SU-SEL-66-126

# ADAPTIVE FILTERS I: FUNDAMENTALS

by

Bernard Widrow

December 1966

Technical Report No. 6764-6

Prepared under

U.S. Army Materiel Command
NIKE-X Project Office
Redstone Arsenal, Alabama
Contract No. DA-01-021 AMC-90015(Y)

and also under

Naval Ship Systems Command
Department of the Navy
Contract NOBsr-95038

Systems Theory Laboratory
Stanford Electronics Laboratories
Stanford University Stanford, California

# ABSTRACT

An adaptive filter consisting of a tapped delay line, variable weights, a signal summer, and a means for automatically adjusting the weights (the LMS algorithm, based on the method of steepest descent) has been presented and analysed. Feedback is used to control the "design" (the weight adjustments) of the system rather than to control the signals being filtered.

A mean-square-error performance criterion is used. In many cases, the mean-square error is a quadratic function of the filter adjustments. In such cases, recursive gradient optimization by the method of steepest descent produces exponential-like transients in the adjustment parameters during adaptation. A linear theory of adaptation based on state-space methods is developed which relates the stability, rate of adaptation, and expected filter performance to signal statistics and to parameters of the adaptation algorithm. Simplifications in analysis have been realized by expressing adaptation transient phenomena in terms of the normal coordinates of the system. The adaptive time constants are given by

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

where $\lambda_p$ is the $p^{th}$ eigenvalue of the input-signal correlation matrix $[\Phi(x,x)]$ and $k_s$ is a design parameter that determines the rate at which adaptation is accomplished.

The LMS algorithm, which is simple and practical, uses measured gradient estimates that are "noisy" but unbiased. Noise in the filter adjustments causes loss in system performance. This loss can be minimized by slow adaptation. The per unit amount by which the mean-square error of an adaptive filter exceeds that of an optimal-least-squares filter is derived as

$$M = \frac{1}{2} \sum_{p=1}^{n} \frac{1}{\tau_p} \quad .$$

Applications of the principles of adaptive filtering have been made in the laboratory to automatic control, automatic modeling, prediction,

noise filtering, and pattern recognition; also to equalizers for communication channels and to antenna arrays capable of developing a high degree of directivity for noise rejection. Adaptive filters and adaptive systems in general will find their strongest applications in situations where inputs are nonstationary and where little or no *a priori* statistical information about input signals is available.

## CONTENTS

## CONTENTS (Cont)

# ILLUSTRATIONS

# ACKNOWLEDGMENTS

The work reported here had its beginnings while the author was a faculty member at MIT from 1956 to 1959. The work was continued when the author went to Stanford University in 1959, and the work was then supported under a Tri-Service contract. More recently the work has been supported by Army and Navy contracts as noted on the title page.

Special thanks go to P. E. Mantey for many useful technical interchanges.

## 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 even predict the next signal from moment to moment.

This paper presents an approach to filtering of statistically stationary or nonstationary signals, using an adaptive filter that is in some sense self-designing (really self-optimizing). This approach does not require complete a priori knowledge of the statistics of the signals to be filtered. Thus the method has novel and significant applications in the fields of noise filtering for communication channels, automatic control, pattern recognition, adaptive antenna design, and many others.

### Previous Developments in Filter Design

Pioneering work in the field of filter design was done by Norbert Wiener [1] more than two decades ago. His efforts made possible the design of linear filters for noise elimination and for predicting and smoothing statistically stationary signals. Wiener filters are simple to implement, and the design is optimal in the least-squares sense.

More recent work by Kalman and Bucy [2] has led to the design of optimal time-variable linear filters for nonstationary signals. For such signals, Kalman-Bucy filters can deliver substantially better performance than Wiener filters.

Both the Wiener and the Kalman-Bucy filters must be designed on the basis of a priori information or assumptions about the statistics of the signals to be processed. These filters are optimal in practice only when the statistical characteristics of the actual input signals match the a priori information on the basis of which the filters were designed. When the a priori information is not known perfectly, these filters will not deliver optimal performance.

## Advantages of the Adaptive Filter

The adaptive filter described in the present paper bases its own "design" (its internal adjustment settings) upon estimated (measured) statistical characteristics of input 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 which automatically updates the 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.

The form of adaptive filter described in this paper is almost as simple to implement as the Wiener filter, and should perform nearly as well as the Kalman-Bucy filter (given perfect a priori information.) Under circumstances in which the a priori information is not perfectly known, it is quite possible that the performance of an adaptive filter could exceed that of either a Wiener or a Kalman-Bucy filter.

When almost no a priori information is available, the use of an adaptive filter may be the only reasonable possibility.

## AN ADAPTIVE FILTER

Adaptive filters can be continuous or discrete. One particular form of adaptive filter to be considered is a discrete (sampled-data) type. It 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 automatically adjust the weights. 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 most of the components of an adaptive filter, which is 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 in the tapped delay line. 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 the adaptive filter. Nevertheless, in many applications, such as those mentioned above, the adaptive process is useful. An example illustrating the use of the desired-response signal is that

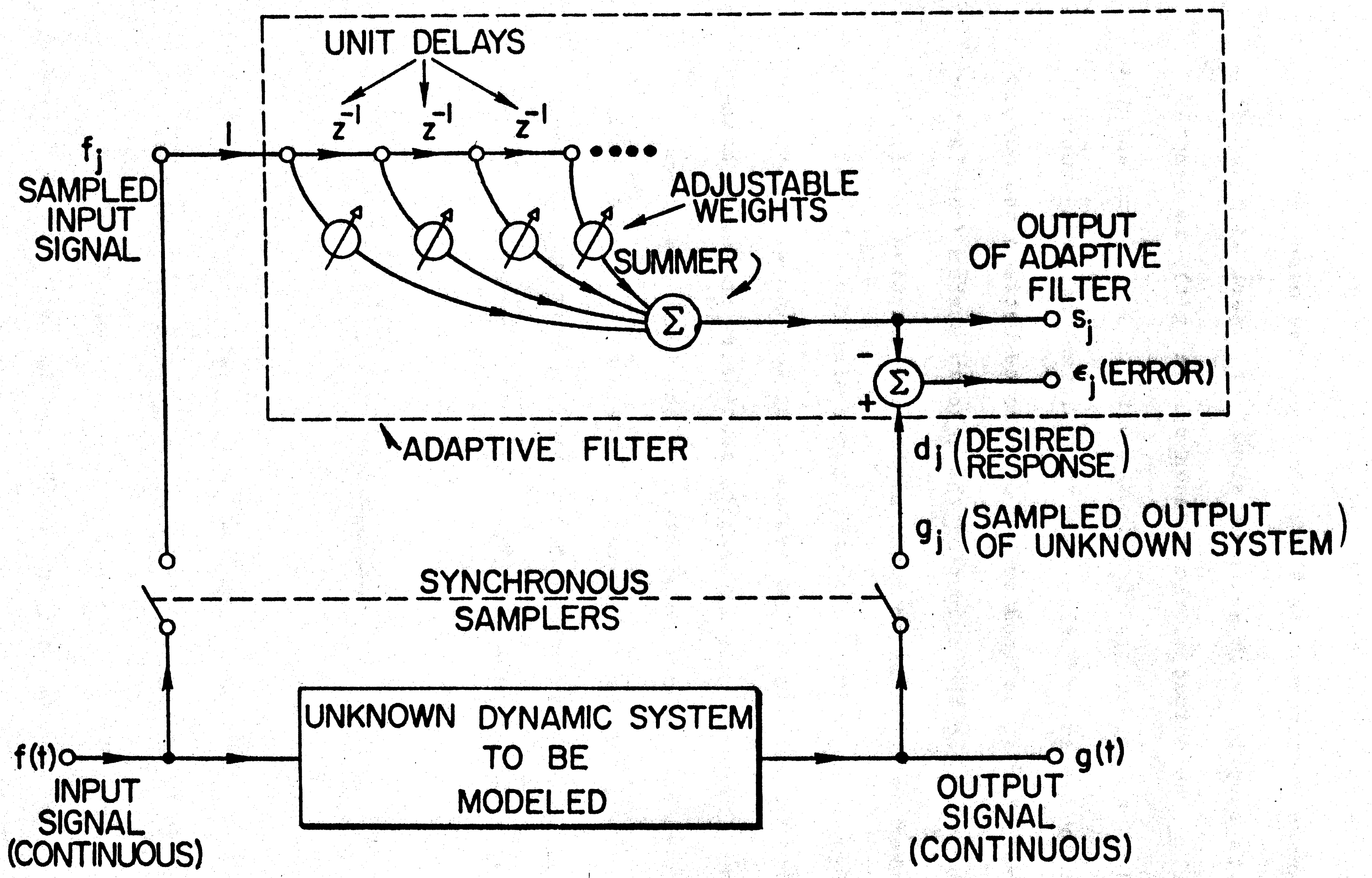


Fig. 1. Modeling an unknown system by a discrete adaptive filter.

shown in Fig. 1. Here a continuous signal f(t) is indicated as an input to an unknown system that is 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 $s_j$ of the adaptive model. The latter 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).

The analyses to be presented in this paper will show that if the input and output signals of the system being modeled are statistically stationary, the error signal is also stationary and 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. It can be seen that this combinatorial system is imbedded in the adaptive filter shown in Fig. 1, and indeed the combinatorial adaptive system is the

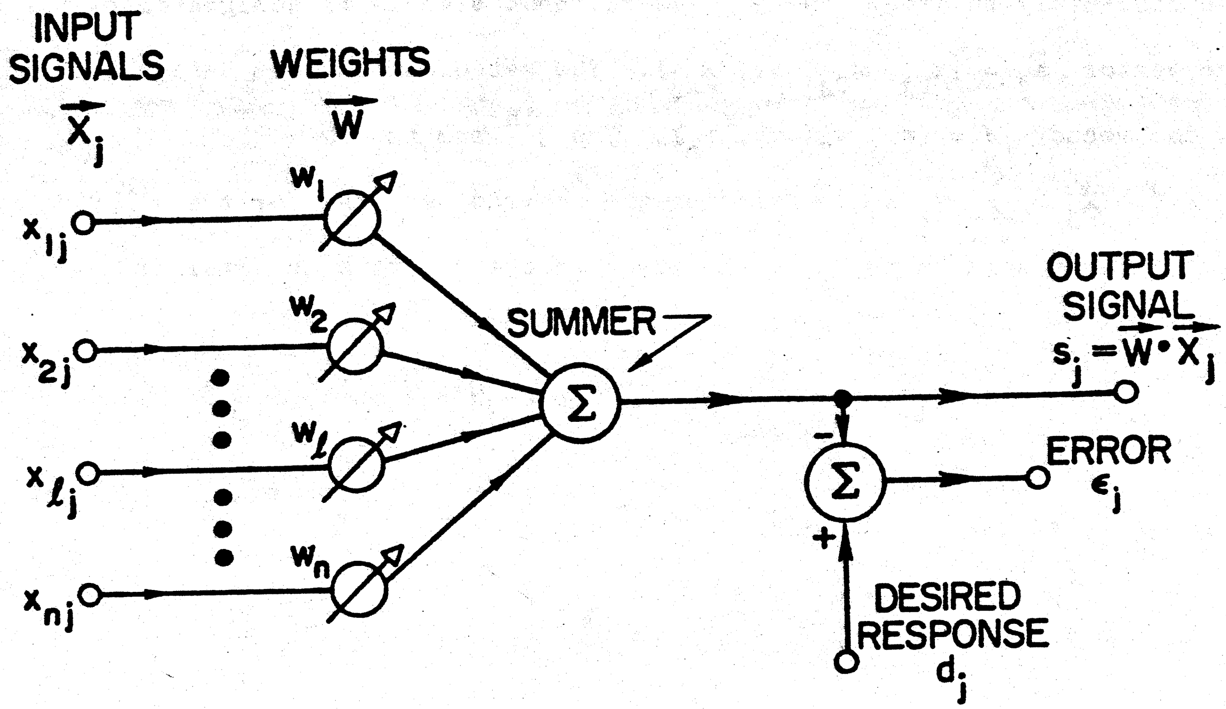


Fig. 2. Adaptive linear combinatorial system.

most significant portion of the adaptive filter.[1] The analysis of adaptive filter performance will be based on a study of the system of Fig. 2, assuming stationary input signals.

A set of 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 $j^{th}$ set of input signals is designated by the vector $\vec{X}_j = (x_{1j}, x_{2j}, \ldots, x_{nj})$. The set of weights is designated by the vector $\vec{W} = (w_1, w_2, \ldots, w_n)$. The $j^{th}$ sum is $s_j = \vec{W} \cdot \vec{X}_j = \sum_{\ell=1}^{n} w_\ell \, x_{\ell j}$. Denoting the desired response for the $j^{th}$ set of input signals as $d_j$, the error at the $j^{th}$ time interval is given by

$$\epsilon_j = d_j - \sum_{\ell=1}^{n} w_\ell \, x_{\ell j} = d_j - \vec{W} \bullet \vec{X}_j \; . \tag{1}$$

From Eq. (1), the square of the $j^{th}$ error is

$$\epsilon_j^2 = d_j^2 - 2 \sum_{\ell=1}^{n} w_\ell \, x_{\ell j} \, d_j + \sum_{\ell=1}^{n} \sum_{m=1}^{n} w_\ell \, w_m \, x_{\ell j} \, x_{mj} \; . \tag{2}$$

Note that the product of sums is written as a double sum and that two separate dummy summation indices are required. From this point onward, all sums will be taken from 1 to n unless otherwise noted.

The expected value of the error squared (the mean-square error) is given by

$$\overline{\epsilon_j^2} = \overline{d^2} - 2 \sum_{\ell} w_\ell \, \phi(x_\ell, d) + \sum_{\ell} \sum_{m} w_\ell \, w_m \, \phi(x_\ell, x_m) \; , \tag{3}$$

[1] This combinatorial system can also be connected to the elements of a phased array antenna to make an adaptive antenna [3]; to a quantizer to form an adaptive threshold element ("Adaline" [4] or TLU [5]) for use in adaptive logic and pattern-recognition systems; or it can be used as the adaptive portion of certain learning control systems [6],[7].

where correlations are defined as

$$\phi(x_\ell,\ d) \triangleq \overline{x_{\ell j}\ d_j}, \quad \text{and} \quad \phi(x_\ell,\ x_m) \triangleq \overline{x_{\ell j}\ x_{mj}}\ . \qquad (4)$$

The expectation is taken over j, the input-vector index number.

Equation (3) may be expressed in matrix form in the following way:

$$\overline{\epsilon_j^2} = \overline{d_j^2} - 2\ \underline{\phi(x,d)}\ W] + \underline{W}\ [\phi(x,x)\ W]\ , \qquad (5)$$

where $W]$ and $\underline{W}$ are the column and row vectors of weights, respectively, and

$$\underline{\phi(x,d)} \triangleq \underline{\phi(x_1,d) \qquad \phi(x_2,d)\ \ldots\ \phi(x_\ell,d)\ \ldots\ \phi(x_n,d)}\ , \qquad (6)$$

$$[\phi(x,x)] \triangleq \begin{bmatrix} \phi(x_1,x_1) & \phi(x_1,x_2) & \cdots & \\ \phi(x_2,x_1) & & \cdots & \vdots \\ \vdots & & & \phi(x_n,x_n) \end{bmatrix}\ . \qquad (7)$$

Equation (6) defines a crosscorrelation vector, an array of crosscorrelations between the individual input signal components and the desired-response signal. Equation (7) defines the correlation matrix of the input-signal components. This is the covariance matrix of the input-signal components when all of their means are zero.

It may be observed from Eq. (3) that for stationary input signals, the mean-square error is precisely a second-order function of the weights, the w's. 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 [8], [9] 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 behavior of the adaptive process when the inputs are nonstationary will be approached by studying first this behavior with stationary inputs. It will be assumed that the input and desired-response signals are stationary unless otherwise noted.

## THE GRADIENT AND THE LEAST-SQUARES-OPTIMAL WEIGHTS

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 Eq. (3). The $i^{th}$ gradient component is

$$\frac{\partial \overline{\epsilon_j^2}}{\partial w_i} = -2\phi\,(x_i,d) + 2 \sum_{\ell} w_\ell\, \phi\,(x_i,x_\ell) \;. \tag{8}$$

The entire gradient vector may therefore be written as

$$\underbracket{\nabla\overline{\epsilon_j^2}} = -2\,\underbracket{\phi(x,d)} + 2\,\underbracket{W}\,[\,\phi(x,x)\,] \;. \tag{9}$$

To find the "optimal" set of weights, $W_{LMS}$, that minimizes $\overline{\epsilon_j^2}$, set $\nabla\overline{\epsilon_j^2} = 0$. Accordingly,

$$\underbracket{\phi(x,d)} = \underbracket{W_{LMS}}\,[\,\phi(x,x)\,] \;, \tag{10a}$$

$$\underbracket{W_{LMS}} = \underbracket{\phi\,(x,d)}\;[\,\phi(x,x)\,]^{-1} \;. \tag{10b}$$

The least-mean-square (LMS) error is achieved by choosing the optimal weight vector given by Eq. (10b). This equation may be recognized as a matrix version of the Wiener-Hopf equation [1].

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

$$\overline{\epsilon^2_{\text{min}}} = \overline{d^2_j} - \underline{\phi(x,d)}\; W_{LMS}\Big] \;. \tag{11}$$

## THE METHOD OF STEEPEST DESCENT; A FEEDBACK MODEL

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 or initial values for the weights. The gradient vector is measured at the point on the performance surface corresponding to these initial weights. The next guess is then obtained from the present guess by making a change in the weight vector in the direction of the negative of the gradient vector--i.e., in the opposite direction to 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 3b. 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

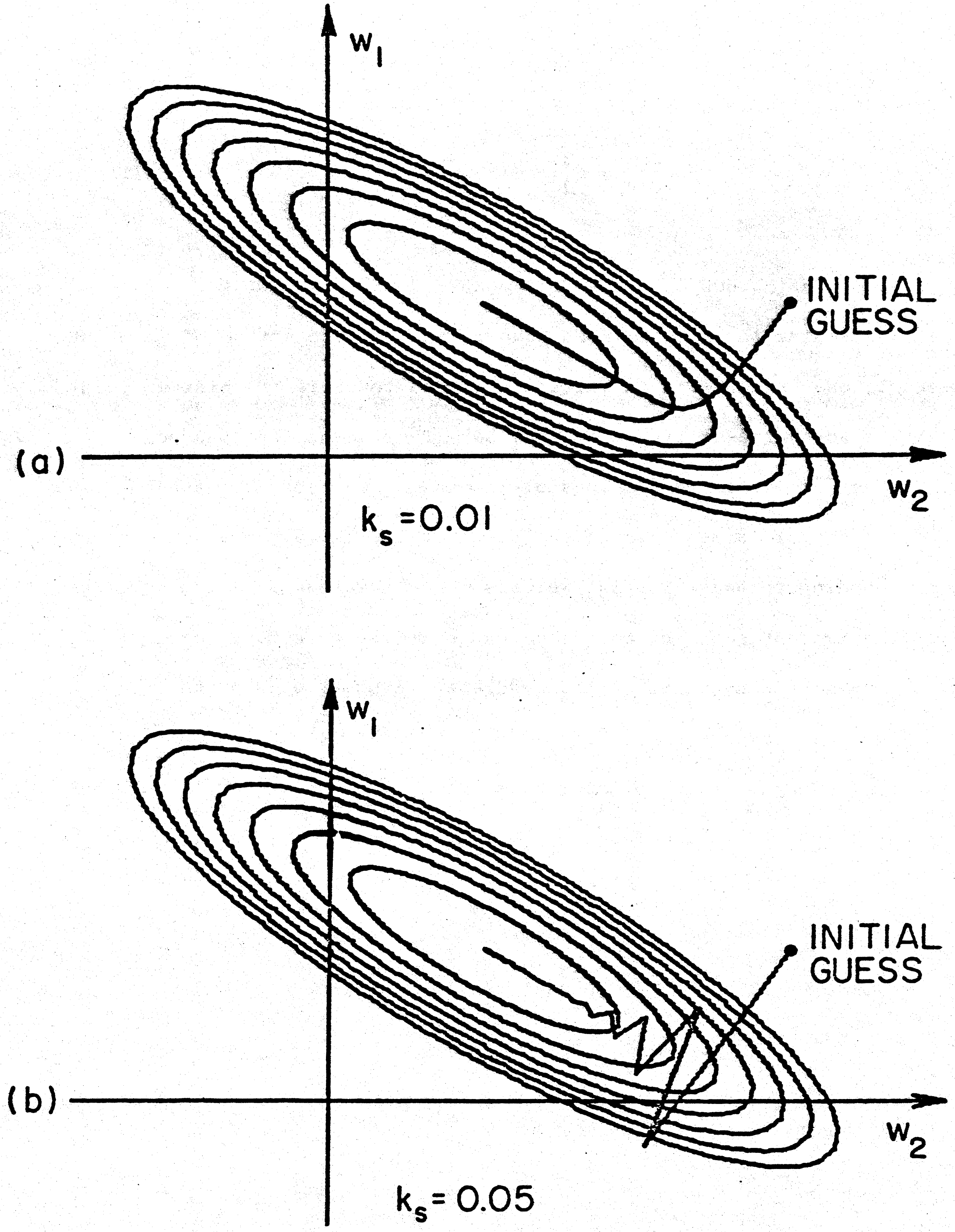


Fig. 3. Illustration of method of steepest descent: (a) overdamped; (b) underdamped.

begins. It will be shown later that the weights undergo geometric (discrete exponential) transients in relaxing toward the surface minimum. "Overdamping" is illustrated in Fig. 3a, while "underdamping" is illustrated in Fig. 3b.

In the procedure described here, it will be assumed that the weight vector is changed after the incidence of each new input-signal vector $\underline{X_j}, \underline{X_{j+1}}, \ldots .$ Therefore the weight vectors will be correspondingly indexed--viz., $\underline{W_j}, \underline{W_{j+1}}, \ldots .$ If each change in the weight vector is made proportional to the error-surface gradient vector, the method of steepest descent can be described by the following relations:

$$\underline{W_{j+1}} = \underline{W_j} + k_s \, \underline{\nabla \overline{\epsilon_j^2}} \; . \tag{12}$$

This equation states that the "next guess" equals the "present guess" plus the gradient vector multiplied by a constant $k_s$. But where did the "present guess" come from? It was calculated during the previous iteration cycle as the "next guess." That is,

$$\underline{(W_j)}_{\substack{\text{present}\\ \text{cycle}}} = \underline{(W_{j+1})}_{\substack{\text{previous}\\ \text{cycle}}}$$

or

$$\underline{W_j} = D \, \underline{W_{j+1}} \tag{13}$$

The operator D is a time-domain delay of one iteration cycle. The method of steepest descent is characterized by Eqs. (12) and (13). An expression for $\underline{\nabla \overline{\epsilon_j^2}}$ can be obtained by using Eq. (9) with suitable indexing.

$$\underline{\nabla \overline{\epsilon_j^2}} = -2 \, \underline{\phi(x,d)} + 2 \, \underline{W_j} [\Phi(x,x)] \; . \tag{14}$$

From this the gradient vector $\underline{\overline{\nabla\epsilon_j^2}}$ is to be interpreted as the gradient of the expected error-squared function when the weight vector is $\underline{W_j}$ .

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.

The analysis of steepest-descent adaptation is facilitated by making use of the familiar feedback flow graph [10],[11], used by control and communications engineers, in a multidimensional sense to express relations (12), (13), and (14) in an equivalent manner. A feedback model is highly appropriate, since in a real sense the gradient is like the "error" signal in an n-dimensional servomechanism which controls the adjustment or design of the adaptive filter. The bigger the gradient, the greater is the required weight-vector correction; when the gradient is zero, make no correction, and turn off the actuator since the "error" in the weight settings is zero. This form of feedback has been called "*performance feedback*" by this author in previous papers [12], [13].

### Flow-Graph Model

A flow graph incorporating relations (12), (13), and (14) is shown in Fig. 4. The "signals" at the nodes are indicated by row 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

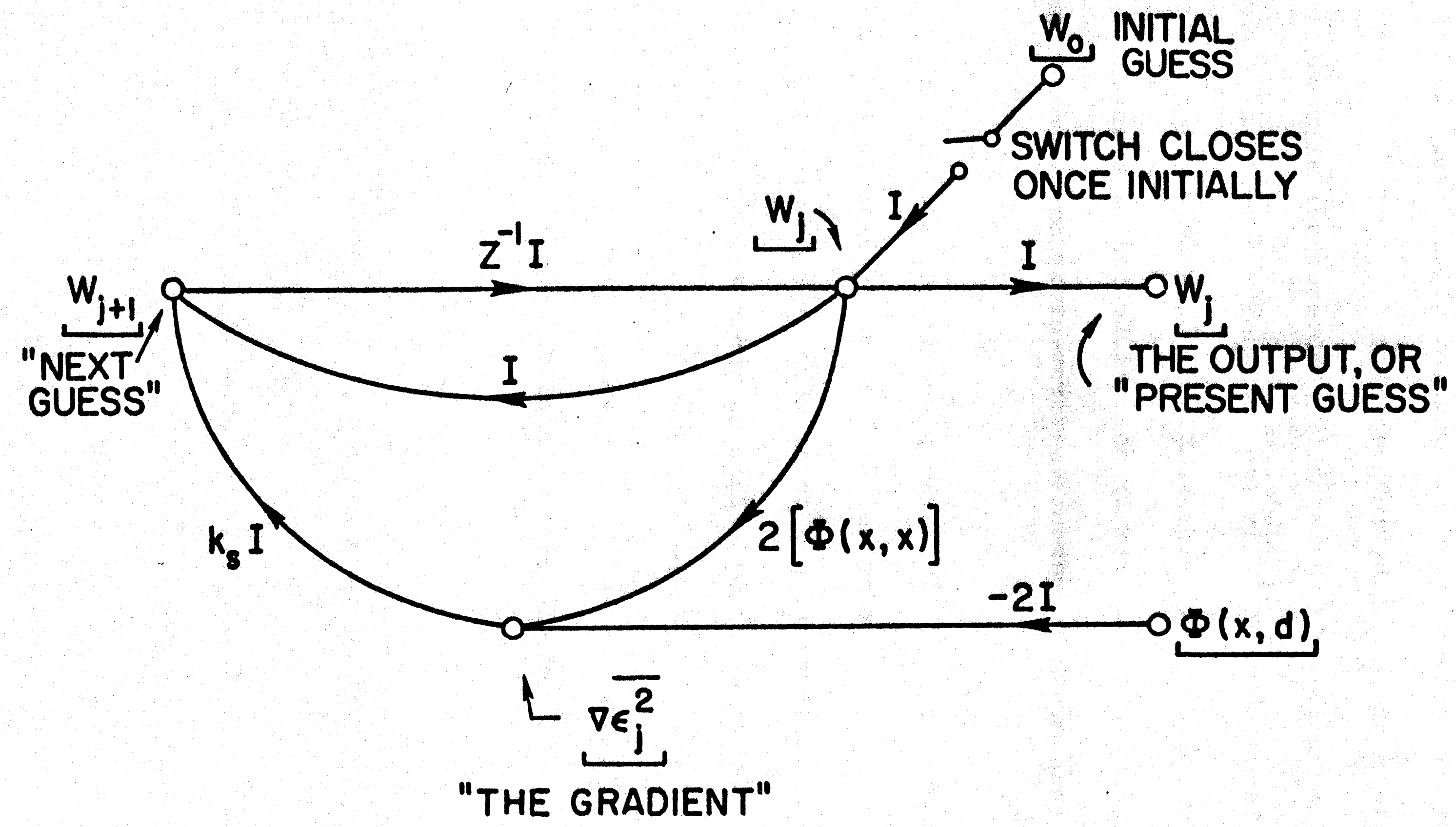


Fig. 4. Flow-graph model of method of steepest descent.

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 represented by row vectors.

In the flow graph of Fig. 4, the symbol I represents a unit matrix transfer function. The symbol $Z^{-1}$ is the "frequency domain" or Z-transform [14], [15], [16], [17] representation of a delay of one iteration cycle; $Z^{-1}\ I$ is the matrix transfer function of a unit delay branch, etc. This graph represents a first-order multidimensional sampled-data system.

Transient phenomena in $\underline{W_j}$ will take place in the flow-graph model exactly as they will in the actual hill-climbing process if the flow graph is initially quiescent, and if at zero time the initial guess is injected into the $\underline{W_j}$ node. This is indicated by the once-only closure of the switch at the start of the zeroth cycle. 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 $\underline{W_j}$ .

If the flow graph is stable, the hill-climbing process is stable. The steady-state conditions in the graph represent the steady-state values of the weight components. These conditions can be determined by inspection of the graph. The signal at the node labeled "the gradient" will be zero in steady state. Under this condition, $\underline{W_{j+1}}$ will be equal to $\underline{W_j}$ , indicating no change in $\underline{W_j}$ , since the signal flowing through the branch labeled $k_s I$ will be zero. In order for this to be so, the following must be true:

$$2\ \underline{W_j}\ [\Phi(x,x)] = 2\ \underline{\phi(x,d)}\ .$$

If $\underline{W_j}$ is set equal to $\underline{W_{LMS}}$, this expression corresponds to equation (10a), verifying that in equilibrium the flow graph (and the actual hill-climbing process) produces the LMS-optimal weights.

Each branch in the flow graph of Fig. 4 has a diagonal-matrix transfer function except for the branch labeled $2[\Phi(x,x)]$. In general, this latter branch matrix will have finite off-diagonal elements. As a result, transients will crosscouple 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 Flow Graph

The first step in the analysis of transients is to diagonalize the flow graph. To do this, return to the original expression for mean-square error given by Eq. (5).

$$\overline{\epsilon_j^2} = \overline{d_j^2} - 2\,\underline{\Phi(x,d)}\,W\Big] + \underline{W}\,[\Phi(x,x)]\,W\Big] \; . \qquad (5)$$

Using (5), (10b), and (11), the mean-square error may be expressed in the following way:

$$\overline{\epsilon_j^2} = \overline{\epsilon_{min}^2} + \underline{(W - W_{LMS})}\,[\Phi(x,x)]\,(W - W_{LMS})\Big] \; . \qquad (15)$$

The $[\Phi(x,x)]$ matrix is real, symmetric, and of a positive semidefinite quadratic form, since $\overline{\epsilon_j^2} \geq 0$.

The characteristic equation of the $[\Phi(x,x)]$ matrix is

$$|\Phi(x,x) - \lambda I| = 0 \; . \qquad (16)$$

The eigenvalues of $[\Phi(x,x)]$ are

$$\lambda_1, \lambda_2, \ldots \lambda_p, \ldots \lambda_n \; . \qquad (17)$$

Assume that they are distinct. The $p^{th}$ eigenvector $\underline{q_p}$ is then determined by

$$\underbrace{q_p}\,[\Phi(x,x)] = \lambda_p \underbrace{q_p} \,. \tag{18}$$

Normalize each eigenvector to have unit length. A square matrix of normalized eigenvectors, the <u>normalized modal matrix</u>, is given by

$$[Q] \triangleq \begin{bmatrix} q_1 \\ \vdots \\ q_p \\ \vdots \\ q_n \end{bmatrix} \tag{19}$$

All eigenvectors of the symmetric matrix $[\Phi(x,x)]$ are mutually orthogonal, and since all rows of $[Q]$ are normalized, the matrix $[Q]$ is <u>orthonormal</u>. Generalizing on Eq. (18) yields

$$[Q]\,[\Phi(x,x)] = [\Lambda]\,[Q] \,, \tag{20}$$

where the matrix $[\Lambda]$ is the diagonal eigenvalue matrix.

$$[\Lambda] \triangleq \begin{bmatrix} \lambda_1 & & & & & \\ & \lambda_2 & & & & \\ & & \ddots & & & \\ & & & \lambda_p & & \\ & & & & \ddots & \\ & & & & & \lambda_n \end{bmatrix} . \tag{21}$$

Premultiplying both sides of (20) with $[Q]^{-1}$ allows the $[\Phi(x,x)]$ matrix to be expressed as

$$[\Phi(x,x)] = [Q]^{-1}\,[\Lambda]\,[Q] \,. \tag{22}$$

Even when the eigenvalues of $[\Phi(x,x)]$ are not all distinct, this matrix can still be expressed as in (22). Then a more elaborate procedure [18], [19] than the one above is needed to find [Q].

Since the modal matrix [Q] is orthonormal,

$$[Q]^{-1} = [Q]^T \, , \tag{23}$$

and therefore

$$[\phi(x,x)] = [Q]^T[\Lambda]\,[Q] \quad . \tag{24}$$

Now substitute relation (24) into Eq. (15). This yields

$$\overline{\epsilon_j^2} = \overline{\epsilon_{\text{min}}^2} + \underbrace{(W - W_{LMS})}\,[Q]^T[\Lambda][Q]\,(W - W_{LMS})\Big] \, . \tag{25}$$

A new set of coordinates may be defined as follows:

$$\begin{aligned} \underbrace{W}\,[Q]^T &\triangleq \underbrace{W'} \, , \\ \underbrace{(W - W_{LMS})}\,[Q]^T &\triangleq \underbrace{(W' - W'_{LMS})} \\ [Q]\,W\Big] &\triangleq W'\Big] \, , \\ [Q]\,(W - W_{LMS})\Big] &\triangleq (W' - W'_{LMS})\Big] \quad . \end{aligned} \tag{26}$$

The mean-square error may now be expressed in terms of the primed coordinates as

$$\overline{\epsilon_j^2} = \overline{\epsilon_{\text{min}}^2} + \underbrace{(W' - W'_{LMS})}\,[\Lambda]\,(W' - W'_{LMS})\Big] \, . \tag{27}$$

Since [Λ] is a diagonal matrix, Eq. (27) expresses the mean-square error in diagonal canonical form [20], [21], [22] in the primed coordinates (normal coordinates). The primed coordinates are the <u>principal</u> <u>axes</u> of the quadratic mean-square-error surface. The primed coordinates are an orthogonal set, rotated from the original set of coordinates by the linear transformation $[Q]^T$, as represented by Eqs. (26).

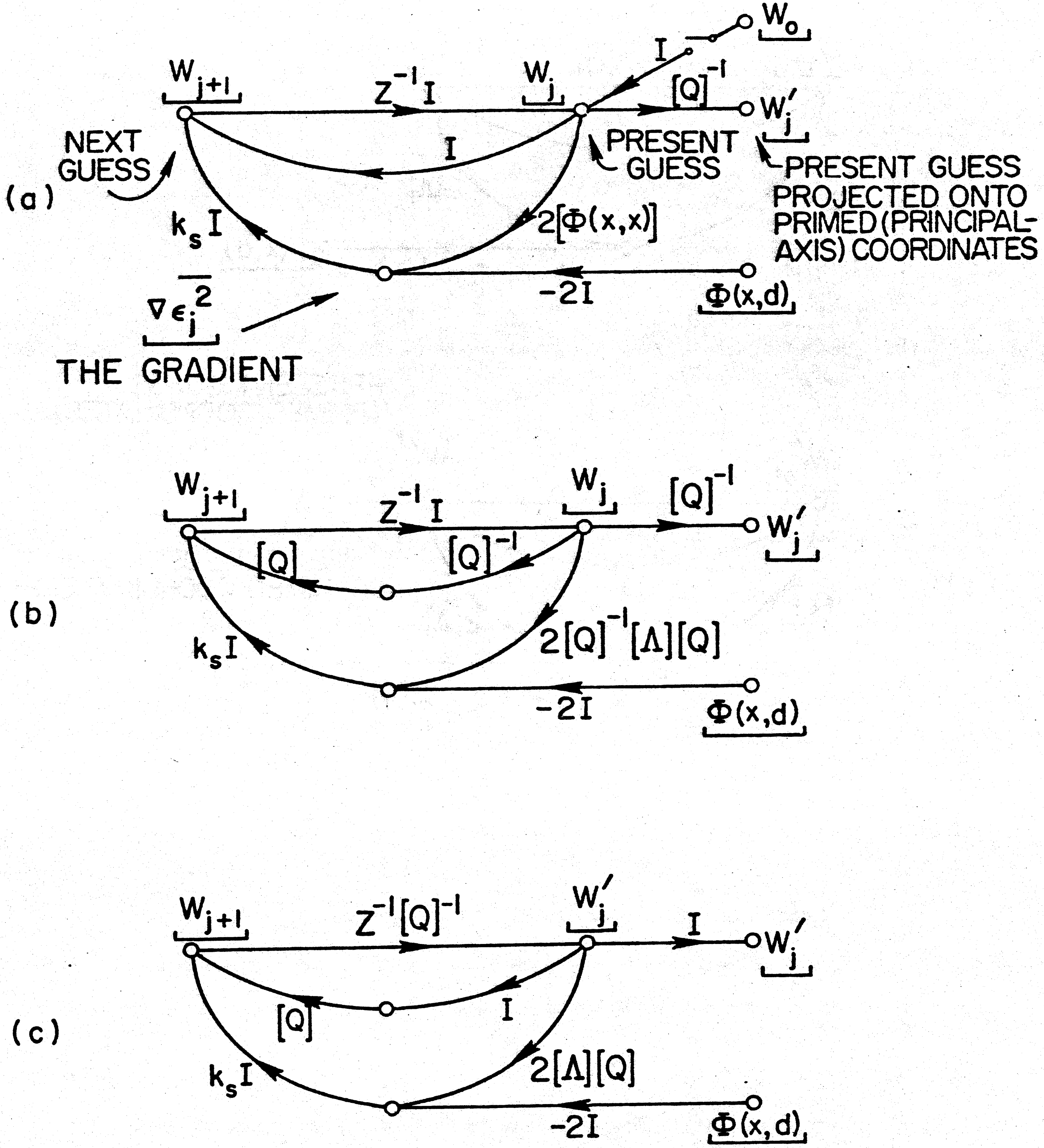
### Normal State Variables

The weight variables comprise a set of state variables, since the complete past history of the adaptive process as it relates to future weight values is contained in the present set of weight values. The weight variables may therefore be called the "weight state variables". The primed variables are derived from the weight state variables by the linear orthogonal transformation $[Q]^T$, and they too comprise a set of state variables, which may be called the "primed weight state variables" or the "normal state variables." The natural modes of the steepest-descent process will be shown to be isolated in the normal or primed coordinates.

### Transients in the Normal State Variables

Refer once again to the flow-graph model of steepest descent, shown in Fig. 4. It is desirable to determine the frequencies or decay rates of the natural modes of the process represented by this graph. The graph is orthogonalized by expressing the transients along the primed coordinates--i.e., in terms of the normal state variables. In the primed-coordinate system, each primed variable has its own natural decay rate.

Figure 5 shows a series of steps involving reduction (simplification) of the multidimensional flow graph of Fig. 4 to an orthogonalized graph. Each step proceeds from the previous step, with care taken to preserve the proper ordering of the matrix multiplications. The algebraic expressions (22), (23), and (26) have been used. Figure 5a is essentially the same as Fig. (4), except that the transformation $[Q]^{-1}$ is shown which projects the transients in $\underline{W_j}$ into the transients in


(a)
$W_0$
I
$W_{j+1}$
$Z^{-1}I$
$W_j$
$[Q]^{-1}$
$W_j'$
NEXT GUESS
I
PRESENT GUESS
PRESENT GUESS PROJECTED ONTO PRIMED (PRINCIPAL-AXIS) COORDINATES
$k_s I$
$2[\Phi(x,x)]$
$-2I$
$\Phi(x,d)$
$\nabla \overline{\epsilon_j^2}$
THE GRADIENT
(b)
$W_{j+1}$
$Z^{-1}I$
$W_j$
$[Q]^{-1}$
$W_j'$
$[Q]$
$[Q]^{-1}$
$k_s I$
$2[Q]^{-1}[\Lambda][Q]$
$-2I$
$\Phi(x,d)$
(c)
$W_{j+1}$
$Z^{-1}[Q]^{-1}$
$W_j'$
I
$W_j'$
$[Q]$
I
$k_s I$
$2[\Lambda][Q]$
$-2I$
$\Phi(x,d)$

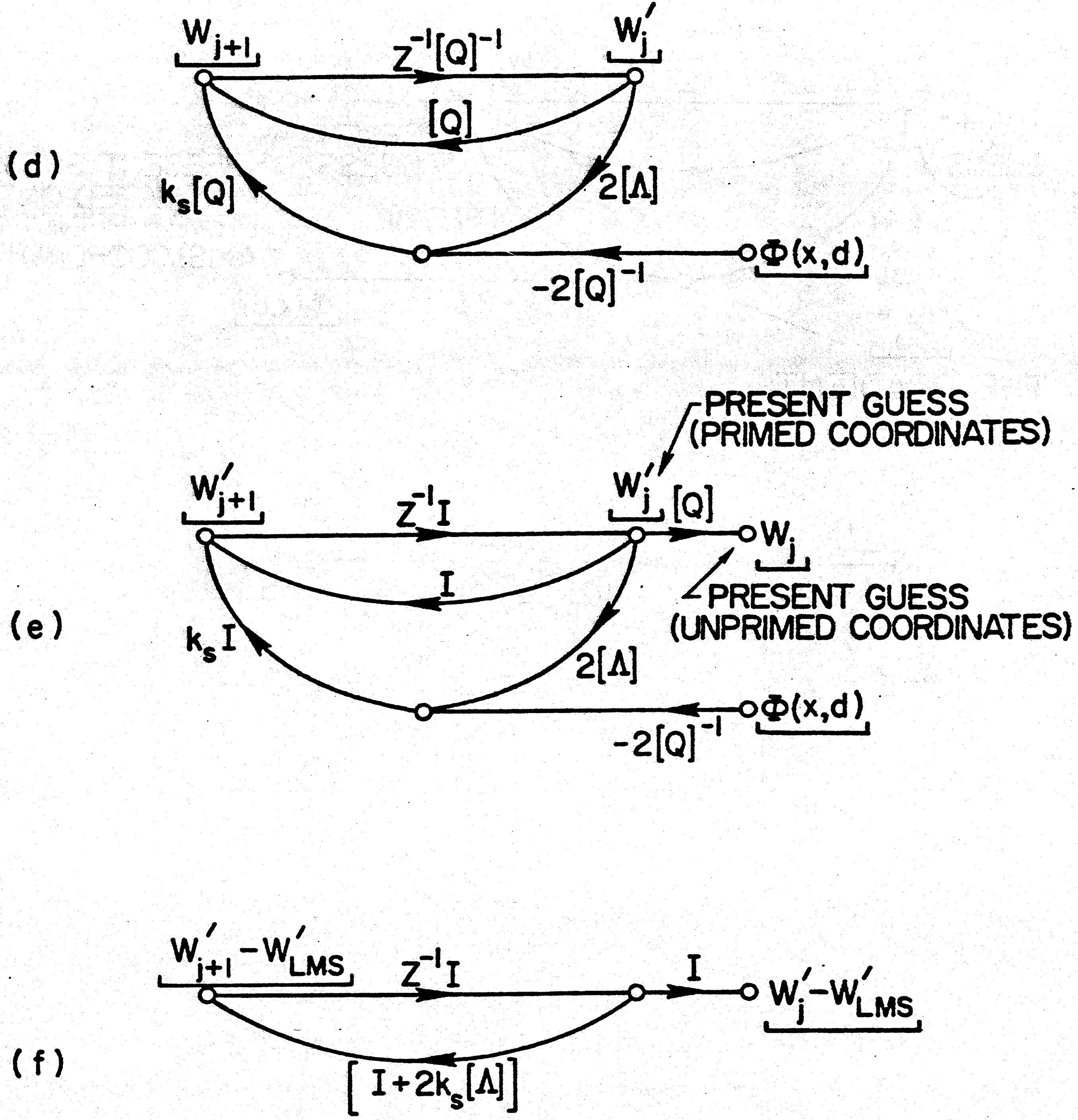


Fig. 5. Transformation steps from unprimed to primed coordinates (normal state variables).

the normal state variables $\underline{W'_j}$. Note that in Fig. 5e, all cross-coupling is removed from within the feedback loop.

In order to study transient phenomena, it is convenient to eliminate the constant "reference" input $\underline{\Phi(x,d)}$ , which has no bearing on transients in any event. This has been done in Fig. 5f. The two parallel feedback branches of Fig. 5e have been combined there. Note that the output variable in Fig. 5f is $\underline{W'_j - W'_{LMS}}$ rather than $\underline{W'_j}$ . This is due to the fact that the constant input $\underline{\Phi(x,d)}$ has been omitted.

The individual components of the vector $\underline{W'_j}$ do not interact. This is clear from Fig. (5f). The diagonalized flow graph corresponds to n independent one-dimensional flow graphs. Each component of $\underline{W'_j}$ is one dimensional and can be treated separately, in the manner Fig. 6.[2]

In Fig. 6a, a method of exciting artificial transients in a one-dimensional sampled-data flow graph is shown. If a unit impulse is applied at the input terminal, a sequence of impulses as shown in Fig. 6b will appear at the output terminal [14],[15],[16],[17]. The amplitudes of succeeding pulses attenuate in geometric progression, and the common ratio r is the feedback-loop gain.

The purpose of inducing the artificial transients is to permit study of time constants in and stability of the processes represented by the flow graphs.

[2] Figures 3a and 3b illustrate that transients in the weights are independent and simple discrete exponentials along the principal axes of the mean-square-error surfaces.

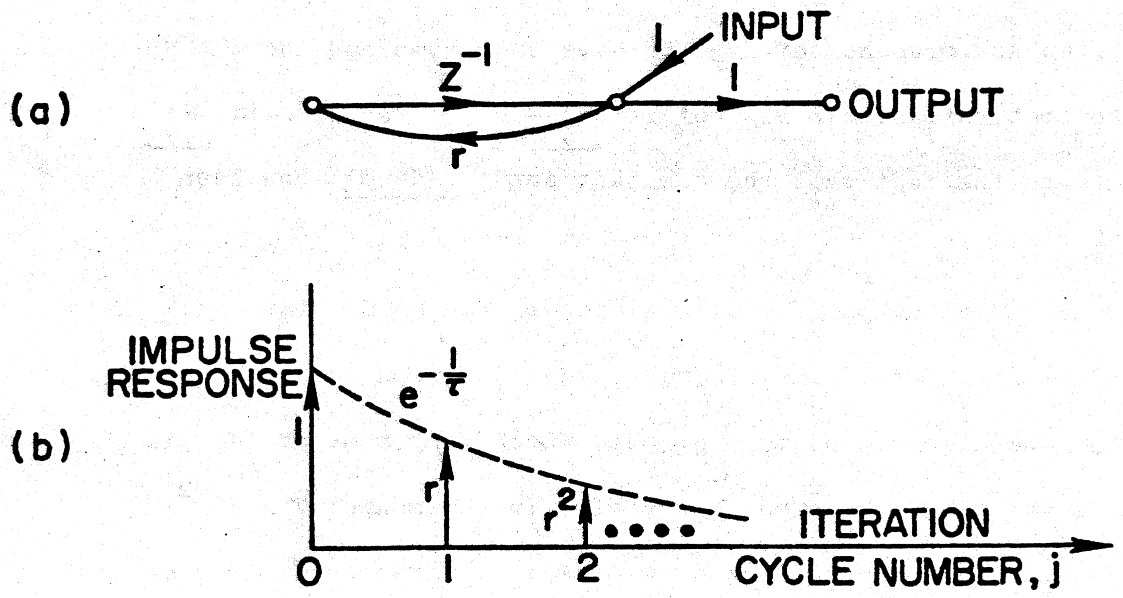


Fig. 6. A one-dimensional flow graph (a), and its impulse response (b).

## Time Constants

If the unit of time is taken to be one iteration cycle, a "time constant" can be defined for the one-dimensional flow graph as the time constant of an exponential envelope of the geometric pulse sequence. As such,

$$r = e^{-1/\tau} = 1 - \frac{1}{\tau} + \frac{1}{2!\tau^2} - \dots \tag{28}$$

Let the time constant be large, i.e., $\tau \gg 1$. Then

$$r \approx 1 - \frac{1}{\tau} \quad .$$

For purposes of analysis, assume that

$$r = 1 - \frac{1}{\tau} \quad . \tag{29}$$

This is Assumption 1.

It can be seen from Fig. 5f that the $p^{th}$ geometric ratio between successive pulse amplitudes is equal to the $p^{th}$ feedback-loop gain, or

$$r_p = 1 + 2\, k_s\, \lambda_p \quad . \tag{30}$$

From (29) and (30),

$$r_p = 1 + 2\, k_s\, \lambda_p = 1 - \frac{1}{\tau_p}$$

$$\tau_p = \frac{-1}{2k_s\lambda_p} \quad . \tag{31}$$

This is the time constant of the $p^{th}$ normal state variable. The number of natural modes is equal to the number of coordinates $n$.

## Stability

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

$$|r| < 1 \quad . \tag{32}$$

It may therefore be concluded that the flow graph of Fig. 5f is stable if for all p,

$$|r_p|_{max} < 1 \quad . \tag{33}$$

The $p^{th}$ geometric ratio is given by Eq. (30). The eigenvalues of $[\Phi(x,x)]$ are such that $\lambda_p \geq$ for all p. Therefore, the only way that the stability condition (33) could be met is for

$$\begin{gathered} k_s < 0, \quad \text{and} \\ |k_s \lambda_{max}| < 1 \quad , \end{gathered} \tag{34}$$

where $\lambda_{max}$ is defined as the maximum eigenvalue of $[\Phi(x,x)]$. From these considerations, it follows that a necessary and sufficient condition for the stability of the steepest-descent adaptation process is that

$$-\frac{1}{\lambda_{max}} < k_s < 0 \quad . \tag{35}$$

It should be observed from (31) and (35) that the rate of adaptation and stability can be controlled by setting $k_s$.

## THE LMS ADAPTATION ALGORITHM

The method of steepest descent requires determination of the gradient vector at successive points on the performance surface (mean-square-error surface). In practice, the true values of these gradients are seldom available. To overcome this difficulty, the "LMS adaptation algorithm" (least-mean-squares-error algorithm)[4], [23] offers a practical procedure for implementing the method of steepest descent.

This algorithm uses measured gradient estimates in place of true gradient values. These estimates may be "noisy" --i.e., contain errors--but the effect of the gradient-measurement errors ("noise") can be minimized through careful application of the adaptation algorithm.

## Method of Gradient Estimation

A method of measuring gradients of the mean-square-error performance surface which does not require squaring, averaging, or differentiating is as follows:

Let the mean-square error $\overline{\epsilon_j^2}$ be represented approximately by the single sample $\epsilon_j^2$, the square of the $j^{th}$ error value. Accordingly, the $i^{th}$ component of the gradient is approximately given by the $i^{th}$ partial derivative of $\epsilon_j^2$ with respect to the weight $w_i$.

$$\frac{\partial \overline{\epsilon_j^2}}{\partial w_i} \approx \frac{\partial \epsilon_j^2}{\partial w_i} = 2\,\epsilon_j\,\frac{\partial \epsilon_j}{\partial w_i} \quad . \tag{36}$$

Differentiating Eq. (1) with respect to $w_i$ gives

$$\frac{\partial \epsilon_j}{\partial w_i} = -x_{ij} \quad . \tag{37}$$

Accordingly,

$$\frac{\partial \overline{\epsilon_j^2}}{\partial w_i} \approx \frac{\partial \epsilon_j^2}{\partial w_i} = -2\,\epsilon_j\,x_{ij} \quad . \tag{38}$$

The entire gradient vector may therefore be approximated as

$$\underline{\nabla \overline{\epsilon_j^2}} \approx \underline{\nabla \epsilon_j^2} = -2\;\epsilon_j\,\underline{X_j} \quad . \tag{39}$$

Thus all one needs to know in order to estimate the gradient is the present input-signal vector $\underline{X_j}$ and its associated scalar error $\epsilon_j$.

The $i^{th}$ component of the gradient estimate $\underline{\nabla \epsilon_j^2}$ may be obtained in summation form by differentiating Eq. (2) with respect to $w_i$.

$$\frac{\partial \epsilon_j^2}{\partial w_i} = -2\, d_j\, x_{ij} + 2 \sum_{\ell} w_\ell\, x_{ij}\, x_{\ell j} \quad . \qquad (40)$$

The expected value of this expression, taken over $j$, is

$$\overline{\left(\frac{\partial \epsilon_j^2}{\partial w_i}\right)} = -2\, \overline{d_j\, x_{ij}} + 2 \sum_{\ell} w_\ell\, \overline{x_{ij}\, x_{\ell j}}$$

$$= -2\, \phi(x_i,d) + 2 \sum_{\ell} w_\ell\, \phi(x_i,x_\ell) \quad . \qquad (41)$$

Inspection of Eqs. (8) and (41) makes it clear that

$$\overline{\left(\frac{\partial \epsilon_j^2}{\partial w_i}\right)} = \frac{\partial \overline{\epsilon_j^2}}{\partial w_i} \quad . \qquad (42)$$

Therefore,

$$\overline{\left(\underline{\nabla \epsilon_j^2}\right)} = \underline{\nabla \overline{\epsilon_j^2}} \quad . \qquad (43)$$

Thus the gradient estimate $\underline{\nabla \epsilon_j^2}$ has an expected value, $\overline{\left(\underline{\nabla \epsilon_j^2}\right)}$, which is exactly the same as the true gradient $\underline{\nabla \overline{\epsilon_j^2}}$ given by Eq. (9). Therefore, the gradient estimate given by Eq. (39) <u>is an unbiased estimate</u>.

<u>Procedure in Using the LMS Algorithm</u>

The LMS algorithm applies the fundamental steepest-descent relationships given by Eqs. (12), (13), and (14), except that now the <u>approximate</u> gradient vector (measured gradient estimate) $\underline{\nabla \epsilon_j^2}$ is used in place of the true gradient vector $\underline{\nabla \overline{\epsilon_j^2}}$. Thus Eq. (12) is replaced by (44):

$$\underline{W_{j+1}} = \underline{W_j} + k_s\, \underline{\nabla \epsilon_j^2} \quad , \qquad (44)$$

and Eq. (14) is replaced by (39):

$$\underline{\nabla \overline{\epsilon_j^2}} \approx \underline{\nabla \epsilon_j^2} = -2\, \epsilon_j\, \underline{X_j} \quad . \tag{39}$$

An adaptation cycle will proceed with the arrival of each new input vector $\underline{X_j}$, $\underline{X_{j+1}}$, ... . From Eqs. (39) and (44), the weight-changing (adaptation) procedure comprising the LMS algorithm is completely represented by (45a) or (45b):

### LMS ALGORITHM

<u>Matrix form</u> $$\underline{W_{j+1}} = \underline{W_j} - 2\, k_s\, \epsilon_j\, \underline{X_j} \quad . \tag{45a}$$

<u>Vector form</u> $$\vec{W}_{j+1} = \vec{W}_j - 2\, k_s\, \epsilon_j\, \vec{X}_j \quad . \tag{45b}$$

In other words, to compute the next weight vector, add the input vector scaled by the product of the error (before adaptation) and a constant $2(-k_s)$ to the present weight vector. In accord with condition (35), it is necessary that $k_s < 0$ for stability. Its magnitude controls the rate of adaptation. The time constants of the adaptive process using the LMS algorithm are given by Eq. (31).

A useful parameter is $\mu_j$, the fraction of the error corrected in each adaptation cycle. The change in error is

$$\Delta\epsilon_j \triangleq - \mu_j\, \epsilon_j \quad . \tag{46}$$

The minus sign in Eq. (46) is necessary in order for $\mu_j$ to be defined as a positive error-reduction factor. Since the error is

$$\epsilon_j = d_j - \vec{W} \cdot \vec{X}_j \quad , \tag{1}$$

the change in error $\Delta\epsilon_j$ due to weight-vector change is

$$\Delta\epsilon_j = - \overrightarrow{\Delta W} \cdot \vec{X} \qquad (47a)$$

$$\triangleq - (\vec{W}_{j+1} - \vec{W}_j) \cdot \vec{X}_j \quad . \qquad (47b)$$

Substituting the value of $(\vec{W}_{j+1} - \vec{W}_j)$ from the LMS algorithm, Eq. (45b),

$$\Delta\epsilon_j = 2k_s \, \epsilon_j \, \vec{X}_j \cdot \vec{X}_j$$

$$= 2 \, k_s \, \epsilon_j \, \|\vec{X}_j\|^2 \quad . \qquad (48)$$

From (46) and (48) it is now possible to relate $\mu_j$, $k_s$, and $\|\vec{X}_j\|^2$:

$$\mu_j = - 2 \, k_s \, \|\vec{X}_j\|^2 \quad . \qquad (49)$$

Adaptation with $k_s$ fixed generally requires $\mu_j$ to vary from cycle to cycle: $\mu_j$ remains fixed only when $\|\vec{X}_j\|^2$ is constant for all j, i.e. when all input vectors are of constant magnitude. Binary input-signal vectors whose component values are either +1 or -1 are examples of constant-magnitude input-signal vectors. Since the $\mu$ parameter is generally variable, it is convenient to work in terms of its average (over j). From Eq. (49),

$$\mu_{avg} = - 2 \, k_s \, \overline{\|\vec{X}_j\|^2} \qquad (50)$$

Time Constants

The time constants may be expressed in terms of $\mu_{avg}$. From Eq. (31) the p^th time constant is

$$\tau_p = \frac{1}{\lambda_p \mu_{avg}} \left( \overline{\|\vec{X}_j\|^2} \right) .$$

Note that the expected value of $\|\vec{X}_j\|^2$ is

$$\overline{\|\vec{X}_j\|^2} = \sum_i \overline{x_{ij}^2} = \sum_i \phi(x_i,x_i) = \text{Tr}[\Phi(x,x)] \quad ,$$

where $\text{Tr}[\Phi(x,x)]$ means the trace of $[\Phi(x,x)]$. The $p^{th}$ time constant can now be expressed as

$$\tau_p = \frac{1}{\lambda_p \mu_{avg}} \text{Tr}\ [\Phi(x,x)] \quad . \tag{51}$$

The time-constant expression becomes especially simple for the case where all input-signal components are mutually uncorrelated and have equal mean squares. In this case, the $[\Phi(x,x)]$ matrix is diagonal with all elements equal and is equal to its own matrix of eigenvalues. Accordingly, the unique time constant is

$$\tau = \frac{1}{\lambda\ \mu_{avg}}\ \text{Tr}\ [\Phi(x,x)] = \frac{\lambda\ n}{\lambda\ \mu_{avg}} = \frac{n}{\mu_{avg}} \quad . \tag{52}$$

Although expression (52) is precisely derived for special circumstances, this expression gives one an approximate measure of the rate of learning for a given average value of the error-correction constant $\mu_{avg}$ for a wide range of $[\Phi(x,x)]$ matrices. For a given $\mu_{avg}$, the learning time generally increases linearly with the number of weights, n.

## Stability of the LMS Algorithm

When the LMS algorithm is utilized, stability is completely determined by condition (35). In order to apply this condition, however, one would need to know the maximum eigenvalue $\lambda_{max}$ of the input-signal correlation matrix $[\Phi(x,x)]$. Sometimes this eigenvalue can be computed, but in most cases such a computation is difficult or impossible to perform. Stability can be assured without knowing $\lambda_{max}$, however, as long as the error correction factor $\mu_j$ is kept within certain bounds. These bounds do not depend on $[\Phi(x,x)]$.

If the LMS algorithm is stable, transients must die out. If the algorithm is unstable on the other hand, the weight-vector magnitude will grow without bound. Stability conditions on $\mu_j$ will now be developed that will prevent the weight-vector magnitude from growing, and indeed will force it to diminish if the iterative process is started with a weight vector of large magnitude.

Let the $j^{th}$ weight vector magnitude $\| \vec{W}_j \|$ become arbitrarily large, i.e., $\| \vec{W}_j \| \rightarrow \infty$. It will be shown that the result of the $j^{th}$ adaptation cycle will then be a diminution of $\| \vec{W}_j \|$, that is,

$$\| \vec{W}_j + \Delta\vec{W}_j \| < \| \vec{W}_j \|$$

under the following conditions:

(a) $2 > \mu_j > 0$.

(b) $d_j$ is finite and bounded.

It can be seen from Eq. (46) that

$$\epsilon_j + \Delta \epsilon_j = \epsilon_j (1 - \mu_j) \quad .$$

The magnitude of the error $\epsilon_j$ will always be reduced after adaptation when $2 > \mu_j > 0$. When $\mu_j$ is chosen in this range,

$$\left| \epsilon_j + \Delta \epsilon_j \right| = \left| \epsilon_j \; (1 - \mu_j) \right| < \left| \epsilon_j \right| \quad . \tag{53}$$

From (1) and (47a)

$$\epsilon_j + \Delta \epsilon_j = d_j - (\vec{W}_j + \Delta\vec{W}_j) \cdot \vec{X}_j \quad .$$

It follows that

$$|d_j - (\vec{W}_j + \Delta\vec{W}_j) \cdot \vec{X}_j| < |d_j - \vec{W}_j \cdot \vec{X}_j| \quad . \tag{54}$$

By the original hypothesis, $|d_j|$ is finite and bounded and $\| W_j \| \to \infty$. Therefore,

$$|\vec{W}_j \cdot \vec{X}_j| >>> |d_j| \ , \quad \text{and}$$

$$|(\vec{W}_j + \Delta\vec{W}_j) \cdot \vec{X}_j| >>> |d_j| \ . \tag{55}$$

The only exceptions to (55) are the cases where $\| \vec{X}_j \| = 0$, and where $\vec{X}_j$ is perfectly orthogonal to $\vec{W}_j$. The first case is of no interest, since no adaptation takes place when $\| \vec{X}_j \| = 0$. The second case occurs with probability zero when input signals are taken from natural processes.

As a consequence of (55), inequality (54) may be written as

$$|(\vec{W}_j + \overrightarrow{\Delta W}_j) \cdot \vec{X}_j| < |\vec{W}_j \cdot \vec{X}_j| \ . \tag{56}$$

The input vector $\vec{X}_j$, the weight vector $\vec{W}_j$, and the weight-change vector $\Delta\vec{W}_j$ are pictured in Fig. 7 in the hyperplane determined by $\vec{X}_j$ and $\vec{W}_j$. Note that the weight-change vector $\overrightarrow{\Delta W}_j$ is parallel to the input-signal vector $\vec{X}_j$, in accord with the basic LMS algorithm. Let a circle be drawn in the hyperplane through the tip of $\vec{W}_j$, as shown in Fig. 7. The weight-change vector $\overrightarrow{\Delta W}_j$ must lie along the dotted chord (parallel to $\vec{X}_j$) of this circle, and within this circle in order to satisfy (56). It is clear from the figure that whether the angle between $\vec{W}_j$ and $\vec{X}_j$ is acute (as drawn) or obtuse,

$$\|\vec{W}_j + \overrightarrow{\Delta W}_j\| < \|\vec{W}_j\| \ . \tag{57}$$

This indicates stability of the LMS adaptation process when $\mu_j$ is positive and always less than 2, when $|d_j|$ is bounded, and when the input-signal vectors $\vec{X}_j$ are not orthogonal to the weight vectors $\vec{W}_j$.

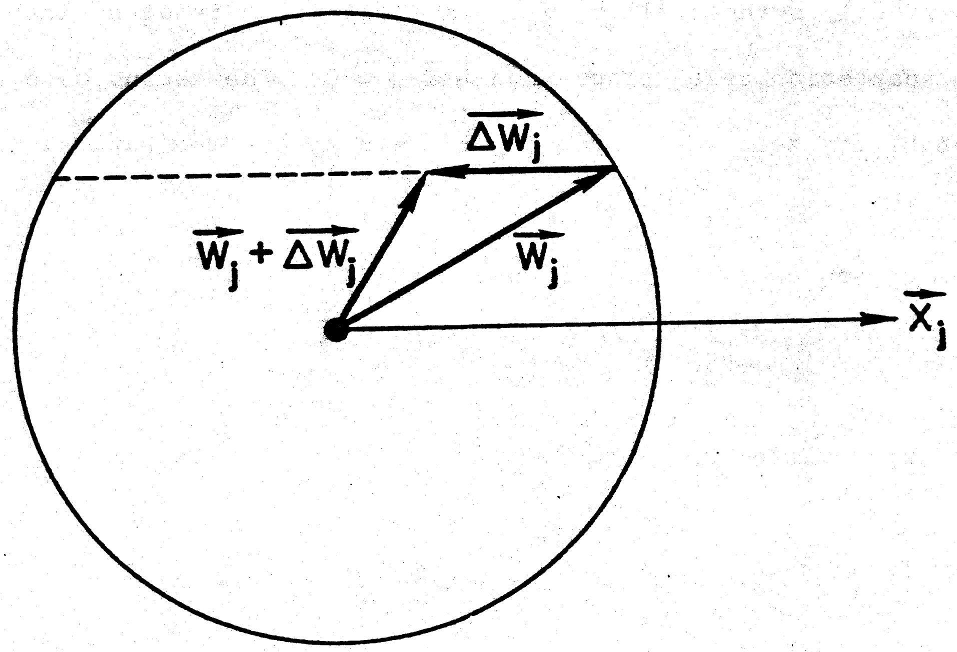

$\overrightarrow{\Delta W_j}$
$\vec{W}_j + \overrightarrow{\Delta W_j}$
$\vec{W}_j$
$\vec{X}_j$

Fig. 7. Geometrical relations among $\vec{X}_j$, $\vec{W}_j$, and $\overrightarrow{\Delta W}_j$ .

(It should be noted that the above demonstration does *not* prove that the LMS algorithm is necessarily unstable if these conditions are not all met.)

In practice, the boundedness of $|d_j|$ and nonorthogonality of $\vec{X}_j$ and $\vec{W}_j$ occur naturally in almost every circumstance. Insuring that $2 > \mu_j > 0$ is usually not difficult. From (49), it is clear that a value of $k_s$ should be chosen in the range

$$0 > k_s > - \frac{1}{\| X_j \|^2_{max}} \quad . \tag{58}$$

If the maximum input-vector magnitude is not known *a priori*, it can be estimated and updated as more and more input-signal observations are made. Other considerations than just stability alone are usually involved in the choice of $k_s$. This parameter controls the time constants of the adaptive process; the time constants in turn affect the rate of learning of the system and the quality of performance. In the next sections, formulas will be derived which relate the rate of adaptation to system performance.

## GRADIENT-MEASUREMENT NOISE

It has been pointed out that the gradient estimates used in the LMS algorithm, though unbiased, are not perfect. Differences exist between the measured estimates $\underbrace{\nabla \epsilon_j^2}$ and the true values of the gradients $\underbrace{\nabla \overline{\epsilon_j^2}}$. These differences will be referred to as *gradient-measurement noise*.

When the LMS adaptive process is stable, transients in the adjustments essentially die out after three to five time constants of the slowest mode elapse. In steady state, the weight values will experience random fluctuations about the respective LMS-optimal weight values, and the amplitudes of these random excursions will depend on the rate of adaptation. The random fluctuations are caused by gradient-measurement noise. It will be shown that slowness in adaptation can serve as a noise filtering process to reduce the deleterious effects of weight-adjustment fluctuations upon system performance.

The mean-square error, as a function of the weights projected in the primed (principal-axis) coordinates, may be expressed in a manner similar to Eq. (27) as follows:

$$\overline{\epsilon_j^2} = \overline{\epsilon_{min}^2} + \underline{(W'_j - W'_{LMS})}[\Lambda] \left.(W'_j - W'_{LMS})\right] \quad . \qquad (59)$$

The mean-square error, a function of the present $\underline{W'_j}$ , is the expected value of the squared error. Let this expected value be represented by the symbol $y_j$. Then (59) may be expressed alternatively as the following summation:

$$y_j \triangleq \overline{\epsilon_j^2} = \overline{\epsilon_{min}^2} + \sum_p \lambda_p (W'_{pj} - w'_{p_{LMS}})^2 \quad , \qquad (60)$$

where p indexes the normal-state-variable number.

Excess Mean-Square Error Due to Adaptation

Any departure at any time of one or more of the weights from its optimal value will cause an increase in $y_j$. In steady state, after gross adapting transients have died out, increases in $y_j$ above $\overline{\epsilon_{min}^2}$ will be caused by random excursions in the weights about their optimal values. The expected value of $y_j$ is (once again taking expectation over j):

$$\overline{y}_j = \overline{\epsilon^2_{\min}} + \sum_p \lambda_p \overline{(w'_{pj} - w'_{p_{LMS}})^2} \; . \qquad (61)$$

In steady state, the mean-square error exceeds the minimum mean-square error by the sum of variances of the fluctuations of the normal state variables weighted by the respective eigenvalues. Thus the excess mean-square error due to the adaptive process is

$$\left(\overline{y}_j - \overline{\epsilon^2_{\min}}\right) = \sum_p \lambda_p \overline{(w'_{pj} - w'_{p_{LMS}})^2} \; . \qquad (62)$$

To evaluate this excess mean-square error, one needs to compute $\overline{(w'_{pj} - w'_{p_{LMS}})^2}$, the variance in the fluctuation of the $p^{th}$ normal state variable, for $p = 1, 2, \ldots, n$.

Covariance Matrix of Gradient-Measurement Noise

Define a gradient-measurement noise vector $\underline{n}_j$ with components $\nu_{1j}, \nu_{2j}, \ldots . \nu_{nj}$ . Then from (38), the error (noise) in the estimate of the $i^{th}$ component of the gradient is

$$\left(\frac{\partial\overline{\epsilon^2_j}}{\partial w_i} - \frac{\partial \epsilon^2_j}{\partial w_i}\right) \triangleq \nu_{ij} \; , \qquad (63)$$

where $\nu_{ij}$ will be referred to as the $i^{th}$ gradient-measurement noise component for the $j^{th}$ input vector. Since the gradient estimates are unbiased,

$$\overline{\nu_{ij}} = 0 \quad \text{for} \quad i = 1, 2, \ldots n.$$

The questions to be considered next are,

$$\overline{\nu^2_{ij}} = ? \qquad \overline{\nu_{ij}\,\nu_{\ell j}} = ?$$

That is, what are the variances of the gradient-noise components and what are their covariances?

Assume that the weight adjustments are set very close to the minimum of the mean-square-error function, that is, that the adaptive process is close to convergence, so that

$$\frac{\partial \overline{\epsilon_j^2}}{\partial w_i} \cong 0, \quad \text{for all} \quad w_i \quad .$$

This is Assumption 2.

From Assumption 2, definition (63), and Eq. (37):

$$\nu_{ij} = - \frac{\partial \epsilon_j^2}{\partial w_i} = - 2\, \epsilon_j \frac{\partial \epsilon_j}{\partial w_i} = 2\, \epsilon_j\, x_{ij} \quad . \qquad (64)$$

The square of $\nu_{ij}$ is

$$\nu_{ij}^2 = 4\, \epsilon_j^2\, x_{ij}^2 \quad .$$

The variance of $\nu_{ij}$ is therefore

$$\overline{\nu_{ij}^2} = 4\, \overline{\epsilon_j^2\, x_{ij}^2} \quad . \qquad (65)$$

The covariances of two gradient-measurement noise components may be expressed by using Eq. (64).

$$\overline{\nu_{ij}\, \nu_{\ell j}} = 4\, \overline{\epsilon_j^2\, x_{ij}\, x_{\ell j}} \quad . \qquad (66)$$

It is desirable to express the variances and covariances of the gradient-measurement-noise components somewhat differently. To do this, assume that

$$\overline{\epsilon_j^2\, x_{ij}^2} = \left(\overline{\epsilon_j^2}\right) \left(\overline{x_{ij}^2}\right) , \qquad (67)$$

and more generally

$$\overline{\epsilon_j^2\, x_{ij}\, x_{\ell j}} = \left(\overline{\epsilon_j^2}\right) \left(\overline{x_{ij}\, x_{\ell j}}\right) \quad . \qquad (68)$$

This is Assumption 3. The assumption will be met with a high degree of consistency when there are a large number of input-signal components. In this case, the error minus its mean, $\left(\epsilon_j - \overline{\epsilon_j}\right)$, and the square thereof, $\left(\epsilon_j - \overline{\epsilon_j}\right)^2$, tend to be uncorrelated with the individual input signal minus its mean, $\left(x_{ij} - \overline{x_{ij}}\right)$, and with products of pairs $\left(x_{ij} - \bar{x}_{ij}\right)\left(x_{\ell j} - \overline{x_{\ell j}}\right)$.

Using (65) and (66) and Assumption 3, the variance and covariance of the gradient-measurement noise components can be written as

$$\overline{\nu_{ij}^2} = 4\left(\overline{\epsilon_j^2}\right)\left(\overline{x_{ij}^2}\right)$$

$$\overline{\nu_{ij}\,\nu_{\ell j}} = 4\left(\overline{\epsilon_j^2}\right)\left(\overline{x_{ij}\,x_{\ell j}}\right) . \tag{69}$$

When operating in the vicinity of the minimum mean-square error (consistent with Assumption 2, expressions (69) can be written as

$$\overline{\nu_{ij}^2} = 4\left(\overline{\epsilon_{\min}^2}\right)\left(\overline{x_{ij}^2}\right) = 4\left(\overline{\epsilon_{\min}^2}\right)\phi(x_i, x_i) \quad . \tag{70}$$

The covariance is therefore

$$\overline{\nu_{ij}\,\nu_{\ell j}} = 4\left(\overline{\epsilon_{\min}^2}\right)\left(\overline{x_{ij}\,x_{\ell j}}\right) = 4\left(\overline{\epsilon_{\min}^2}\right)\phi(x_i, x_\ell) \; . \tag{71}$$

The gradient-measurement-noise covariance matrix can therefore be expressed as

$$[\Phi(\nu,\nu)] \triangleq \overline{\mathcal{N}_j\rbrack\,\lfloor\mathcal{N}_j\rfloor} = \begin{bmatrix} \nu_{1j}\nu_{1j} & \nu_{1j}\nu_{2j} & \nu_{1j}\nu_{3j} & \cdots \\ \nu_{2j}\nu_{1j} & \cdots & & \\ \vdots & & \cdots & \nu_{nj}\nu_{nj} \end{bmatrix} . \tag{72}$$

Using (71) and (7),

$$[\Phi(\nu,\nu)] = 4\left(\overline{\epsilon_{\min}^2}\right)[\Phi(x,x)] \quad . \tag{73}$$

## Propagation Path of Gradient-Measurement Noise

Deriving the gradient-measurement noise covariance matrix is an important step toward calculating the variances of the steady-state fluctuations in the normal state variables, as required for determining the excess mean-square error, Eq. (62). The gradient-measurement noise components propagate and interact with the normal state variables in a way that can be modeled by means of signal flow graphs.

The flow graph of Fig. 5a is a precise model for the method of steepest descent when acting upon a quadratic performance surface. During each iteration cycle, use is made of the true gradient values. In order to use this model to represent steepest descent via the LMS algorithm, however, the differences between the true and the estimated gradient values must be accounted for. This is accomplished in the flow graph of Fig. 8a where the gradient-measurement noise $\underline{n_j}$ is shown added into the "gradient node" to represent the actual measured "noisy gradient."

Fig. 8b can be obtained from Fig. 8a by series of flow-graph reduction steps which are analagous to the steps shown in Fig. 5 which connect Fig. 5a to Fig. 5e. Notice that in Fig. 8b, the gradient measurement noise $\underline{n_j}$ propagates into the graph via a branch of transfer function $[Q]^{-1}$.

Now define a "primed gradient-measurement noise" $\underline{n'_j}$ such that

$$\underline{n'_j} \triangleq \underline{n_j}\,[Q]^{-1} . \qquad (74)$$

The vector $\underline{n'_j}$ is therefore the projection of $\underline{n_j}$ onto the normal or principal-axis set of coordinates. In accord with (74), the noise $\underline{n_j}$ is replaced in Fig. 8c by the noise $\underline{n'_j}$ which in turn propagates into

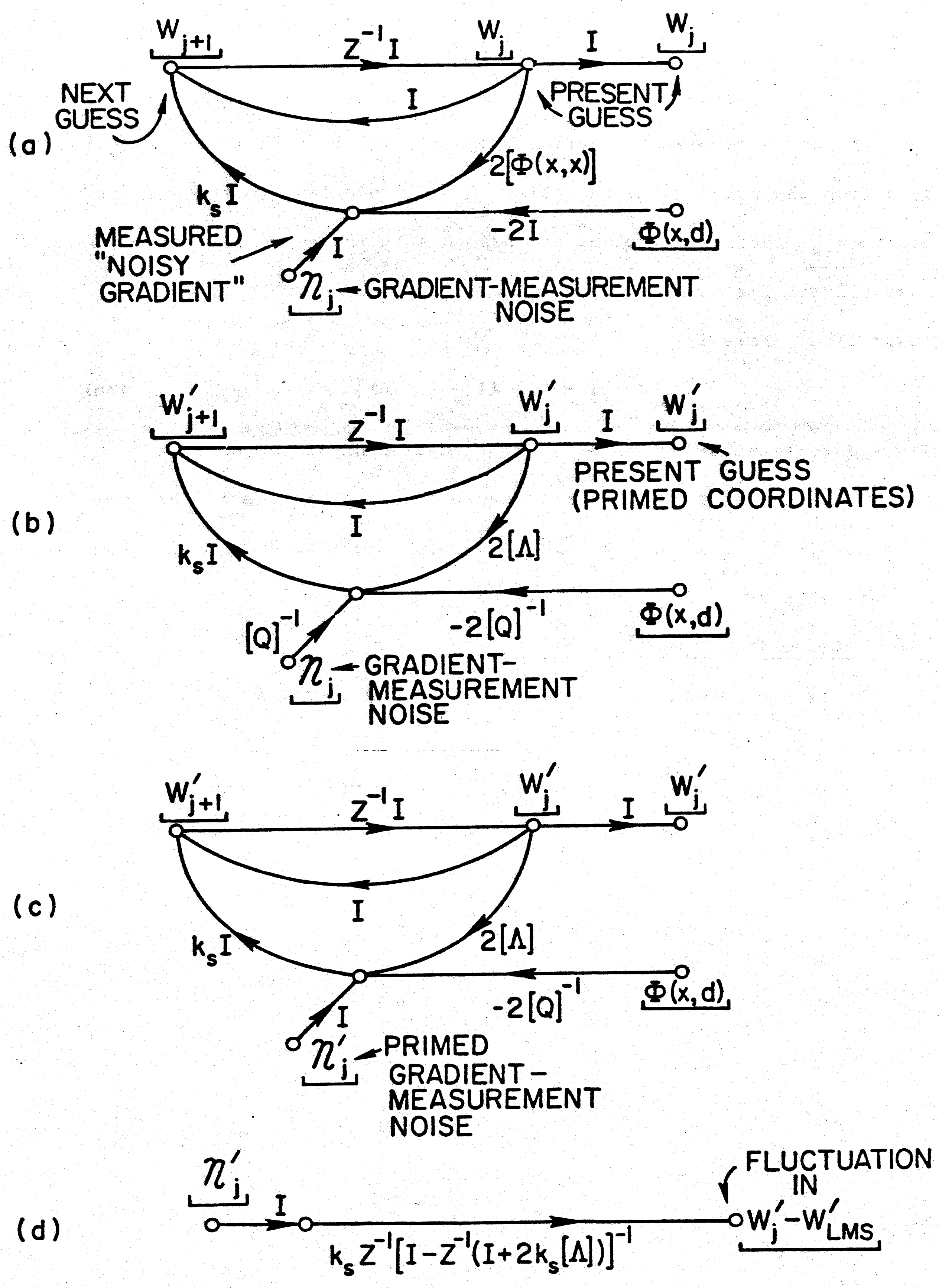


Fig. 8. The gradient-measurement-noise propagation path.

the graph via a branch whose transfer function is I.

It can be seen from the flow graph of Fig. 8c that the signal flow path from the point where the noise $\underline{n'_j}$ is injected, to the "output" point $\underline{W'_j}$, does not include any branches in which crosscoupling could take place. The transfer function of this path can be obtained by inspection. This is

$$k_s \, Z^{-1} [I - Z^{-1} \, (I + 2k_s \, \Lambda)]^{-1} \; . \tag{75}$$

Fig. 8d represents symbolically the transmission of the noise $\underline{n'_j}$ through this transfer function to cause the fluctuations in the components of $\underline{W'_j - W'_{LMS}}$. The variances of these fluctuations will be calculated next.

Fluctuations in Normal State Variables

It is now necessary to obtain the covariance matrix $[\Phi(\nu',\nu')]$ of the projected (primed) noise $\underline{n'_j}$. The covariance matrix of $\underline{n_j}$ is known from (72) and (73). From these, with (22) and (26), it is possible to calculate $[\Phi(\nu',\nu')]$:

$$\begin{aligned} [\Phi(\nu',\nu')] &\triangleq \overline{n'_j] \; \underline{n'_j}} = [Q] \; \overline{n_j] \; \underline{n_j}} \; [Q]^{-1} \\ &= 4\left(\overline{\epsilon^2_{\min}}\right)[Q][\Phi(x,x)][Q]^{-1} \\ &= 4\left(\overline{\epsilon^2_{\min}}\right) [\Lambda] \quad . \end{aligned} \tag{76}$$

It is interesting to note from (76) that the components of $\underline{n'_j}$ are mutually uncorrelated, since the covariance matrix $[\Phi(\nu',\nu')]$ is diagonal. The noise $\underline{n'_j}$ propagates through the transfer function (75), as was pointed out previously. Since the transfer-function matrix (75) is diagonal, each component of $\underline{n'_j}$ propagates independently, so that there are simply n mutually uncorrelated noises propagating in n independent one-dimensional linear discrete paths. Thus, it can be seen from (75) that the $p^{th}$ component of the noise vector $\underline{\mathfrak{n}'_j}$ propagates through a one-dimensional transfer function

$$\frac{k_s Z^{-1}}{1 - Z^{-1}(1 + 2 k_s \lambda_p)} \tag{77}$$

to reach the $p^{th}$ normal state variable.

It will be assumed that each of the individual components of the noise $\underline{\mathfrak{n}'_j} \triangleq \nu'_{1j} \; \nu'_{2j} \; \dots \; \nu'_{nj}$ is "white", i.e.,

$$\overline{(\nu'_{pj+1})(\nu'_{pj})} = 0, \quad p = 1,2, \dots n \; . \tag{78}$$

This is <u>Assumption 4</u>. This assumption is easily met when input-signal vectors are chosen at random. When they are segments of a time series, the assumption is still met approximately because of the variability of the input signals, and also the variability of the weight vector due to the adaptation process.

The $p^{th}$ component of $\underline{\mathfrak{n}'_j}$ is thus an uncorrelated discrete process which is an input to the discrete transfer function (77). This transfer function has unit impulse response represented by the sequence

$$0, k_s, k_s r_p, k_s r_p^2, k_s r_p^3, \dots , \tag{79}$$

where the geometric ratio $r_p$ is given by

$$r_p = 1 + 2 k_s \lambda_p \; . \tag{30}$$

It has been shown that when the input signal to a linear discrete (sampled-data) system is "white", the variance of the output signal equals the variance of the input signal multiplied by the sum of the squares of the amplitudes of the impulses in the unit impulse response [14][15]. The sum of the squares of the sequence (79) is

$$(\mathrm{SUM\ SQ})_p \triangleq k_s^2\,[1 + r_p^2 + r_p^4 + \ldots] = \frac{k_s^2}{1-r_p^2} \quad . \tag{80}$$

From Eq. (31), $k_s$ can be expressed as

$$k_s = \frac{-1}{2\tau_p\,\lambda_p} \quad . \tag{81}$$

Combining (80) and (81) gives

$$(\mathrm{SUM\ SQ})_p = \frac{1}{4\,\tau_p^2\,\lambda_p^2\left(\frac{2}{\tau_p} - \frac{1}{\tau_p^2}\right)} \quad . \tag{82}$$

In accord with Assumption 1, $\tau_p \gg 1$. The sum of squares may therefore be written as

$$(\mathrm{SUM\ SQ})_p = \frac{1}{8\tau_p\,\lambda_p^2} \quad . \tag{83}$$

The variance of the $p^{th}$ component of $\underline{\nu}'_j$ as obtained from (76) is

$$\overline{(\nu'_{pj})^2} = 4\left(\overline{\epsilon^2_{min}}\right)\lambda_p \quad .$$

The "sum of squares" for the $p^{th}$ propagation path is given by (83). The variance of the steady-state fluctuation in the $p^{th}$ normal state variable is therefore

$$\overline{(w'_{pj} - w'_{p_{LMS}})^2} = \overline{(\nu'_{pj})^2}\ (\text{SUM SQ})_p$$

$$= 4\left(\overline{\epsilon^2_{min}}\right) \lambda_p \frac{1}{8\tau_p \lambda_p^2}$$

$$= \frac{\overline{\epsilon^2_{min}}}{2\tau_p \lambda_p} \quad . \tag{84}$$

It is now possible to calculate the excess mean-square error due to adaptation. Refer to Eq. (62). Substitute (84) into (62). The excess mean-square error is

$$\bar{y}_j - \overline{\epsilon^2_{min}} = \sum_p \frac{\lambda_p\left(\epsilon^2_{min}\right)}{2\tau_p \lambda_p} = \frac{1}{2}\left(\overline{\epsilon^2_{min}}\right)\sum_p \frac{1}{\tau_p} \quad . \tag{85}$$

Notice from this simple formula that the excess mean-square error depends only on the value of the minimum mean-square error and the adaptive time constants. The excess mean-square error could be made as small as one pleases by adapting slowly, i.e., by making the $\tau_p$'s large.

## MISADJUSTMENT

Where the purpose of adaptation is the minimization of mean-square error, the excess mean-square error is an important factor. However, it alone does not have as much physical meaning and/or usefulness as the relative excess mean-square error, i.e., the excess mean-square error normalized with respect to the minimum mean-square error. This fundamental dimensionless measure of the performance of an adaptive system has been called the "misadjustment" [12][13]. Thus

$$\text{Misadjustment} \quad M \triangleq \frac{\left(\overline{y_j} - \overline{\epsilon^2_{\min}}\right)}{\overline{\epsilon^2_{\min}}} \tag{86}$$

The misadjustment allows one to compare the performance of an adaptive system with that of an "ideal" system whose mean-square error is $\overline{\epsilon^2_{\min}}$ --i.e., with a Wiener filter designed on the basis of perfect _a priori_ knowledge of the second-order statistics of the signals to be processed.

## Misadjustment Formulas

A simple formula for misadjustment results from substituting (85) in (86).

$$M = \frac{1}{2} \sum_{p=1}^{n} \frac{1}{\tau_p} \quad , \tag{87}$$

where $n$ is the number of adjustment variables (number of weights). The time constants are controlled by the parameter $k_s$, in accord with Eq. (31).

Some insight can be obtained by considering the important special case in which all time constants are equal. This case results when $[\Phi(x,x)]$ is diagonal with all elements equal; $\tau_p = \tau$ for all $p$. Then

$$M = \frac{1}{2} \sum_{p=1}^{n} \frac{1}{\tau_p} = \frac{n}{2\tau} \quad . \tag{88}$$

From this special case, it can be seen that $M$ increases linearly with the number of weights and varies inversely with the time constant of the adaptation process. As the speed of adaptation approaches zero, $M$ approaches zero, and the mean-square error therefore approaches $\overline{\epsilon^2_{\min}}$. For this special case, $M$ may be expressed in terms of $k_s$. Using

(31) and (88),

$$M = \frac{n}{2\tau} = - k_s \lambda n \quad . \qquad (89)$$

This formula may also be expressed in terms of the average error correction factor $\mu_{avg}$, in accord with Eq. (52).

$$M = \frac{n}{2\tau} = \frac{\mu_{avg}}{2} \quad . \qquad (90)$$

Equation (88) can be used as an approximate, fairly general relationship between speed of adaptation and quality of performance, without requiring detailed knowledge of the individual time constants. As an example, let the adaptive filter shown in Fig. 1 have 25 taps on its tapped delay line. The question is, how fast could such a filter be adapted? Assume that a steady-state misadjustment of 10 percent would be satisfactory. Using Eq. (88),

$$M = 0.1 = \frac{n}{2\tau} = \frac{25}{2\tau} \quad ;$$

$$\tau = 125 \text{ iteration cycles, approximately.}$$

Assuming that adapting transients would die out in three time-constant intervals, the settling time of the adaptive process for $\tau = 125$ cycles would be approximately 375 adaptation cycles, or 375 input-sample periods.

## Efficiencies of Adaptation Processes

It is useful to define a figure of merit, or efficiency measure, for adaptive processes. Such a definition will be made here which allows the effectiveness of several specific adaptation schemes to be compared. Define the figure of merit as

$$\text{F.M.} \triangleq \frac{\text{number of weights}}{\left(\begin{array}{c}\text{settling}\\ \text{time}\end{array}\right)\left(M\right)} \quad , \qquad (91)$$

where "settling time" can arbitrarily but reasonably be defined in terms of the largest adaptive time constant $\tau_{\max}$ as

$$\begin{pmatrix}\text{settling}\\ \text{time}\end{pmatrix} \triangleq 3\ \tau_{\max} \ . \tag{92}$$

The figure of merit was defined above so that with fixed M, F.M. increases with reduction of settling time per number of weights.

The figure of merit of the LMS adaptation process when all time constants turn out to be equal is, using (91) and (88)

$$\text{F.M.} = \frac{n}{(3\tau)\,(M)} = \frac{n}{3\tau\left(\frac{n}{2\tau}\right)} = \frac{2}{3} \ . \tag{93}$$

The figure of merit of the LMS adaptation process in general is, using (91), (92), and (87):

$$\text{F.M.} = \frac{n}{\left(3\ \tau_{\max}\right)\left(\frac{1}{2}\sum_{p=1}^{n}\frac{1}{\tau_p}\right)} = \frac{2}{3}\ \frac{1}{\frac{1}{n}\sum_{p=1}^{n}\frac{\tau_{\max}}{\tau_p}} \ . \tag{94}$$

Since

$$\frac{1}{n}\sum_{p=1}^{n}\frac{\tau_{\max}}{\tau_p} \geq 1 \ ,$$

it follows that

$$\text{F.M.} \leq \frac{2}{3} \ . \tag{95}$$

The figure of merit of the LMS process is less when the time constants differ from one another, than when they are all the same. Thus the greater the disparity among the eigenvalues of $[\Phi(x,x)]$ the lower will be the efficiency of the adaptation process. Lower efficiency means longer settling time for the same level of misadjustment for the same number of weights.

At the cost of increased complexity, the LMS algorithm can be implemented by using Newton's method [8][9] instead of steepest descent; Newton's method causes all time constants to be equal, thus improving the figure of merit to the value 2/3. It will be recalled that the steepest-descent LMS algorithm requires that the input vector $\underline{X_j}$ be scaled by the error $\epsilon_j$ and then added to the weight vector. The use of Newton's method would require in addition that the input vector $\underline{X_j}$ be postmultiplied by $[\Phi(x,x)]^{-1}$. This matrix multiplication is costly and in addition, knowledge of $[\Phi(x,x)]$ and its inverse are required. In most cases, the loss of efficiency of the conventional (steepest-descent) LMS algorithm is more than offset by its inherent simplicity, stability, and ease of implementation.

If a finite number of input-vector samples N are available and if these data can be stored and repeated over and over again to train the adaptive system, it has been shown that adaptation with small $k_s$ will cause the weights to approach a solution $\underline{W}$ given by

$$\underline{W} = \underline{\Phi(x,d)}_N \, [\Phi(x,x)]_N^{-1} \,, \tag{96}$$

where $\underline{\Phi(x,d)}_N$ is the sample crosscorrelation vector and $[\Phi(x,x)]^{-1}$ is the inverse of the sample input correlation matrix [23]. It can be shown without making any new assumptions [13][24] that a system so adapted will have a misadjustment of

$$M = \frac{n}{N} \,. \tag{97}$$

In this case, the system settling time, or averaging time, is N sample periods. The figure of merit of this "data-repeated-again-and-again" adaptive process is therefore

$$\text{F.M.} = \frac{n}{(N)\left(\frac{n}{N}\right)} = 1 \quad . \tag{98}$$

It is interesting to note that the LMS adaptation process described herein, which uses its input data on a one-pass, one-pattern-at-a-time basis, has a figure of merit which is only one third less than that of a process which could require the storage of all past input data. However, to achieve the same level of misadjustment (the same signal-processing performance) as is achieved in the data-repeating process, the settling time or averaging window of the one-pattern-at-a-time process would have to be fifty percent longer than the averaging time of the data-repeating process.

## APPLICATION OF ADAPTIVE FILTERS TO NONSTATIONARY SIGNALS

When the input to an adaptive system is a stationary process, as has been assumed in the preceding analysis, the mean-square-error surface is fixed in shape, orientation, and position. When on the other hand the system input is nonstationary, it is possible to define a quasi-static mean-square-error surface whose characteristics would vary relatively slowly with changes in the statistical characteristics of the input signal.

When the LMS algorithm is implemented in connection with a slow adaptive process (small $k_s$, long time constants), it causes the adjustments of an adaptive system to approach very closely the minimum of a stationary mean-square-error surface. The slower the adaptive process, the nearer the system adjustments come to, and the more closely they remain at, the "bottom of the bowl." In steady state, the excess mean-square error is proportional to the speed of adaptation [see Eq. (85)]. This type of excess mean-square error may be considered to be due to adapting too rapidly.

In the nonstationary-input case, the position of the instantaneous mean-square-error surface minimum is constantly changing, and if the time constant of the adaptive process is not too great, the LMS algorithm will allow the adjustments of an adaptive system continually to "track" the surface minimum. This process is analysed in some detail in [13], and it is shown there that if the position of the surface minimum changes slowly and randomly, another excess mean-square-error component develops (due to lag in the tracking process) whose amplitude is directly proportional to the square of the time constant of adaptation. This type of excess mean-square error may be considered to be due to adapting too slowly. Thus in the nonstationary case, there are two components and causes of excess mean-square error. It is shown in [13] that their sum is minimized by setting $\tau$ to satisfy the following criterion:

"The rate of adaptation is optimized when the loss of performance resulting from adapting too rapidly equals twice the loss in performance resulting from adapting too slowly."

## OTHER FORMS OF ADAPTIVE FILTERS

The adaptive filter shown in Fig. 1, to which the analysis in this paper relates, has the following attributes: It is discrete, quasi-statically linear, of finite memory (having a finite number of delay-line taps), and it has no signal feedback paths from its output $s_j$ to any of its inputs $\underline{X_j}$. Several other kinds of filters are described in this section which differ from the filter of Fig. 1 is one or more of these attributes.

<u>A Continuous Adaptive Filter</u>

Figure 9a shows a possible configuration for a continuous, quasi-statically-linear filter that uses the adaptive combinatorial system of Fig. 2. The inputs to the variable weights come from a series of continuous filters that may be passive RLC electric circuits. Each of these circuits is assumed to have different sets of poles and zeros. The entire system could have all the poles of all the RLC filters. The variable weights can be regarded as controlling the residues of these poles.

For this system, the mean-square error is a quadratic function of the weights. The LMS algorithm can be used directly to minimize mean-square error. However, the algorithm needs to be expressed in continuous rather than in discrete form as it was in Eq. (45a). The latter could be written:

$$\underline{W_{j+1}} - \underline{W_j} = - 2\, k_s\, \epsilon_j\, \underline{X_j} \quad . \tag{99}$$

The difference equation (99) can be transformed to a differential equation to give a continuous form of the LMS algorithm:

$$\frac{d}{dt}\underline{W(t)} = - 2\, k_s\, \epsilon(t)\, \underline{X(t)} \quad . \tag{100}$$

The weights are then obtained by integration.

$$\underline{W(t)} = - 2\, k_s \int \epsilon(t)\, \underline{X(t)}\, dt \quad . \tag{101}$$

This algorithm can be implemented using analog computing techniques.

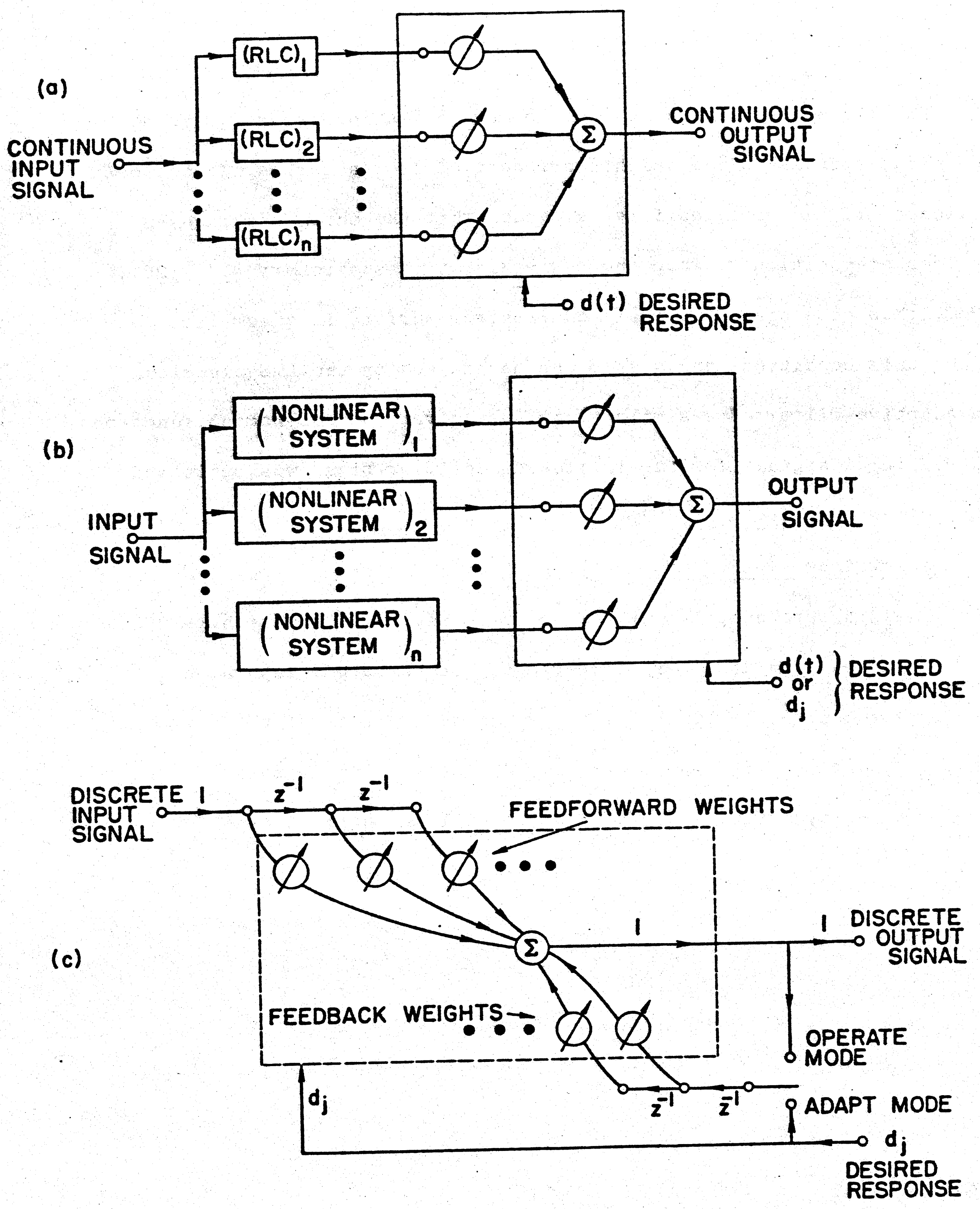


Fig. 9. Additional forms of adaptive filters: (a) continuous; (b) nonlinear; (c) feedback.

## A Nonlinear Adaptive Filter

The filter form shown in Fig. 9b may be continuous or discrete. The input signals to the weights are derived from a series of nonlinear devices. If the nonlinearities are such that the outputs of these devices are stationary when the system input is stationary [1], [25], [26], then once again the mean-square-error surface is quadratic. Under this condition, the system can be adapted by the LMS algorithm. An adaptive filter of a similar form, using various polynomial functions of the input signals for the indicated nonlinearities, was built and analysed by Gabor *et al* [27].

## A Feedback Adaptive Filter

The discrete adaptive filter shown in Fig. 9c has variably-weighted signal feedback paths. This filter does not exhibit a quadratic performance surface. Indeed, the performance surface of this system generally has relative minima. A method for adapting such a filter has been devised by Mantey [28], [29] which will converge on a unique minimum. This method involves breaking the feedback link during the "adapt mode" and exciting the feedback branch with the *desired* response as shown in the figure. In this mode, adaptation can proceed using the LMS algorithm. Mantey has shown that this adaptation yields weights which give the best estimate of the desired response based on weighting the past inputs and the desired responses. If the minimum mean-square error of this minimization is small, then the system when returned to the "operate mode" yields very satisfactory performance.

In the specialized case of a "feedback-only" filter, where only the present input is used in conjunction with the past outputs to form the new output, Mantey has shown for white inputs that use of the LMS algorithm, with the desired response exciting the feedback branch, guarantees a stable system when restored to the "operate mode". The resultant system in this case has a frequency response which is the best least-squares approximation to that implied by the ratio of the transforms of input and desired output. In this sense the result is said to be optimal. For cases where the input is not white and/or feedforward weights are used, the only instabilities generated in the numerous cases which have been simulated could be attributed to problems of computer accuracy.

## CONCLUSIONS

It has been shown that a simple linear combinatorial system coupled to a tapped delay line can be used as the basis for an adaptive filter that adjusts its own parameters to fit the given problem. In this filter, the principle of "performance feedback" is used to control, not the signals, but the "design" of the adaptive filter--i.e., the actual values of the variable parameters (weights). In this form of feedback process, the "feedback error signal" is the gradient of the mean-square-error performance surface, which in many important cases is a quadratic function of the filter adjustments.

The LMS algorithm, based on the method of steepest descent, is used to search the mean-square error surface for a minimum. State-space methods, which are widely used in modern control theory, have been applied to the analysis of stability and time constants. Considerable simplifications in analysis have been realized by expressing transient phenomena of the system adjustments (which take place during the adaptation process) in terms of the normal coordinates of the system. The time constants are given by

$$\tau_p = \frac{1}{2(-k_s)\ \lambda_p} \quad , \quad p = 1, 2, \ldots n ,$$

where $\lambda_p$ is the $p^{th}$ eigenvalue of the matrix $[\Phi(x,x)]$, the input-signal correlation matrix, and $k_s$ is a design parameter which controls the rate of adaptation.

The LMS algorithm uses measured gradient estimates which are noisy but unbiased. The effect of random fluctuations in the weight values due to "gradient-measurement noise" in the LMS-steepest descent process can be minimized by using a sufficiently slow rate of adjustment of the filter parameters.

A measurement M of relative excess mean-square error caused by the adaptation process has been derived and evaluated as

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

The value of the misadjustment depends on the time constants (settling times) of the filter adjustment weights. When all of these time constants are equal, M is proportional to the number of dimensions and inversely proportional to the time constant. That is,

$$M = \frac{n}{2\tau}$$

Although the above results specifically apply to statistically stationary processes, the LMS algorithm can also be used with non-stationary processes, in which case "the rate of adaptation is optimized when the loss of performance resulting from adapting too rapidly equals twice the loss in performance resulting from adapting too slowly."

## Advantages, Limitations, Applications

Adaptive filtering techniques will probably turn out to be most useful under circumstances when almost no *a priori* statistical information is available. In such circumstances, the powerful methods of Wiener and of Kalman and Bucy could not be used well, and the adaptive approach may present the only reasonable possibility.

The LMS algorithm presented here is quite simple to implement, but it does require a "desired response" input signal, at least during adaptation. This is a limitation; however, the desired response can be made available in a number of applications such as noise filtering and prediction, modeling, pattern recognition, certain adaptive control-systems applications, adaptive antennas, adaptive equalizers for telephone systems, and many others. Several of these applications have been successfully realized in the laboratory, either by computer simulation or by actual physical realization of digital and/or analog adaptive circuits. These applications and other extensions of adaptive filtering techniques will be discussed in subsequent papers.

## Suggestions for Future Work

Much work remains to be done in the aforementioned areas of application. New areas of application should be explored both from a theoretical standpoint and from a practical one. The applicability of adaptive filters to statistically nonstationary processes presents some highly challenging mathematical and statistical problems, and perhaps is the area in which the strongest applications of adaptive techniques will be made.

# REFERENCES

1. N. Wiener, Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications. New York: Wiley, 1949.

2. R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory", J. Basic Engineering (Trans. ASME), vol. 83D, 1961.

3. B. Widrow, P. E. Mantey and L. Griffiths, "Adaptive antennas", in preparation.

4. B. Widrow and M. E. Hoff, Jr., "Adaptive switching circuits", 1960 WESCON Conv. Rec., Inst. Radio Engnrs., Part 4, pp. 96-104. First presentation of LMS algorithm; not called this in the paper however.

5. N. G. Nilsson, Learning Machines. New York: McGraw-Hill, 1965.

6. B. Widrow and F. W. Smith, "Pattern-recognizing control systems", 1963 Computer and Information Sciences (COINS) Symposium Proceedings, Washington D. C.: Spartan, 1964.

7. F. W. Smith, "Design of quasi-optimal minimum-time controllers", IEEE Trans. on Automatic Control, vol. AC-11, pp. 71-77, January 1966.

8. R. V. Southwell, Relaxation Methods in Engineering Science. London: Oxford University Press, 1940.

9. D. J. Wilde, Optimum Seeking Methods. Englewood Cliffs, N. J.: Prentice-Hall, 1964.

10. S. J. Mason, "Feedback theory: further properties of signal flow graphs", Proc. IRE, vol. 44, pp. 920-926, July 1956.

11. J. Mason and H. J. Zimmerman, Electronic Circuits, Signals, and Systems. New York: Wiley, 1960.

12. B. Widrow, "Adaptive sampled-data systems--a statistical theory of adaptation", 1959 WESCON Conv. Rec., Inst. Radio Engnrs., Part 4, 1959.

13. B. Widrow, "Adaptive sampled-data systems", Proc. First International Congress of the International Federation of Automatic Control, Moscow, 1960.

14. F. R. Ragazzini and G. F. Franklin, Sampled-data Control Systems. New York: McGraw-Hill, 1958.

15. E. I. Jury, Sampled-data Control Systems. New York: Wiley, 1958.

16. J. T. Tou, Digital and Sampled-data Control Systems. New York: McGraw-Hill, 1959.

References contd.

17. H. Freeman, Discrete-time Systems. New York: Wiley, 1965.

18. C. Lanzos, Applied Analysis. Englewood Cliffs, N. J.: Prentice-Hall, 1956.

19. R. Bellman, Introduction to Matrix Analysis. New York: McGraw-Hill, 1960.

20. F. B. Hildebrand, Methods of Applied Mathematics. Englewood Cliffs, N. J.:Prentice-Hall, 1952.

21. L. A. Zadeh and C. A. DeSoeur, Linear System Theory: The State Space Approach. New York: McGraw-Hill, 1963.

22. P. M DeRusso, R. J. Roy, and C. M. Close, State Variables for Engineers. New York: Wiley, 1965.

23. J. S. Koford and G. F. Groner, "The use of an adaptive threshold element to design a linear optimal pattern classifier," IEEE Trans. on Information Theory, vol. IT-12, January 1966.

24. B. Widrow, Class notes, EE 249, Stanford University, 1966.

25. N. Wiener, Nonlinear Problems in Random Theory. Cambridge, Mass. M.I.T. Press, 1958.

26. A. G. Bose, "A theory of nonlinear systems," Tech. Rept. 309, Research Laboratory of Electronics, M.I.T., Cambridge, Mass., May 15, 1956.

27. D. Gabor, W.P.L. Wilby, and R. Woodcoch, "A universal nonlinear filter predictor and simulator which optimizes itself by a learning process," Proc. Institution of Electrical Engineers, London, vol. 108 B, July 1960.

28. P. E. Mantey, "Convergent automatic-synthesis procedures for sampled-data networks with feedback," Rept. SEL-64-112 (TR No. 6773-1), Stanford Electronics Laboratories, Stanford, Calif. October 1964.

29. P. E. Mantey, "System modeling with discrete recursive systems," a paper submitted to Trans. IEEE, PG-AC.