The Least Mean Fourth (LMF) Adaptive Algorithm and Its Family

EUGENE WALACH AND BERNARD WIDROW, FELLOW, IEEE

Abstract—New steepest descent algorithms for adaptive filtering and have been devised which allow error minimization in the mean fourth and mean sixth, etc., sense. During adaptation, the weights undergo exponential relaxation toward their optimal solutions. Time constants have been derived, and surprisingly they turn out to be proportional to the time constants that would have been obtained if the steepest descent least mean square (LMS) algorithm of Widrow and Hoff had been used. The new gradient algorithms are insignificantly more complicated to program and to compute than the LMS algorithm. Their general form is

$$W_{j+1} = W_j + 2\mu K \epsilon_j^{2K-1} X_j,$$

where W_j is the present weight vector, W_{j+1} is the next weight vector, ϵ_j is the present error, X_j is the present input vector, μ is a constant controlling stability and rate of convergence, and $2K$ is the exponent of the error being minimized. Conditions have been derived for weight-vector convergence of the mean and of the variance for the new gradient algorithms. The behavior of the least mean fourth (LMF) algorithm is of special interest. In comparing this algorithm to the LMS algorithm, when both are set to have exactly the same time constants for the weight relaxation process, the LMF algorithm, under some circumstances, will have a substantially lower weight noise than the LMS algorithm. It is possible, therefore, that a minimum mean fourth error algorithm can do a better job of least squares estimation than a mean square error algorithm. This intriguing concept has implications for all forms of adaptive algorithms, whether they are based on steepest descent or otherwise.

I. Introduction

MANY signal processing problems such as plant modeling, noise canceling, channel equalization, etc., can be represented in the form depicted in Fig. 1. Shown is a linear plant which can be represented by the polynomial transfer function $P(z)$, whose output is corrupted by additive independent zero-mean noise n_j. Our aim is to find, preferably in an adaptive iterative way, a plant model $\hat{P}(z)$. This can be done by minimizing a certain statistical measure of the error ϵ_j. Usually the minimization is done in the mean square sense, i.e., one minimizes the expected value of the square of the error $E[\epsilon_j^2]$. This choice of performance measure is usually due to its utility, simplicity, and relative ease of analysis. Nonmean-square error criteria have appeared in the literature, generally in the context of analysis of Gaussian processes [1]–[4].

In this paper we consider the more general problem of minimizing $E[\epsilon_j^{2K}]$ for $K = 1, 2, \cdots$. Assuming that the noise n_j is independent of the input signal x_j, it is easy to see that the optimal solutions for the problem of Fig. 1 will be the same for all the choices of integer $K \geq 1$. Namely,

$$E\left[\epsilon_j^{2K}\right] \text{ is minimal} \Leftrightarrow P(z) = \hat{P}(z). \tag{1}$$

The choice of any particular K, however, will influence the performance of the adaptive algorithm employed in order to find the solution (1).

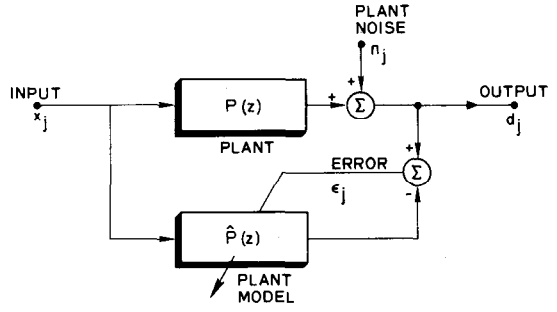

Fig. 1. Block diagram of adaptive plant modeling.

In this paper we consider the steepest descent LMS (least mean square) algorithm of Widrow and Hoff which is probably the simplest and most widely used. In Section II we reintroduce briefly well-known formulas which govern the behavior of the "conventional" LMS algorithm. In Section III we derive an extension of the Widrow–Hoff algorithm which enables us to minimize $E[\epsilon_j^{2K}]$ for $K \geq 1$. Then we analyze the performance of this new algorithm. The LMS case can, of course, be viewed as a particular case of the general algorithm for $K = 1$. Based on this analysis, we show that for certain problems the choice of $K > 1$ is considerably advantageous over the "conventional" choice of $K = 1$. Section IV is dedicated to the special case of the LMF algorithm (i.e., algorithm which minimizes the mean fourth of the error ϵ_j, with $K = 2$). Computer simulations are presented to illustrate the potential advantages of the new algorithm.

II. The LMS Algorithm

Consider the schematic structure pictured in Fig. 2. This is a more detailed version of the plant modeling scheme of Fig. 1. The adaptive plant model $\hat{P}(z)$ is built of a tapped-delay line of length n. We assume that n is equal to the order of the plant $P(z)$ so that if all the weights $W^T = (w_1, \cdots, w_n)$ of the adaptive model were frozen at the proper values $(W^*)^T = (w_1^*, \cdots, w_n^*)$, then the model $\hat{P}(z)$ will match exactly the transfer function $P(z)$ of the plant.

Manuscript received April 15, 1983; revised September 9, 1983.

E. Walach is with IBM Corp., Box 218, Watson Research Center, Yorktown Heights, NY.

B. Widrow is with the Department of Electrical Engineering, Stanford University, CA 94305.

0018-9448/84/0300-0275$01.00

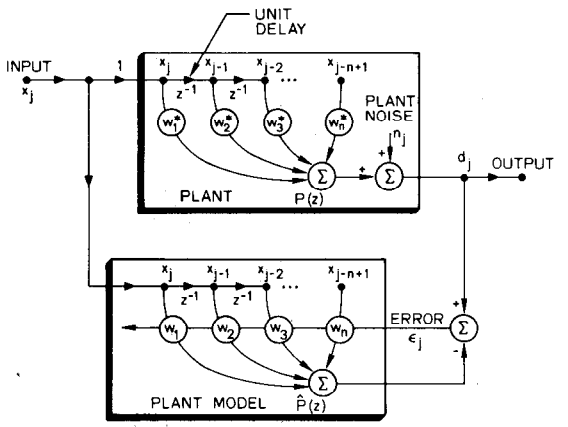

Fig. 2. Schematic structure of the tapped-delay-line used for plant modeling.

It is clear that

$$P(z) = w_1^* + w_2^* z^{-1} + w_3^* z^{-2} + \cdots + w_n^* z^{-n+1}. \quad (1a)$$

Subsequently we will assume that the input x_j and the plant noise n_j are independent of each other. Moreover we will assume that both x_j and n_j are distributed symmetrically around zero (i.e., all the odd moments of x_j and n_j are equal to zero).

Referring to Fig. 2, the system error at the jth moment ϵ_j can be found as

$$\epsilon_j = d_j - \boldsymbol{W}_j^T \boldsymbol{X}_j = n_j + (\boldsymbol{W}^*)^T \boldsymbol{X}_j - \boldsymbol{W}_j^T \boldsymbol{X}_j = n_j - \boldsymbol{V}_j^T \boldsymbol{X}_j, \quad (2)$$

where

$$\boldsymbol{X}_j^T = (x_j, x_{j-1}, \cdots, x_{j-n+1}) \quad (3)$$

represents the vector of the last n samples of the input signal,

$$\boldsymbol{W}_j^T = (w_{1j}, w_{2j}, \cdots, w_{nj}) \quad (4)$$

denotes the current values of the adaptive weights and

$$\boldsymbol{V}_j = \boldsymbol{W}_j - \boldsymbol{W}^* \quad (5)$$

represents the difference between the current adaptive weight vector and the optimal solution. This is the weight vector error. Adaptation is done by moving, at each iteration, a certain small step in the direction opposite to the current estimation of the gradient.

The Widrow–Hoff algorithm uses an instantaneous estimation of the gradient of $E[\epsilon_j^2]$:

$$\nabla(\epsilon_j^2) = -2\epsilon_j \boldsymbol{X}_j. \quad (6)$$

Hence the adaptation rule is

$$\boldsymbol{W}_{j+1} = \boldsymbol{W}_j + 2\mu\epsilon_j \boldsymbol{X}_j. \quad (7)$$

The constant μ controls stability and rate of convergence.

The behavior of this algorithm has been analyzed extensively in the literature (see, for instance, [5]). It was shown that if the adaptation constant μ was chosen such that

$$0 < \mu < \frac{1}{nE[x_j^2]}, \quad (8)$$

then the mean of the weight vector $\boldsymbol{W}_j$ will converge from any initial guess to the optimal solution $\boldsymbol{W}^*$. Assuming that vectors $\boldsymbol{V}_j$ and $\boldsymbol{X}_j$ are independent of each other, the weight vector error $\boldsymbol{V}_j$ will obey the matrix equation

$$E[\boldsymbol{V}_{j+1}] = (I - 2\mu R)E[\boldsymbol{V}_j] \quad (9)$$

where R is the autocorrelation matrix of the input signal

$$R = E[\boldsymbol{X}_j \boldsymbol{X}_j^T]. \quad (10)$$

Hence there will be, generally, n different modes of convergence and n different relaxation time constants for the weights given by

$$\tau_i = \frac{1}{2\mu\lambda_i}, \quad (11)$$

where λ_i, $i = 1, \cdots, n$, represent the eigenvalues of R.

After convergence, the system error ϵ_j comes close to being equal to the plant noise n_j. However, due to the noise in the estimate (6) of the gradient, the weights of the model will be also noisy. Therefore the system error power will be higher than the optimal (Wiener) power $E[n_j^2]$. It is of interest to consider the ratio between this excess mean square error and the optimal error power. This dimensionless ratio is called misadjustment M. It was shown in [5] that

$$M = \mu n E[x_j^2] = \mu \operatorname{tr}(R). \quad (12)$$

Substituting (11) into (12) yields

$$M = \frac{1}{2}\sum_{i=1}^{n} \frac{1}{\tau_i} = \left(\frac{n}{2}\right)\left(\frac{1}{\tau_i}\right)_{\text{avg}}. \quad (13)$$

Clearly, the adaptive algorithm performs better if the misadjustment is lower. However, the time constants τ_i cannot be increased indefinitely because eventually the adaptive algorithm will lose its ability to react to the possible fluctuations in the parameters of the plant. n is equal to the fixed order of the plant. Therefore we will be able to compare the efficiency of various algorithms by measuring the corresponding misadjustment for the given values of the time constant τ and the plant order n.

III. Modified Steepest Descent Algorithm

In this section we will extend the Widrow–Hoff algorithm in order to be able to minimize $E[\epsilon_j^{2K}]$ for arbitrary choice of $K = 1, 2, \cdots$. We can estimate the instantaneous gradient as follows. The instantaneous error, given by (2), is raised to the $2K$ power and differentiated with respect to the weight vector $\boldsymbol{W}_j$. The instantaneous gradient is therefore

$$\nabla(\epsilon_j^{2K}) = -2K\epsilon_j^{2K-1} \boldsymbol{X}_j. \quad (14)$$

Using this gradient, the new adaptation rule will be

$$\boldsymbol{W}_{j+1} = \boldsymbol{W}_j + 2\mu K \epsilon_j^{2K-1} \boldsymbol{X}_j. \quad (15)$$

Examination of expression (15) shows that if the proposed algorithm converges ($E[\boldsymbol{W}_{j+1}] = E[\boldsymbol{W}_j]$), then the

point of convergence must obey the equation

$$E\left[\epsilon_j^{2K-1}\boldsymbol{X}_j\right] = \mathbf{0}. \tag{16}$$

Substituting (2) into the above,

$$E\left[\left(d_j - \boldsymbol{W}_j^T\boldsymbol{X}_j\right)^{2K-1}\boldsymbol{X}_j\right] = 0. \tag{17}$$

For $K = 1$ (conventional LMS case) the system of (17) is linear in $\boldsymbol{W}_j$ and has only one (optimal) solution. Generally, however, (17) constitutes a system of n polynomial equations of degree $2K-1$ in n unknowns w_i, $i = 1,\cdots,n$. One has to consider the possibility of convergence to a local minimum. However, the mean of the error to the $2K$ power is a convex function of the weight vector and therefore cannot have local minima. Indeed the Hessian matrix of the error-to-the-$2K$-power function can be shown to be positive definite or positive semidefinite. The basic argument is presented by Gersho [4].

Next we derive conditions for convergence of the mean for $K \geq 1$. We will prove that $E[\boldsymbol{V}_j] \to 0$. We will limit our analysis, here, to the relatively simple case of small deviations from the Wiener solution when the weight error vector $\boldsymbol{V}_j$ is close to zero. A more general convergence proof is presented in the Appendix, where the stronger question of convergence of the variance is demonstrated. We begin by subtracting the optimal (Wiener) solution from both sides of (15). Accordingly,

$$\boldsymbol{V}_{j+1} = \boldsymbol{V}_j + 2\mu K\epsilon_j^{2K-1}\boldsymbol{X}_j. \tag{18}$$

Substituting (2) into (18) yields

$$\boldsymbol{V}_{j+1} = \boldsymbol{V}_j + 2\mu K\boldsymbol{X}_j\left(\sum_{i=0}^{2K-1}\binom{2K-1}{i}n_j^i\left(-\boldsymbol{X}_j^T\boldsymbol{V}_j\right)^{2K-1-i}\right). \tag{19}$$

Since we assumed that $\boldsymbol{V}_j$ is close to zero, we can neglect the impact of terms on the right side of (19) which include high powers of $\boldsymbol{V}_j$. Hence

$$\boldsymbol{V}_{j+1} \cong \boldsymbol{V}_j + 2\mu K\boldsymbol{X}_j\left(n_j^{2K-1} - (2K-1)n_j^{2K-2}\boldsymbol{X}_j^T\boldsymbol{V}_j\right). \tag{20}$$

Take expectations of both sides of (20). We have assumed that n_j is independent of $\boldsymbol{X}_j$. For small μ we can assume that n_j is independent of $\boldsymbol{V}_j$.[1] Therefore the second term on the right side of (20) will vanish under expectation. Hence

$$E\left[\boldsymbol{V}_{j+1}\right] = \left(I - 2\mu K(2K-1)E\left[\eta_j^{2K-2}\right]R\right)E\left[\boldsymbol{V}_j\right], \tag{21}$$

where R is positive definite input autocorrelation matrix defined by (10). Denote

$$\Omega \triangleq I - 2\mu K(2K-1)E\left[n_j^{2K-2}\right]\cdot R. \tag{22}$$

[1] $\boldsymbol{V}_j$ is independent of n_j only when n_j is white. However, for small values of μ we can assume that $\boldsymbol{V}_j$ is determined, mainly, by long past samples $n_{j-\Delta}$, which are not correlated with the current value of n_j. In such a case, relations between n_j and $\boldsymbol{V}_j$ are weak, and they can be viewed as two independent random variables (see also [5] and [6]). For the simple case of $K = 1$, more rigorous proofs of convergence which do not assume independence of $\boldsymbol{V}_j$, n_j and $\boldsymbol{V}_j$, also exist (see [7] and [8]).

Substitution of (22) into (21) yields

$$E\left[\boldsymbol{V}_{j+1}\right] = \Omega\cdot E\left[\boldsymbol{V}_j\right]. \tag{23}$$

Since R was assumed to be positive definite, we can choose an adaptation constant μ

$$0 < \mu < \frac{1}{K(2K-1)E\left[n_j^{2K-2}\right]\gamma_{\max}}, \tag{24a}$$

where

$$\gamma_{\max} \triangleq \text{maximal eigenvalue of } R, \tag{24b}$$

so that all eigenvalues of the matrix Ω will have absolute values smaller than 1. For choice of μ in accord with (24) the normal form factorization will be

$$\Omega = A\cdot\begin{pmatrix}\delta_1 & & 0\\ & \ddots & \\ 0 & & \delta_n\end{pmatrix}\cdot A^T$$

$$\delta \triangleq \sup_i|\delta_i| < 1, \qquad i = 1,\cdots,n; \quad AA^T = I. \tag{25}$$

Now we are able to evaluate the impact of each iteration on the weight vector $\boldsymbol{V}_j$. Denote by v_{j+1} the norm of the vector $E[\boldsymbol{V}_{j+1}]$:

$$v_{j+1} \triangleq E\left[\boldsymbol{V}_{j+1}^T\right]\cdot E\left[\boldsymbol{V}_{j+1}\right]. \tag{26}$$

From (23), (25), and (26) we can find

$$v_{j+1} = E\left[\boldsymbol{V}_j^T\right]A\begin{pmatrix}\delta_1^2 & & 0\\ & \ddots & \\ 0 & & \delta_n^2\end{pmatrix}A^TE\left[\boldsymbol{V}_j\right] \leq \delta^2 v_j. \tag{27}$$

The recursive inequality (27) shows that the adaptation process will cause the convergence $E[\boldsymbol{V}_j] \to 0$, i.e., algorithm (15) will provide an unbiased estimate of the Wiener solution.

In order to complete the convergence analysis of algorithm (15), we have to show that the variance $\boldsymbol{V}_j^T\boldsymbol{V}_j$ also converges; in fact we show that for some choices of μ there exists a finite superior limit to $E(\boldsymbol{V}_j^T\boldsymbol{V}_j)$. This is somewhat more complex than the proof of convergence of the mean presented above and has been deferred to the Appendix.

Convergence of the mean is contingent, of course, on compliance with condition (24a). In practice this condition might be difficult to check. However, we can bound the maximal eigenvalue of a positive definite matrix by its trace, $\text{tr}(R) = nE(x_j^2)$, and thus find the easily applied sufficient condition for convergence of the mean of the new adaptive algorithm:

$$0 < \mu < \frac{1}{Kn(2K-1)E\left[n_j^{2K-2}\right]\cdot E\left[x_j^2\right]}. \tag{28}$$

Conditions for convergence of the variance are given in the Appendix.

We proceed next to the evaluation of the time constants of the adaptive process. Once again we assume that the current weight vector guess is in the vicinity of the optimal solution, so that approximation (21) holds. Comparing (21)

to (9), it is clear that all the modified algorithms ($K > 1$) will behave in a way similar to that of the conventional LMS algorithm ($K = 1$). This means that, generally, there will be n different modes of convergence corresponding to the n different eigenvalues of R. Accordingly there will be n different relaxation time constants for the weights,

$$\tau_i = \frac{1}{2\mu K(2K-1)E\left[n_j^{2K-2}\right]\lambda_i}, \qquad i = 1,\cdots,n. \tag{29}$$

Comparing (29) to (11), it is clear that for arbitrary choice of K, the time constants will be proportional to the time constants of the LMS algorithm. This is a surprising result! What it means is that the time constants in the weight relaxation process when minimizing mean fourth error, for example, are proportional to the time constants of the weight relaxation process when minimizing mean square error.

The last step in our analysis of the adaptive process (15) will be the evaluation of the misadjustment $M(K)$. Since the misadjustment is defined only for the adaptive processes in steady state (after adaptive transients have died out), we can assume that the error vector V_j is small, close to zero. Therefore we can use once more the basic expression (20). Multiplying each side of (20) by the transposed form of itself yields

$$\begin{aligned} &V_{j+1}V_{j+1}^T \\ &= \left(I - 2\mu K(2K-1)n_j^{2K-2}X_jX_j^T\right) \\ &\quad \cdot V_jV_j^T\left(I - 2\mu K(2K-1)n_j^{2K-2}X_jX_j^T\right) \\ &\quad + 4\mu^2K^2n_j^{4K-2}X_jX_j^T + 2\mu Kn_j^{2K-1} \\ &\quad \cdot X_jV_j^T\left(I - 2\mu K(2K-1)n_j^{2K-2}X_jX_j^T\right) \\ &\quad + 2\mu Kn_j^{2K-1}\left(I - 2\mu K(2K-1)n_j^{2K-2}X_jX_j^T\right)V_jX_j^T. \end{aligned} \tag{30}$$

The last two terms on the right side are multiplied by the odd degrees of the noise n_j. Therefore, since n_j was assumed to be independent of X_j and V_j (for small values of μ) and its odd moments were assumed to be equal to zero, these terms vanish when we take expectations of both sides of (30). Hence

$$\begin{aligned} &E\left(V_{j+1}V_{j+1}^T\right) \\ &= E\left(V_jV_j^T\right) \\ &\quad - 2\mu K(2K-1)E\left[n_j^{2K-2}\left(X_jX_j^TV_jV_j^T + V_jV_j^TX_jX_j^T\right)\right] \\ &\quad + 4\mu^2K^2(2K-1)^2E\left[n_j^{4K-4}\left(X_jX_j^TV_jV_j^TX_jX_j^T\right)\right] \\ &\quad + 4\mu^2K^2E\left[n_j^{4K-2}X_jX_j^T\right]. \end{aligned} \tag{31}$$

For small μ the third term on the right side of (31) can be neglected since for any V_j it will be small relative to the second term. Moreover, assuming that the algorithm has converged and is in steady state:

$$E\left(V_{j+1}V_{j+1}^T\right) = E\left(V_jV_j^T\right). \tag{32}$$

Hence

$$\begin{aligned} &-2\mu K(2K-1)E\left[n_j^{2K-2}\left(X_jX_j^TV_jV_j^T + V_jV_j^TX_jX_j^T\right)\right] \\ &\qquad + 4\mu^2K^2E\left[n_j^{4K-2}X_jX_j^T\right] = 0. \end{aligned} \tag{33}$$

Again using the assumption of independence of n_j, X_j and V_j, (33) becomes a Lyapunov equation that has the unique solution:

$$E\left[V_jV_j^T\right] = \frac{\mu KE\left[n_j^{4K-2}\right]}{(2K-1)E\left[n_j^{2K-2}\right]}I. \tag{34}$$

Eq. (34) implies that the noises in the various weights are uncorrelated with each other. This conclusion holds for arbitrary K (for $K = 1$ this is a well-known feature of the LMS algorithm, see [5] and [6]).

Now we evaluate the power of the additional noise at the system output due to noise in the weights.

$$E\left[\left(V_j^TX_j\right)^2\right] = E\left[\left(\sum_{i=1}^{n} v_{ij}x_{j-i+1}\right)^2\right]. \tag{35}$$

Since, according to (34), all the components v_{ij} of the vector V_j are uncorrelated with each other, we can neglect all the "cross terms" in the expression (35). Hence

$$E\left[\left(V_j^TX_j\right)^2\right] = nE\left[v_{ij}^2 \cdot x_{j-i+1}\right)^2\right] = nE\left[v_{ij}^2\right]E\left[x_j^2\right]. \tag{36}$$

Substituting (34) into (36) yields

$$E\left[\left(V_j^TX_j\right)^2\right] = \frac{\mu KnE\left[n_j^{4K-2}\right]E\left[x_j^2\right]}{(2K-1)E\left[n_j^{2K-2}\right]}. \tag{37}$$

Substituting (37) into the definition of the misadjustment $M(K)$ we find for an arbitrary choice of K:

$$\begin{aligned} M(K) &= \frac{\begin{pmatrix}\text{error power}\\ \text{due to weight noise}\end{pmatrix}}{\begin{pmatrix}\text{Wiener error}\\ \text{power}\end{pmatrix}} = \frac{E\left[\left(V_j^TX_j\right)^2\right]}{E\left[n_j^2\right]} \\ &= \frac{\mu KnE\left[n_j^{4K-2}\right]E\left[x_j^2\right]}{(2K-1)E\left[n_j^2\right]E\left[n_j^{2K-2}\right]}. \end{aligned} \tag{38}$$

Combining (29) and (38) yields

$$M(K) = \frac{E\left[n_j^{4K-2}\right]}{2(2K-1)^2E\left[n_j^2\right]\left(E\left[n_j^{2K-2}\right]\right)^2}\sum_{i=1}^{n}\frac{1}{\tau_i}. \tag{39}$$

It is easy to see that expressions (8), (11), and (13) can be viewed as a special case of expressions (28), (29), and (39) for $K = 1$. Comparing various algorithms for $K = 1, 2, \cdots$ and keeping corresponding time constants equal from one algorithm to another, we can define $\alpha(K)$, using (13) and (39), as

$$\alpha(K) \triangleq \frac{M(1)}{M(K)} = \frac{(2K-1)^2E\left[n_j^2\right]\left(E\left[n_j^{2K-2}\right]\right)^2}{E\left[n_j^{4K-2}\right]}. \tag{40}$$

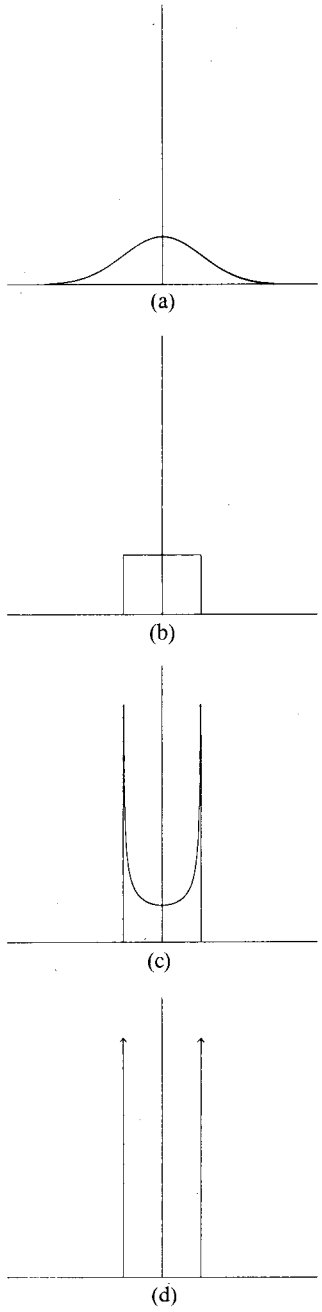

Fig. 3. Several probability densities for various forms of plant noise n_j.

The use of algorithms with $K > 1$ will be advantageous over the use of the conventional LMS when $\alpha(K) > 1$. This means that lower misadjustment can be expected for the same speed of convergence when $\alpha(K) > 1$.

The optimal choice of K can be determined when the moments of the plant noise n_j are known, since $\alpha(K)$ only depends on these moments. A set of special probability densities for n_j has been studied. These densities are pictured in Fig. 3. They have been selected for their practical importance, and they are shown in a logical ordering in Fig. 3. The Gaussian density is shown in Fig. 3(a), the uniform density in Fig. 3(b), the probability density of a sinusoidal signal is shown in Fig. 3(c), and that of a square wave is shown in Fig. 3(d). The corresponding values of $\alpha(K)$ are given in Table I for $K = 1, 2, 3, 4$.

TABLE I
Values of $\alpha(K)$ for Several Probability Densities for Plant Noise n_j

	GAUSSIAN	UNIFORM	SINE WAVE	SQUARE WAVE
K = 1	1	1	1	1
K = 2	0.6	2.3	3.6	9
K = 3	0.24	3.67	7.14	25
K = 4	0.08	5	11.4	49

IV. Least Mean Fourth Error Algorithm

This algorithm can be viewed as a special case of the general algorithm, analyzed in the previous section, for the choice of $K = 2$. The LMF algorithm is

$$\boldsymbol{W}_{j+1} = \boldsymbol{W}_j + 4\mu\epsilon_j^3\boldsymbol{X}_j. \tag{41}$$

Example 1: If the plant noise n_j is a random process uniformly distributed between ± 1 then, from Table I, $\alpha(2) = 2.33$. In this case, one can expect that the use of LMF algorithm will enable an improvement of about 3 dB over the LMS algorithm. There will be about 3 dB less noise in the weights for the same speed of convergence. In order to illustrate this case, a computer simulation was performed. The LMS and LMF adaptive algorithms were used to model the plant

$$A(z) = 0.1 + 0.2z^{-1} + 0.3z^{-2} + 0.4z^{-3} + 0.5z^{-4} + 0.4z^{-5} + 0.3z^{-6} + 0.2z^{-7} + 0.1z^{-8}. \tag{42}$$

The input signal was white and of unit power. The plant output noise was simulated as a uniformly white random process of power 100. A ten-weight tap-delay line adaptive model was used. Two simulations were performed using exactly the same data, the first using LMS and the second using LMF. In each case the power of weight noise (averaged over the last 125 samples) was plotted versus the number of iterations performed. The adaptive weights were initialized by adding to each one of the weights of the plant itself random components drawn from a population having power equal to 0.56. Results of ten independent experiments were averaged.

For the LMS algorithm the adaptation constant μ was chosen to be 9×10^{-4}, giving a theoretical time constant of ≈ 550 samples. All time constants were equal because x_j was white. After about two time constants the error power indeed decreased to its steady state value. The misadjustment was measured to be 0.959×10^{-2} (compared to a theoretical value of 0.9×10^{-2}).

For the LMF algorithm the adaptation constant was chosen to be 1.5×10^{-6}. Hence, according to (29), the expected time constant was once again equal to 550. After about two time constants the error power decreased to the steady state value. But this time the misadjustment was only 0.445×10^{-2} (theoretically, according to expression (39), the misadjustment should have been 0.386×10^{-2}).

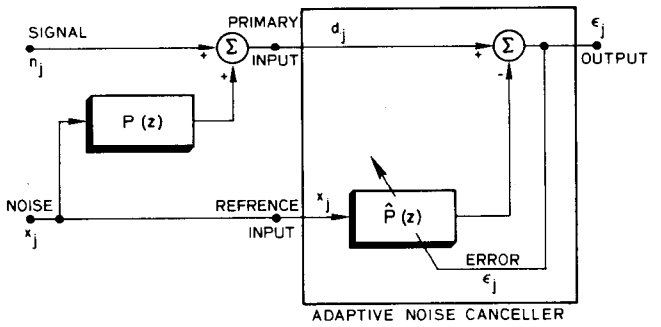

Fig. 4. Block diagram of adaptive noise canceler.

As expected, for the given time constant, the LMF algorithm had about 3 dB less weight noise due to adaptivity than the conventional LMS algorithm, giving about 3 dB less excess mean square error.

Example 2: Another important case of advantageous implementation of the LMF algorithm (or more generally implementation of algorithms for $K > 1$) is the one where the plant noise is deterministic rather than random. Such a situation is frequently encountered, for instance, when the adaptive system of Fig. 1 is employed for the purpose of noise canceling. In this case the formulation of the problem is somewhat changed. (Refer to Fig. 4). The adaptive noise canceler shown in this figure is identical to the adaptive plant modeling scheme shown in Fig. 1, and the same notation is used. In the system of Fig. 4, the "primary input" d_j consists of the sum of the "signal" n_j and a noise originating from the "noise" x_j filtered by $P(z)$. The "reference input" is the noise x_j. The objective is to eliminate, if possible, the additive noise of the primary input to obtain the signal n_j at the noise canceler "output." The noise x_j, correlated with the primary noise, is adaptively filtered by $\hat{P}(z)$. Adapting $\hat{P}(z)$ to minimize the mean square of the error ϵ_j causes the noise canceler output, also ϵ_j, to be a best least squares estimate of the signal n_j. The signal n_j could be stochastic or deterministic. The concept of adaptive noise canceling is presented in [5].

Despite a changed interpretation in Fig. 4, the mechanism of the adaptive process remains the same as in Fig. 1. Hence we can still use the expressions developed in the previous section in order to evaluate the performance. For this example, let

$$n_j = a\cos(\omega j). \tag{43}$$

Accordingly, Table I shows that $\alpha(2) = 3.6$. Therefore the LMF algorithm is expected to outperform the LMS algorithm by almost 6 dB in this case.

The conditions used for the computer simulations were exactly the same as in the previous example except for the fact that n_j was obtained from (43) for $a = 10\sqrt{2}$ and $\omega = \pi/4$. The performance of the LMS algorithm resulted in a misadjustment of 0.892×10^{-2} (compared to the theoretical value of 0.9×10^{-2}). For the LMF algorithm

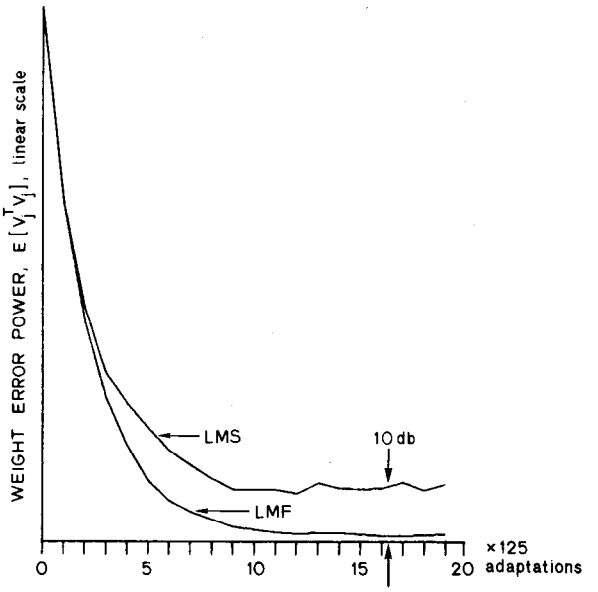

Fig. 5. Learning curves for LMS and LMF algorithms with rectangularly distributed error.

the misadjustment was only 0.26×10^{-2} (theoretical value 0.25×10^{-2}) for the same time constant.

Example 3: For this example, the configuration of Fig. 4 is used. We let n_j consist of randomly distributed samples of $\pm a$. Accordingly, Table I shows that $\alpha(2) = 9$. In this case the improvement due to the implementation of the LMF algorithm is close to 10 dB!

The results of the corresponding computer simulation are presented in Fig. 5. Once more exactly the same conditions were used as during the previous two examples; n_j was a randomly distributed sequence of samples of amplitude ± 10. (A square-wave n_j would have given the same result.) For the LMS algorithm the misadjustment of 0.960×10^{-2} was measured (compared to the theoretical value of 0.9×10^{-2}). The lower curve in Fig. 5 corresponds to the LMF algorithm. The misadjustment was only 0.099×10^{-2} (compared to the theoretical value of 0.1×10^{-2}) for the same time constant.

All three examples have verified the following:

a) that expressions (28), (29), and (39) for stability, time constants, and misadjustment closely agree with the computer simulations of the performance of the steepest descent algorithms for $K \geq 1$ for small values of adaptation constant μ;
b) that in certain important cases, the choice of $K > 1$ enables considerable improvement of the performance of the adaptive algorithms.

However, though potentially advantageous, the implementation of "higher order error algorithms" requires a certain degree of caution. First of all it should be noted that in certain cases the choice of $K > 1$ might cause deterioration in the performance. For instance if n_j is Gaussian then $\alpha(2) = (9/15) < 1$. Therefore in that particular case, LMS will outperform LMF.

Even when $\alpha(k) > 1$, the choice of large K is generally associated with decrease in the degree of stability of the algorithm for the given initial conditions.[2] In some cases it might be advisable (if feasible from the computational point of view) to employ simultaneously both the LMS and LMF algorithms. Both algorithms could be designed to work with the same time constant. If the difference between the errors of the two algorithms is low, then the LMF algorithm should be used to operate with the lower misadjustment. However, if the difference between the two algorithms exceeds a certain preprogrammed threshold, then a system controller will conclude that the LMF algorithm has gone unstable and lost track of the optimal solution. The controller will switch to the LMS, and the LMF algorithm will restart its adaptation using the current weights of the LMS process as an initial condition. Then the controller will switch back to using the LMF output.

V. Conclusion

A new family of algorithms was introduced to adjust the weights of an adaptive filter so that the expected value of the error to the degree $2K$ would be minimized. The concept of steepest descent was used as a basis. Hence all these algorithms can be viewed as an extension of the Widrow–Hoff LMS algorithm.

The performance of the new algorithms was analyzed and it was shown that in certain cases the choice of $K > 1$ will outperform the conventional LMS algorithm by a considerable margin. By "outperform" we mean less noise in the weights for the same speed of convergence. Examples were presented in which the LMF algorithm performs better than the LMS algorithm by margin of 3–10 dB. Use of LMF is not universally indicated, however, for one case (Gaussian plant noise) presented showed LMS to outperform LMF by about 3 dB. The best value of K can be chosen by using expressions (39) and (40). Simulation results were presented to illustrate and verify the theoretical results.

The above analysis was limited to the steepest descent algorithms. However, the results obtained have even more general implications. Corresponding to the steepest descent LMS and LMF algorithms are the "exact" least squares algorithms [9]–[11] and "exact" least fourth algorithms which are yet to appear in the literature. Minimization of the mean fourth error instead of minimization of mean square error can be shown to offer similar benefits (lower estimation variance for the same amount of input data) in similar cases when using "exact" algorithms. This will be the subject of a future paper.

[2] In order to illustrate this fact, Example 1 was simulated with the choice of $K > 2$ and initial weight error increased 16 times. Theoretically, for uniformly distributed n_j, $\alpha(K) = (4K - 1)/3$. Hence, the misadjustment should decrease proportionally to K. Indeed, for $K = 4$ the misadjustment decreased to the 0.18×10^{-2}. However, for the given initial conditions and $K = 5$ the algorithm "blew up." Then the simulation was performed again, starting this time with lower initial error. This time the algorithm (for $K = 5$) converged without any difficulty. Hence, the choice of the optimal K must be a compromise between the best misadjustment (as defined by (39)) and convergence limits induced by the expected error in the initial conditions.

Acknowledgment

We would like to thank John M. Cioffi for many fruitful discussions of these subjects and Dr. Odile Macchi for her many valuable suggestions and patient editing. We have found the reviewer's comments to be especially helpful.

Appendix

Our purpose is to analyze the convergence of the variance of the LMS family of algorithms given by (15) (or (18)). It should be noted that generally (for $K > 1$) the convergence properties of LMF algorithms depend on the choice of initial conditions. Thus our stability analysis must be limited to a certain bounded domain D around the Wiener solution. Let D be a sphere of radius d

$$\boldsymbol{V}_j \in D \Leftrightarrow \boldsymbol{V}_j^T \boldsymbol{V}_j < d^2. \tag{A1}$$

The radius d must be finite, although it can be chosen to be arbitrary large so that domain D will encompass all the possible choices of the initial position of the adaptation process.

We will prove that for every finite D, there exists a nonempty range of adaptation constants μ

$$0 \le \mu \le \mu_0 \tag{A2}$$

that inside D, each iteration decreases the variance of the weight error vector, i.e.,

$$\limsup_{j \to \infty} E\left[\boldsymbol{V}_j^T \boldsymbol{V}_j\right] = c(\mu),$$

where $c(\mu)$ is a finite scalar. Moreover we will show that for $\mu \to 0$ the convergence point $c(\mu) \to 0$. In addition we will develop an approximate evaluation of the stable range $[0, \mu_0]$. Our derivation will be based on the fundamental assumption that the sequences $\boldsymbol{V}_j$, n_j, and $\boldsymbol{X}_j$ are independent random variables.[3]

Using (19) once more, the impact of each iteration can be evaluated from

$$\boldsymbol{V}_{j+1}^T \boldsymbol{V}_{j+1} = \boldsymbol{V}_j^T \boldsymbol{V}_j - 4\mu K \left(\sum_{i=0}^{2K-1} \binom{2K-1}{i} n_j^i \left(-\boldsymbol{X}_j^T \boldsymbol{V}_j\right)^{2K-i} \right) + 4\mu^2 K^2 \boldsymbol{X}_j^T \boldsymbol{X}_j \left(\sum_{i=0}^{2K-1} \binom{2K-1}{i} n_j^i \left(-\boldsymbol{X}_j \boldsymbol{V}_j\right)^{2K-1-i} \right)^2. \tag{A3}$$

Take expectations of both sides of (A3). These expectations can be done in two stages: assume first that $\boldsymbol{V}_j$ is given, then find conditional expectation of both sides of (A3). Then take expectation over all possible $\boldsymbol{V}_j \in D$. Since n_j was assumed to be independent of $\boldsymbol{X}_j$ and $\boldsymbol{V}_j$, and to have zero odd moments, all terms on the right-hand side of (A3) which include an odd power

[3] Strictly speaking this assumption is not realistic, however, it is widely accepted and used (see, for instance, [6]). For the simple case of $K = 1$, more rigorous proofs of convergence which do not assume independence of $\boldsymbol{V}_j$, and $\boldsymbol{X}_j$, also exist (see [7] and [8]).

of n_j will vanish under expectation. Hence the conditional expectation is

$$E_{V_j}\left[V_{j+1}^T V_{j+1}\right] = V_j^T V_j - 4\mu K E_{V_j}\left[\sum_{i=0}^{K-1}\binom{2K-1}{2i} n_j^{2i}\left(X_j^T V_j\right)^{2(K-i)}\right] + 4\mu^2 K^2 E_{V_j}\left[X_j^T X_j\left(\sum_{i=0}^{2K-1}\binom{2K-1}{i} n_j^i\left(-X_j V_j\right)^{2K-1-i}\right)^2\right]. \tag{A4}$$

For a bounded V_j, the last term on the right side of (A4) must also be bounded. Assume that X_j and n_j have finite higher order moments. As such, we can find a positive scalar a so that for every V_j in the domain D,

$$0 \le E_{V_j}\left[X_j^T X_j\left(\sum_{i=0}^{2K-1}\binom{2K-1}{i} n_j^i\left(-X_j^T V_j\right)^{2K-1-i}\right)^2\right] < a. \tag{A5}$$

Denote by B_j the following $n \times n$ dimensional matrix,

$$B_j \triangleq E_{V_j}\left[\sum_{i=0}^{K-1}\binom{2K-1}{2i} n_j^{2i}\left(X_j^T V_j\right)^{2(K-i-1)} X_j X_j^T\right]. \tag{A6}$$

This matrix is positive definite for arbitrary V_j, assuming as before that $R = E[X_j X_j^T]$ is positive definite. Substitution of (A5) and (A6) into (A4) yields

$$E_{V_j}\left[V_{j+1}^T V_{j+1}\right] < V_j^T\left(I - 4\mu K B_j\right) V_j + 4\mu^2 K^2 a. \tag{A7}$$

Moreover all the terms in (A6) are nonnegative definite matrices and thus for all V_j

$$B_j \ge C_j \triangleq (2K-1)\, E\left[n_j^{2K-2}\right] R, \tag{A8}$$

which is the $i = K-1$ term in (A6). Hence we can find a positive scalar $b_{\min}$ such that for arbitrary V_j all the eigenvalues of B_j will be greater than $b_{\min}$. Furthermore, for arbitrary $V_j \in D$, the maximal eigenvalue of B_j can be bounded by a certain scalar $b_{\max}$. Hence

$$0 < b_{\min} < \left\{\begin{array}{l}\text{all eigenvalues of } B_j \\ \text{for arbitrary } V_j \in D\end{array}\right\} < b_{\max}. \tag{A9}$$

Choose the adaptation constant μ such that

$$0 < \mu < \frac{1}{4Kb_{\max}}. \tag{A10}$$

Then all the eigenvalues of the matrix $I - 4\mu K B_j$ will be positive but smaller than 1. Hence

$$V_j^T\left(I - 4\mu K B_j\right) V_j < \left(1 - 4\mu K b_{\min}\right) V_j^T V_j. \tag{A11}$$

Substitution of (A11) into (A7) yields

$$E_{V_j}\left[V_{j+1}^T V_{j+1}\right] < \left(1 - 4\mu K b_{\min}\right) V_j^T V_j + 4\mu^2 K^2 a. \tag{A12}$$

Relation (A12) holds for every $V_j \in D$. Therefore we can average it over all possible choices of $V_j \in D$:

$$E\left[V_{j+1}^T V_{j+1}\right] < \left(1 - 4\mu K b_{\min}\right) E\left[V_j^T V_j\right] + 4\mu^2 K^2 a. \tag{A13}$$

Since $(1 - 4\mu K b_{\min}) < 1$ from (A10), $E[V_j^T V_j]$ has to converge in the following sense

$$\limsup_{j\to\infty} E\left[V_j^T V_j\right] < \frac{\mu K a}{b_{\min}}. \tag{A14}$$

Therefore if μ was chosen small enough the upper limit of the variance (limited by (A14)) can be brought arbitrary close to zero.

So far we have established that, subject to the assumptions stated above, there is a nonempty range of adaptation constant μ in which algorithm (15) will converge inside the given bounded domain D. However, in practice verification of stability condition (A10) might be quite cumbersome. Therefore it is useful to have a simple, practical approximation for finding the stable range of μ. We will derive such a range based on the assumption of small deviations from the Wiener solution, i.e.,

$$\|V_j\| \approx 0. \tag{A15}$$

Recall that (A3) and (A4) which describe the impact of iteration j on the variance of V_j. In the vicinity of the Wiener solution, when (A15) is true, we can neglect in (A4) all terms which depend on V_j to the power higher than two. Hence

$$E_{V_j}\left[V_{j+1}^T V_{j+1}\right] \cong V_j^T V_j - 4\mu K(2K-1) E_{V_j}\left[n_j^{2K-2}\left(V_j^T X_j\right)^2\right] + 4\mu^2 K^2 E_{V_j}\left[X_j^T X_j\left(n_j^{4K-2} + (2K-1)\cdot(4K-3) n_j^{4K-4}\left(V_j^T X_j\right)^2\right)\right]. \tag{A16}$$

The product $(V_j^T X_j)^2$ can be represented as $V_j^T X_j X_j^T V_j$. Rearranging the terms, and using once more the assumption of independence of X_j, V_j, and n_j, yields

$$E_{V_j}\left[V_{j+1}^T V_{j+1}\right] = V_j^T G V_j + 4\mu^2 K^2 E\left[X_j^T X_j\right] E\left[n_j^{4K-2}\right], \tag{A17}$$

where

$$G \triangleq I - 4\mu K(2K-1) E\left[n_j^{2K-2}\right] E\left[X_j X_j^T\right] + 4\mu^2 K^2 (2K-1)(4K-3) E\left[n_j^{4K-4}\right] E\left[X_j^T X_j X_j X_j^T\right]. \tag{A18}$$

By inspection of (A17), it is clear that convergence properties depend solely on the nature of the matrix G: the algorithm will converge if and only if the magnitudes of all eigenvalues of G are less than one. In order to evaluate the matrix G we will need an additional approximation:

$$X_j^T X_j \cong \text{const.} \cong nE\left[x_j^2\right], \tag{A19}$$

which is quite reasonable for large values of n. Substitution of (A19) into (A18) yields

$$G = I - 4\mu K(2K-1)\left\{E\left[n_j^{2K-2}\right] - \mu K(4K-3) nE\left[x_j^2\right] E\left[n_j^{4K-4}\right]\right\} R. \tag{A20}$$

Since the autocorrelation matrix R is assumed to be positive definite, all eigenvalues of G will have absolute values smaller than one if and only if

$$0 < \mu < \frac{E\left[n_j^{2K-2}\right]}{K(4K-3) nE\left[x_j^2\right] E\left[n_j^{4K-4}\right]} \tag{A21}$$

and

$$1 - \gamma_{\max} 4\mu K(2K-1)\left(E\left[n_j^{2K-2}\right] - \mu K(4K-3) nE\left[x_j^2\right] E\left[n_j^{4K-4}\right]\right) > -1 \tag{A22}$$

where $\gamma_{\max}$ is the maximal eigenvalue of R. Condition (A22) will be always satisfied, regardless of the choice of μ. Hence, (A21) is a sufficient condition for stability of the variance of the weight vector for small deflection from the Wiener solution.

It should be noted that use of stability condition (A21) is quite straightforward from a practical point of view, since only knowledge of input power and plant noise moments is required.

In the derivation of the stability condition (A21), several simple assumptions were made: (A15) and (A19). In our experience, however, this stability condition turned out to be quite robust and to provide an excellent approximation of the stable range of adaptation constant μ.

References

[1] S. Sherman, "Non-mean-square error criteria," *IEEE Trans. Inform. Theory*, pp. 125–26, Sept. 1958.

[2] M. Zakai, "General error criteria," *IEEE Trans. Inform. Theory*, pp. 94–95, Jan. 1964.

[3] J. L. Brown, Jr., "Asymmetric non-mean-square error criteria," *IRE Trans. Automat. Contr.*, Jan. 1962.

[4] A. Gersho, "Some aspects of linear estimation with non-mean-square error criteria," in *Proc. Asilomar Ckts. and Systems Conf.*, 1969.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

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

[11] J. M. Cioffi and T. Kailath, "Fast, recursive-least-squares, transversal filters for adaptive filtering," *IEEE Acoust., Speech, Signal Processing*, vol. ASSP-32, Apr.'84.

Analysis and Performance Evaluation of an Adaptive Notch Filter

BENJAMIN FRIEDLANDER, SENIOR MEMBER, IEEE, AND JULIUS O. SMITH, MEMBER, IEEE

Abstract—**An adaptive notch filter is derived by using a general prediction error framework. The proposed infinite impulse response filter has a special structure that guarantees the desired transfer characteristics. The filter coefficients are updated by a version of the recursive maximum likelihood algorithm. The convergence properties of the algorithm and its asymptotic behavior are discussed, and its performance is evaluated by simulation results.**

1. Introduction

DETECTION, estimation, and filtering of narrow-band signals in the presence of noise, are some of the most common problems in signal processing. In some applications such as interference rejection and correlation processing, it is desired to remove the narrow-band signal components while leaving the broad-band energy unchanged. This can be achieved by passing the signals through a notch filter of the type depicted in Fig. 1, where the notches are centered on the narrow-band signals.

When the frequencies of the narrow-band components are known, the design of such notch filter is straightforward (Fig. 1). Standard time-domain and frequency-domain implementations are available [1]. A simple time-domain filter will have pole-zero pairs, with zeros located on the unit circle at the desired notch frequencies. The situation is more complicated when the narrow-band frequencies are not known *a priori*. In this case, it is necessary to design an adaptive filter in which the notch frequencies and bandwidths will be adjusted based on the input data.

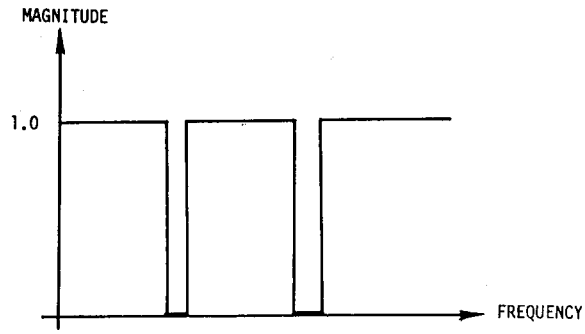

Fig. 1. Transfer function of an ideal notch filter.

In this paper we develop an adaptive infinite impulse–response (IIR) notch filter, based on the general prediction error framework used by Ljung [2], [3] to de-

Manuscript received March 30, 1983; revised August 8, 1983. This work was supported by the Office of Naval Research under Contract N00014-82-C-0476.

The authors are with Systems Control Technology, Inc., 1801 Page Mill Road, Palo Alto, CA 94304.

0018-9448/84/0300-0283$01.00