On the Statistical Efficiency of the LMS Algorithm with Nonstationary Inputs

BERNARD WIDROW, FELLOW, IEEE, AND EUGENE WALACH

***Abstract*—A fundamental relationship exists between the quality of an adaptive solution and the amount of data used in obtaining it. Quality is defined here in terms of "misadjustment," the ratio of the excess mean square error (mse) in an adaptive solution to the minimum possible mse. The higher the misadjustment, the lower the quality is. The quality of the exact least squares solution is compared with the quality of the solutions obtained by the orthogonalized and the conventional least mean square (LMS) algorithms with stationary and nonstationary input data. When adapting with noisy observations, a filter trained with a finite data sample using an exact least squares algorithms will have a misadjustment given by**

$$M = \frac{n}{N} = \frac{\text{number of weights}}{\text{number of training samples}}.$$

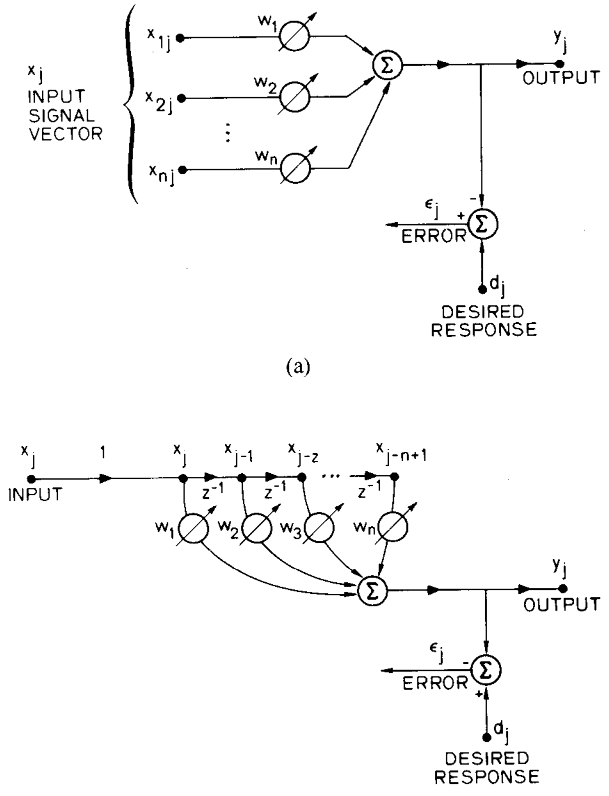
If the same adaptive filter were trained with a steady flow of data using an ideal "orthogonalized LMS" algorithm, the misadjustment would be

$$M = \frac{n}{4\tau_{\text{mse}}} = \frac{\text{number of weights}}{\text{number of training samples}}.$$

Thus, for a given time constant τ_{mse} of the learning process, the ideal orthogonalized LMS algorithm will have about as low a misadjustment as can be achieved, since this algorithm performs essentially as an exact least squares algorithm with exponential data weighting. It is well known that when rapid convergence with stationary data is required, exact least squares algorithms can in certain cases outperform the conventional Widrow–Hoff LMS algorithm. It is shown here, however, that for an important class of nonstationary problems, the misadjustment of conventional LMS is the same as that of orthogonalized LMS, which in the stationary case is shown to perform essentially as an exact least squares algorithm.

I. Introduction

The basic component of most adaptive filtering and signal processing systems is the adaptive linear combiner [1]–[4] shown in Fig. 1. An output signal is formed which is the weighted sum of a set of input signals. The output would be a simple linear combination of the inputs only if the weights were fixed. In actual practice, the weights are adjusted or adapted purposefully; the resulting weight values are signal dependent. This process causes the system behavior during adaptation to differ significantly from that of a linear system. However, when the adaptive process converges and the weights settle to essentially fixed values with only minor random fluctuations about the equilibrium solution, the converged system exhibits essentially linear behavior.



(a)

(b)

Fig. 1. Adaptive linear combiner and its application in an adaptive filter.

Adaptive linear combiners have been successfully used in the modeling of unknown systems [2], linear prediction [2], [5], adaptive noise cancelling [4], adaptive antenna systems [3], channel equalization systems for high-speed digital communications [6]–[9], echo cancellation [10]–[12], systems for instantaneous frequency estimation [13], receivers of narrow-band signals buried in noise (the "adaptive line enhancer") [4], [14], [15], adaptive control systems [16]–[18], and in many other applications.

In Fig. 1(a), the interpretation of the input signal vector $X_j = (x_{1j}, \cdots, x_{nj})^T$, and the desired response d_j might vary depending on how the adaptive linear combiner is used. In Fig. 1(b), an application to adaptive filtering is shown. In turn, an application of adaptive filtering to plant modeling or system identification is shown in Fig. 2. Here, we can view the desired response d_j as a linear combination of the last n samples of the input signal vector, corrupted by a certain independent zero-mean plant noise n_j. Our

Manuscript received May 11, 1983; revised July 4, 1983.

B. Widrow is with Department of Electrical Engineering, Stanford University, CA 94305.
E. Walach is with IBM Corp., Watson Research Center, Yorktown Heights, NY.

0018-9448/84/0300-0211$01.00 ©1984 IEEE

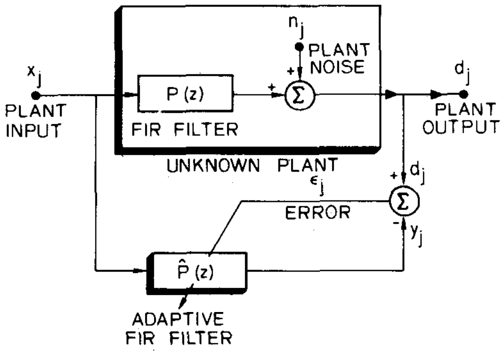


Fig. 2. Adaptive plant identification.

aim in this application is to estimate an unknown plant (represented by its transfer function $P(z) = w_1^* + \cdots + w_n^* z^{-n+1}$) through the minimization of the output error ϵ_j in the mean-square sense. For purposes of analysis, we consider the plant to be a transversal finite impulse response (FIR) filter. Referring to Figs. 1(a) and (b), the input signal vector at the jth sampling instant is designated by

$$X_j = (x_j, \cdots, x_{j-n+1})^T, \tag{1}$$

and the set of the weights of adaptive transversal filter is designated by

$$W = (w_1, \cdots, w_n)^T. \tag{2}$$

The jth output signal is

$$y_j = \sum_{i=1}^{n} w_i x_{j-i+1} = W^T X_j = X_j^T W. \tag{3}$$

The input signals and desired response are assumed to be stationary ergodic processes. Denoting the desired response as d_j, the error at the jth time is

$$\epsilon_j = d_j - y_j = d_j - W^T X_j = d_j - X_j^T W. \tag{4}$$

The square of this error is

$$\epsilon_j^2 = d_j^2 - 2d_j X_j^T W + W^T X_j X_j^T W. \tag{5}$$

The mean-square error (mse) ξ, the expected value of ϵ_j^2, is

$$\begin{aligned} \xi &\triangleq E[\epsilon_j^2] \\ &= E[d_j^2] - 2E[d_j X_j^T] W + W^T E[X_j X_j^T] W \\ &= E[d_j^2] - 2P^T W + W^T R W, \end{aligned} \tag{6}$$

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

$$E[d_j X_j] = E \begin{bmatrix} d_j x_j \\ \vdots \\ d_j x_{j-n+1} \end{bmatrix} \triangleq P, \tag{7}$$

and the input autocorrelation matrix R is defined as

$$E[X_j X_j^T] = E \begin{bmatrix} x_j x_j & \cdots & x_j x_{j-n+1} \\ x_{j-1} x_j & \cdots & x_{j-1} x_{j-n+1} \\ \vdots & & \vdots \\ x_{j-n+1} x_j & \cdots & x_{j-n+1} x_{j-n+1} \end{bmatrix} \triangleq R. \tag{8}$$

It can be observed from (6) that with stationary ergodic inputs the mean square error performance function is a quadratic function of the weights, a paraboloidal "bowl" that has a unique minimal point for

$$W = W^* = R^{-1} P. \tag{9}$$

In practice, of course, we do not know the exact statistics R and P. One way of finding an estimate of the optimal weight vector W^* would be to estimate R and P for the given input and given desired response. This approach would lead to what is called an exact least mean square solution. This approach is optimal in a sense that the sum of square errors will be minimal for the given data sample. However, such solutions are generally somewhat complex from the computational point of view [19]–[21]. On the other hand one can use one of the simpler gradient search algorithms such as the least mean square (LMS) steepest descent algorithm of Widrow and Hoff. However, this algorithm is sometimes associated with a certain deterioration in performance with problems for which there exists great spread between the eigenvalues of the autocorrelation matrix R (see, for instance, [20]–[21]).

In order to establish a bridge between the LMS and the exact least squares approaches mentioned above, we will introduce an idealized orthogonalized LMS algorithm. For the implementation of this algorithm, we will have to assume perfect knowledge of the autocorrelation matrix R. Naturally that means that this idealized algorithm cannot be used in practice. However, its performance provides a convenient theoretical benchmark for the sake of comparison.[1]

In the next section we will analyze briefly the performance of the exact least squares solution when the weights are obtained with a finite data sample. Then in Sections III and IV we will analyze the idealized LMS algorithm and show, at least heurstically, that its performance is equivalent to that of an exact least squares algorithm. Based on this heuristic argument, we will view the idealized orthogonalized LMS process as an "optimal" gradient search algorithm.

In Section V we will define a class of nonstationary problems: problems in which an unknown plant $P(z)$ varies in a certain random way. Once again, the adaptive filter performs a modeling task. For this class of frequently encountered problems, we will analyze and compare the performances of the orthogonal LMS and the conventional steepest descent LMS algorithms. We will show that both perform equivalently (in the mean) for this class of nonstationary problems.

II. Misadjustment of Finite - Data Exact Least - Squares Processes

Suppose that the adaptive linear combiner in Fig. 2 is fed N independent zero-mean $n \times 1$ data vectors

[1] It should be noted that there are numerous algorithms in the literature which recursively estimate the autocorrelation matrix R and use this estimation for the purpose of orthogonalization of the input data (see, for instance, [28]–[32]). These algorithms asymptotically converge to the idealized algorithm discussed here.

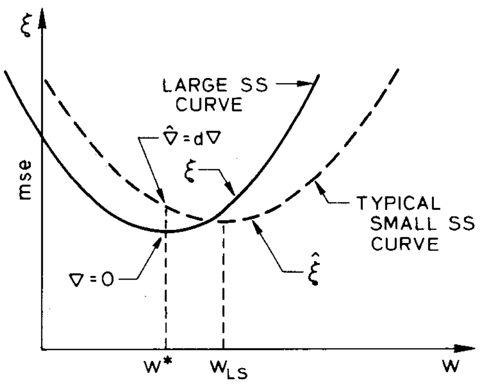


Fig. 3. Small and large sample-size mean square error curves.

$X_1, X_2, \cdots, X_N$ and their respective scalar desired responses $d_1, d_2, \cdots, d_N$, all drawn from a stationary ergodic process. Keeping the weights fixed, a set of N error equations can be written as

$$\epsilon_i = d_i - X_i^T W, \qquad i = 1, 2, \cdots, N. \tag{10}$$

The objective is to find a weight vector that minimizes the sum of the squares of the error values based on the finite sample of N items of data.

Eq. (10) can be written in matrix form as

$$\boldsymbol{\epsilon} = \boldsymbol{D} - \chi \boldsymbol{W}, \tag{11}$$

where χ is an $N \times n$ rectangular matrix

$$\chi \triangleq [X_1, X_2, \cdots, X_N]^T, \tag{12}$$

where $\boldsymbol{\epsilon}$ is an N element error vector

$$\boldsymbol{\epsilon} \triangleq [\epsilon_1, \epsilon_2, \cdots, \epsilon_N]^T, \tag{13}$$

and where $\boldsymbol{D}$ is an N element vector of desired responses

$$\boldsymbol{D} \triangleq [d_1, d_2, \cdots, d_N]^T. \tag{14}$$

A unique solution of (11), a weight vector $\boldsymbol{W}$ that would bring $\boldsymbol{\epsilon}$ to zero, would exist only if χ is square and nonsingular. However, the case of greatest interest is that of $N \gg n$. As such, (11) would typically be overconstrained and one would generally seek a best least squares solution. The sum of the squares of the errors is

$$\boldsymbol{\epsilon}^T \boldsymbol{\epsilon} = \boldsymbol{D}^T \boldsymbol{D} + \boldsymbol{W}^T \chi^T \chi \boldsymbol{W} - 2\boldsymbol{D}^T \chi \boldsymbol{W}. \tag{15}$$

This sum multiplied by $1/N$ is an estimate $\hat{\xi}$ of the mse ξ. Thus

$$\hat{\xi} = \frac{1}{N} \boldsymbol{\epsilon}^T \boldsymbol{\epsilon}$$

and

$$\lim_{N \to \infty} \hat{\xi} = \xi. \tag{16}$$

Note that $\hat{\xi}$ is a quadratic function of the weights, the parameters of the quadratic form being related to properties of the N data samples. $(\chi^T \chi)$ is square and is assumed to be positive definite. $\hat{\xi}$ is a small sample-size mse function, while ξ is the large sample-size "true" mse function. Fig. 3 shows a comparative sketch of these functions. Many small sample-size curves are possible, but there is only one large sample-size curve. The large sample-size curve is the average of the many small sample-size curves.

The minimum of the small sample-size function can be found by differentiating (15), and the result is

$$\boldsymbol{W}_{\mathrm{LS}} = (\chi^T \chi)^{-1} \chi^T \boldsymbol{D}. \tag{17}$$

This is the exact least squares solution for the given data sample. The Wiener solution $\boldsymbol{W}^*$ is the expected value of $\boldsymbol{W}_{\mathrm{LS}}$.[2]

Each of the small sample-size curves of Fig. 3 is an ensemble member. Let the ensemble be constructed in the following manner. Assume that the vectors $X_1, X_2, \cdots, X_N$ are the same for all ensemble members, but that the associated desired responses $d_1, d_2, \cdots, d_N$ differ from one ensemble member to another because of the stochastic character of plant noise (refer to Fig. 2). Over this ensemble therefore, the χ matrix is constant while the desired response vector $\boldsymbol{D}$ is stochastic. In order to evaluate the excess mean square error due to adaptation with the finite amount of data available, we have to find

$$\xi_{\mathrm{excess}} = \frac{1}{N} E[\boldsymbol{V}^T \chi^T \chi \boldsymbol{V}], \tag{18}$$

where

$$\boldsymbol{V} \triangleq \boldsymbol{W}_{\mathrm{LS}} - \boldsymbol{W}^*. \tag{19}$$

Expectation is taken over the above-described ensemble. Eq. (18) can be written as

$$\begin{aligned} \xi_{\mathrm{excess}} &= \frac{1}{N} \operatorname{tr}(E[\boldsymbol{V}^T \chi^T \chi \boldsymbol{V}]) = \frac{1}{N} E[\operatorname{tr}(\boldsymbol{V}^T \chi^T \chi \boldsymbol{V})] \\ &= \frac{1}{N} E[\operatorname{tr}(\boldsymbol{V}\boldsymbol{V}^T \chi^T \chi)] = \frac{1}{N} \operatorname{tr}(E[\boldsymbol{V}\boldsymbol{V}^T] \cdot \chi^T \chi). \end{aligned} \tag{20}$$

The covariance matrix of the weight error vector $\boldsymbol{V}$ is known to be (see, for instance, [22])[3]

$$E(\boldsymbol{V}\boldsymbol{V}^T) = [\chi^T \chi]^{-1} \cdot \xi_{\min}, \tag{21}$$

where $\xi_{\min}$ is the minimum mean square error, the minimum of the true mse function.

Substitution of (21) into (20) yields

$$\xi_{\mathrm{excess}} = \frac{n}{N} \cdot \xi_{\min}. \tag{22}$$

It is important to note that this formula does not depend on χ. The above described ensemble can be generalized to an ensemble of ensembles, each having its own χ, without changing formula (22). Hence this formula is valid for a very wide class of inputs.

It is useful to consider a dimensionless ratio between the excess mean square error and the minimum mean square error. This ratio is called in the literature (see, for instance, [1], [2], [4]) the misadjustment M. For the exact least square

[2] Note that validity of this statement is contingent on our earlier assumption that the plant $P(z)$ is FIR and both plant and plant model have the same order n.

[3] For the sake of simplicity we assumed here that the plant noise n_j in Fig. 2 is white and that the adaptive filter has enough weights to match the unknown plant.

solution we can find the misadjustment (due to the finite length of input data) from (22) as

$$M = \frac{n}{N} = \frac{\text{(number of weights)}}{\text{(number of independent training samples)}}. \tag{23}$$

This misadjustment formula was first presented without detailed proof by Widrow and Hoff [1] in 1960, and has been used for many years in pattern recognition studies. For small values of M (less than 25 percent), it has proven an excellent approximation. A formula similar to (23), based on somewhat different assumptions, was derived by Davisson [23] in 1970.

III. Stochastic Gradient Search by Steepest Descent

Gradient methods are commonly used to adjust adaptive parameters in order to search the quadratic mean square error performance function for its minimum. Most widely used is the method of steepest descent. With this method, a sequence of changes is made in the weight vector along the direction of the negative gradient. Thus the next weight vector $\boldsymbol{W}_{j+1}$ is made equal to the present weight vector $\boldsymbol{W}_j$ plus a change proportional to the negative gradient at the jth iteration:

$$\boldsymbol{W}_{j+1} = \boldsymbol{W}_j + \mu(-\nabla_j). \tag{24}$$

The parameter μ controls stability and rate of convergence. An "instantaneous gradient" $\hat{\nabla}$, an estimate of the true gradient ∇_j, can be found by differentiation of (5) with respects to $\boldsymbol{W}$:

$$\hat{\nabla}_j = -2\epsilon_j \boldsymbol{X}_j. \tag{25}$$

Using the instantaneous gradient in place of the true gradient in (24) yields the LMS algorithm of Widrow and Hoff,

$$\boldsymbol{W}_{j+1} = \boldsymbol{W}_j + 2\mu\epsilon_j \boldsymbol{X}_j. \tag{26}$$

The behavior of algorithm (26) has been analyzed extensively in the literature (see, for instance, [2], [3], [4], [18], [24], [25]). It was proved in [2], [4] that if the adaptation constant μ was chosen such that

$$0 < \mu < \frac{1}{\operatorname{tr}(\boldsymbol{R})}, \tag{27}$$

then the adaptive weights will relax from their initial condition to the Wiener solution[4] $\boldsymbol{W}^*$. That means that the weight error vector

$$\boldsymbol{V}_j = \boldsymbol{W}_j - \boldsymbol{W}^* \tag{28}$$

will converge to zero in the mean. The relaxation process will be governed by the relation

$$E\left[\boldsymbol{V}_{j+1}\right] = (I - 2\mu\boldsymbol{R})E\left[\boldsymbol{V}_j\right]. \tag{29}$$

Therefore, there will be n different modes of convergence corresponding to n eigenvalues of the autocorrelation matrix $\boldsymbol{R}$. Using normal decomposition of the matrix $\boldsymbol{R}$:

$$\boldsymbol{R} = Q\Lambda Q^T \tag{30}$$

$$QQ^T = I, \qquad \Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix}, \tag{31}$$

we can find the corresponding n time constants τ_i of the weight relaxation process as

$$\mu\lambda_i \ll 1, \quad \tau_i = \frac{1}{2\mu\lambda_i}, \qquad 1 \le i \le n. \tag{32a}$$

As the weights relax toward the Wiener solution, the mean square error, a quadratic function of the weights, undergoes a geometric progression toward $\xi_{\min}$. The "learning curve" is a plot of mse versus number of adaptation cycles. The natural modes of the learning curve have time constants half as large as the corresponding time constants of the weights [2]–[4]. Accordingly, the mse (associated with the error $E[\boldsymbol{W}_j] - \boldsymbol{W}^*$) learning curve time constants are

$$\tau_{i_{\text{mse}}} = \frac{1}{4\mu\lambda_i}, \qquad 1 \le i \le n. \tag{32b}$$

After convergence takes place, there remains noise in the weights due to the noise in the estimation (25) of the gradient. An approximate value of the covariance of the weight noise, valid for small μ, was derived in [4, app. D]:

$$E\left[\boldsymbol{V}_j\boldsymbol{V}_j^T\right] = \mu \cdot \xi_{\min} \cdot I. \tag{33}$$

The noise in the weights will cause an excess error in the system output (in addition to the Wiener error):

$$\begin{aligned}\xi_{\text{excess}} &= E\left[\left(\boldsymbol{V}_j^T\boldsymbol{X}_j\right)^2\right] \\ &= E\left[\operatorname{tr}\left(\boldsymbol{V}_j\boldsymbol{V}_j^T\boldsymbol{X}_j\boldsymbol{X}_j^T\right)\right].\end{aligned} \tag{34}$$

Assuming, as has been done before [2]–[4], that $\boldsymbol{V}_j$ and $\boldsymbol{X}_j$ are independent, expression (33) can be substituted into (34) to obtain

$$\xi_{\text{excess}} = \mu\xi_{\min} nE\left[x_j^2\right] = \mu \operatorname{tr}(\boldsymbol{R})\xi_{\min}. \tag{35}$$

Therefore, we can compute the misadjustment, defined as a ratio between the excess mean square and the minimum mean square error:

$$M = \mu \operatorname{tr}(\boldsymbol{R}). \tag{36}$$

The adaptation constant μ should be kept low in order to keep the misadjustment low. However, low μ is associated with slow adaptation in accordance with (32).

Expressions (27)–(36) illustrate the potential vulnerability of the steepest descent algorithm. The speed of convergence will depend on the choice of initial conditions. In the worst case, the convergence will be dominated by the lowest eigenvalue

$$\lambda_{\min} = \min(\lambda_1, \cdots, \lambda_n). \tag{37}$$

This implies that even if we choose the maximal value allowable for the adaptation constant μ (due to the stability

[4] In [2]–[4] it was proved that (27) is sufficient for the convergence in the mean. However it can be shown that this condition is both sufficient and necessary for the convergence of the variance. See also [33]–[35].

constraint (27)), the slowest time constant for the weights would be

$$\tau \geq \frac{\operatorname{tr}(\boldsymbol{R})}{2\lambda_{\min}}. \tag{38}$$

For the class of problems for which there exists a great spread of eigenvalues of the autocorrelation matrix $\boldsymbol{R}$, this number will be high resulting in long convergence times (at least in the worst case).

IV. Gradient Search by the Orthogonalized LMS Algorithm

In order to eliminate a potential deficiency of the Widrow–Hoff algorithm due to eigenvalue spread of the input autocorrelation matrix, we can prefilter the input signal in such a way that it will become orthogonal. Such a process would require perfect knowledge of the autocorrelation matrix $\boldsymbol{R}$. Hence such an algorithm is not able to be implemented practically, although it is important from a theoretical point of view.

The block diagram of the idealized orthogonal LMS algorithm used in a modeling process is presented in the Fig. 4. In Fig. 4(a), each input signal vector to the adaptive process is preprocessed by the orthogonalization matrix

$$\boldsymbol{R}^{-1/2} = Q\begin{pmatrix} \lambda_1^{-1/2} & & 0 \\ & \ddots & \\ 0 & & \lambda_n^{-1/2} \end{pmatrix} Q^T = Q\Lambda^{-1/2}Q^T. \tag{39}$$

Refer next to Fig. 4(b). Clearly this system is equivalent to that of Fig. 4(a) because the matrices $\boldsymbol{R}^{1/2}$ and $\boldsymbol{R}^{-1/2}$ incorporated into the unknown plant cancel each other. Denoting

$$\boldsymbol{X}_j^\dagger \triangleq \boldsymbol{R}^{-1/2}\boldsymbol{X}_j, \tag{40a}$$

$$\boldsymbol{W}_j^\dagger \triangleq \boldsymbol{R}^{1/2}\boldsymbol{W}_j, \tag{40b}$$

and

$$\boldsymbol{W}^{\dagger *} \triangleq \boldsymbol{R}^{1/2}\boldsymbol{W}^*, \tag{41}$$

we can transform the system of Fig. 4(b) to that of Fig. 4(c). The adaptive process of Fig. 4(c) is, in turn, equivalent to the conventional LMS adaptive process of Fig. 1 and 2 with input signal $\boldsymbol{X}_j^\dagger$ guaranteed to be orthogonalized. The plant impulse response is now modified to be $\boldsymbol{W}^{\dagger *}$. From the systems equivalences of Fig. 4, we conclude that the orthogonalized LMS algorithm of Fig. 4(a) will perform exactly as the conventional steepest descent algorithm of Widrow and Hoff fed by an input signal with autocorrelation matrix

$$\boldsymbol{R}^\dagger \triangleq E\left[\boldsymbol{X}_j^\dagger\left(\boldsymbol{X}_j^\dagger\right)^T\right] = \boldsymbol{R}^{-1/2}E\left[\boldsymbol{X}_j\boldsymbol{X}_j^T\right]\boldsymbol{R}^{-1/2} = \boldsymbol{I}. \tag{42}$$

Therefore we can, evaluate the performance of the idealized orthogonalized LMS algorithm, using the well-known expressions (25)–(36).

The adaptation rule of the orthogonalized algorithm will be

$$\boldsymbol{W}_{j+1}^\dagger = \boldsymbol{W}_j^\dagger + 2\mu^\dagger\epsilon_j\boldsymbol{R}^{-1/2}\boldsymbol{X}_j = \boldsymbol{W}_j^\dagger + 2\mu^\dagger\epsilon_j\boldsymbol{X}_j^\dagger. \tag{43}$$

According to (27), for a choice of adaptation constant in the range

$$0 < \mu^\dagger < \frac{1}{\operatorname{tr}(\boldsymbol{R}^\dagger)} = \frac{1}{n}, \tag{44}$$

the algorithm will converge to the optimal (Wiener) solution $\boldsymbol{W}^{\dagger *}$. Since all the eigenvalues of $\boldsymbol{R}^\dagger$ are equal to unity there will be only one mode of convergence and only one time constant. According to (32a) and (32b),

$$\tau = \frac{1}{2\mu^\dagger} \tag{45a}$$

$$\tau_{\text{mse}} = \frac{1}{4\mu^\dagger}. \tag{45b}$$

These time constants will hold for an arbitrary choice of initial conditions. Therefore an idealized algorithm would indeed solve the problem of worst case slow convergence of the conventional steepest descent algorithm.

Next we will analyze the orthogonalized LMS algorithm to determine how close it brings us to the performance of the exact least mean square algorithm described in Section II.

According to (44) and (45a) the minimal possible time constant for the weights which is consistent with algorithm stability is

$$\tau_{\min} = \frac{n}{2}. \tag{46}$$

Completion of the convergence process will require several time constants or, in other words, a number of input samples which will be of the order of magnitude of n. This is consistent with the fact that the exact least mean square algorithm receives at least n samples in order to compute an estimation of n weights of the unknown plant.

After convergence takes place, the weights of the model remain noisy (due to the noise in the instantaneous estimation of the gradient). We will denote by $\boldsymbol{V}_j^\dagger$ the weight error vector

$$\boldsymbol{V}_j^\dagger = \boldsymbol{W}_j^\dagger - \boldsymbol{W}^{\dagger *} = \boldsymbol{W}_j^\dagger - \boldsymbol{R}^{1/2}\boldsymbol{W}^*. \tag{47}$$

According to (33), the weight error covariance matrix is

$$E\left[\boldsymbol{V}_j^\dagger\left(\boldsymbol{V}_j^\dagger\right)^T\right] = \mu^\dagger \cdot \xi_{\min} \cdot I. \tag{48}$$

The noise in the adaptive weights will cause misadjustment which can be found from (36) as the ratio of excess mse to minimum mse:

$$M^\dagger = \frac{\begin{pmatrix}\text{excess mean}\\ \text{square error}\end{pmatrix}}{\begin{pmatrix}\text{Wiener error}\\ \text{power}\end{pmatrix}} = \mu^\dagger \operatorname{tr}(\boldsymbol{R}^\dagger) = \mu^\dagger \cdot n. \tag{49}$$

Substituting (45b) into (49) yields

$$M^\dagger = \frac{n}{4\tau_{\text{mse}}}. \tag{50}$$

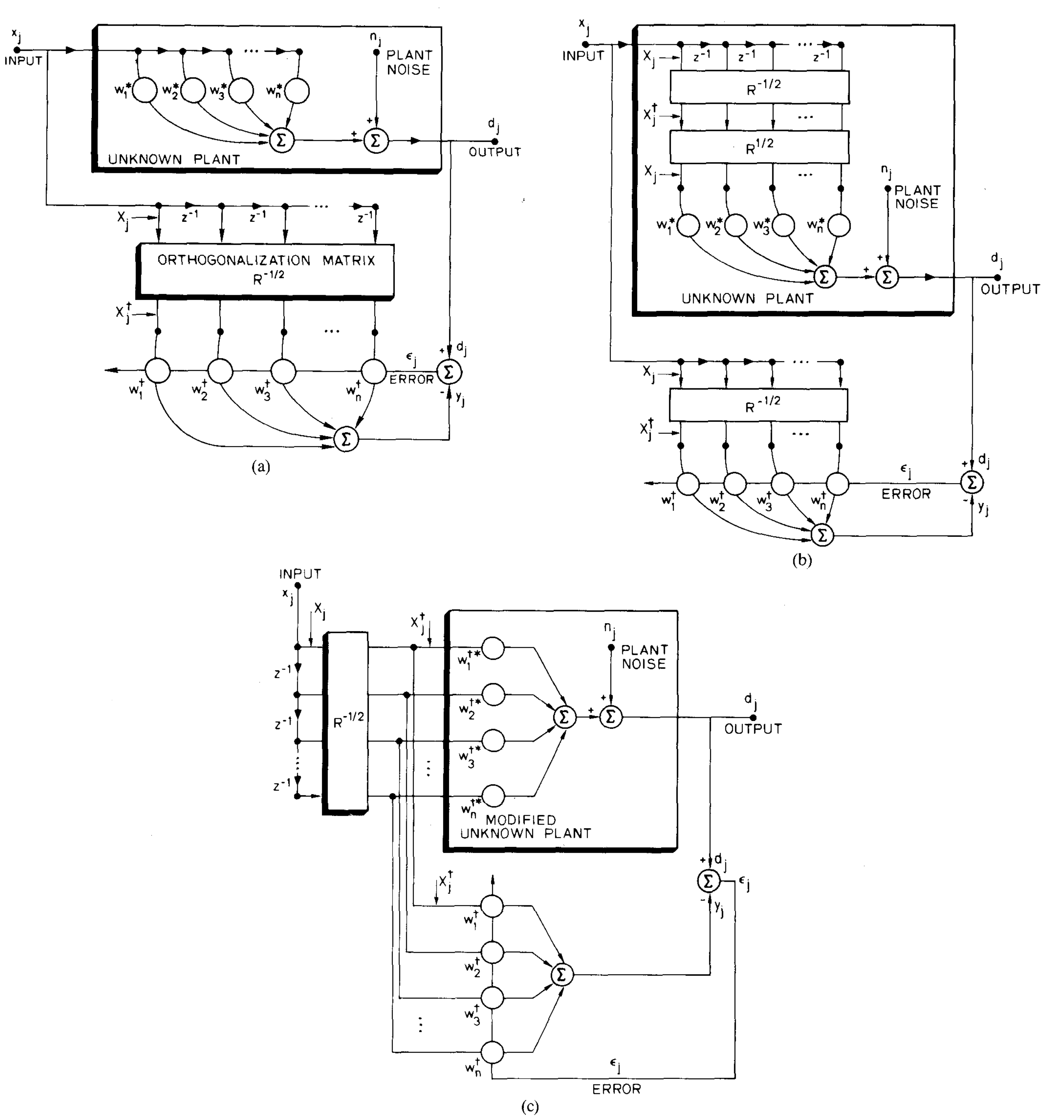


Fig. 4. Orthogonalized LMS algorithm applied to plant identification. (a) Adaptation with orthogonalized inputs. (b) Adaptation with transformed plant inputs. (c) Modeling modified plant.

The question is, how does this compare with the misadjustment $M = n/N$ of the exact least mean squares algorithm found in Section II. Since we wish to compare the misadjustment of an exact least squares process that adapts with a finite equally weighted data sample to the orthogonalized LMS algorithm which weights its input data exponentially, we are to some extent comparing "apples with oranges." However, at least from a heuristic point of view, both (23) and (50) are equivalent.

The orthogonalized LMS algorithm exponentially weights its input data over time as it establishes its weight values. An exponential averaging window moves with time. The settling time of the adaptive process is of the order of four time constants of the mse learning curve. At any moment, the weights are determined by adaptation taken place over the last four time constants worth of data. Thus, in a steady flow situation, the training data "consumed" at any time by the orthogonalized LMS algorithm is essentially

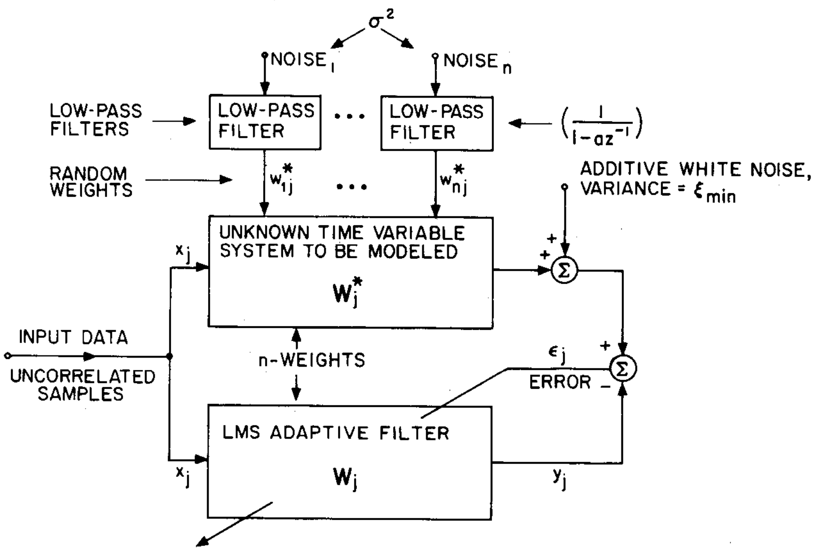


Fig. 5. Modeling an unknown time-variable system.

the most recent $4\tau_{mse}$ samples. The misadjustment of the orthogonalized LMS can therefore be expressed as

$$M^\dagger = \frac{n}{4\tau_{mse}} = \frac{\text{(number of weights)}}{\text{(number of independent training samples)}}. \tag{51}$$

It is clear from this expression that the orthogonalized LMS algorithm uses its training data about as efficiently as an exact least squares process.

V. Orthogonalized LMS Versus Conventional LMS: Nonstationary Inputs

Filtering nonstationary signals is a major area of application for adaptive systems. When the statistical character of an input signal changes gradually, randomly, and unpredictably, a filtering system that can automatically optimize its input–output response in accord with the requirements of the input signal could yield superior performance relative to that of a fixed, nonadaptive system. The performance of the conventional steepest descent LMS algorithm is compared here with orthogonalized steepest descent LMS (which, as demonstrated in the previous section, possesses certain optimality qualities), when both algorithms are used to adapt transversal filters with nonstationary inputs. The nonstationary situations to be studied are highly simplified, but they retain the essence of the problem that is common to more complicated and realistic situations.

The example considered here involves modeling or identifying an unknown time-variable plant, assumed to be transversal, of length n, whose weights (impulse response values) undergo independent stationary ergodic first-order Markov processes, as indicated in Fig. 5. The plant input signal x_j is assumed to be stationary, ergodic, and, in general, colored. Additive plant output noise, assumed to be stationary, of mean zero, and of variance ξ_{min}, prevents a perfect match between the unknown system and the adaptive system. The minimum mse is, therefore, ξ_{min}, achieved whenever the n weights of the adaptive filter W_j match the corresponding n weights of the unknown plant. The latter are at every instant the optimal values for the corresponding weights of the adaptive filter and are designated W_j^*, the subscript j indicating that the unknown "target" to be tracked is time variable. It is tempting to call the target W_j^* a time variable Wiener solution, but this would be improper since the nonstationary nature of the problem is beyond classical Wiener theory.

According to the scheme of Fig. 5, minimizing mse causes the adaptive weight vector W_j to attempt to best match the unknown W_j^* on a continual basis. The $\boldsymbol{R}$ matrix, dependent only on the statistics of x_j, is constant even as W_j^* varies. The desired response of the adaptive filter d_j is nonstationary, containing the output of a time-variable system. The minimum mse ξ_{min} is constant. Thus the mse function, a quadratic bowl, varies in position while its eigenvalues, eigenvectors, and ξ_{min} remain constant. The adaptive process has the task of tracking the bottom of a randomly moving bowl in the presence of gradient noise.

In order to study this form of nonstationary adaptation both analytically and by computer simulation, a model comprising an ensemble of nonstationary adaptive processes has been defined and constructed as illustrated in Fig. 6. The ensemble of unknown filters to be modeled are all identical copies of the unknown plant and have the same time-variable weight vector W_j^*. Each ensemble member has its own independent stationary input signal going to both the unknown system and the corresponding adaptive system. The effect of output noise in the unknown systems is obtained by the addition of independent noise of variance ξ_{min}. All of the adaptive filters are assumed to start with the same initial weight vector W_o, each develops its own weight vector over time in attempting to pursue the Markovian target W_j^*.

The class of problems defined above was analyzed in [26], using the conventional steepest descent algorithm as an adaptation rule. Presently our aim will be to modify this line of reasoning in order to evaluate the nonstationary

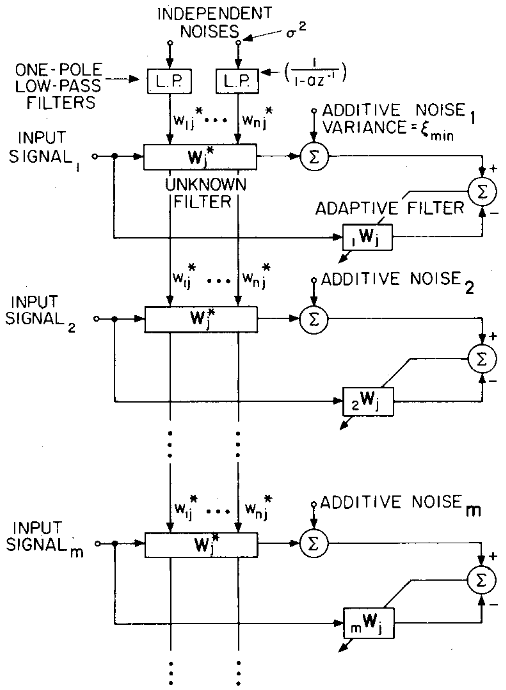


Fig. 6. Ensemble of nonstationary adaptive processes.

performance of the orthogonalized LMS algorithm. We use once more the equivalent representations of Fig. 4. Clearly the adaptive model will attempt to track the moving target

$$\boldsymbol{W}_j^{\dagger *} \triangleq \boldsymbol{R}^{1/2}\boldsymbol{W}_j^*. \tag{52}$$

For a given adaptive filter, the weight vector tracking error at the jth instant is $(\boldsymbol{W}_j^\dagger - \boldsymbol{W}_j^{\dagger *})$. This error is due to both the effects of gradient noise and weight vector lag, and may be expressed as

$$\begin{aligned}(\text{weight vector error})_j &= \left(\boldsymbol{W}_j^\dagger - \boldsymbol{W}_j^{\dagger *}\right)\\ &= \underbrace{\left(\boldsymbol{W}_j^\dagger - E\left[\boldsymbol{W}_j^\dagger\right]\right)}_{\substack{\text{weight vector}\\ \text{noise}}} + \underbrace{\left(E\left[\boldsymbol{W}_j^\dagger\right] - \boldsymbol{W}_j^{\dagger *}\right)}_{\substack{\text{weight vector}\\ \text{lag}}}.\end{aligned} \tag{53}$$

The expectations are averages over the ensemble. Time averages are impossible to use with nonstationary statistics.

The components of error are identified in (53). Any difference between the ensemble mean of the adaptive weight vectors and the target value $\boldsymbol{W}_j^{\dagger *}$ is due to lag in the adaptive process, i.e., due to the adapting weight vectors lagging behind the moving target $\boldsymbol{W}_j^{\dagger *}$. The deviation of the individual adaptive weight vectors about their ensemble mean is due to gradient noise.

Weight vector error causes an excess mse. The ensemble average excess mse at the jth instant is

$$\begin{pmatrix}\text{ensemble}\\ \text{avg excess}\\ \text{mse}\end{pmatrix}_j = E\left[\left(\boldsymbol{W}_j^\dagger - \boldsymbol{W}_j^{\dagger *}\right)^T \boldsymbol{R}^\dagger \left(\boldsymbol{W}_j^\dagger - \boldsymbol{W}_j^{\dagger *}\right)\right]. \tag{54a}$$

The average excess mse at time j is

$$\begin{aligned}\xi_{\text{excess}} &= \begin{pmatrix}\text{avg excess}\\ \text{mse}\end{pmatrix}\\ &= \left\langle E\left[\left(\boldsymbol{W}_j^\dagger - \boldsymbol{W}_j^{\dagger *}\right)^T \boldsymbol{R}^\dagger \left(\boldsymbol{W}_j^\dagger - \boldsymbol{W}_j^{\dagger *}\right)\right]\right\rangle.\end{aligned} \tag{54b}$$

Using (53), this can be expanded as follows:

$$\begin{aligned}&\begin{pmatrix}\text{avg excess}\\ \text{mse}\end{pmatrix}\\ &= \Big\langle E\left[\left(\boldsymbol{W}_j^\dagger - E\left[\boldsymbol{W}_j^\dagger\right]\right)^T \boldsymbol{R}^\dagger \left(\boldsymbol{W}_j^\dagger - E\left[\boldsymbol{W}_j^\dagger\right]\right)\right]\\ &\quad + E\left[\left(E\left[\boldsymbol{W}_j^\dagger\right] - \boldsymbol{W}_j^{\dagger *}\right)^T \boldsymbol{R}^\dagger \left(E\left[\boldsymbol{W}_j^\dagger\right] - \boldsymbol{W}_j^{\dagger *}\right)\right]\\ &\quad + 2E\left[\left(\boldsymbol{W}_j^\dagger - E\left[\boldsymbol{W}_j^\dagger\right]\right)^T \boldsymbol{R}^\dagger \left(E\left[\boldsymbol{W}_j^\dagger\right] - \boldsymbol{W}_j^{\dagger *}\right)\right]\Big\rangle.\end{aligned} \tag{55}$$

Expanding the last term of (55) and simplifying since $\boldsymbol{W}_j^\dagger$ is constant over the ensemble,

$$\begin{aligned}&\Big\langle 2E\left[\boldsymbol{W}_j^{\dagger T}\boldsymbol{R}^\dagger E\left[\boldsymbol{W}_j^\dagger\right] - \boldsymbol{W}_j^{\dagger T}\boldsymbol{R}^\dagger \boldsymbol{W}_j^{\dagger *}\right.\\ &\qquad \left. - E\left[\boldsymbol{W}_j^\dagger\right]^T \boldsymbol{R}^\dagger E\left[\boldsymbol{W}_j^\dagger\right] + E\left[\boldsymbol{W}_j^\dagger\right]^T \boldsymbol{R}^\dagger \boldsymbol{W}_j^{\dagger *}\right]\Big\rangle\\ &\quad = \Big\langle 2E\left[\boldsymbol{W}_j^\dagger\right]^T \boldsymbol{R}^\dagger E\left[\boldsymbol{W}_j^\dagger\right] - E\left[\boldsymbol{W}_j^\dagger\right]^T \boldsymbol{R}^\dagger E\left[\boldsymbol{W}_j^\dagger\right]\\ &\qquad - E\left[\boldsymbol{W}_j^\dagger\right]^T \boldsymbol{R}^\dagger \boldsymbol{W}_j^{\dagger *} + E\left[\boldsymbol{W}_j^\dagger\right]^T \boldsymbol{R}^\dagger \boldsymbol{W}_j^{\dagger *}\Big\rangle\\ &\quad = 0.\end{aligned} \tag{56}$$

Therefore, (55) becomes

$$\begin{aligned}&\begin{pmatrix}\text{avg excess}\\ \text{mse}\end{pmatrix}\\ &= \Big\langle E\left[\left(\boldsymbol{W}_j^\dagger - E\left[\boldsymbol{W}_j^\dagger\right]\right)^T \boldsymbol{R}^\dagger \left(\boldsymbol{W}_j^\dagger - E\left[\boldsymbol{W}_j^\dagger\right]\right)\right]\\ &\quad + E\left[\left(E\left[\boldsymbol{W}_j^\dagger\right] - \boldsymbol{W}_j^{\dagger *}\right)^T \boldsymbol{R}^\dagger \left(E\left[\boldsymbol{W}_j^\dagger\right] - \boldsymbol{W}_j^{\dagger *}\right)\right]\Big\rangle.\end{aligned} \tag{57}$$

The average excess mse is thus a sum of components due to both gradient noise and lag:

$$\begin{aligned}&\begin{pmatrix}\text{avg excess}\\ \text{mse due to lag}\end{pmatrix}\\ &= \left\langle E\left[\left(E\left[\boldsymbol{W}_j^\dagger\right] - \boldsymbol{W}_j^{\dagger *}\right)^T \boldsymbol{R}^\dagger \left(E\left[\boldsymbol{W}_j^\dagger\right] - \boldsymbol{W}_j^{\dagger *}\right)\right]\right\rangle\\ &= \left\langle E\left[\left(E\left[\boldsymbol{W}_j^\dagger\right] - \boldsymbol{W}_j^{\dagger *}\right)^T \left(E\left[\boldsymbol{W}_j^\dagger\right] - \boldsymbol{W}_j^{\dagger *}\right)\right]\right\rangle,\end{aligned} \tag{58}$$

$$\begin{aligned}&\begin{pmatrix}\text{avg excess}\\ \text{mse due to}\\ \text{gradient noise}\end{pmatrix}_j\\ &= E\left[\left(\boldsymbol{W}_j^\dagger - E\left[\boldsymbol{W}_j^\dagger\right]\right)^T \boldsymbol{R}^\dagger \left(\boldsymbol{W}_j^\dagger - E\left[\boldsymbol{W}_j^\dagger\right]\right)\right]\\ &= E\left[\left(\boldsymbol{W}_j^\dagger - E\left[\boldsymbol{W}_j^\dagger\right]\right)^T \left(\boldsymbol{W}_j^\dagger - E\left[\boldsymbol{W}_j^\dagger\right]\right)\right].\end{aligned} \tag{59}$$

Using (48) can can evaluate the expression (59):[5]

$$\begin{pmatrix}\text{avg excess}\\ \text{mse due to}\\ \text{gradient noise}\end{pmatrix}_j = \mu^\dagger n \xi_{\min} = \frac{n\xi_{\min}}{2\tau}. \tag{60}$$

The next step is an evaluation of (58), the excess mse due to lag. Statistical knowledge of $(E[\mathbf{W}_j^\dagger] - \mathbf{W}_j^{\dagger *})$ will be required. In finding lag effects, we may eliminate gradient noise from consideration so that $E[\mathbf{W}_j^\dagger] = \mathbf{W}_j^\dagger$. Knowledge of $(\mathbf{W}_j^\dagger - \mathbf{W}_j^{\dagger *})$ will be sufficient.

Without gradient noise, orthogonalized LMS is represented by (43) as

$$\begin{aligned}\mathbf{W}_{j+1}^\dagger &= \mathbf{W}_j^\dagger + 2\mu^\dagger E\left[\epsilon_j \mathbf{X}_j^\dagger\right]\\ &= \mathbf{W}_j^\dagger + 2\mu^\dagger E\left[\left(\left(\mathbf{X}_j^\dagger\right)^T \mathbf{W}_j^{\dagger *} - \left(\mathbf{X}_j^\dagger\right)^T \mathbf{W}_j^\dagger\right)\mathbf{X}_j^\dagger\right]\\ &= \left(1 - 2\mu^\dagger\right)\mathbf{W}_j^\dagger + 2\mu^\dagger \mathbf{W}_j^{\dagger *}. \end{aligned}\tag{61}$$

Substitution of (52) into (61) yields

$$\mathbf{W}_{j+1}^\dagger - \left(1 - 2\mu^\dagger\right)\mathbf{W}_j^\dagger = 2\mu^\dagger \mathbf{W}_j^{\dagger *} = 2\mu^\dagger \mathbf{R}^{1/2}\mathbf{W}_j^*. \tag{62}$$

The random vector $\mathbf{W}_j^*$ is the driving function for this vector difference equation. Notice that there is no cross-coupling from any one coordinate to any other. Furthermore, all components of $\mathbf{W}_j^*$ have been assumed to be stationary, ergodic, independent of each other, first-order Markov, and they have all been assumed to have the same variances and the same autocorrelation functions. Therefore, (62) may be treated as an array of n independent first-order linear difference equations.

We will next derive an expression for the covariance of $(\mathbf{W}_j^\dagger - \mathbf{W}_j^{\dagger *})$ resulting from the random driving function $\mathbf{W}_j^*$. Taking z transforms of both sides of (62) yields

$$z\mathbf{W}^\dagger(z) - (1 - 2\mu)\mathbf{W}^\dagger(z) = 2\mu \mathbf{R}^{1/2}\mathbf{W}^*(z) \tag{63}$$

or, equivalently,

$$\mathbf{W}^\dagger(z) = \left[\frac{2\mu z^{-1}}{1 - (1 - 2\mu)z^{-1}}\right]\mathbf{R}^{1/2}\mathbf{W}^*(z) \tag{64}$$

and

$$\begin{aligned}\mathbf{W}^\dagger(z) - \mathbf{W}^{\dagger *}(z) &= \left[\frac{z^{-1} - 1}{1 - (1 - 2\mu)z^{-1}}\right]\mathbf{R}^{1/2}\mathbf{W}^*(z)\\ &\triangleq \mathbf{R}^{1/2}\mathbf{A}(z).\end{aligned}\tag{65}$$

The vector $\mathbf{A}(z)$, conveniently defined in (65), will be used below.

The random vector $\mathbf{W}_j^*$ is assumed to be first-order Markov. A model for the generation of $\mathbf{W}_j^*$ is shown in Fig. 5. Each of its components is generated by passing uncorrelated ("white") samples of variance σ^2 through a one-pole discrete filter of transfer function $1/(1 - az^{-1})$. This transfer function is in cascade with the above described transfer function. Thus, the transfer function from uncorrelated samples to the components of $\mathbf{A}(z)$ is

$$T(z) = \left(\frac{1}{1 - az^{-1}}\right)\left(\frac{z^{-1} - 1}{1 - (1 - 2\mu)z^{-1}}\right), \tag{66}$$

where a is the geometric ratio of the nonstationarity. The variance of each of the independent components of $\mathbf{A}(z)$ is equal to σ^2 multiplied by $S_t \triangleq$ sum of the squares of the impulses of the impulse response corresponding to the transfer function $T(z)$ given by (66). Transforming to the time domain and summing squares yields

$$S_t = \left(\frac{1}{a - 1 + 2\mu}\right)^2\left(\frac{1 - a}{1 + a} + \frac{\mu}{1 - \mu} - \frac{4(1 - a)\mu}{1 - a + 2\mu a}\right). \tag{67}$$

The covariance of $\mathbf{A}(z)$ may be expressed as

$$\operatorname{cov}[\mathbf{A}(z)] = S_t \sigma^2 \mathbf{I}. \tag{68}$$

Hence, the covariance of $(\mathbf{W}_j^\dagger - \mathbf{W}_j^{\dagger *})$ may be found as

$$\operatorname{cov}\left[\left(\mathbf{W}_j^\dagger - \mathbf{W}_j^{\dagger *}\right)\right] = \mathbf{R}^{1/2}(\operatorname{cov}[\mathbf{A}(z)])\mathbf{R}^{1/2} = S_t \sigma^2 \mathbf{R}. \tag{69}$$

Substituting (69) into (58) yields

$$\begin{pmatrix}\text{avg excess}\\ \text{mse due to lag}\end{pmatrix} = S_t \sigma^2 \operatorname{tr}(\mathbf{R}). \tag{70}$$

In order to use (70), we have to evaluate the value of the constant S_t in expression (67). In practice it might be difficult, but in certain special cases of interest this task might be simplified. If the adaptive process is slow relative to the changes in the plant, we can assume $\mu^\dagger = 0$. Then

$$S_t = \frac{1}{1 - a^2}. \tag{71}$$

A more common case occurs when the adaptive process is rapid relative to the time variation of the plant. As such,

$$1 - a \ll \mu^\dagger \ll 1, \tag{72}$$

and therefore

$$S_t \approx \frac{1}{4\mu^\dagger} = \frac{\tau}{2}. \tag{73}$$

Expression (73) can be used as an excellent approximation of S_t. Substitution of (73) into (70) yields

$$\begin{pmatrix}\text{avg excess}\\ \text{mse due to lag}\end{pmatrix} = \frac{\tau\sigma^2}{2}\operatorname{tr}(\mathbf{R}) = \frac{\sigma^2}{4\mu^\dagger}\operatorname{tr}(\mathbf{R}). \tag{74}$$

Substitution of (60) and (74) into (57) yields

$$\begin{aligned}\begin{pmatrix}\text{avg excess}\\ \text{mse}\end{pmatrix} &= \mu^\dagger n \xi_{\min} + \frac{\sigma^2}{4\mu^\dagger}\operatorname{tr}(\mathbf{R})\\ &= \frac{n\xi_{\min}}{2\tau} + \frac{\tau\sigma^2}{2}\operatorname{tr}(\mathbf{R}).\end{aligned}\tag{75}$$

Normalizing with respect to $\xi_{\min}$, we can compute the net misadjustment due to weight noise and weight lag as

$$\begin{aligned}M_{\text{sum}}^\dagger &= \frac{n}{2\tau} + \frac{\tau\sigma^2}{2\xi_{\min}}\operatorname{tr}(\mathbf{R})\\ &= \mu^\dagger n + \frac{\sigma^2}{4\mu^\dagger \xi_{\min}}\operatorname{tr}(\mathbf{R}).\end{aligned}\tag{76}$$

[5] Strictly speaking, in the nonstationary case, $\xi_{\min}$ includes the power of the plant noise and the additional error power due to lag. For the sake of simplicity, we assume here that the adaptive process tracks well. Fluctuations in the plant parameters and error due to lag is assumed to be negligible relative to the $\xi_{\min}$ of stationary plant.

The optimal choice of $\mu^\dagger$ that minimizes $M^\dagger_{\text{sum}}$ is obtained by differentiating (76) with respect to $\mu^\dagger$, and setting the derivative to zero. The result is that the adaptive process is optimized when the component of $M^\dagger_{\text{sum}}$ due to the noise in the gradient is equal to the component of $M^\dagger_{\text{sum}}$ due to the lag:

$$\mu^{\dagger *} n = \frac{\sigma^2}{4\mu^{\dagger *}\xi_{\min}} \operatorname{tr}(\boldsymbol{R}). \tag{77}$$

Hence the optimal value of $\mu^\dagger$ is[6]

$$\mu^{\dagger *} = \sqrt{\frac{\sigma^2 \operatorname{tr}(\boldsymbol{R})}{4n\xi_{\min}}}. \tag{78}$$

Substitution of (78) into (76) yields the minimal misadjustment

$$M^{\dagger *}_{\text{sum}} = \sqrt{\frac{\sigma^2 n \operatorname{tr}(\boldsymbol{R})}{\xi_{\min}}}. \tag{79}$$

With *a priori* knowledge of the input signal power (to get $\operatorname{tr}(\boldsymbol{R})$), and with knowledge of $\xi_{\min}$ and σ^2, these formulas could be used to set μ and predict misadjustment. Without such knowledge, one could slew the value of μ seeking the lowest system mse.

These results apply to orthogonalized LMS. An analogous problem was of considered in [26], where the performance of stochastic gradient LMS algorithm of Widrow and Hoff in a nonstationary environment was evaluated. Using as a model the same configuration presented in Figs. 5 and 6, the excess mean square error was computed. The approach used was similar to the one described above, i.e., the system error was divided into two uncorrelated parts: error due to the gradient noise and error due to the lag in the adaptation process. The overall misadjustment M_{sum} equals the sum of misadjustments caused by each one of these errors:

$$M_{\text{sum}} = \mu \operatorname{tr}(\boldsymbol{R}) + \frac{1}{\mu}\frac{n\sigma^2}{4\xi_{\min}}. \tag{80}$$

Optimizing the choice of μ results in the minimum M_{sum} when the two right-hand terms are equal,

$$\mu^* \operatorname{tr}(\boldsymbol{R}) = \frac{1}{\mu^*}\frac{N\sigma^2}{4\xi_{\min}}. \tag{81}$$

Hence for the steepest descent LMS algorithm on optimal choice of adaptation constant would be

$$\mu^* = \sqrt{\frac{n\sigma^2}{4\xi_{\min}\operatorname{tr}(\boldsymbol{R})}} = \frac{n}{\operatorname{tr}(\boldsymbol{R})}\mu^{\dagger *}. \tag{82}$$

In other words, on optimal value μ^* for the adaptation constant of the steepest descent LMS algorithm is equal to the optimal value of the adaptation constant $\mu^{\dagger *}$ for the orthogonalized algorithm normalized by the factor of input signal power.

For the steepest descent LMS algorithm, the misadjustment will be minimal for $\mu = \mu^*$. Using (82) and (80) we have

$$M^*_{\text{sum}} = \sqrt{\frac{n\sigma^2 \operatorname{tr}(\boldsymbol{R})}{\xi_{\min}}}. \tag{83}$$

Comparing (83) with (79), we note that

$$M^*_{\text{sum}} = M^{\dagger *}_{\text{sum}}. \tag{84}$$

The optimized misadjustment of the orthogonalized LMS algorithm turns out to be equal to the optimized misadjustment of the original Widrow–Hoff steepest descent LMS algorithm, regardless of the eigenvalue spread. The important conclusion is that, for the assumed form of nonstationary input, the two algorithms give identical mean square error performance although the adaptive steps of the two algorithms generally differ in detail.

VI. Conclusion

Using an exact least squares algorithm to determine the n weights of an adaptive filter from N independent samples of data, we have shown the misadjustment to be $M = n/N$.

We have devised an ideal form of the steady flow LMS algorithm called orthogonalized LMS. To implement this algorithm, one would need perfect knowledge of the input covariance matrix $\boldsymbol{R}$. Since this would not generally be known, the orthogonalized LMS algorithm is primarily of theoretical interest. We have shown that with a stationary input, this algorithm develops a misadjustment equal to $M^\dagger = n/4\tau_{\text{mse}}$. Comparing this result with the misadjustment formula for exact least squares, one sees great similarity. The amounts of data at any time being used to determine the weights is of the order of four time constants of the learning curve, the "settling time" of the adaptive process. One concludes that for a given speed of adaptation, the misadjustment of the orthogonalized LMS algorithm is about as low as can be obtained by any algorithm, and that this algorithm uses its input data about as efficiently as an exact least squares process.

Next, we have analyzed the behavior of the orthogonalized LMS algorithm in a nonstationary plant identification application. Comparing the results of this study with those obtained by Widrow and McCool for the same application but using the conventional steepest descent LMS algorithm, it was determined that the mean square error performance of the conventional LMS algorithm is on the average identical to that of the orthogonalized LMS algorithm, regardless of eigenvalue spread. For the nonstationary problem considered, we conclude that the conventional LMS algorithm performs as efficiently as exact least squares.

It is not always true that the performance of the LMS algorithm with nonstationary inputs will be independent of the eigenvalue spread. As a matter of fact, a form of nonstationarity studied by Macchi and Eweda [27] was

[6] When optimizing $\mu^\dagger$ is accord with the above procedure, one must keep this parameter within the stable range (44). Formula (78) applies as long as $\mu^{\dagger *}$ is in the stable range. Otherwise, $\mu^\dagger$ is set to its maximal stable value. The minimal misadjustment is determined either by (79), or by (76) when $\mu^\dagger$ is limited by stability considerations. Low plant noise would be the cause of $\mu^{\dagger *}$ exceeding the stable limit.

such that the LMS algorithm did show sensitivity to eigenvalue spread and the performance of an orthogonalizing algorithm for this case could have been better than that of conventional LMS. However, from the analysis conducted above, one can conclude that the steepest descent LMS algorithm of Widrow and Hoff, devised in 1960, not only is the simplest algorithm but is a highly efficient one for use with a variety of nonstationary inputs.

Acknowledgment

We would like to thank John M. Cioffi, John M. McCool, and Timothy Saxe for many useful discussions.

References

[1] B. Widrow and M. E. Hoff, "Adaptive switching circuits," in *IRE WESCON Conv. Rec.*, pt. 4, pp. 96–104, 1960.

[2] B. Widrow, "Adaptive filters," in *Aspects of Network and System Theory*, R. Kalman and N. DeClaris, Eds. New York: Holt, Rinehart, and Winston, 1971, pp. 563–587.

[3] B. Widrow, P. Mantey, L. Griffiths, and B. Goode, "Adaptive antenna systems," *Proc. IEEE*, vol. 55, no. 12, pp. 2143–2159, Dec. 1967.

[4] B. Widrow, J. R. Glover, Jr., J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong, Jr., and R. C. Goodlin, "Adaptive noise cancelling: Principles and applications," *Proc. IEEE*, vol. 63, no. 12, pp. 1692–1716, Dec. 1975.

[5] J. Makhoul, "Linear prediction: A tutorial review," *Proc. IEEE*, vol. 63, no. 4, pp. 561–580, Apr. 1975.

[6] R. W. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, pp. 547–588, Apr. 1965.

[7] R. W. Lucky, J. Salz, and E. J. Weldon, Jr., *Principles of Data Communication*. New York: McGraw Hill, 1968, ch. 6.

[8] R. D. Gitlin, E. Y. Ho, and J. E. Mazo, "Passband equalization of differentially phase-modulated data signals," *Bell Syst. Tech. J.*, vol. 52, no. 2, pp. 219–238, Feb. 1973.

[9] S. Qureshi, "Adaptive equalization—A comprehensive review," *IEEE Commun. Magazine*, pp. 9–16, Mar. 1982.

[10] M. M. Sondhi and A. J. Presti, "A self-adaptive echo canceller," *Bell Syst. Tech. J.*, vol. 46, no. 3, pp. 497–511, Mar. 1967.

[11] D. D. Falconer, K. H. Mueller, and S. B. Weinstein, "Simultaneous two-way data transmission over a two-wire circuit," presented at Proc. NTC, Dallas, TX, Dec. 1976.

[12] D. L. Duttweiler, "A twelve-channel digital echo canceller," *IEEE Trans. Comm.*, pp. 647–563, May 1978.

[13] L. J. Griffiths, "Rapid measurement of instantaneous frequency," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 209–222, Apr. 1975.

[14] J. R. Zeidler, E. H. Satorius, D. M. Chabries, and H. T. Wexler, "Adaptive enhancement of multiple sinusoids in uncorrelated noise," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, pp. 240–254, June 1978.

[15] J. T. Rickard, J. R. Zeidler, M. J. Dentino, and M. Shensa, "A performance analysis of adaptive line enhancer-augmented spectral detectors," *IEEE Trans. Circuits Syst.*, vol. CAS-28, no. 6, pp. 534–541, June 1981.

[16] B. Widrow, J. M. McCool, and B. P. Medoff, "Adaptive control by inverse modeling," presented at the Twelfth Asilomar Conference on Circuits, Systems, and Computers, Nov. 1978.

[17] B. Widrow, D. Schur, and S. Shaffer, "On adaptive inverse control," presented at the Fifteenth Asilomar Conf. Circuits, Syst., Comput., pp. 185–189, Nov. 1981.

[18] B. D. O. Anderson and R. M. Johnstone, "Convergence results for Widrow's adaptive controller," presented at the IFAC Conf. System Identification, 1981.

[19] D. T. L. Lee, M. Morf, and B. Friedlander, "Recursive least squares ladder estimation algorithms," *IEEE Trans. Circuits Syst.*, vol. CAS-28, no. 6, pp. 467–481, June 1981.

[20] L. J. Griffiths, "A continuously adaptive filter implemented as a lattice structure," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Hartford, CT, May 1977, pp. 683–686.

[21] B. Friedlander, "Lattice filters for adaptive processing," *Proc. IEEE*, vol. 70, no. 8, pp. 829–867, Aug. 1982.

[22] G. C. Goodwin and R. L. Payne, "Dynamic system identification: Experiment design and data analysis," New York: Academic, 1977.

[23] L. D. Davisson, "Steady-state error in adaptive mean-square minimization," *IEEE Trans. Inform. Theory*, vol. IT-16, pp. 382–385, July 1970.

[24] R. R. Bitmead, "Convergence in distribution of LMS-type adaptive parameter estimates," *IEEE Trans. Automat. Contr.*, vol. AC-28, no. 1, pp. 54–60, Jan. 1983.

[25] O. Macchi and E. Eweda, "Second-order convergence analysis of stochastic adaptive linear filtering," *IEEE Trans. Automat. Contr.*, vol. AC-28, no. 1, pp. 76–85, Jan. 1983.

[26] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, Jr., "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, no. 8, pp. 1151–1162, Aug. 1976.

[27] O. Macchi and E. Eweda, "Tracking properties of adaptive nonstationary filtering," submitted for publication.

[28] R. W. Chang, "A new equalizer structure for fast start-up digital communication," *Bell Syst. Tech. J.*, vol. 50, no. 6, pp. 1969–2014, July–Aug. 1971.

[29] K. H. Mueller, "A new fast converging mean-square algorithm for adaptive equalizers with partial-response signaling," *Bell Syst. Tech. J.*, vol. 54, no. 1, pp. 143–153, Jan. 1975.

[30] D. Godard, "Channel equalization using a Kalman filter for fast data transmission," *IBM J. Res. Develop.*, pp. 267–273, May 1974.

[31] R. D. Gitlin and F. R. Magee, Jr., "Self-orthogonalizing adaptive equalization algorithms," *IEEE Trans. Commun.*, vol. COM-25, no. 7, pp. 666–672, July 1977.

[32] H. Sari, "Simplified algorithms for adaptive channel equalization," *Philips J. Res.*, vol. 37, pp. 56–77, 1982.

[33] L. L. Horowitz and K. D. Senne, "Performance advantage of complex LMS for controlling narrow-band adaptive arrays," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 562–576, June 1981.

[34] L. J. Griffiths, "Rapid measurements of digital instantaneous frequency," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 207–222, Apr. 1975.

[35] A. Nehorai and D. Malah, "On the stability and performance of adaptive line enhancer," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, 1980, pp. 478–481.