatrix} \tag{11-5-4}$$

We make several observations. First of all, we note that when the data vector $\mathbf{I}$ (or the data matrix $\mathbf{A}$) is known, as is the case when a training sequence is available at the receiver, the ML channel impulse response estimate obtained by minimizing (11-5-3) over $\mathbf{f}$ is

$$\mathbf{f}_{ML}(\mathbf{I}) = (\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}'\mathbf{v} \tag{11-5-5}$$

On the other hand, when the channel impulse response $\mathbf{f}$ is known, the optimum ML detector for the data sequence $\mathbf{I}$ performs a trellis search (or tree search) by utilizing the Viterbi algorithm for the ISI channel.

When neither $\mathbf{I}$ nor $\mathbf{f}$ are known, the minimization of the performance index $DM(\mathbf{I}, \mathbf{f})$ may be performed jointly over $\mathbf{I}$ and $\mathbf{f}$. Alternatively, $\mathbf{f}$ may be estimated from the probability density function $p(\mathbf{v} \mid \mathbf{f})$, which may be obtained by averaging $p(\mathbf{v}, \mathbf{f} \mid \mathbf{I})$ over all possible data sequences. That is,

$$\begin{aligned} p(\mathbf{v} \mid \mathbf{f}) &= \sum_m p(\mathbf{v}, \mathbf{I}^{(m)} \mid \mathbf{f}) \\ &= \sum_m p(\mathbf{v} \mid \mathbf{I}^{(m)}, \mathbf{f}) P(\mathbf{I}^{(m)}) \end{aligned} \tag{11-5-6}$$

where $P(\mathbf{I}^{(m)})$ is the probability of the sequence $\mathbf{I} = \mathbf{I}^{(m)}$, for $m = 1, 2, \ldots, M^N$ and M is the size of the signal constellation.

Channel Estimation Based on Average over Data Sequences As indicated in the above discussion, when both $\mathbf{I}$ and $\mathbf{f}$ are unknown, one approach is to estimate the impulse response $\mathbf{f}$ after averaging the probability density $p(\mathbf{v}, \mathbf{I} \mid \mathbf{f})$ over all possible data sequences. Thus, we have

$$\begin{aligned} p(\mathbf{v} \mid \mathbf{f}) &= \sum_m p(\mathbf{v} \mid \mathbf{I}^{(m)}, \mathbf{f}) P(\mathbf{I}^{(m)}) \\ &= \sum_m \left[\frac{1}{(2\pi\sigma^2)^N} \exp\left(-\frac{\|\mathbf{v} - \mathbf{A}^{(m)}\mathbf{f}\|^2}{2\sigma^2} \right) \right] P(\mathbf{I}^{(m)}) \end{aligned} \tag{11-5-7}$$

Then, the estimate of $\mathbf{f}$ that maximizes $p(\mathbf{v} \mid \mathbf{f})$ is the solution of the equation

$$\begin{aligned} \frac{\partial p(\mathbf{v} \mid \mathbf{f})}{\partial \mathbf{f}} &= \sum_m P(\mathbf{I}^{(m)}) \\ (\mathbf{A}^{(m)t}\mathbf{A}^{(m)}\mathbf{f} - \mathbf{A}^{(m)t}\mathbf{v}) &\exp\left(-\frac{\|\mathbf{v} - \mathbf{A}^{(m)}\mathbf{f}\|^2}{2\sigma^2} \right) = 0 \end{aligned} \tag{11-5-8}$$

Hence, the estimate of $\mathbf{f}$ may be expressed as

$$\begin{aligned} \mathbf{f} = &\left[\sum_m P(\mathbf{I}^{(m)}) \mathbf{A}^{(m)t}\mathbf{A}^{(m)} g(\mathbf{v}, \mathbf{A}^{(m)}, \mathbf{f}) \right]^{-1} \\ &\times \sum_m P(\mathbf{I}^{(m)}) g(\mathbf{v}, \mathbf{A}^{(m)}, \mathbf{f}) \mathbf{A}^{(m)t}\mathbf{v} \end{aligned} \tag{11-5-9}$$

where the function $g(\mathbf{v}, \mathbf{A}^{(m)}, \mathbf{f})$ is defined as

$$g(\mathbf{v}, \mathbf{A}^{(m)}, \mathbf{f}) = \exp\left(-\frac{\|\mathbf{v} - \mathbf{A}^{(m)}\mathbf{f}\|^2}{2\sigma^2} \right) \tag{11-5-10}$$

The resulting solution for the optimum $\mathbf{f}$ is denoted by $\mathbf{f}_{ML}$.

Equation (11-5-9) is a nonlinear equation for the estimate of the channel impulse response, given the received signal vector $\mathbf{v}$. It is generally difficult to obtain the optimum solution by solving (11-5-9) directly. On the other hand, it is relatively simple to devise a numerical method that solves for $\mathbf{f}_{ML}$ recursively. Specifically, we may write

$$\begin{aligned} \mathbf{f}^{(k+1)} = &\left[\sum_m P(\mathbf{I}^{(m)}) \mathbf{A}^{(m)t}\mathbf{A}^{(m)} g(\mathbf{v}, \mathbf{A}^{(m)}, \mathbf{f}^{(k)}) \right]^{-1} \\ &\times \sum_m P(\mathbf{I}^{(m)}) g(\mathbf{v}, \mathbf{A}^{(m)}, \mathbf{f}^{(k)}) \mathbf{A}^{(m)t}\mathbf{v} \end{aligned} \tag{11-5-11}$$

Once $\mathbf{f}_{ML}$ is obtained from the solution of (11-5-9) or (11-5-11), we may simply use the estimate in the minimization of the metric $DM(\mathbf{I}, \mathbf{f}_{ML})$, given by (11-5-3), over all the possible data sequences. Thus, $\mathbf{I}_{ML}$ is the sequence $\mathbf{I}$ that mimimizes $DM(\mathbf{I}, \mathbf{f}_{ML})$, i.e.,

$$\min_{\mathbf{I}} DM(\mathbf{I}, \mathbf{f}_{ML}) = \min_{\mathbf{I}} \|\mathbf{v} - \mathbf{A}\mathbf{f}_{ML}\|^2 \tag{11-5-12}$$

We know that the Viterbi algorithm is the computationally efficient algorithm for performing the minimization of $DM(\mathbf{I}, \mathbf{f}_{ML})$ over $\mathbf{I}$.

This algorithm has two major drawbacks. First, the recursion for $\mathbf{f}_{LM}$ given by (11-5-11) is computationally intensive. Second, and, perhaps, more importantly, the estimate $\mathbf{f}_{ML}$ is not as good as the maximum-likelihood estimate $\mathbf{f}_{ML}(\mathbf{I})$ that is obtained when the sequence $\mathbf{I}$ is known. Consequently, the error rate performance of the blind equalizer (the Viterbi algorithm) based on the estimate $\mathbf{f}_{ML}$ is poorer than that based on $\mathbf{f}_{ML}(\mathbf{I})$. Next, we consider joint channel and data estimation.

Joint Channel and Data Estimation Here, we consider the joint optimization of the performance index $DM(\mathbf{I}, \mathbf{f})$ given by (11-5-3). Since the elements of the impulse response vector $\mathbf{f}$ are continuous and the elements of the data vector $\mathbf{I}$ are discrete, one approach is to determine the maximum-likelihood estimate of $\mathbf{f}$ for each possible data sequence and, then, to select the data sequence that minimizes $DM(\mathbf{I}, \mathbf{f})$ for each corresponding channel estimate. Thus, the channel estimate corresponding to the mth data sequence $\mathbf{I}^{(m)}$ is

$$\mathbf{f}_{ML}(\mathbf{I}^{(m)}) = (\mathbf{A}^{(m)t}\mathbf{A}^{(m)})^{-1}\mathbf{A}^{(m)t}\mathbf{v}. \tag{11-5-13}$$

For the mth data sequence, the metric $DM(\mathbf{I}, \mathbf{f})$ becomes

$$DM(\mathbf{I}^{(m)}, \mathbf{f}_{ML}(\mathbf{I}^{(m)})) = \|\mathbf{v} - \mathbf{A}^{(m)}\mathbf{f}_{ML}(\mathbf{I}^{(m)})\|^2 \tag{11-5-14}$$

Then, from the set of M^N possible sequences, we select the data sequence that minimizes the cost function in (11-5-14), i.e., we determine

$$\min_{\mathbf{I}^{(m)}} DM(\mathbf{I}^{(m)}, \mathbf{f}_{ML}(\mathbf{I}^{(m)})) \tag{11-5-15}$$

The approach described above is an exhaustive computational search method with a computational complexity that grows exponentially with the length of the data block. We may select $N = L$, and, thus, we shall have one channel estimate for each of the M^L surviving sequences. Thereafter, we may continue to maintain a separate channel estimate for each surviving path of the Viterbi algorithm search through the trellis.

A similar approach has been proposed by Seshadri (1991). In essence, Seshadri's algorithm is a type of generalized Viterbi algorithm (GVA) that retains $K \geqslant 1$ best estimates of the transmitted data sequence into each state of the trellis and the corresponding channel estimates. In Seshadri's GVA, the search is identical to the conventional VA from the beginning up to the L stage of the trellis, i.e., up to the point where the received sequence $(v_1, v_2, \ldots, v_L)$ has been processed. Hence, up to the L stage, an exhaustive search is performed. Associated with each data sequence $\mathbf{I}^{(m)}$, there is a corresponding channel estimate $\mathbf{f}_{ML}(\mathbf{I}^{(m)})$. From this stage on, the search is modified, to retain $K \geq 1$ surviving sequences and associated channel estimates per state instead of only one sequence per state. Thus, the GVA is used for processing the received signal sequence $\{v_n, n \geq L+1\}$. The channel estimate is updated recursively at each stage using the LMS algorithm to further reduce the computational complexity. Simulation results given in the paper by Seshadri (1991) indicate that this GVA blind equalization algorithm performs rather well at moderate signal-to-noise ratios with $K = 4$. Hence, there is a modest increase in the computational complexity of the GVA compared with that for the conventional VA. However, there are additional computations involved with the estimation and updating of the channel estimates $\mathbf{f}(\mathbf{I}^{(m)})$ associated with each of the surviving data estimates.

An alternative joint estimation algorithm that avoids the least-squares computation for channel estimation has been devised by Zervas *et al.* (1991). In this algorithm, the order for performing the joint minimization of the performance index $DM(\mathbf{I}, \mathbf{f})$ is reversed. That is, a channel impulse response, say $\mathbf{f} = \mathbf{f}^{(1)}$ is selected and then the conventional VA is used to find the optimum sequence for this channel impulse response. Then, we may modify $\mathbf{f}^{(1)}$ in some manner to $\mathbf{f}^{(2)} = \mathbf{f}^{(1)} + \Delta\mathbf{f}^{(1)}$ and repeat the optimization over the data sequences $\{\mathbf{I}^{(m)}\}$.

Based on this general approach, Zervas developed a new ML blind equalization algorithm, which is called a *quantized-channel algorithm*. The algorithm operates over a grid in the channel space, which becomes finer and finer by using the ML criterion to confine the estimated channel in the neighborhood of the original unknown channel. This algorithm leads to an efficient parallel implementation, and its storage requirements are only those of the VA.

11-5-2 Stochastic Gradient Algorithm

Another class of blind equalization algorithms are stochastic-gradient iterative equalization schemes that apply a memoryless nonlinearity in the output of a linear FIR equalization filter in order to generate the "desired response" in each iteration.

Let us begin with an initial guess of the coefficients of the optimum equalizer, which we denote by $\{c_n\}$. Then, the convolution of the channel response with the equalizer response may be expressed as

$$\{c_n\} \star \{f_n\} = \{\delta_n\} + \{e_n\} \tag{11-5-16}$$

where $\{\delta_n\}$ is the unit sample sequence and $\{e_n\}$ denotes the error sequence that results from our initial guess of the equalizer coefficients. If we convolve the equalizer impulse response with the received sequence $\{v_n\}$, we obtain

$$\begin{aligned}\{\hat{I}_n\} &= \{v_n\} \star \{c_n\} \\ &= \{I_n\} \star \{f_n\} \star \{c_n\} + \{\eta_n\} \star \{c_n\} \\ &= \{I_n\} \star (\{\delta_n\} + \{e_n\}) + \{\eta_n\} \star \{c_n\} \\ &= \{I_n\} + \{I_n\} \star \{e_n\} + \{\eta_n\} \star \{c_n\} \end{aligned} \tag{11-5-17}$$

The term $\{I_n\}$ in (11-5-17) represents the desired data sequence, the term $\{I_n\} \star \{e_n\}$ represents the residual ISI, and the term $\{\eta_n\} \star \{c_n\}$ represents the additive noise. Our problem is to utilize the deconvolved sequence $\{\hat{I}_n\}$ to find the "best" estimate of a desired response, denoted in general by $\{d_n\}$. In the case of adaptive equalization using a training sequence, $\{d_n\} = \{I_n\}$. In a blind equalization mode, we shall generate a desired response from $\{\hat{I}_n\}$.

The mean square error (MSE) criterion may be employed to determine the "best" estimate of $\{I_n\}$ from the observed equalizer output $\{\hat{I}_n\}$. Since the transmitted sequence $\{I_n\}$ has a nongaussian pdf, the MSE estimate is a nonlinear transformation of $\{\hat{I}_n\}$. In general, the "best" estimate $\{d_n\}$ is given by

$$\begin{aligned} d_n &= g(\hat{I}_n) && \text{(memoryless)} \\ d_n &= g(\hat{I}_n, \hat{I}_{n-1}, \ldots, \hat{I}_{n-m}) && (m\text{th-order memory}) \end{aligned} \tag{11-5-18}$$

where $g(\ )$ is a nonlinear function. The sequence $\{d_n\}$ is then used to generate an error signal, which is fed back into the adaptive equalization filter, as shown in Fig. 11-5-1.

A well-known classical estimation problem is the following. If the equalizer output $\hat{I}_n$ is expressed as

$$\hat{I}_n = I_n + \tilde{\eta}_n \tag{11-5-19}$$

where $\tilde{\eta}_n$ is assumed to be zero-mean gaussian (the central limit theorem may

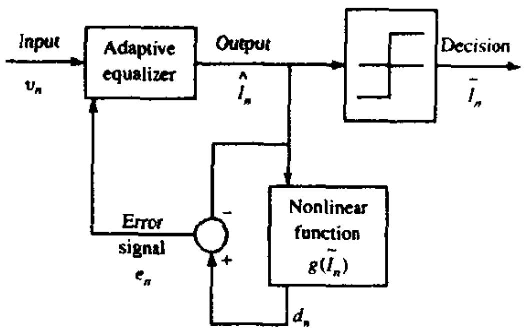


FIGURE 11-5-1 Adaptive blind equalization with stochastic gradient algorithms.

TABLE 11-5-1 STOCHASTIC GRADIENT ALGORITHMS FOR BLIND EQUALIZATION

Equalizer tap coefficients	$\{c_n, 0 \le n \le N-1\}$
Received signal sequence	$\{v_n\}$
Equalizer output sequence	$\{\hat{I}_n\} = \{v_n\} \star \{c_n\}$
Equalizer error sequence	$\{e_n\} = g(\hat{I}_n) - \hat{I}_n$
Tap coefficient update equation	$\mathbf{c}_{n+1} = \mathbf{c}_n + \Delta \mathbf{v}_n^* e_n$

Algorithm	Nonlinearity: $g(\tilde{I}_n)$
Godard	$\frac{\hat{I}_n}{\lvert\tilde{I}_n\rvert}(\lvert\tilde{I}_n\rvert + R_2\lvert\hat{I}_n\rvert - \lvert\hat{I}_n\rvert^3),\ R_2 = \frac{E\{\lvert I_n\rvert^4\}}{E\{\lvert I_n\rvert^2\}}$
Sato	$\zeta \operatorname{csgn}(\hat{I}_n),\ \zeta = \frac{E\{[\mathrm{Re}(I_n)]^2\}}{E\{\lvert\mathrm{Re}(I_n)\rvert\}}$
Benveniste–Goursat	$\hat{I}_n + k_1(\hat{I}_n - I_n) + k_2\lvert\hat{I}_n - \tilde{I}_n\rvert\,[\zeta \operatorname{csgn}(\hat{I}_n) - \tilde{I}_n]$, k_1 and k_2 are positive constants
Stop-and-Go	$\hat{I}_n + \frac{1}{2}A(\hat{I}_n - \tilde{I}_n) + \frac{1}{2}B(\hat{I}_n - \tilde{I}_n)^*$ $(A, B) = (2, 0)$, $(1, 1)$, $(1, -1)$, or $(0, 0)$, depending on the signs of decision-directed error $\hat{I}_n - \tilde{I}_n$ and the error $\zeta \operatorname{csgn}(\hat{I}_n) - \tilde{I}_n$

be invoked here for the residual ISI and the additive noise), $\{I_n\}$ and $\{\tilde{\eta}_n\}$ are statistically independent, and $\{I_n\}$ are statistically independent and identically distributed random variables, then the MSE estimate of $\{I_n\}$ is

$$d_n = E(I_n \mid \tilde{I}_n) \tag{11-5-20}$$

which is a nonlinear function of the equalizer output when $\{I_n\}$ is nongaussian.

Table 11-5-1 illustrates the general form of existing blind equalization algorithms that are based on LMS adaptation. We observe that the basic difference among these algorithms lies in the choice of the memoryless nonlinearity. The most widely used algorithm in practice is the *Godard algorithm*, sometimes also called the *constant-modulus algorithm* (CMA).

It is apparent from Table 11-5-1 that the output sequence $\{d_n\}$ obtained by taking a nonlinear function of the equalizer output plays the role of the desired response or a training sequence. It is also apparent that these algorithms are simple to implement, since they are basically LMS-type algorithms. As such, we expect that the convergence characteristics of these algorithms will depend on the autocorrelation matrix of the received data $\{v_n\}$.

With regard to convergence, the adaptive LMS-type algorithms converge in the mean when

$$E[v_n g^*(\hat{I}_n)] = E[v_n \hat{I}_n^*] \tag{11-5-21}$$

and, in the mean square sense, when (superscript H denotes the conjugate transpose)

$$\begin{aligned} E[\mathbf{c}_n^H v_n g^*(\hat{I}_n)] &= E[\mathbf{c}_n^H v_n \hat{I}_n^*] \\ E[\hat{I}_n g^*(\hat{I}_n)] &= E[\lvert\hat{I}_n\rvert^2] \end{aligned} \tag{11-5-22}$$

Therefore, it is required that the equalizer output $\{\hat{I}_n\}$ satisfy (11-5-22). Note that (11-5-22) states that the autocorrelation of $\{\hat{I}_n\}$ (the right-hand side) equals the cross-correlation between $\hat{I}_n$ and a nonlinear transformation of $\hat{I}_n$ (left-hand side). Processes that satisfy this property are called *Bussgang* (1952), as named by Bellini (1986). In summary, the algorithms given in Table 11-5-1 converge when the equalizer output sequence $\hat{I}_n$ satisfies the Bussgang property.

The basic limitation of stochastic gradient algorithms is their relatively slow convergence. Some improvement in the convergence rate can be achieved by modifying the adaptive algorithms from LMS-type to recursive-least-square (RLS) type.

Godard Algorithm As indicated above, the Godard blind equalization algorithm is a steepest-descent algorithm that is widely used in practice when a training sequence is not available. Let us describe this algorithm in more detail.

Godard considered the problem of combined equalization and carrier phase recovery and tracking. The carrier phase tracking is performed at baseband, following the equalizer as shown in Fig. 11-5-2. Based on this structure, we may express the equalizer output as

$$\hat{I}_k = \sum_{n=-K}^{K} c_n v_{k-n} \tag{11-5-23}$$

and the input to the decision device as $\hat{I}_k \exp(-j\hat{\phi}_k)$, where $\hat{\phi}_k$ is the carrier phase estimate in the kth symbol interval.

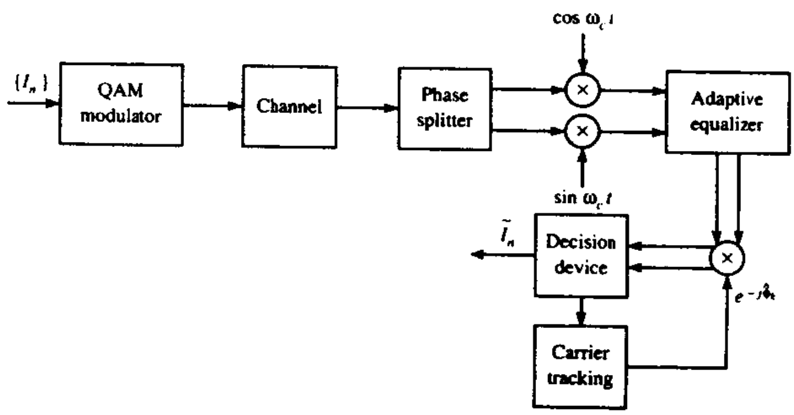
If the desired symbol were known, we could form the error signal

$$\varepsilon_k = I_k - \hat{I}_k e^{-j\hat{\phi}_k} \tag{11-5-24}$$

and minimize the MSE with respect to $\hat{\phi}_k$ and $\{c_n\}$, i.e.,

$$\min_{\hat{\phi}_k, \mathbf{C}} E(|I_k - \hat{I}_k e^{-j\hat{\phi}_k}|^2) \tag{11-5-25}$$

FIGURE 11-5-2 Godard scheme for combined adaptive (blind) equalization and carrier phase tracking.



This criterion leads us to use the LMS algorithm for recursively estimating $\mathbf{C}$ and ϕ_k. The LMS algorithm based on knowledge of the transmitted sequence is

$$\hat{\mathbf{C}}_{k+1} = \hat{\mathbf{C}}_k + \Delta_c(I_k - \hat{I}_k e^{-j\hat{\phi}_k})\mathbf{V}_k^* e^{j\hat{\phi}_k} \tag{11-5-26}$$

$$\hat{\phi}_{k+1} = \hat{\phi}_k + \Delta_\phi \operatorname{Im}(I_k \hat{I}_k^* e^{j\hat{\phi}_k}) \tag{11-5-27}$$

where Δ_c and Δ_ϕ are the step-size parameters for the two recursive equations. Note that these recursive equations are coupled together. Unfortunately, these equations will not converge, in general, when the desired symbol sequence $\{I_k\}$ is unknown.

The approach proposed by Godard is to use a criterion that depends on the amount of intersymbol interference at the output of the equalizer but one that is independent of the QAM signal constellation and the carrier phase. For example, a cost function that is independent of carrier phase and has the property that its minimum leads to a small MSE is

$$G^{(p)} = E(|\hat{I}_k|^p - |I_k|^p)^2 \tag{11-5-28}$$

where p is a positive and real integer. Minimization of $G^{(p)}$ with respect to the equalizer coefficients results in the equalization of the signal amplitude only. Based on this observation, Godard selected a more general cost function, called the *dispersion of order p*, defined as

$$D^{(p)} = E(|\hat{I}_k|^p - R_p)^2 \tag{11-5-29}$$

where R_p is a positive real constant. As in the case of $G^{(p)}$, we observe that $D^{(p)}$ is independent of the carrier phase.

Minimization of $D^{(p)}$ with respect to the equalizer coefficients can be performed recursively according to the steepest-descent algorithm

$$\mathbf{C}_{k+1} = \mathbf{C}_k - \Delta_p \frac{dD^{(p)}}{d\mathbf{C}_k} \tag{11-5-30}$$

where Δ_p is the step-size parameter. By differentiating $D^{(p)}$ and dropping the expectation operation, we obtain the following LMS-type algorithm for adjusting the equalizer coefficients:

$$\hat{\mathbf{C}}_{k+1} = \hat{\mathbf{C}}_k + \Delta_p \mathbf{V}_k^* \hat{I}_k \, |\hat{I}_k|^{p-2}(R_p - |\hat{I}_k|^p) \tag{11-5-31}$$

where Δ_p is the step-size parameter and the optimum choice of R_p is

$$R_p = \frac{E(|I_k|^{2p})}{E(|I_k|^p)} \tag{11-5-32}$$

As expected, the recursion in (11-5-31) for $\hat{\mathbf{C}}_k$ does not require knowledge of the carrier phase. Carrier phase tracking may be carried out in a decision-directed mode according to (11-5-27).

Of particular importance is the case $p = 2$, which leads to the relatively simple algorithm

$$\begin{aligned} \hat{\mathbf{C}}_{k+1} &= \hat{\mathbf{C}}_k + \Delta_p \mathbf{V}_k^* \hat{I}_k (R_2 - |\hat{I}_k|^2) \\ \hat{\phi}_{k+1} &= \hat{\phi}_k + \Delta_\phi \operatorname{Im} (\tilde{I}_k \hat{I}_k^* e^{j\hat{\phi}_k}) \end{aligned} \tag{11-5-33}$$

where $\tilde{I}_k$ is the output decision based on $\hat{I}_k$, and

$$R_2 = \frac{E(|I_k|^4)}{E(|I_k|^2)} \tag{11-5-34}$$

Convergence of the algorithm given in (11-5-33) was demonstrated in the paper by Godard (1980). Initially, the equalizer coefficients were set to zero except for the center (reference) tap, which was set according to the condition

$$|c_0|^2 > \frac{E\,|I_k|^4}{2\,|x_0|^2\,[E(|I_k|^2)]^2} \tag{11-5-35}$$

which is sufficient, but not necessary, for convergence of the algorithm. Simulation results performed by Godard on simulated telephone channels with typical frequency response characteristics and transmission rates of 7200–12 000 bits/s indicate that the algorithm in (11-5-31) performs well and leads to convergence in 5000–20 000 iterations, depending on the signal constellation. Initially, the eye pattern was closed prior to equalization. The number of iterations required for convergence is about an order of magnitude greater than the number required to equalize the channels with a known training sequence. No apparent difficulties were encountered in using the decision-directed phase estimation algorithm in (11-5-33) from the beginning of the equalizer adjustment process.

11-5-3 Blind Equalization Algorithms Based on Second- and Higher-Order Signal Statistics

It is well known that second-order statistics (autocorrelation) of the received signal sequence provide information on the magnitude of the channel characteristics, but not on the phase. However, this statement is not correct if the autocorrelation function of the received signal is periodic, as is the case for a digitally modulated signal. In such a case, it is possible to obtain a measurement of the amplitude and the phase of the channel from the received signal. This cyclostationarity property of the received signal forms the basis for a channel estimation algorithm devised by Tong *et al.* (1993).

It is also possible to estimate the channel response from the received signal by using higher-order statistical methods. In particular, the impulse response of a linear, discrete-time-invariant system can be obtained explicitly from cumulants of the received signal, provided that the channel input is nongaussian. We describe the following simple method for estimation of the channel impulse response from fourth-order cumulants of the received signal sequence. The fourth-order cumulant is defined as

$$\begin{aligned} c(v_k, v_{k+m}, v_{k+n}, v_{k+l}) \equiv c_v(m, n, l) \\ &= E(v_k v_{k+m} v_{k+n} v_{k+l}) \\ &\quad - E(v_k v_{k+m})E(v_{k+n} v_{k+l}) \\ &\quad - E(v_k v_{k+n})E(v_{k+m} v_{k+l}) \\ &\quad - E(v_k v_{k+l})E(v_{k+m} v_{k+n}) \end{aligned} \tag{11-5-36}$$

(The fourth-order cumulant of a gaussian signal process is zero.) Consequently, it follows that

$$c_r(m, n, l) = c(I_k, I_{k+m}, I_{k+n}, I_{k+l}) \sum_{k=0}^{\infty} f_k f_{k+m} f_{k+n} f_{k+l} \tag{11-5-37}$$

For a statistically independent and identically distributed input sequence $\{I_n\}$ to the channel, $c(I_k, I_{k+m}, I_{k+n}, I_{k+l}) = k$, a constant, called the *kurtosis*. Then, if the length of the channel response is $L + 1$, we may let $m = n = l = -L$ so that

$$c_r(-L, -L, -L) = kf_L f_0^3 \tag{11-5-38}$$

Similarly, if we let $m = 0$, $n = L$ and $l = p$, we obtain

$$c_r(0, L, p) = kf_L f_0^2 f_p \tag{11-5-39}$$

If we combine (11-5-38) and (11-5-39), we obtain the impulse response within a scale factor as

$$f_p = f_0 \frac{c_r(0, L, p)}{c_r(-L, -L, -L)}, \qquad p = 1, 2, \ldots, L \tag{11-5-40}$$

The cumulants $c_r(m, n, l)$ are estimated from sample averages of the received signal sequence $\{v_n\}$.

Another approach based on higher-order statistics is due to Hatzinakos and Nikias (1991). They have introduced the first polyspectra-based adaptive blind equalization method named the *tricepstrum equalization algorithm* (TEA). This method estimates the channel response characteristics by using the complex cepstrum of the fourth-order cumulants (tricepstrum) of the received signal sequence $\{v_n\}$. TEA depends only on fourth-order cumulants of $\{v_n\}$ and is capable of separately reconstructing the minimum-phase and maximum-phase characteristics of the channel. The channel equalizer coefficients are then computed from the measured channel characteristics. The basic approach used in TEA is to compute the tricepstrum of the received sequence $\{v_n\}$, which is the inverse (three-dimensional) Fourier transform of the logarithm of the trispectrum of $\{v_n\}$. (The *trispectrum* is the three-dimensional discrete Fourier transform of the fourth-order cumulant sequence $c_r(m, n, l)$). The equalizer coefficients are then computed from the cepstral coefficients.

By separating the channel estimation from the channel equalization, it is possible to use any type of equalizer for the ISI, i.e., either linear, or decision-feedback, or maximum-likelihood sequence detection. The major disadvantage with this class of algorithms is the large amount of data and the inherent computational complexity involved in the estimation of the higher-order moments (cumulants) of the received signal.

In conclusion, we have provided an overview of three classes of blind equalization algorithms that find applications in digital communications. Of the three families of algorithms described, those based on the maximum-likelihood criterion for jointly estimating the channel impulse response and the data sequence are optimal and require relatively few received signal samples for performing channel estimation. However, the computational complexity of the algorithms is large when the ISI spans many symbols. On some channels, such as the mobile radio channel, where the span of the ISI is relatively short, these algorithms are simple to implement. However, on telephone channels, where the ISI spans many symbols but is usually not too severe, the LMS-type (stochastic gradient) algorithms are generally employed.

11-6 BIBLIOGRAPHICAL NOTES AND REFERENCES

Adaptive equalization for digital communications was developed by Lucky (1965, 1966). His algorithm was based on the peak distortion criterion and led to the zero-forcing algorithm. Lucky's work was a major breakthrough, which led to the rapid development of high-speed modems within five years of publication of his work. Concurrently, the LMS algorithm was devised by Widrow (1966), and its use for adaptive equalization for complex-valued (in-phase and quadrature components) signals was described and analyzed in a tutorial paper by Proakis and Miller (1969).

A tutorial treatment of adaptive equalization algorithms that were developed during the period 1965–1975 is given by Proakis (1975). A more recent tutorial treatment of adaptive equalization is given in the paper by Qureshi (1985). The major breakthrough in adaptive equalization techniques, beginning with the work of Lucky in 1965 coupled with the development of trellis-coded modulation, which was proposed by Ungerboeck and Csajka (1976), has led to the development of commercially available high speed modems with a capability of speeds of 9600–28 800 bits/s on telephone channels.

The use of a more rapidly converging algorithm for adaptive equalization was proposed by Godard (1974). Our derivation of the RLS (Kalman) algorithm, described in Section 11-4-1, follows the approach outlined by Picinbono (1978). RLS lattice algorithms for general signal estimation applications were developed by Morf *et al.* (1977, 1979). The applications of these algorithms have been investigated by several researchers, including Makhoul (1978), Satorius and Pack (1981), Satorius and Alexander (1979), and Ling and Proakis (1982, 1984a–c, 1985). The fast RLS Kalman algorithm for adaptive equalization was first described by Falconer and Liung (1978). The above references are just a few of the important papers that have been published on RLS algorithms for adaptive equalization and other applications.

Sato's (1975) original work on blind equalization was focused on PAM (one-dimensinal) signal constellations. Subsequently it was generalized to two-dimensional and multidimensional signal constellations in the algorithms devised by Godard (1980), Benveniste and Goursat (1984), Sato (1986), Foschini (1985), Picchi and Prati (1987), and Shalvi and Weinstein (1990). Blind equalization methods based on the use of second- and higher-order moments of the received signal were proposed by Hatzinakos and Nikias (1991) and Tong *et al.* (1994). The use of the maximum-likelihood criterion for joint channel estimation and data detection has been investigated and treated in papers by Seshadri (1991), Ghosh and Weber (1991), Zervas *et al.* (1991) and Raheli *et al.* (1995). Finally, the convergence characteristics of stochastic gradient blind equalization algorithms have been investigated by Ding (1990), Ding *et al.* (1989), and Johnson (1991).

PROBLEMS

11-1 An equivalent discrete-time channel with white gaussian noise is shown in Fig. P11-1.

a Suppose we use a linear equalizer to equalize the channel. Determine the tap coefficients c_{-1}, c_0, c_1 of a three-tap equalizer. To simplify the computation, let the AWGN be zero.

b The tap coefficients of the linear equalizer in (a) are determined recursively via the algorithm

$$\mathbf{C}_{k+1} = \mathbf{C}_k - \Delta \mathbf{g}_k, \qquad \mathbf{C}_k = [c_{-1k} \quad c_{0k} \quad c_{1k}]'$$

where $\mathbf{g}_k = \Gamma \mathbf{C}_k - \mathbf{b}$ is the gradient vector and Δ is the step size. Determine the range of values of Δ to ensure convergence of the recursive algorithm. To simplify the computation, let the AWGN be zero.

c Determine the tap weights of a DFE with two feedforward taps and one feedback gap. To simplify the computation, let the AWGN be zero.

11-2 Refer to Problem 10-18 and answer the following questions.

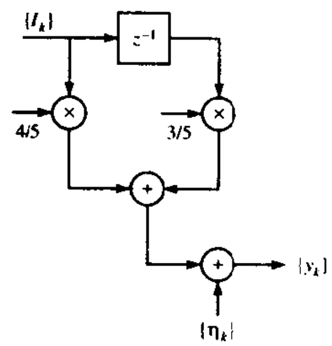


FIGURE P11-1

a Determine the maximum value of Δ that can be used to ensure that the equalizer coefficients converge during operation in the adaptive mode.
b What is the variance of the self-noise generated by the three-tap equalizer when operating in an adaptive mode, as a function of Δ? Suppose it is desired to limit the variance of the self-noise to 10% of the minimum MSE for the three-tap equalizer when $N_0 = 0.1$. What value of Δ would you select?
c If the optimum coefficients of the equalizer are computed recursively by the method of steepest descent, the recursive equation can be expressed in the form

$$\mathbf{C}_{(n+1)} = (\mathbf{I} - \Delta\mathbf{\Gamma})\mathbf{C}_{(n)} + \Delta\boldsymbol{\xi}$$

where $\mathbf{I}$ is the identity matrix. The above represents a set of three coupled first-order difference equations. They can be decoupled by a linear transformation that diagonalizes the matrix $\mathbf{\Gamma}$. That is, $\mathbf{\Gamma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}'$ where $\mathbf{\Lambda}$ is the diagonal matrix having the eigenvalues of Γ as its diagonal elements and $\mathbf{U}$ is the (normalized) modal matrix that can be obtained from your answer to 10-18(b). Let $\mathbf{C}' = \mathbf{U}'\mathbf{C}$ and determine the steady-state solution for $\mathbf{C}'$. From this, evaluate $\mathbf{C} = (U')^{-1}\mathbf{C}' = \mathbf{U}\mathbf{C}'$ and, thus, show that your answer agrees with the result obtained in 10-18(a).

11-3 When a periodic pseudo-random sequence of length N is used to adjust the coefficients of an N-tap linear equalizer, the computations can be performed efficiently in the frequency domain by use of the discrete Fourier transform (DFT). Suppose that $\{y_n\}$ is a sequence of N received samples (taken at the symbol rate) at the equalizer input. Then the computation of the equalizer coefficients is performed as follows.
a Compute the DFT of one period of the equalizer input sequence $\{y_n\}$, i.e.,

$$Y_k = \sum_{n=0}^{N-1} y_n e^{-j2\pi nk/N}$$

b Compute the desired equalizer spectrum

$$C_k = \frac{X_k Y_k^*}{|Y_k|^2}, \qquad k = 0, 1, \ldots, N-1$$

where $\{X_i\}$ is the precomputed DFT of the training sequence.
c Compute the inverse DFT of $\{C_k\}$ to obtain the equalizer coefficients $\{c_n\}$. Show that this procedure in the absence of noise yields an equalizer whose frequency response is equal to the frequency response of the inverse folded channel spectrum at the N uniformly spaced frequencies $f_k = k/NT$, $k = 0, 1, \ldots, N-1$.

11-4 Show that the gradient vector in the minimization of the MSE may be expressed as

$$\mathbf{G}_k = -E(\varepsilon_k \mathbf{V}_k^*)$$

where the error $\varepsilon_k = I_k - \hat{I}_k$, and the estimate of $\mathbf{G}_k$, i.e.,

$$\hat{\mathbf{G}}_k = -\varepsilon_k \mathbf{V}_k^*$$

satisfies the condition that $E(\hat{\mathbf{G}}_k) = \mathbf{G}_k$.

11-5 The *tap-leakage LMS algorithm* proposed in the paper by Gitlin *et al.* (1982) may be expressed as

$$\mathbf{C}_N(n+1) = w\mathbf{C}_N(n) + \Delta\varepsilon(n)\mathbf{V}_N^*(n)$$

where $0<w<1$, Δ is the step size, and $\mathbf{V}_N(n)$ is the data vector at time n. Determine the condition for the convergence of the mean value of $\mathbf{C}_N(n)$.

11-6 Consider the random process

$$x(n)=gv(n)+w(n), \qquad n=0,1,\ldots,M-1$$

where $v(n)$ is a known sequence, g is a random variable with $E(g)=0$, and $E(g^2)=G$. The process $w(n)$ is a white noise sequence with

$$\gamma_{ww}(m)=\sigma_w^2\,\delta_m$$

Determine the coefficients of the linear estimator for g, that is,

$$\hat{g}=\sum_{n=0}^{M-1} h(n)x(n)$$

that minimize the mean square error

11-7 A digital transversal filter can be realized in the frequency-sampling form with system function (see Problem 10-25)

$$H(z)=\frac{1-z^{-M}}{M}\sum_{k=0}^{M-1}\frac{H_k}{1-e^{j2\pi k/M}z^{-1}}$$
$$=H_1(z)H_2(z)$$

where $H_1(z)$ is the comb filter, $H_2(z)$ is the parallel bank of resonators, and $\{H_k\}$ are the values of the discrete Fourier transform (DFT).

a Suppose that this structure is implemented as an adaptive filter using the LMS algorithm to adjust the filter (DFT) parameters $\{H_k\}$. Give the time-update equation for these parameters. Sketch the adaptive filter structure.

b Suppose that this structure is used as an adaptive channel equalizer in which the desired signal is

$$d(n)=\sum_{k=0}^{M-1} A_k\cos\omega_k n, \qquad \omega_k=\frac{2\pi k}{M}$$

With this form for the desired signal, what advantages are there in the LMS adaptive algorithm for the DFT coefficients $\{H_k\}$ over the direct-form structure with coefficients $\{h(n)\}$? (see Proakis, 1970).

11-8 Consider the performance index

$$J=h^2+40h+28$$

Suppose that we search for the minimum of J by using the steepest-descent algorithm

$$h(n+1)=h(n)-\tfrac{1}{2}\Delta g(n)$$

where $g(n)$ is the gradient.

a Determine the range of values of Δ that provides an overdamped system for the adjustment process.

b Plot the expression for J as a function of n for a value of Δ in this range.

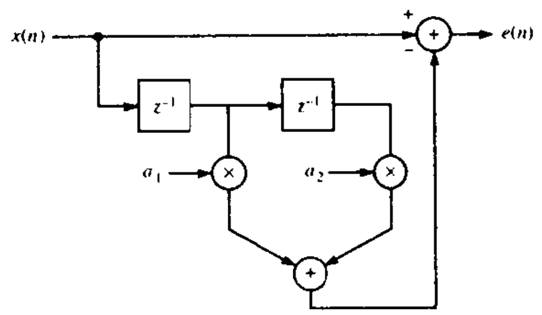


FIGURE P11-9

11-9 Determine the coefficients a_1 and a_2 for the linear predictor shown in Fig. P11-9, given that the autocorrelation $\gamma_{xx}(m)$ of the input signal is

$$\gamma_{xx}(m) = b^{|m|}, \qquad 0 < b < 1$$

11-10 Determine the lattice filter and its optimum reflection coefficients corresponding to the linear predictor in Problem 11-9.

11-11 Consider the adaptive FIR filter shown in Fig. P11-11. The system $C(z)$ is characterized by the system function

$$C(z) = \frac{1}{1 - 0.9z^{-1}}$$

Determine the optimum coefficients of the adaptive transversal (FIR) filter $B(z) = b_0 + b_1 z^{-1}$ that minimize the mean square error. The additive noise is white with variance $\sigma_w^2 = 0.1$.

11-12 An $N \times N$ correlation matrix Γ has eigenvalues $\lambda_1 > \lambda_2 > \ldots > \lambda_N > 0$ and associated eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N$. Such a matrix can be represented as

$$\boldsymbol{\Gamma} = \sum_{i=1}^{N} \lambda_i \mathbf{v}_i \mathbf{v}_i^{*t}$$

a If $\Gamma = \Gamma^{1/2}\Gamma^{1/2}$, where $\Gamma^{1/2}$ is the square root of Γ, show that $\Gamma^{1/2}$ can be represented as

$$\boldsymbol{\Gamma}^{1/2} = \sum_{i=1}^{N} \lambda_i^{1/2} \mathbf{v}_i \mathbf{v}_i^{*t}$$

b Using this representation, determine a procedure for computing $\Gamma^{1/2}$.

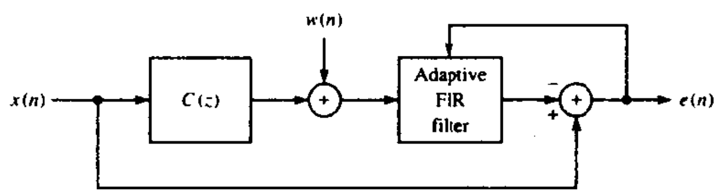


FIGURE P11-11