From Aspects of Network and System Theory, edited by
R.E. Kalman & N. DeClaris, Holt, Rinehart and Winston, Inc.,

1971.

Adaptive Filters

Bernard Widrow

INTRODUCTION

The term "filter" is often applied to any device or system that processes incoming signals or other data in such a way as to eliminate noise, or smooth the signals, or identify each signal as belonging to a particular class, or predict the next input signal from moment to moment.

This paper presents an approach to signal filtering using an *adaptive filter* that is in some sense self-designing (really self-optimizing). The adaptive filter described here bases its own "design" (its internal adjustment settings) upon *estimated* (measured) statistical characteristics of in put and output signals. The statistics are not measured explicitly and then used to design the filter; rather, the filter design is accomplished in a single process by a recursive algorithm that automatically updates the system adjustments with the arrival of each new data sample.

Inevitable errors in the statistics estimates prevent the adaptive filter from delivering optimal performance, but the loss in performance can often be made quite small. This loss will be related to the averaging time (which in turn is related to the speed of adaptation) and to the number of internal adjustments.

AN ADAPTIVE FILTER STRUCTURE

Many forms of adaptive filters have been described in the literature, some of which have been shown to be optimal in certain applications. The special form of filter to be treated here, if its adjustments were fixed, would be a simple linear discrete feedback-free system. Although there exist applications for which the use of this filter would be optimal, a principal reason for considering such a filter is its relatives implicity. Its form is basic and fundamental, and analysis of its behavior during adaptation gives insight to other more complicated adaptation processes.

The filter to be considered here consists of a tapped delay line, variable weights (variable gains) whose input signals are the signals at the delay-line taps, a summer to add the weighted signals, and machinery to adjust the weights automatically. The impulse response of such a discrete system is completely controlled by the weight settings. The adaptation process automatically seeks an optimal filter impulse response by adjusting the weights. Figure 1 illustrates schematically the adaptive filter used in this case for modeling an unknown dynamic system.

Two kinds of processes take place in the adaptive filter: training and operating. The training (adaptation) process is concerned with adjusting the weights. The operating process consists in forming output signals by weighting the delay-line tap signals, using the weights resulting from the training process.

During the training process, an additional input signal, the "desired response," must be supplied to the adaptive filter along with the usual input signals. This requirement may in some cases restrict the use of this particular form of adaptive filter.

An example illustrating the use of the desired-response signal is that shown in Fig. 1. Here a continuous input signal $f(t)$ is applied to an unknown system to be modeled. The discrete adaptive model is supplied with an input signal $f(j)$, derived from samples of $f(t)$. The output of the unknown system $g(t)$ is sampled, and these samples $g(j)$ are compared with the output $y(j)$ of the adaptive-system model. This system can self-adapt to minimize the mean-square error, where the error is defined as the difference between the output of the adaptive model and the output of the unknown system (the latter output being taken as the desired response for the adaptive model).

It will be shown that if the input and output signals of the system being modeled are statistically stationary, the error signal has a mean-square value which is a quadratic function of the weight settings. Thus, the mean-square-error function may be viewed as a "performance surface" for the adaptive process. Automatic minimization of mean-square error can be accomplished by "hill-climbing" methods. For the adaptive filter shown in Fig. 1, the performance surface has a unique stationary point (a minimum) which can be sought using gradient techniques.

THE PERFORMANCE SURFACE

The analysis of the adaptive filter can be developed by considering the adaptive linear combinatorial system shown in Fig. 2. This combinatorial system is embedded in the adaptive filter of Fig. 1, and indeed is its most significant part.[1]

In the system of Fig. 2, a set of stationary input signals is weighted and summed to form an output signal. The input signals in the set are assumed to occur simultaneously and discretely in time. The jth set of input signals is designated by the vector $\mathbf{X}^T(j) = [x_1(j), x_2(j), \cdots, x_l(j), \cdots, x_n(j)]$. The set of weights is designated by the vector

[1] This combinatorial system can also be connected to the elements of a phased array antenna to make an adaptive antenna,[1] as will be shown subsequently; below; to a quantizer to form an adaptive threshold element ("Adaline"[2] or TLU[3]) for use in adaptive logic and pattern-recognition systems; or it can be used as the adaptive portion of certain learning control systems.[4],[5]

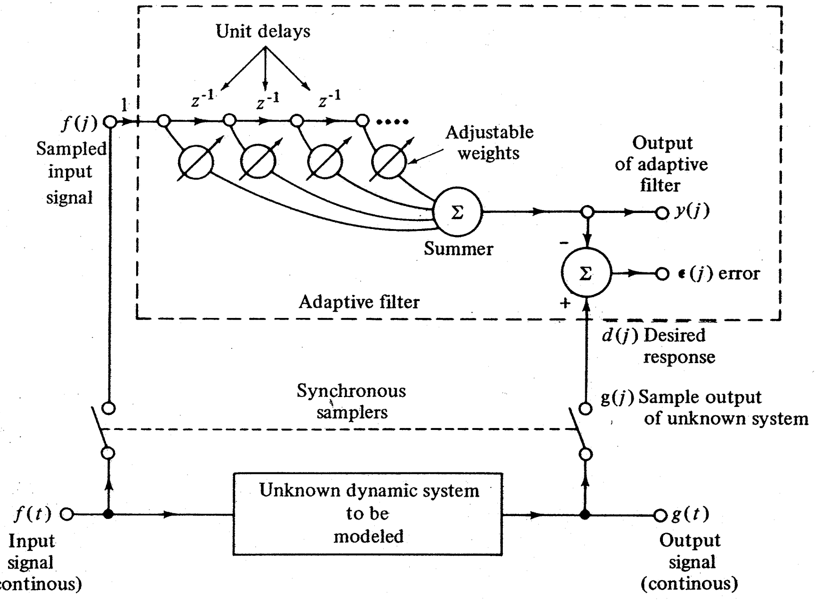

FIG. 1. Modeling an unknown system by a discrete adaptive filter.

$\mathbf{W}^T(j) = [w_1(j), w_2(j), \cdots, x_l(j), \cdots, x_n(j)]$. The jth output signal is

$$y(j) = \sum_{l=1}^{n} w_l(j)x_l(j) \tag{1}$$

This can be written in matrix form as

$$y(j) = \mathbf{W}^T(j)\mathbf{X}(j) = \mathbf{X}^T(j)\mathbf{W}(j) \tag{2}$$

Denoting the desired response for the jth set of input signals as $d(j)$, the error at the jth time is

$$\epsilon(j) = d(j) - y(j) = d(j) - \mathbf{W}^T(j)\mathbf{X}(j) \tag{3}$$

The square of this error is

$$\epsilon^2(j) = d^2(j) - 2d(j)\mathbf{X}^T(j)\mathbf{W}(j) + \mathbf{W}^T(j)\mathbf{X}(j)\mathbf{X}^T(j)\mathbf{W} \tag{4}$$

The mean-square error, the expected value of $\epsilon^2(j)$, is

$$E[\epsilon^2(j)] = \bar{d}^2(j) - 2\mathbf{\Phi}(x, d)\mathbf{W}(j) + \mathbf{W}^T(j)\mathbf{\Phi}(x, x)\mathbf{W}(j) \tag{5}$$

where the vector of cross-correlations between the input signals and the desired response is defined as

$$E[d(j)\mathbf{X}(j)] = E\begin{bmatrix} x_1(j)d(j) \\ x_2(j)d(j) \\ \cdot \\ \cdot \\ \cdot \\ x_n(j)d(j) \end{bmatrix} \triangleq \mathbf{\Phi}(x, d) \tag{6}$$

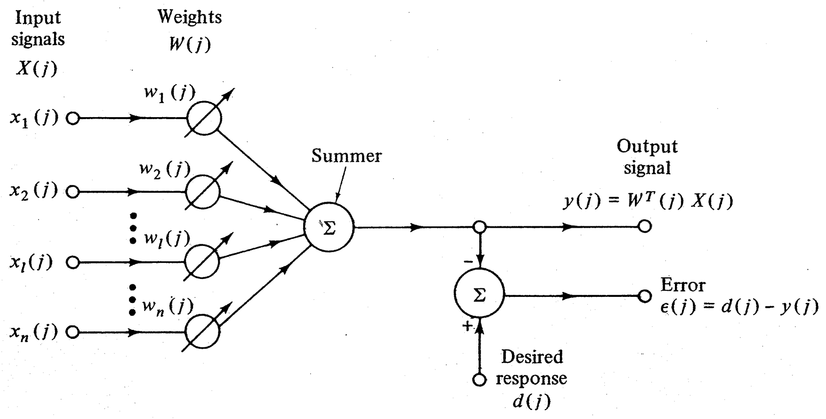

FIG. 2. Adaptive linear combinatorial system.

and where the correlation matrix of the input signals is defined as

$$E[\mathbf{X}(j)\mathbf{X}^T(j)] = E\begin{bmatrix} x_1(j)x_1(j) & x_1(j)x_2(j) & \cdots & \\ x_2(j)x_1(j) & x_2(j)x_2(j) & \cdots & \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ & & \cdots & x_n(j)x_n(j) \end{bmatrix} \triangleq \mathbf{\Phi}(x, x) \tag{7}$$

It may be observed from (5) that for stationary input signals, the mean-square error is precisely a second-order function of the weights. The mean-square-error performance function may be visualized as a bowl-shaped surface, a parabolic function of the weight variables. The adaptive process has the job of continually seeking the "bottom of the bowl." A means of accomplishing this by the well-known method of steepest descent[6],[7] is discussed below.

In the nonstationary case, the bottom of the bowl may be moving, while the orientation and curvature of the bowl may be changing. The adaptive process has to track the bottom of the bowl when inputs are nonstationary. Detailed analysis of the adaptive process when the input statistics are time-variable is beyond the scope of this paper and is a subject of strong current research interest.

It will be assumed that the input and desired-response signals are stationary. This paper is concerned with transient phenomena that take place when a system is adapting to an unknown stationary input process, and in addition, it is concerned with steady-state behavior after adaptive transients die out.

THE GRADIENT AND THE WIENER SOLUTION

The method of steepest descent uses gradients of the performance surface in seeking its minimum. The gradient at any point on the performance surface may be obtained by differentiating the mean-square-error function of equation (5) with respect to the weight vector. The gradient is

$$\nabla[\bar{\epsilon}^2(j)] = -2\mathbf{\Phi}(x, d) + 2\mathbf{\Phi}(x, x)\mathbf{W}(j) \tag{8}$$

To find the "optimal" weight vector $\mathbf{W}_{\mathrm{LMS}}$ that yields the least mean-square error, set the gradient to zero. Accordingly,

$$\mathbf{\Phi}(x, d) = \mathbf{\Phi}(x, x)\mathbf{W}_{\mathrm{LMS}} \tag{9}$$

$$\mathbf{W}_{\mathrm{LMS}} = \mathbf{\Phi}^{-1}(x, x)\mathbf{\Phi}(x, d) \tag{10}$$

Equation (10) is the Wiener–Hopf equation in matrix form.

An expression for the minimum mean-square error may be obtained by substituting (10) into (5):

$$\bar{\epsilon}_{\min}^2 = \bar{d}^2(j) - \mathbf{W}_{\mathrm{LMS}}^T \mathbf{\Phi}(x, d) \tag{11}$$

THE METHOD OF STEEPEST DESCENT

In seeking the minimum mean-square error by the method of steepest descent, one begins with an initial guess as to where the minimum point of the mean-square-error surface may be. This means that one begins with a set of initial conditions for the weights. The gradient vector is then measured, and the next guess is obtained from the present guess by making a change in the weight vector in the direction of the negative of the gradient vector—that is, in the opposite direction of the gradient vector. If the mean-square error is reduced with each change in the weight vector, the process will converge on the stationary point (minimum) regardless of the choice of initial weights.

A plan view of a two-dimensional (two-weight) quadratic performance surface is shown in Figs. 3A and B. The mean-square error is assumed to be measured along a coordinate normal to the plane of the paper. The computer-drawn ellipses represent contours of constant mean-square error, spaced at equal increments. The gradient must be orthogonal to these contours everywhere on the surface. A series of small steps undertaken by the weight vector, starting with an initial guess, is illustrated in Fig. 3A. These steps are so small that they appear to comprise a continuous chain. A series of larger steps is shown in Fig. 3B. Each step is taken normal to the error contour from which it begins. It will be shown that the weights undergo geometric (discrete exponential) transients in

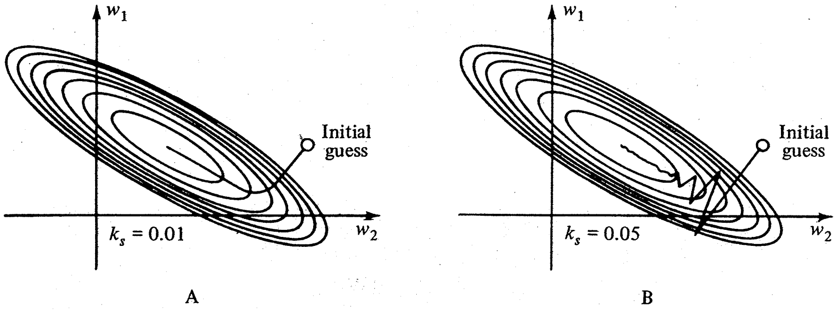

FIG. 3. Illustration of method of steepest descent. A—Overdamped. B—Underdamped.

relaxing toward the surface minimum. "Overdamping" and "underdamping" are illustrated in Figs. 3A and B, respectively.

The method of steepest descent makes each change in the weight vector proportional to the gradient vector; the method of steepest descent can thus be described by the following relation:

$$\mathbf{W}(j+1) = \mathbf{W}(j) + k_s \nabla[\bar{\epsilon}^2(j)] \tag{12}$$

An expression for $\nabla[\bar{\epsilon}^2(j)]$ can be obtained by using equation (8), so

$$\mathbf{W}(j+1) = \mathbf{W}(j) + 2k_s \mathbf{W}^T(j)\mathbf{\Phi}(x, x) - 2k_s \mathbf{\Phi}(x, d) \tag{13}$$

The gradient vector $\nabla[\bar{\epsilon}^2(j)]$ is the gradient of the expected error-squared function when the weight vector is $\mathbf{W}(j)$.

When, as in the present case, the performance function is quadratic, the gradient is a *linear* function of the weights. The beauty of working with the quadratic performance surface lies both in this linear relation and in the freedom from relative minima that is a characteristic of such a surface.

FEEDBACK MODEL OF STEEPEST DESCENT

The analysis of steepest-descent adaptation is facilitated by making use of the familiar feedback flow graph,[8],[9] used in a multidimensional sense, to express relations (12) and (13). A feedback model is highly appropriate, since the gradient is like an "error" signal in an n-dimensional servomechanism that controls the adjustments of the adaptive filter. The bigger the gradient, the greater is the required weight–vector correlation; when the gradient is zero, no correction is needed since the "error" in the weight settings is zero. This form of feedback has been called "*performance feedback*" by the writer.[10],[11]

The flow graph incorporating relations (12) and (13) is shown in Fig. 4. The "signals" at the nodes are indicated by row vectors rather than by column vectors. The transfer function of each branch is a matrix, as indicated on the flow graph. The signal vector flowing out of each branch is that flowing in multiplied by the matrix transfer function of the branch. The matrix transfer function of two parallel branches of such a graph is the sum of the matrix transfer functions of the branches. The matrix transfer function of two branches in cascade is the product of the matrix transfer functions arranged in the order of signal flow, since the signal vectors are row vectors. The symbol Z^{-1} is the "frequency domain" or Z-transform[12]–[15] representation of a delay of one iteration cycle; $Z^{-1}I$ is the matrix transfer function of a unit delay branch, and so forth. The graph represents a first-order multidimensional sampled-data system.

Transient phenomena in $\mathbf{W}(j)$ will take place in the flow-graph model exactly as they will in the actual hill-climbing process if the initial weight vector $\mathbf{W}(0)$ in the flow graph is set to the initial guess. Transients in the weight components can be studied by examining the natural behavior of the flow graph. The "output" of the graph is the present weight vector $\mathbf{W}(j)$. Assume, for the moment, that precise gradient measurements are available during the hill-climbing process, so the source of additive gradient measurement noise may be ignored. The reader can verify that equilibrium conditions in the graph are

$$\mathbf{W}(\infty) = \mathbf{\Phi}^{-1}(x, x)\mathbf{\Phi}(x, d) = \mathbf{W}_{\text{LMS}} \tag{14}$$

Each branch transfer function in the flow graph of Fig. 4 is a diagonal matrix except for the feedback branch labeled $2\mathbf{\Phi}(x, x)$. In general, this branch matrix will have finite off-diagonal elements since the input signals may be mutually correlated. As a result, transients will cross-couple from one component of the weight vector to the next. This somewhat complicates the study of transient phenomena in the hill-climbing process.

DIAGONALIZATION OF THE FEEDBACK MODEL: THE NATURAL MODES

To facilitate the analysis of adaptive transients, the flow graph may be diagonalized. Consider the expression for the mean-square error given

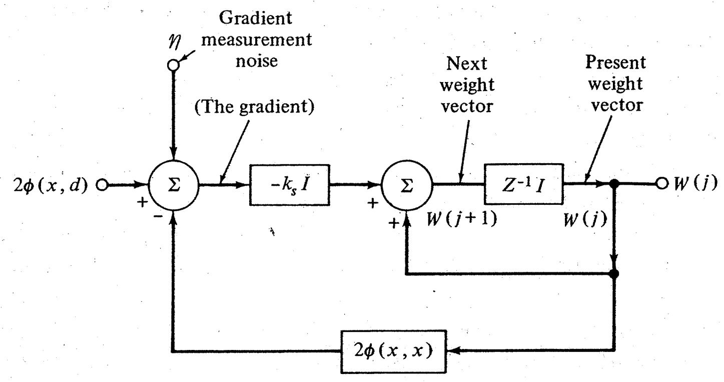

FIG. 4. Feedback model of the method of steepest descent with quadratic performance surface.

by (5). Combining this with (10) and (11), the mean-square error may be expressed as

$$\bar{\epsilon}^2(j) = \bar{\epsilon}_{\min}{}^2 + [\mathbf{W}(j) - \mathbf{W}_{\mathrm{LMS}}]^T \mathbf{\Phi}(x, x)[\mathbf{W}(j) - \mathbf{W}_{\mathrm{LMS}}] \tag{15}$$

The $\mathbf{\Phi}(x, x)$ matrix is real, symmetric, and positive definite. It may be expanded in normal form:

$$\mathbf{\Phi}(x, x) = \mathbf{Q}^{-1}\mathbf{\Lambda}\mathbf{Q} \tag{16}$$

The diagonal matrix of eigenvalues is $\mathbf{\Lambda}$, and the square matrix of eigenvectors is the modal matrix $\mathbf{Q}$. Let the latter matrix be constructed or normalized eigenvectors, thus making $\mathbf{Q}$ orthonormal; therefore $\mathbf{Q}^{-1} = \mathbf{Q}^T$. The mean-square error may now be expressed as

$$\bar{\epsilon}^2(j) = \bar{\epsilon}_{\min}{}^2 + [\mathbf{W}(j) - \mathbf{W}_{\mathrm{LMS}}]^T \mathbf{Q}^T\mathbf{\Lambda}\mathbf{Q}[\mathbf{W}(j) - \mathbf{W}_{\mathrm{LMS}}] \tag{17}$$

A new set of coordinates may be defined as follows:

$$\begin{aligned} \mathbf{W}^T(j)\mathbf{Q}^T &\triangleq \mathbf{W}'^T(j) \\ \mathbf{Q}\mathbf{W}(j) &= \mathbf{W}'(j) \end{aligned} \tag{18}$$

Substituting these into (17) yields

$$\bar{\epsilon}^2(j) = \bar{\epsilon}_{\min}{}^2 + [\mathbf{W}'(j) - \mathbf{W}_{\mathrm{LMS}}']^T \mathbf{\Lambda}[\mathbf{W}'(j) - \mathbf{W}_{\mathrm{LMS}}'] \tag{19}$$

The transformation $\mathbf{Q}$ projects $\mathbf{W}(j)$ into $\mathbf{W}'(j)$—that is, projects $\mathbf{W}(j)$ onto the primed coordinates. It can be observed from (19) that since $\mathbf{\Lambda}$ is diagonal, the primed coordinates must comprise the principal axes of the quadratic mean-square-error performance surface.

Refer once again to the feedback model of steepest descent, shown in Fig. 4. By inserting the normal form (16) for the transfer function $\mathbf{\Phi}(x, x)$ and by some manipulation of the feedback model, the simpler but completely equivalent feedback model of Fig. 5 results. All cross-couplings within the feedback paths have now been eliminated.

The natural modes of steepest descent are the natural modes of the feedback portion of the model of Fig. 5. Since the transients are isolated and each of the primed coordinates has its own natural mode, the natural behavior of steepest descent can be completely explored by considering the action within a single primed coordinate.

Consider, therefore, the isolated one-dimensional feedback model for the pth normal coordinate, shown in Fig. 6. The transfer function of this feedback system is

$$\frac{-k_s Z^{-1}}{1 - (1 + 2k_s\lambda_p)Z^{-1}} \tag{20}$$

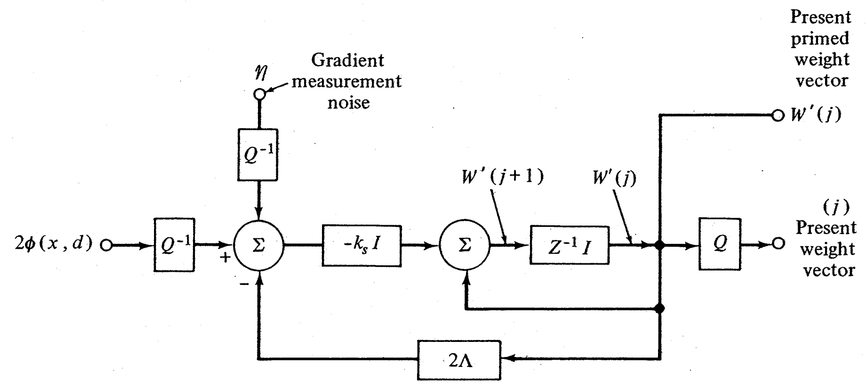

FIG. 5. Feedback model of steepest descent, using normal (primed) coordinates.

where λ_p is the pth eigenvalue of $\mathbf{\Phi}(x, x)$. The impulse response is geometric, having the geometric ratio

$$r_p = 1 + 2k_s\lambda_p \tag{21}$$

An exponential envelope of time constant τ_p can be fitted to the discrete impulse response by considering the unit of time to be one iteration (adaptation) cycle and by making the time constant such that

$$r_p = e^{-1/\tau_p} \tag{22}$$

The pth time constant can be expressed in terms of the constant k_s and the eigenvalue λ_p as

$$\tau_p = \frac{-1}{\ln(1 + 2k_s\lambda_p)} \tag{23}$$

In most practical situations, k_s is small, so $r_p \ll 1$. Accordingly, for slow adaptation

$$\tau_p = \frac{1}{2(-k_s)\lambda_p}$$

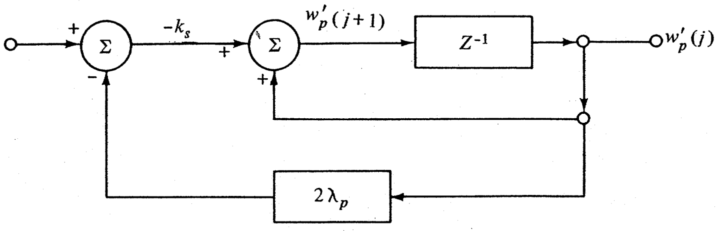

FIG. 6. One-dimensional feedback model (for the pth normal coordinate).

The stability of the one-dimensional flow graph is assured when the magnitude of the geometric ratio is less than one.

$$|r_p| < 1 \tag{24}$$

It may therefore be concluded that the multidimensional flow graph of Fig. 5 is stable iff

$$|r_p|_{\max} < 1 \tag{25}$$

The eigenvalues of $[\mathbf{\Phi}(x, x)]$ are such that $\lambda_p \geq 0$ for all p. Therefore, the only way that the stability condition (25) could be met is for

$$k_s < 0$$

and

$$|k_s\lambda_{\max}| < 1 \tag{26}$$

where $\lambda_{\max}$ is the maximum eigenvalue of $[\mathbf{\Phi}(x, x)]$. It follows that a necessary and sufficient condition for the stability of the steepest-descent adaptation process in the absence of gradient measurement noise[2] is that

$$-\frac{1}{\lambda_{\max}} < k_s < 0 \tag{27}$$

It should be observed from (23) and (27) that the rate of adaptation and stability can be controlled by setting k_s.

THE LMS ADAPTATION ALGORITHM

The purpose of the adaptation process is to find an exact or an approximate solution to the Wiener–Hoff equation (10). One way of finding the optimum weight vector is simply to solve (10). Although this solution is generally straightforward, it could present serious computational problems when the number of weights n is large and when input data rates are high. In addition to the necessity of inverting an $n \times n$ matrix, this method may require as many as $[n(n + 1)]/2$ autocorrelation and cross-correlation measurements to be made to obtain the elements of $\mathbf{\Phi}(x, x)$ and $\mathbf{\Phi}(x, d)$. Furthermore, this process needs to be continually repeated in most practical situations where the input-signal statistics may change slowly. No perfect solution of equation (10) is possible in practice because of the fact that an infinite statistical sample would be required to estimate perfectly the elements of the correlation matrices.

[2] This assumes that true values of the gradient are used each adaptation cycle. In practice, measured rather than exact values are used. Nevertheless, the same stability conditions apply, as will be discussed below.

A method for finding approximate solutions to (10) will be presented below. The accuracy of this method is limited by statistical sample size, since weight values are found that are based on finite-time measurements of input-data signals. This method does not require explicit measurements of correlation functions, nor does it require matrix inversion. It is the "LMS" algorithm,[1,2] based on the method of steepest descent. This algorithm does not even require squaring, averaging, or differentiation in order to make use of gradients of mean-square-error functions.

When using the LMS algorithm, changes in the weight vector are made along the direction of the estimated gradient vector. Accordingly,

$$\mathbf{W}(j+1) = \mathbf{W}(j) + k_s \hat{\nabla}[\bar{\epsilon}^2(j)] \tag{28}$$

where

$\mathbf{W}(j) \triangleq$ weight vector before adaptation
$\mathbf{W}(j+1) \triangleq$ weight vector after adaptation
$k_s \triangleq$ scalar constant controlling rate of convergence and stability ($k_s < 0$)
$\hat{\nabla}[\bar{\epsilon}^2(j)] \triangleq$ estimate of gradient of $E[\epsilon^2] = \bar{\epsilon}^2$ with respect to $\mathbf{W}$, with $\mathbf{W} = \mathbf{W}(j)$

One method for obtaining the estimated gradient of the mean-square-error function is to take the gradient of a single time sample of the squared error; that is,

$$\hat{\nabla}[\bar{\epsilon}^2(j)] = \nabla[\epsilon^2(j)] = 2\epsilon(j)\nabla[\epsilon(j)] \tag{29}$$

From equation (3),

$$\nabla[\epsilon(j)] = \nabla[d(j) - \mathbf{W}^T(j)\mathbf{X}(j)] = -\mathbf{X}(j)$$

Thus,

$$\hat{\nabla}[\bar{\epsilon}^2(j)] = -2\epsilon(j)\mathbf{X}(j) = -2[d(j) - \mathbf{W}^T(j)\mathbf{X}(j)]\mathbf{X}(j) \tag{30}$$

The gradient estimate of (30) is unbiased, as will be shown by the following argument: For a given weight vector $\mathbf{W}(j)$, the expected value of the gradient estimate is

$$\begin{aligned} E\hat{\nabla}[\bar{\epsilon}^2(j)] &= -2E\{\mathbf{X}(j)[d(j) - \mathbf{X}^T(j)\mathbf{W}(j)]\} \\ &= -2[\mathbf{\Phi}(x, d) - \mathbf{\Phi}(x, x)\mathbf{W}(j)] \end{aligned} \tag{31}$$

Comparing (8) and (31), we see that

$$E\{\hat{\nabla}[\bar{\epsilon}^2(j)]\} = \nabla[\bar{\epsilon}^2(j)]$$

and therefore for a given weight vector, the gradient estimate $\hat{\nabla}[\bar{\epsilon}^2(j)]$ is unbiased.

Using the gradient-estimation formula (30), the weight iteration rule, equation (28), becomes

$$\mathbf{W}(j+1) = \mathbf{W}(j) - 2k_s\epsilon(j)\mathbf{X}(j) \tag{32}$$

and the next weight vector is obtained by adding to the present weight vector the input vector scaled by the value of the error. This is the LMS algorithm.

This algorithm is directly usable as a weight-adaptation formula for digital systems. Figure 7A is a block-diagram representation of this equation in terms of one component $w_i(j)$ of the weight vector $\mathbf{W}(j)$. An equivalent differential equation which can be used in analog implementation of continuous systems [see Fig. 7B] is given by

$$\frac{d}{dt}\mathbf{W}(t) = -2k_s e(t)\mathbf{X}(t)$$

This equation can also be written as

$$\mathbf{W}(t) = -2k_s \int_0^t \epsilon(\xi)\mathbf{X}(\xi)\, d\xi = \mathbf{W}(0)$$

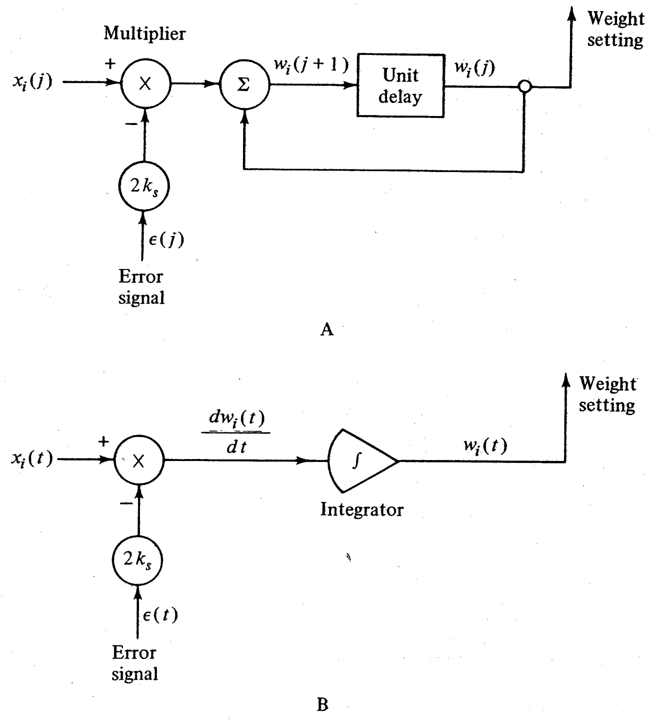

FIG. 7. Block diagram representation of LMS algorithm. A—Digital realization. B—Analog realization.

Figure 8 shows how circuitry of the type indicated in Fig. 7A or 7B might be incorporated into the implementation of the basic adaptive element of Fig. 2.

CONVERGENCE OF THE WEIGHT-VECTOR MEAN

For the purpose of the following discussion, assume that the time between successive iterations of the LMS algorithm is sufficiently long so that the sample input vectors $\mathbf{X}(j)$ and $\mathbf{X}(j+1)$ are uncorrelated. This assumption is common in the field of stochastic approximation.[16],[17]

Because the weight vector $\mathbf{W}(j)$ is a function *only* of the input vectors $\mathbf{X}(j-1), \mathbf{X}(j-2), \cdots, \mathbf{X}(0)$ [see equation (32)] and because the successive input vectors are uncorrelated, $\mathbf{W}(j)$ is independent of $\mathbf{X}(j)$. For stationary input processes meeting this condition, the expected value $E[\mathbf{W}(j)]$ of the weight vector after a large number of iterations can then be shown to converge to the Wiener solution given by (10). Taking the

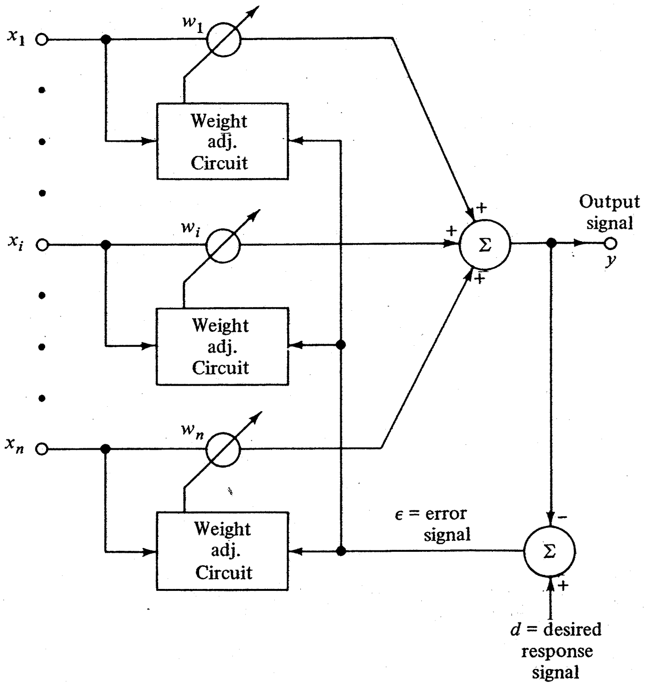

FIG. 8. Analog/digital implementation of LMS weight-adjustment algorithm.

expected value of both sides of (32), we obtain a difference equation in the expected value of the weight vector:

$$\begin{aligned} E[\mathbf{W}(j+1)] &= E[\mathbf{W}(j)] - 2k_s E\{\mathbf{X}(j)[d(j) - \mathbf{X}^T(j)\mathbf{W}(j)]\} \\ &= [\mathbf{I} + 2k_s \mathbf{\Phi}(x, x)]E[\mathbf{W}(j)] - 2k_s \mathbf{\Phi}(x, d) \end{aligned} \tag{33}$$

With an initial weight vector $\mathbf{W}(0)$, $j + 1$ iterations of equation (33) yield

$$\begin{aligned} E[\mathbf{W}(j+1)] = [\mathbf{I} + 2k_s \mathbf{\Phi}(x, x)]^{j+1}\mathbf{W}(0) \\ - 2k_s \sum_{i=0}^{j} [\mathbf{I} + 2k_s \mathbf{\Phi}(x, x)]^i \mathbf{\Phi}(x, d) \end{aligned} \tag{34}$$

Equation (34) may be put in diagonal form by using the normal-form expansion of the matrix $\mathbf{\Phi}(x, x)$; that is,

$$\mathbf{\Phi}(x, x) = \mathbf{Q}^{-1}\mathbf{\Lambda}\mathbf{Q} \tag{16}$$

The eigenvalues are all positive, since $\mathbf{\Phi}(x, x)$ is positive definite. Equation (34) may now be expressed as

$$\begin{aligned} E[\mathbf{W}(j+1)] &= [\mathbf{I} + 2k_s \mathbf{Q}^{-1}\mathbf{\Lambda}\mathbf{Q}]^{j+1}\mathbf{W}(0) \\ &\qquad - 2k_s \sum_{i=0}^{j} [\mathbf{I} + 2k_s \mathbf{Q}^{-1}\mathbf{\Lambda}\mathbf{Q}]^i \mathbf{\Phi}(x, d) \\ &= \mathbf{Q}^{-1}[\mathbf{I} + 2k_s\mathbf{\Lambda}]^{j+1}\mathbf{Q}\mathbf{W}(0) \\ &\qquad - 2k_s \mathbf{Q}^{-1} \sum_{i=0}^{j} [\mathbf{I} + 2k_s \mathbf{E}]^i \mathbf{Q}\mathbf{\Phi}(x, d) \end{aligned} \tag{35}$$

Consider the diagonal matrix $[\mathbf{I} + 2k_s\mathbf{\Lambda}]$. As long as its diagonal terms are all of magnitude less than unity,

$$\lim_{j\to\infty} [\mathbf{I} + 2k_s\mathbf{\Lambda}]^{j+1} \to 0$$

and the first term of (35) vanishes as the number of iterations increases. The second term in (35) generally converges to a nonzero limit. The summation factor

$$\sum_{i=0}^{j} [\mathbf{I} + 2k_s\mathbf{\Lambda}]^i$$

becomes

$$\lim_{j\to\infty} \sum_{i=0}^{j} [\mathbf{I} + 2k_s\mathbf{\Lambda}]^i = -\frac{1}{2k_s}\mathbf{\Lambda}^{-1}$$

where the formula for the sum of a geometric series has been used; that is,

$$\sum_{i=0}^{\infty} (1 + 2k_s\lambda_p)^i = \frac{1}{1 - (1 + 2k_s\lambda_p)} = \frac{-1}{2k_s\lambda_p}$$

Thus, in the limit, equation (35) becomes

$$\lim_{j\to\infty} E[\mathbf{W}(j + 1)] = \mathbf{Q}^{-1}\mathbf{\Lambda}\mathbf{Q}\mathbf{\Phi}(x, d) = \mathbf{\Phi}^{-1}(x, x)\mathbf{\Phi}(x, d)$$

Comparison of this result with equation (10) shows that as the number of iterations increases without limit, the expected value of the weight vector converges to the Wiener solution.

Convergence of the mean of the weight vector to the Wiener solution is insured if and only if the proportionality constant k_s is set within certain bounds. Since the diagonal terms of $[\mathbf{I} + 2k_s\mathbf{\Lambda}]$ must all have magnitude less than unity, and since all eigenvalues in $\mathbf{\Lambda}$ are positive, the bounds on k_s are given by

$$|1 + 2k_s\lambda_{\max}| < 1$$

or

$$\frac{-1}{\lambda_{\max}} < k_s < 0 \tag{36}$$

where $\lambda_{\max}$ is the maximum eigenvalue (27) of $\mathbf{\Phi}(x, x)$. Note that this convergence condition on k_s is identical to the stability condition (27) of the noiseless steepest-descent feedback model. This convergence condition can be related to the total input-signal power as follows: Since

$$\lambda_{\max} \leqq \text{trace } [\mathbf{\Phi}(x, x)] \tag{37}$$

where

$$\text{trace } [\mathbf{\Phi}(x, x)] \triangleq E[\mathbf{X}^T(j)\mathbf{X}(j)]$$
$$= \sum_{i=1}^{n} E[x_i^2] \triangleq \text{total input power}$$

it follows that satisfactory convergence can be obtained with

$$\frac{-1}{\sum_{i=1}^{n} E[x_i^2]} < k_s < 0$$

In practice, when slow, precise adaptation is desired, k_s is usually chosen such that

$$\frac{1}{\sum_{i=1}^{n} E[x_i^2]} \gg |k_s|$$

It is believed that the assumption of independent successive input samples used in the above convergence proof is overly restrictive. That is, convergence of the mean of the weight vector to the Wiener solution can be achieved under conditions of highly correlated input samples. Griffiths[18] has presented an experiment in which it is shown that adaptation using highly correlated successive samples converges to the Wiener solution, but leads to slightly higher steady-state mean-square error than does adaptation using statistically independent successive samples.

TIME CONSTANTS AND LEARNING CURVE WITH LMS ADAPTATION

As shown above, the weights undergo transients during adaptation. The transients consist of sums of exponentials with time constants given by

$$\tau_p = \frac{1}{2(-k_s)\lambda_p} \qquad p = 1, 2, \cdots, n \tag{39}$$

where λ_p is the pth eigenvalue of the input-signal correlation matrix $\mathbf{\Phi}(x, x)$.

In the special case when all eigenvalues are equal, all time constants are equal. Accordingly,

$$\tau = \frac{1}{2(-k_s)\lambda} \tag{40}$$

One very useful way to monitor the progress of an adaptive process is to plot or display its "learning curve." When mean-square error is the performance criterion being used, one can plot the expected mean-square error at each stage of the learning process as a function of the number of adaptation cycles. Since the underlying relaxation phenomenon that takes place in the weight values is of exponential nature, and since from equation (5) the mean-square error is a quadratic form in the weight values, the transients in the mean-square-error function must also be exponential in nature.

When all the time constants are equal, the mean-square-error learning curve is a pure exponential with a time constant

$$\tau_{\text{mse}} = \frac{\tau}{2} = \frac{1}{4(-k_s)\lambda} \tag{41}$$

The basic reason for this is that the square of an exponential function is an exponential with half the time constant. Estimation of the rate of adaptation is more complex when the eigenvalues are unequal.

When actual experimental learning curves are plotted, they are generally of the form of noisy exponentials because of the inherent noise in the adaptation process. The slower the adaptation, the smaller will be the amplitude of the noise apparent in the learning curve.

MISADJUSTMENT WITH LMS ADAPTATION

All adaptive or learning systems capable of adapting at real-time rates experience losses in performance because their system adjustments are based on statistical averages taken with limited sample sizes. The faster a system adapts, in general, the poorer will be its expected performance.

When the LMS algorithm is used with the basic adaptive element of Fig. 2, the expected level of mean-square error will exceed that of the Wiener optimum system whose weights are set in accordance with equation (10). The longer the time constants of adaptation, however, the closer the expected performance comes to the Wiener optimum performance. To get the Wiener performance—that is, to achieve the minimum mean-square error—one would have to know the input statistics a priori, or if (as is usual) these statistics are unknown, they would have to be measured with an arbitrarily large statistical sample.

When the LMS adaptation algorithm is used, an excess mean-square error therefore develops. A measure of the extent to which the adaptive system is misadjusted as compared to the Wiener optimum system is determined in a performance sense by the ratio of the excess mean-square error to the minimum mean-square error. This dimensionless measure of the loss in performance is defined as the "misadjustment" M. For LMS adaptation of the basic adaptive element, it is shown in Ref. [19] that

$$\text{Misadjustment } M = \frac{1}{2}\sum_{p=1}^{n}\frac{1}{\tau_p} \tag{42}$$

In deriving this formula, it is assumed that the Wiener input signal vectors are uncorrelated and that the adapting weight vector $\mathbf{W}(j)$ is close to the Wiener optimal $\mathbf{W}_{\text{LMS}}$.

The value of the misadjustment depends on the time constants (settling times) of the filter adjustment weights. In the special case when all the time constants are equal, *M is proportional to the number of weights and inversely proportional to the time constant;* that is,

$$M = \frac{n}{2\tau} = \frac{n}{4\tau_{\text{mse}}} \tag{43}$$

Although the foregoing results specifically apply to statistically stationary processes, the LMS algorithm can also be used with time-variable input processes. It is shown in Ref. [11] that under certain assumed conditions, the rate of adaptation is optimized when the excess mean-square error resulting from adapting too rapidly equals twice the excess mean-square error resulting from adapting too slowly.

APPLICATIONS OF ADAPTIVE FILTERS

In order to demonstrate how adaptive filters of the type described above can be used in practice, an application example will be presented. It is concerned with signal filtering and prediction. The behavior of adaptive filters in this application will be illustrated both qualitatively and quantitatively by the results of computer simulations.

In applying adaptive techniques to a practical systems problem, the key step lies in providing an appropriate desired-response signal for the adaptation process. In adaptive modeling applications, the desired response is generally available as the output of the unknown system to be modeled. This idea is illustrated in Fig. 1. The modeling situation is a simple one, but what can be done to provide the desired response for a real-time adaptive predictor? If the future were known, the predictor would not be needed. One way of coping with this situation is described below.

Let $x(j)$ be the discrete input signal applied to an adaptive tapped-delay-line predictor. Let the desired response be $d(j) = x(j + \delta)$. The objective is to predict the input signal δ time units into the future. The desired response signal is needed to get the error signal, which is required by the LMS algorithm of (32). An error signal can be obtained by delaying the predictor output $y(j)$ by δ time units, and subtracting $y(j - \delta)$ from the input $x(j)$. This would yield $\epsilon(j - \delta)$, but $\epsilon(j - \delta)$ could be just as useful as $\epsilon(j)$ for the purposes of adaptation. To see this, examine the LMS-algorithm equation,

$$\mathbf{W}(j + 1) = \mathbf{W}(j) - 2k_s\epsilon(j)\mathbf{X}(j) \qquad (32)$$

If $\epsilon(j - \delta)$ is available, the LMS algorithm can be utilized in the following form to realize the same ultimate weight-vector solution.

$$\begin{aligned}\mathbf{W}(j + 1 - \delta) &= \mathbf{W}(j - \delta) - 2k_s\epsilon(j - \delta)\mathbf{X}(j - \delta)\\ &= \mathbf{W}(j - \delta) - 2k_s[d(j - \delta) - \mathbf{W}^T(j - \delta)]\mathbf{X}(j - \delta) \qquad (44)\end{aligned}$$

In this form, the delayed desired response can be obtained, since $d(j - \delta) = x(j)$. An adaptive system based on (44) is shown in Fig. 9.

The output of the adaptive system in Fig. 9 is $y(j - \delta)$, an estimate of the present input $x(j)$. The predictive output $y(j)$, an estimate of $x(j + \delta)$, is developed by a nonadaptive filter, identical in form to the adaptive filter, whose weights are copied from those of the adaptive filter. In the steady state, after adaptive transients die out, the output $y(j)$ will be close to a best least-squares estimate of $x(j + \delta)$.

Suppose now that the input signal is $x(j) = s(j) + n(j)$, where $s(j)$ is "signal" and $n(j)$ is "noise." Let the "noise" be uncorrelated with the "signal." Let $E[n(j)] = 0$. Assume that $E[n(j)n(j + \alpha)] = 0$ for all values of α in the range $\delta \leq \alpha < \infty$, and for any other values of α. The system of Fig. 9 will produce a weight vector whose mean converges to

$$\begin{aligned} \mathbf{W}_{\mathrm{LMS}} &= \mathbf{\Phi}^{-1}(x, x)\mathbf{\Phi}(x, d) \\ &= \mathbf{\Phi}^{-1}(x, x)E[\mathbf{X}(j)d(j)] \\ &= \mathbf{\Phi}^{-1}(x, x)E\{[\mathbf{S}(j) + \mathbf{N}(j)][s(j + \delta) + n(j + \delta)]\} \\ &= \mathbf{\Phi}^{-1}(x, x)E[\mathbf{S}(j)s(j + \delta)] \end{aligned}$$

Thus, the adaptive process illustrated in Fig. 9 is suitable for prediction and for elimination of noise having the properties assumed above. In this case, adapting with the readily available desired response $d(j - \delta) = x(j)$ produces the same mean filter weight vector as would have been obtained if it were possible for the desired response to be $d(j - \delta) = s(j)$.

This technique could be used quite effectively for noise elimination alone. Suppose that the noise is "white" (uncorrelated). The system of Fig. 9 could be used to predict with $\delta = 1$, then the output signal could be delayed by one time unit to produce a good estimate of the current signal $s(j)$.

In a computer simulation of a 5-weight tapped-delay-line adaptive filter, a bandpass signal $s(j)$ and a white noise $n(j)$ were summed to form an input $x(j)$. The signal power was 1, and the noise power was ½. The signal bandwidth was equal to its center frequency. Figure 10 shows the signal $s(j)$ and the filter input $x(j) = s(j) + n(j)$. Figure 11 shows

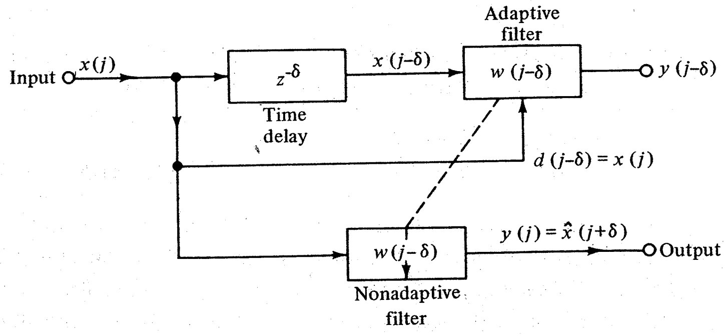

FIG. 9. An adaptive predictor.

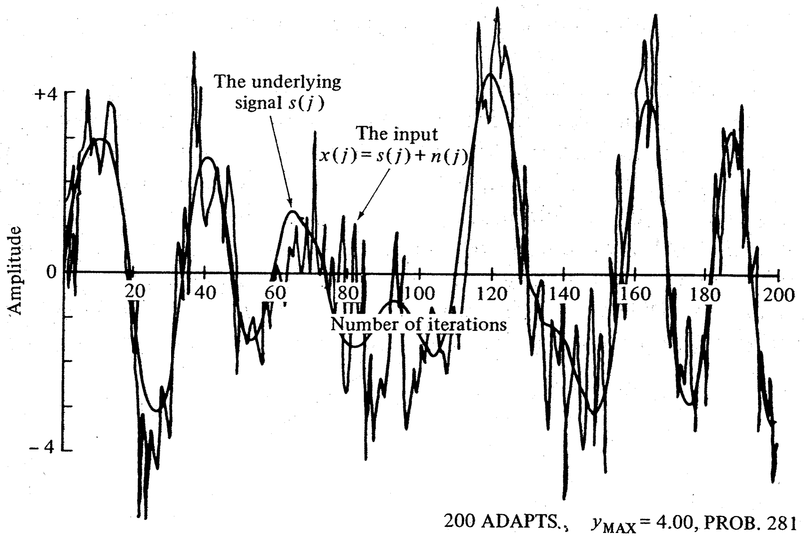

FIG. 10. Five-weight adaptive predictor: underlying signal and noisy input. (Number of sample patterns = number of iterations, j).

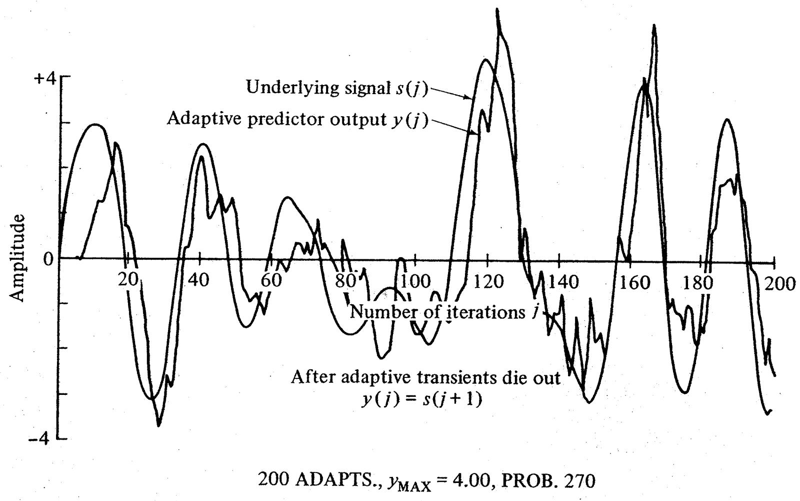

FIG. 11. Five-weight adaptive predictor: underlying signal and predictor output.

the filter output $y(j)$ plotted on the same scale as the underlying signal $s(j)$. The objective was to predict one sample time into the future, and to eliminate noise. Thus, after adapting transients died out, $y(j) = \hat{s}(j+1)$.

The initial transient phenomena can be observed in Fig. 10, as well as in Fig. 12, which shows learning curves for this simulation. Each point on the individual learning curve is a computed value of mean-square error, obtained by using equation (5), corresponding to the current weight vector that resulted from adaptation. The input statistics were known precisely, having been generated by the computer. The smooth learning curve is an average over an ensemble of 150 simulations, each starting with the same initial weight vector $[\mathbf{W}(0) = 0]$ and each having a different input signal derived from the same stochastic process.

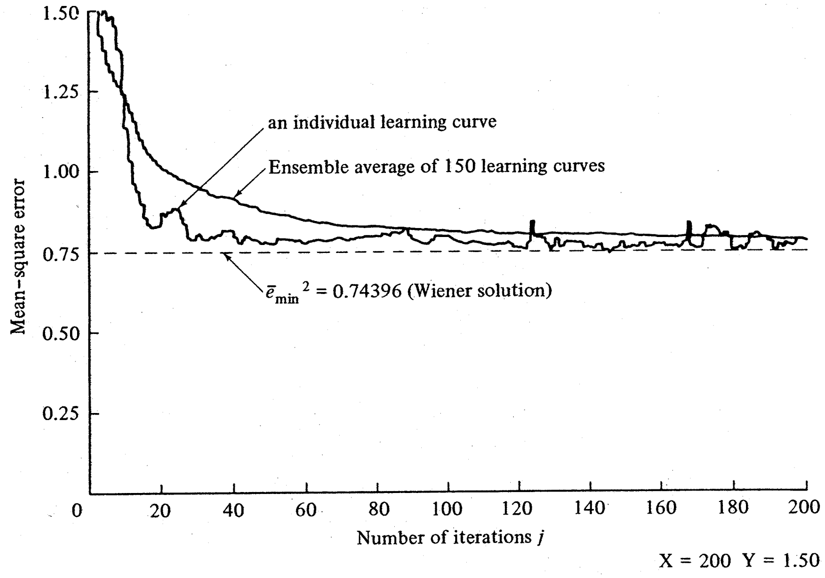

FIG. 12. Individual and ensemble learning curves.

Adaptive Predictor

Five weights
Initial weight values all zero
Constant $k_s = -0.0065$
Misadjustment $\begin{cases}\text{theoretical: } 4.87\% \\ \text{measured: } 5.40\%\end{cases}$

Input Statistics

Signal $s(j)$ $\begin{cases}\text{center freq.} = 0.03 \times \text{sampling freq.} \\ \text{bandwidth} = 0.03 \times \text{sampling freq.} \\ \text{variance} = 1\end{cases}$

Noise $n(j)$ $\{$ "white"; variance = 0.5

Input $x(j)$ $\begin{cases}\text{variance} = 1.5 \\ \text{eigenvalues of } \Phi(x, x) = 5.14, \\ \quad 0.853, 0.502, 0.500, 0.500\end{cases}$

The eigenvalues of $\mathbf{\Phi}(x, x)$ were known to be 5.14, 0.853, 0.502, 0.500, and 0.500. The time constants of the natural modes in the weight-adjustment system are obtained from equation (39). The constant k_s was set equal to -0.0065. The time constants are 15.0, 90.0, 152, 154, and 154 adaptations. Exponential relaxation in the weights causes exponential relaxation in the learning curve. The time constants in the learning curve are half those in the weights. The learning-curve time constants are therefore 7.5, 45, 76, 77, and 77 adaptations. Although the learning curve consists of a linear combination of exponential transients and is not a single exponential, an "eyeball" measurement of time constant of the ensemble-average learning curve shows this to be about 30 adaptations.

Using the set of known eigenvalues of $\mathbf{\Phi}(x, x)$, the misadjustment is calculated using (42). This gives a value $M = 4.87$ percent. The misadjustment was found to be 5.40 percent by direct measurement.

Although the time constants are not all equal, it is interesting to use formula (43), which relates the misadjustment M to the number of weights n, and to the time constant τ_{mse} of the learning curve.

$$M = \frac{n}{4\tau_{\text{mse}}} \tag{43}$$

If τ_{mse} is taken at the "eyeball" value of 30 adaptations,

$$M = \frac{5}{4(30)} = 4.15 \text{ percent}$$

This is a less accurate formula for general use, but it is a lot simpler and does not require detailed knowledge of the eigenvalues of $\mathbf{\Phi}(x, x)$.

The adaptive filter produced about 5 percent more mean-square error than the Wiener filter, but the Wiener filter could have been designed only with complete knowledge of the second-order statistics of the signal and the noise.

In general, formula (43) gives an approximate "rule of thumb" relating speed of adaptation to the number of weights. If a misadjustment of 5 percent is acceptable, then

$$\tau_{\text{mse}} = 5n$$

if a misadjustment of 10 percent is acceptable, then

$$\tau_{\text{mse}} = 2.5n$$

and so on.

The adaptive filtering techniques described here have also been applied to the problem of processing the outputs of the elements of an antenna array so as to provide a main directive lobe in a specified "look" direction while simultaneously rejecting unknown noises that are received by the array. Tapped delay lines, similar to that shown in Fig. 1, are used to filter the output of each element in the array. An array output is formed by summing the outputs of all weighted signals. The weights are adapted using the LMS algorithm presented above. The processor adaptively rejects incident noises whose directions are different from the desired look direction by forming appropriate nulls in the antenna directivity pattern. The results of this application of adaptive-filtering techniques are described in detail in Ref. [1].

Some of the best applications of adaptive filters will doubtless be made to the filtering of nonstationary inputs. Means of describing the behavior of adaptive systems with nonstationary inputs is for the most part an uncharted area and is currently a subject of intensive research.

REFERENCES

[1] Widrow, B., P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive antenna systems," *Proc. IEEE*, vol. 55, pp. 2143–2159, Dec. 1967.

[2] Widrow, B., and M. E. Hoff, Jr., "Adaptive switching circuits," *1960 IRE WESCON Conv. Record*, pt. 4, pp. 96–104. (First presentation of LMS algorithm; not called this in the paper, however.)

[3] Nilsson, N. G., *Learning Machines*. New York: McGraw-Hill, 1965.

[4] Widrow, B., and F. W. Smith, "Pattern-recognizing control systems," *1963 Computer and Information Sciences (COINS) Symp. Proc.* Washington D.C.: Spartan, 1964.

[5] Smith, F. W., "Design of quasi-optimal minimum-time controllers," *IEEE Trans. Automatic Control*, vol. AC-11, pp. 71–77, Jan. 1966.

[6] Southwell, R. V., *Relaxation Methods in Engineering Science*. New York: Oxford, 1940.

[7] Wilde, D. J., *Optimum Seeking Methods*. Englewood Cliffs, N.J.: Prentice-Hall, 1964.

[8] Mason, S. J., "Feedback theory: further properties of signal flow graphs," *Proc. IRE*, vol. 44, pp. 920–926, July 1956.

[9] Mason, J., and H. J. Zimmerman, *Electronic Circuits, Signals, and Systems*. New York: Wiley, 1960.

[10] Widrow, B., "Adaptive sampled-data systems—a statistical theory of adaptation," *1959 IRE WESCON Conv. Record*, pt. 4, pp. 74–85, 1959.

[11] Widrow, B., "Adaptive sampled-data systems," *Proc. First Intern. Cong. Intern. Federation of Automatic Control*, Moscow, 1960.

[12] Ragazzini, F. R., and G. F. Franklin, *Sampled-data Control Systems*. New York: McGraw-Hill, 1958.

[13] Jury, E. I., *Sampled-Data Control Systems*. New York: Wiley, 1958.

[14] Tou, J. T., *Digital and Sampled-data Control Systems*. New York: McGraw-Hill, 1959.

[15] Freeman, H., *Discrete-Time Systems*. New York: Wiley, 1965.

[16] Robbins, H., and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, pp. 400–407, 1951.

[17] Dvorotzky, A., "On stochistastic approximation," *Proc. Third Berkeley Symp. Math. Statist. and Probability*, J. Neyman (ed.), University of California Press, Berkeley, Calif., 1956, pp. 39–55.

[18] Griffiths, L. J., "Signal extraction using real-time adaptation of a linear multichannel filter," Ph.D. dissertation, Stanford University, Dec. 1967.

[19] Widrow, B., "Adaptive filters I: fundamentals," Rept. SEL-66-126 (TR 6764-6), Stanford Electronics Laboratories, Stanford, Calif., Dec. 1966.