Adaptive Blind Equalization Using Second- and Higher Order Statistics

Fang B. Ueng and Yu T. Su

Abstract— **This paper presents two classes of adaptive blind algorithms based on second- and higher order statistics. The first class contains fast recursive algorithms whose cost functions involve second and third- or fourth-order cumulants. These algorithms are stochastic gradient-based but have structures similar to the fast transversal filters (FTF) algorithms. The second class is composed of two stages: the first stage uses a gradient adaptive lattice (GAL) while the second stage employs a higher order-cumulant (HOC) based least mean squares (LMS) filter. The computational loads for these algorithms are all linearly proportional to the number of taps used. Furthermore, the second class, as various numerical examples indicate, yields very fast convergence rates and low steady state mean square errors (MSE) and intersymbol interference (ISI). MSE convergence analyses for the proposed algorithms are also provided and compared with simulation results.**

I. Introduction

THE PURPOSE of blind equalization is to recover the intersymbol interference and noise corrupted signal from the received signal without the help of a training signal. Earlier investigators like Sato [1], Godard [3], and Benvensite and Goursat [2] used different LMS-type algorithms to deal with this problem. Since the second order cumulant (i.e., autocorrelation function) is completely blind to the phase property of the channel to be identified, if the channel is not minimum phase, these algorithms may not be capable of generating correct results. Statistics of higher order must therefore be considered in the realization of blind equalization of nonminimum phase (NMP) channels.

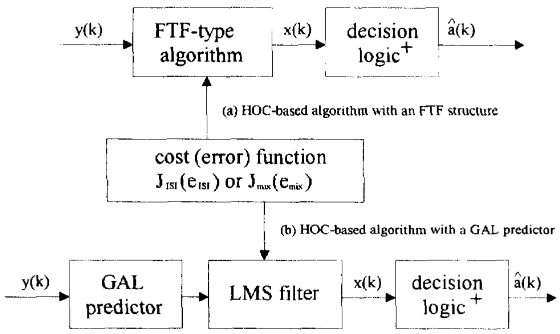
Giannakis [5] has showed that the impulse response of an FIR filter can be determined from the cumulants (third- or fourth-order) of the filter output alone. In other words, cumulants can be used to estimate the parameters of a MA model without any *a priori* knowledge of the transmitted data, if the input distribution is not Gaussian. His result was further extended to identify linear, time-invariant NMP systems with non-Gaussian correlated input sequences [4]. Swami and Mendel [6] derived a recursive algorithm for estimating the coefficients of an MA model of known order using autocorrelations and third-order cumulants. Zheng and McLaughlin [9] proposed an algorithm that uses closed-form formula to obtain an initial estimation and proceed to adaptively minimize the squared estimation error of the third order cumulants. Tugnait [7] used the total squared matching errors of various second and fourth-order statistics as the cost function to identify an ARMA model. Hatzinakos and Nikias [8] presented an adaptive blind equalization method using the complex cepstrum of the fourth-order cumulants (tricepstrum). Alshebeili, Venetsanopoulos, and Enis Çetin [10] suggested the use of second-order and all samples of third-order cumulants or the diagonal slice of bispectrum to identify FIR systems. Porat and Friedlander [11] described a nonlinear algorithm using the second- and fourth-order moments of the symbol sequence for equalizing QAM signals. All these algorithms are based on some closed-form relations between the parameters to be identified and the observed signal's cumulants of various orders or their Fourier transforms called polyceptra. For the application to channel equalization, extra steps are needed to use the estimated system parameters to recover the transmitted signal.

Shalvi and Weinstein (SW) [12] avoided these extra steps by devising new cost functions based on a necessary and sufficient condition for achieving zero ISI in nonminimum phase linear time-invariant channels. Recently, they developed [13] another two blind deconvolution algorithms that render learning speeds much faster than those of their earlier proposals in [12]. Since SW algorithms were aimed at removing ISI, the resulting MSE are often not as small as desired. Moreover, the convergence rate improvement of the second class of SW algorithms [13] was obtained at the expense of higher complexity and less flexibility for real-time implementation. This paper presents two classes of blind equalizers (see Fig. 1) that possess the properties of 1) fast learning speed, 2) small steady-state ISI and MSE, and 3) low computing complexity. A system model is introduced and candidate cost functions for achieving zero ISI are discussed in the next section. In Section III, we propose cost (error) functions for minimizing both ISI and MSE and present a fast transversal filter structure to perform multidimensional minimization of these cost functions. Another class of algorithms (Section IV) divide the equalization process into two stages. The received samples are whitened by a GAL filter in the first stage. The output sequence is then fed into a regular LMS filter (the second stage). In Section V, we provide computer simulation results on the ISI and MSE performance of various proposed algorithms and make comparison with the theoretical MSE behavior, which is analyzed in Appendix A. A brief summary and some conclusions are given in Section VI.

Manuscript received September 15, 1993; revised June 17, 1994. This work was supported in part by the National Science Council of Taiwan under Grant NSC 82-0404-D-009-024. This work was presented in part at the 1993 Wireless Communication Workshop, Chung Cheng University, Chia-Yi, Taiwan, September 17, 1993.

The authors are with the Department of Communications Engineering, National Chiao Tung University, Hsinchu 30035, Taiwan.

IEEE Log Number 9406083.



+ An inverse transform after decision is needed if the cost (error) function used depends on third-order cumulants

Fig. 1. Classification of the proposed blind equalizers; $J_{\rm ISI}(e_{\rm ISI})$ = cost (error) functions for minimizing ISI; $J_{\rm MSE}(e_{\rm MSE})$ = cost (error) functions for minimizing MSE, $J_{\rm mix} = f_1(J_{\rm ISI}, J_{\rm MSE})$, $e_{\rm mix} = f_2(e_{\rm ISI}, e_{\rm MSE})$.

II. Problem Formulation and HOC-Based Criteria

Let an i.i.d. sequence of symbols $\{a(k)\}$ be transmitted through a linear time-invariant channel, then the equivalent baseband output sequence $\{y(k)\}$ can be written as

$$y(k) = \sum_i b(i)a(k-i) + n(k) \tag{1}$$

where the additive noise $\{n(k)\}$ is a white Gaussian sequence and $\{b(i)\}$ is the channel impulse response. Suppose the channel output sequence $y(k)$ is fed through an FIR-type equalizer with impulse response $h(i)$, $i = 0, 1, \cdots, M$ then the filtered sequence becomes

$$x(k) = \sum_{i=0}^{M} h(i)y(k-i). \tag{2}$$

The purpose of a zero-forcing (i.e., intersymbol-interference elimination) equalizer is to find $\mathbf{h} = [h(0), h(1), \cdots, h(M-1)]$ such that the combining of the channel and the equalizer has the effect of a distortionless filter. In other words, the Z-transform of the perfect zero-forcing equalizer is equal to $1/B(z)$, $B(z)$ being the Z-transform of the channel impulse response $\{b(i)\}$. Denoting the Z-transform of $\mathbf{h}$ by $H(z)$, we can describe this condition as

$$H(z) = \frac{1}{B(z)} = \frac{1}{|B(z)|}e^{-j\angle B(z)} \tag{3}$$

where $\angle B(z)$ is the phase of $B(z)$. The magnitude of $H(z)$ can be estimated by using second-order statistics alone but not its phase. Hence, a suitable criterion should be a function of both second-order statistics and HOC's.

On the other hand, zero ISI requires that the taps $\{h(i)\}$ be such that the output $x(k)$ is identical to the input $a(k)$ up to a constant delay. That is, the combined channel-equalizer impulse response

$$s(i) = h(i) \otimes b(i) = \sum_l h(i-l)b(l) \tag{4}$$

where $\otimes$ stands for convolution, must be of the form

$$\mathbf{s} = e^{j\theta}(0 \cdots 1 \cdots 0). \tag{5}$$

When such a condition is satisfied, the distribution (or all cumulants) of the equalizer output is equal to that of the channel input data. In other words, the responsibility of a zero-forcing equalizer is to adjust its tap weights such that the instantaneous distribution of its output converges to the desired distribution [14]. Shalvi and Weinstein [12] simplified this requirement to one that involves only second- and fourth-order cumulants. They consider the combined channel/equalizer system $\{s(i)\}$ and showed

$$E[x^2(k)] = E[a^2(k)]\sum_l |s(l)|^2 \tag{6}$$

and

$$C_4[x(k)] = C_4[a(k)]\sum_l |s(l)|^4 \tag{7}$$

where $E[\cdot]$ denotes the expectation operator and the fourth-order cumulant $C_4[z]$ is defined by

$$C_4[z] = E[z^4] - 3E[z^2]. \tag{8}$$

Since

$$\sum_l |s(l)|^4 \le \left(\sum_l |s(l)|^2\right)^2 \tag{9}$$

with equality holds if and only if $\{s(l)\}$ has at most one nonzero component, perfect equalization implies that $\sum_l |s(l)|^4 = 1$ if and only if $\sum_l |s(l)|^2 = 1$ or equivalently, we can use the following criterion [12]

$$\begin{array}{ll} \text{maximize} & |C_4[x(k)]| \\ \text{subject to} & E[x^2(k)] = E[a^2(k)]. \end{array} \tag{10}$$

A similar criterion involving third-order cumulants can also be derived. From the identity [9]

$$C_3^x(m_1, m_2) = C_3^a(0, 0)\sum_l s(l)s(l+m_1)s(l+m_2) \tag{11}$$

where $C_3^z(m_1, m_2) = E[z(l)z(l+m_1)z(l+m_2)]$, we conclude that, if, as in the previous case, $E[x^2(k)] = E[a^2(k)]$, then

$$\begin{aligned} |C_3^x(0, 0)| &= \left|E[x^3(k)]\sum_l s^3(l)\right| \le |E[a^3(k)]|\sum_l |s(l)|^2 \\ &= |E[a^3(k)]|. \end{aligned} \tag{12}$$

Hence, we have an alternative criterion

$$\begin{array}{ll} \text{maximize} & |C_3[x(k)]| \\ \text{subject to} & E[x^2(k)] = E[a^2(k)]. \end{array} \tag{13}$$

The maximization can be replaced by minimization of

$$J_3 = \left\{\frac{E[|x(k)|^2]}{E[|a(k)|^2]} - \frac{|C_3(x(k))|}{E[a^3(k)]}\right\}^2. \tag{14}$$

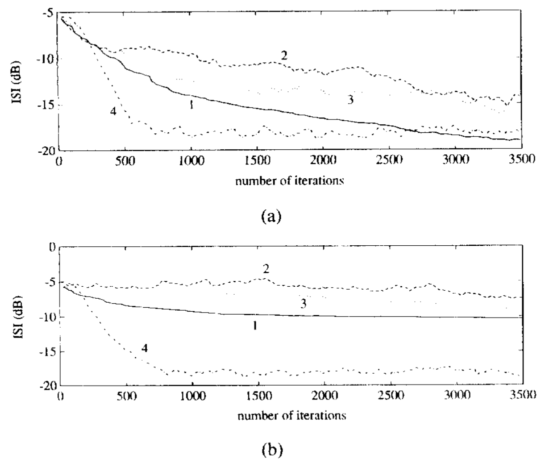


Fig. 2. ISI learning curves for (a) Channel 1 (b) Channel 2; (1) J_4-LMS algorithm, (2) B–G algorithm, (3) Sato algorithm, (4) J_4-FTF algorithm.

Note that the inequality (9) also implies that the object function

$$J_4 = \left[\frac{E[|x(k)|^2]}{E[|a(k)|^2]} - 1\right]^2 + \left[\frac{C_4[x(k)]}{C_4[a(k)]} - 1\right]^2 \qquad (15)$$

is minimized by (5). These facts immediately suggest that blind zero-forcing equalization can be accomplished by applying a stochastic gradient-based method to perform either the constrained maximization (12), (13) or the unconstrained minimization of (15). Another approach suggeted by Shalvi and Weinstein [13] was motivated by the observation that the transformation

$$\mathbf{s}'(n) = \mathbf{s}^p(n)[\mathbf{s}^*(n)]^q$$

where $\mathbf{s}(n)$ is the vector representing the combined channel/equalizer impulse response at the nth iteration and $p+q \geq 2$, followed by the normalization

$$\mathbf{s}(n+1) = \mathbf{s}'(n)/||\mathbf{s}'(n)||$$

causes the combined impulse response $\{s(k)\}$ to converge quickly to the desired response (5). Two algorithms based on this fact, one in batch mode the other in sequential mode, were proposed in [13].

All these algorithms, as mentioned before, were designed to eliminate ISI. Their MSE performance is often not satisfactory and in some cases is unacceptable (see Fig. 2). Moreover, a stochastic gradient algorithm using (10), (13), (14), or (15) as its cost function is sensitive to the characteristic of the transmitting channel (see Fig. 5). The fast algorithms of [13] are more robust but require a complexity of $O(M^2)$ where M is the equalizer length; they are not particularly suitable for real-time operation either. Moreover, the batch-processed super exponential algorithm of [13] may exhibit undesired jittering after it converges [see Fig. 7(a)]. Algorithms proposed below will not have these shortcomings.

III. HOC-Based Algorithms with an FTF Structure

Although the functions defined by (14) and (15) are suitable criteria for minimizing ISI, minimum MSE cannot be achieved by using either of them alone. In fact, it can be shown that in an additive white Gaussian noise channel the minimum MSE solution leads to

$$\sum_l |s(l)|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left|\frac{B(\omega)}{B(\omega)+N_0}\right|^2 d\omega \overset{\text{def}}{=} \sigma^2 < 1 \qquad (16)$$

where $B(\omega)$ is the discrete Fourier transform of $\{b(i)\}$ and N_0 is the one-sided power spectral density of the additive channel. In the presence of noise the zero ISI upper bound $\max\{C_4[x(k)]\} = C_4[a(k)]$ cannot be attained either. As the noise level N_0 increases, the constraint of searching on the surface of the unit ball $\{\mathbf{s}: \sum_l |s(l)|^2 = 1\}$ (i.e., $E[x^2(k)] = E[a^2(k)]$) will drive the tap-weight vector $\mathbf{h}(k)$ further and further away from the minimum MSE solution which must be obtained from the surface $\{\mathbf{s}: \sum_l |s(l)|^2 = \sigma^2\}$. An apprpriate solution is to add to the HOC-based cost (or error) function a term which reflects the magnitude of the MSE under blind circumstances. MSE involves only second-order statistics and cost (error) functions for minimizing the MSE of a blind equalizer have been presented before [1]–[3]. The following error function is a good candidate for minimizing both ISI and MSE:

$$\begin{aligned} e_{2,\,\text{hoc}}(k) &= -k_1|e_{dd}(k)|\frac{\partial J_4}{\partial h_i} \\ &\quad + k_2 e_{dd}(k) + k_3|e_{dd}(k)|e_{sa}(k) \\ &\overset{\text{def}}{=} -k_1|e_{dd}(k)|\frac{\partial J_4}{\partial h_i} + e_2(k) \end{aligned} \qquad (17)$$

where k_i's are appropriate weighting factors, $\hat{a}(k)$ = hard-decision output based on $x(k)$, $e_{dd}(k) = \hat{a}(k) - x(k)$ is the decision-directed error signal, $e_{sa}(k) = x(k) - \alpha\,\text{sgn}\,[x(k)]$, and $\alpha = E[a^2(k)]/E[|a(k)|]$. Note that the inclusion of $e_{dd}(k)$ and $|e_{dd}(k)|$ in $e_{2,\,\text{hoc}}$ serves two related purposes: 1) to measure the quality of the current equalizer output or the "distance" from the current estimation of $\{s(i)\}$ to the desired response and 2) to offer an automatic switch between the start-up period and the standard transmission mode [2].

Another possible error signal can be obtained by considering the error signal used in the super exponential method [13] which updates the equalizer's tap-weight vector by

$$\begin{aligned} \mathbf{h}(k) = \mathbf{h}(k-1) + \frac{\beta}{\delta}Q(k)[x^2(k) - 3m_a^2]x(k) \\ - \delta x(k)\mathbf{y}(k) \end{aligned} \qquad (18)$$

where $m_a^2 = E[a^2(k)]$, β is a constant, $\mathbf{y}(k) = [y(k), y(k-1), \cdots, y(k-M+1)]^T$, $\delta = C_4[a(k)]/E[|a(k)|^2]$ and $Q(k)$ is an estimator of the matrix R_L^{-1}/m_a^2, $R_L = E[\mathbf{y}(k)\mathbf{y}^T(k)]$. In [13], $Q(k)$ is updated by another recursive formula which must be initialized by batch processing a large data segment first. We now present an algorithm that a) does not need initial batch processing, b) has a very small MSE, c) is insensitive to the eigenspread of the received data, and d) requires an $O(M)$

TABLE I
THE SW2-FTF BLIND EQUALIZER

$e_f(n\|n-1) = F^T(n-1)y(n) =$ *a priori* forward prediction error	(a)
$e_f(n\|n) = \gamma_f(n)e_f(n\|n-1) =$ *a posteriori* forward prediction error	(b)
$E_f(n) = E_f(n-1) + e_f(n\|n)e_f(n\|n-1) =$ accumulated forward prediction error	(c)
$\gamma(n) = \gamma_f(n)E_f(n-1)/E_f(n)$	(d)
$k_M(n\|n-1) = \frac{e_f(n\|n-1)}{E_f(n-1)}F(n-1) + \begin{bmatrix} 0 \\ k_{f,\,M-1}(n) \end{bmatrix}$	(e)
$\begin{bmatrix} k_{b,\,M-1}(n) \\ 0 \end{bmatrix} = k_M(n\|n-1) - \frac{e_b(n\|n-1)}{E_b(n-1)}\mathbf{G}(n-1)$	(f)
$e_b(n\|n-1) = k_{M,\,M}(n\|n-1)E_b(n-1) =$ *a priori* backward prediction error	(g)
$\gamma_b(n) = [1 - e_b(n\|n-1)\gamma(n)k_{M,\,M}(n\|n-1)]^{-1}\gamma(n)$	(h)
$e_b(n\|n) = \gamma_b(n)e_b(n\|n-1) =$ *a posteriori* backward prediction error	(i)
$F(n) = F(n-1) - e_f(n\|n-1)\begin{bmatrix} 0 \\ k_{f,\,M-1}(n) \end{bmatrix} =$ forward prediction filter	(j)
$\mathbf{G}(n) = \mathbf{G}(n-1) - e_b(n\|n-1)\begin{bmatrix} k_{b,\,M-1}(n) \\ 0 \end{bmatrix} =$ backward prediction filter	(k)
$x(n) = \mathbf{h}^T(n-1)\mathbf{y}(n) =$ equalizer output signal	(l)
$e(n) = (x^2(n) - 3m_a^2 - \delta_l)x(n) =$ error signal	(m)
$\mathbf{h}(n) = \mathbf{h}(n-1) + n\beta m_a^2 \|e_{dd}(n)\|\gamma(n)e(n)k_M(n\|n-1) + \mu e_2(n)\mathbf{y}(n)$	(n)
$k_{f,\,M-1}(n+1) = k_{b,\,M-1}(n)$	(o)
$\gamma_f(n+1) = \gamma_b(n)$	(p)
$E_b(n) = E_b(n-1) + e_b(n\|n-1)e_b(n\|n) =$ accumulated backward prediction error	(q)

complexity only. b) and c) can be accomplished by replacing (18) with

$$\mathbf{h}(k) = \mathbf{h}(k-1) + \left\{ \frac{\beta}{\delta}|e_{dd}(k)|R_L^{-1}[(x^2(k) - 3m_a^2)x(k) - \delta x(k)] + \mu e_2(k) \right\}\mathbf{y}(k) \quad (19)$$

a) and d) are achieved by first defining the time domain autocorrelation matrix

$$R(k) \stackrel{\text{def}}{=} \frac{1}{k}\sum_{J+!}^{k}\mathbf{y}(j)\mathbf{y}^T(j) \quad (20)$$

and replacing R_L^{-1} in (16) by $R^{-1}(k)$. Noting that the unweighted correlation matrix $kR(k)$ and its inverse can be computed recursively in a way similar to that in a recursive least-squares method [18], we define the forward predictor's Kalman gain vector as

$$k_M(k|k-1) = R^{-1}(k-1)\mathbf{y}(k). \quad (21)$$

It follows [18]

$$R^{-1}(k)\mathbf{y}(k) = \gamma(k)k_M(k|k-1) \quad (22)$$

where $\gamma(k)$ is the so-called conversion factor. Substituting the above equation into (19), using the analogy between the resulting equation and the recursive relation governing the update of the tap-weight vector for the FTF algorithm, and after some algebraic manipulations we obtain the blind equalizer described by Table I. This algorithm will be called the SW2-FTF algorithm henceforth. The same approach can be applied to other cost (error) functions [(10), (13)–(15), or (17)] with proper modifications made on steps (m) and (n) of Table I. The resulting algorithms all have the same order of complexity and enjoy the same advantages [i.e., (a)–(d)].

When the constrained minimization of J_3 is implemented the transmitted data sequence has to be transformed first unless it has a nonzero skewness (i.e., $E[a^3(k)] \neq 0$. If the i.i.d. sequence $\{a(k)\}$ is generated from a (normalized) PAM signal set $\{S_i\}$ defined by

$$\{S_i\} = \{\pm 1, \pm 3, \cdots, \pm(2M-1)\} \quad (23)$$

with $p(S_i) = 1/2M$, then the nonzero cumulant requirement cannot be met. To solve this problem, [9] used the nonlinear transform:

$$S_i' = \ln[K(2M + S_i)] - \mu_0 \quad (24)$$

where $K \in [1/2M, 3/4M]$ is a compression factor and $\mu_0 = E\{[\ln[K(2M + S_i)]\}$. At the receiving end, to restore the original transmitted signal, it is necessary to make the inverse transformation

$$S_i = \frac{1}{K}e^{\mu_0 + S_i'} - 2M \quad (25)$$

at the decision output (see Fig. 1). Instead of (24), (25) we suggest that the following simpler transform pair be used to accomplish the same purpose

$$S_i' = (S_i + 2M)^2 - \mu$$
$$S_i = \sqrt{S_i' + \mu} - 2M \quad (26)$$

where

$$\mu = \frac{1}{2M}\sum_{i=-(2M-1)}^{2M-1}(S_i + 2M)^2. \quad (27)$$

TABLE II
THE L-HOC BLIND EQUALIZER

A. The lattice filter:	
$e_0^+(n) = e_0^-(n) = y(n)$	(I)
$e_p^+(n) = e_{p-1}^+(n) - \gamma_p(n)e_{p-1}^-(n-1)$	(II)
$e_p^-(n) = e_{p-1}^-(n-1) - \gamma_p(n)e_{p-1}^+(n)$	(III)
$D_p(n) = \lambda_1 D_p(n-1) + e_p^-(n-1)^2 + e_p^+(n)^2$	(IV)
$\gamma_p(n) = \gamma_p(n-1) + \frac{\beta}{D_p(n)}[e_p^+(n)e_{p-1}^-(n-1) + e_p^-(n)e_{p-1}^+(n)]$	(V)
B. The transversal filter:	
$h_i(k+1) = h_i(k) - \mu e_{2,\text{hoc}}(k)e_p^+(k-i)$	(VI)
$h_i(k)$ is the ith tap weight at the kth iteration.	
The transversal filter input is the last stage output of the lattice filter.	

IV. HOC-BASED ALGORITHMS USING A GAL PREDICTOR

Although the above HOC-based blind algorithms have low computational load, their convergence time can still be reduced. It has been shown [15] that the start-up period can be shortened if the channel correlation matrix is orthogonalized. This can be attained by employing a whitening filter in front of the equalizer. We already know that [14, ch. 14] if the input of a lattice filter is wide-sense Markov of order M, then the forward prediction error produced at the Nth stage lattice filter, $N \geq M$, is white. Taking the Nth prediction error as the input to an HOC-based adaptive algorithm (like those mentioned in the previous section), we then obtain an equalizer that puts the responsibilities of estimating $1/|B(z)|$ and $\angle B(z)$ separately on two cascaded processes. Such a *division of labor* should be able to increase the learning speed. We now describe a new class of algorithms. All of them use a lattice predictor as a preliminary equalizer.

Let us consider the processing of the predictor output and ignore at first the GAL filtering part. Recall that the multidimensional Newton search results in the recursion [15, ch. 4]

$$\mathbf{h}(n+1) = \mathbf{h}(n) - \mu R_L^{-1}\frac{\partial \mathcal{T}}{\partial \mathbf{h}} \tag{28}$$

where $\mathcal{T}$ is the designed cost function and R_L is the autocorrelation matrix. The major difficulty in implementing this algorithm is the computing of the autocorrelation matrix R_L. The computing complexity can be greatly reduced if the sequence $\{y(n)\}$ is white which, as just mentioned, can be obtained by letting $\{y(n)\}$ pass through a lattice filter and taking the pth ($p \geq q$, q being the length of the channel's impulse response) stage forward prediction error $\{e_p^+(n)\}$ as the output. Redefining the input vector of the Newton algorithm as $\mathbf{y}(n) = [e_p^+(n)e_p^+(n-1)\cdots e_p^+(n-M+1)]^T$ we can express the autocorrelation matrix as

$$R_L = \text{diag}\,(E[e_p^{+2}(n)],\, E[e_p^{+2}(n-1)], \cdots, E[e_p^{+2}(n-M+1)]). \tag{29}$$

The ensemble averages, $E[e_p^{+2}(l)]$, $l = n, n-1, \cdots, n-M+1$, can be estimated by time averages

$$E[e_p^{+2}(l)] \approx \frac{1}{l}\sum_{k=1}^{l} e_p^{+2}(k) \stackrel{\text{def}}{=} \hat{E}_l. \tag{30}$$

For n sufficiently large all these estimators will approach a constant λ and therefore $R_L = \lambda I$, I is the identity matrix. Now the recursion (28) can be written as

$$h_i(n+1) = h_i(n) - \mu'\frac{\partial \mathcal{T}}{\partial h_i} \tag{31}$$

where $\mu' = \mu\lambda$. Replacing $-\mu'(\partial\mathcal{T}/\partial h_i)$ in the above equation by the product of $\mathbf{y}(n)$ and the error function $e_{2,\text{hoc}}$ defined by (17), we then obtain the algorithm presented in Table II. This algorithm will be referred to as the lattice-$J_{2,\text{hoc}}$ or the L-HOC algorithm. Again, the same approach can be employed to generate algorithms with different cost (error) functions for the second stage filter. We will omit the extensions to these cases and use L-HOC as a representative of its class.

As a remark, it is well-known that if $\{b(i)\}$ is minimum phase then the prediction error (innovation) $y(k) - E[y(k)|y(k-1), y(k-2), \cdots]$ is equal to $ca(k)$ where c is a constant. So if the transmitting channel is minimum phase, an equalizer with a GAL predictor (or any other whitening prefilter) should converge faster than one without (see Fig. 6).

V. SIMULATION AND NUMERICAL RESULTS

To demonstrate the usefulness of the proposed algorithms, computer simulations for equalizing the following channels are performed.

Channel 1 (eigenspread = 10.5, zeros at 7.1, $-0.24 \pm 0.19i$): $\mathbf{B_1} = [-0.15\ 1\ 0.5\ 0.1]$.

Channel 2 (eigenspread = 41.5, zeros at -2.1, -0.48): $\mathbf{B_2} = [0.3887\ 1\ 0.3887]$.

Channel 3 (eigenspread = 95.9, zeros at -1.8, -0.55): $\mathbf{B}_3 = [0.42\ 1\ 0.42]$.

Channel 4 (eigenspread = 150, zeros at -1.67, -0.6): $\mathbf{B}_4 = [0.44\ 1\ 0.44]$.

Channel 5 (all-pass channel):

$$h_i = \begin{cases} 0, & i < 0 \\ -0.4, & i = 0\,. \\ 0.84 * 0.4^{i-1}, & i > 0 \end{cases}$$

We use both the MSE and ISI measures in assessing the proposed algorithms' performance. ISI is defined as

$$\text{ISI} = \frac{\sum_l |s(l)|^2 - |s|^2_{\max}}{|s|^2_{\max}} \tag{32}$$

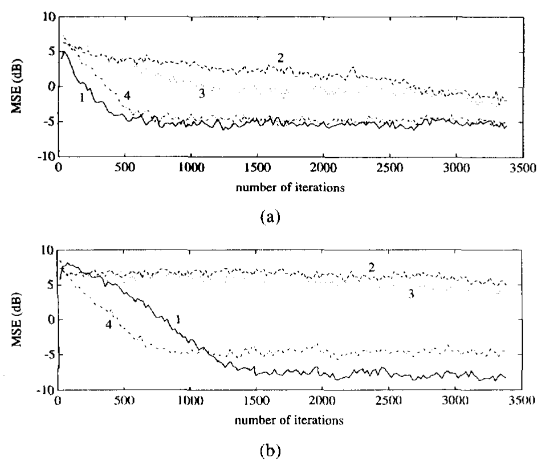


Fig. 3. MSE learning curves for (a) Channel 1 (b) Channel 2; (1) L-J_4 algorithm, (2) B–G algorithm, (3) Sato algorithm, (4) J_4-FTF algorithm.

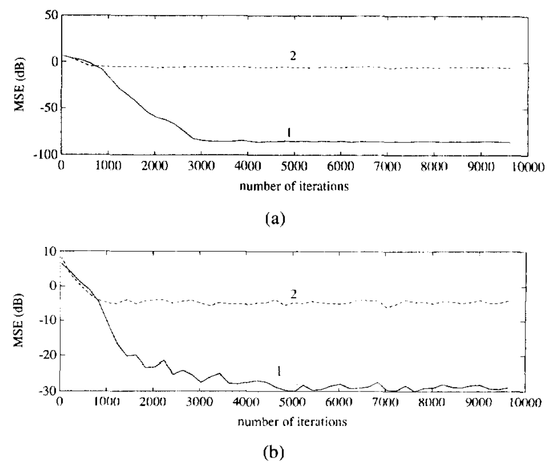


Fig. 4. Thermal noise free MSE performance comparison of (1) L-HOC and (2) L-J_4 algorithms; (a) Channel 1, (b) Channel 2.

where $|s|_{\max}$ is the maximum absolute value of the impulse response of the combined channel/equalizer system $\{s(l)\}$. All numerical results were obtained with a hundred runs. For HOC-based lattice or transversal filter we choose $\mu = 0.00001$ and $\mu = 0.0002$ is used for conventional blind algorithms. All filters hav a length of 15 taps. Besides the SW2–FTF and the L-HOC algorithms, we also compare the Sato algorithm [1], the Beniste–Goursat (B–G) algorithm [2], the J_4-LMS algorithm (one that uses the cost function J_4 with LMS filtering), and the L-J_4 algorithm (GAL predictor followed by J_4-LMS). The performance of the SW algorithms of [12] is not included since they and the J_4-LMS algorithm have similar learning behavior.

Fig. 2 exhibits ISI behaviors for the J_4-LMS, the J_4-FTF algorithms and two conventional blind equalizers at SNR $= 30$ dB. Obviously, the algorithms using J_4 as the cost function have a learning speed faster than conventional blind equalizers. Convergence speed improvement brought about by the FTF-based algorithm in identifying NMP channels can be found in Fig. 2 as well. In Fig. 3 the MSE performances of 1) the L-J_4 algorithm, 2) the B–G algorithm, 3) the Sato algorithm, and 4) the J_4-FTF algorithm are compared. It can be seen that the two HOC-based algorithms far outperform the other two. At SNR $= 30$ dB, convergence (MSE ≤ -5 dB) can be expected within 1500 iterations. Unfortunately, the steady-state MSE's of both HOC-based algorithms are relatively high. This drawback is removed by the addition to the original error signal of a term which measures the MSE, as can be seen from Fig. 4 where improvements of 80 (Channel 1) and 25 dB (Channel 2) are obtained. On the other hand, it indicates that the improvement is a decreasing function of the eigenspread of R_L.

The influence of the channel eigenspread is also shown in Fig. 5: when the eigenspread is large, the learning speed of the J_4-LMS algorithm (or those proposed in [12]) becomes so slow that the algorithm is of no practical use any more. Fig. 6 compares the learning curves obtained from both simulation and analysis. These curves confirm the correctness of our MSE analysis; they also indicate that the recursive formulae derived in the Appendix are useful in predicting the MSE learning behavior. The effect of GAL filtering can be found from Fig. 6(b): curve 3 represents MSE performance of the L-J_4 algorithm when equalizing a channel that resulted from cascading a minimum phase channel with Channel 5—an all-pass channel, and curve 4 is corresponding MSE performance of the J_4-LMS algorithm. Finally, in Fig. 7, we show ISI and MSE learning curves for SNR $= 10, 20, 30$ dB, respectively. We find out that both the steady-state ISI and MSE are sensitive to thermal noise. The effectiveness of $e_2(k)$ is clear: a 30 dB and 25 dB degradation on ISI and MSE performance results when the SW2-FTF algorithm is replaced by either the superexponential algorithm [Fig. 7(a)] or the J_4-FTF algorithm [see Fig. 3(b) and Fig. 7(b)]. Also shown in Fig. 7 is the ISI behavior (circled points) of the batch-processed super exponential algorithm [13]. We notice that its ISI does not remain stable after the algorithm converges. This is because this approach uses estimations of equalizer output HOCs to evaluate desired tap-weights $\{h(i)\}$ and the ISI measure is sensitive to estmation errors when the equalizer is at equilibrium. Such a jittering phenomenon can be avoided if we increase the batch size; but then the fast convergence advantage of this algorithm will no longer exist.

VI. Conclusions

This paper presents two classes of adaptive blind algorithms based on second- and higher order statistics. The first class consists of FTF based algorithms whose cost functions involve second and third- or fourth-order cumulants. The second class uses a gradient adaptive lattice predictor cascaded with an HOC-based stochastic gradient algorithm. The FTF-type algorithm used by the first class necessitates a numerical stabilization scheme [20] when implemented in finite-precision environment. The second class, on the other hand, is numerically stable, for both its first stage (GAL) and second-stage filters are. The computational loads for these algorithms are all linearly proportional to the filter length: the first class

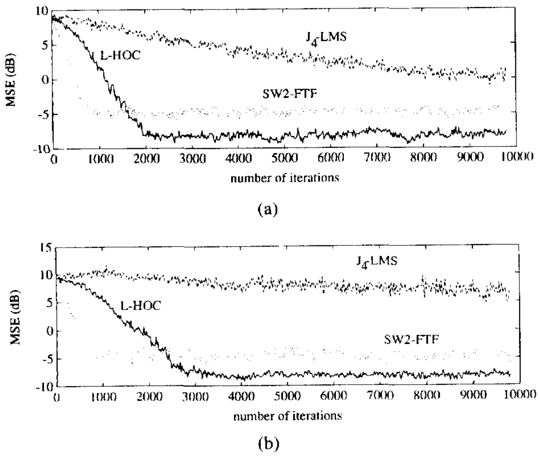


Fig. 5. The effect of channel characteristics on the equalizer's MSE performance; (a) Channel 3; (b) Channel 4.

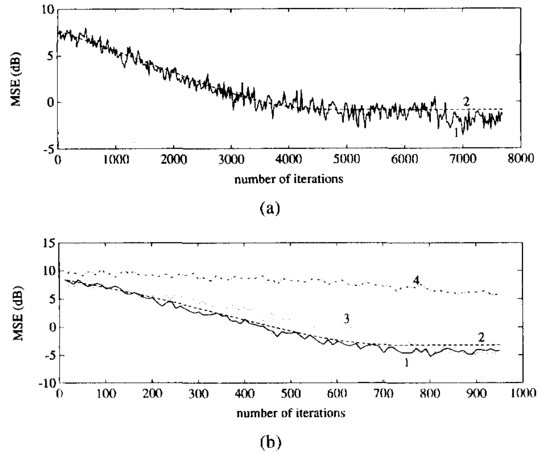


Fig. 6. J_4-LMS algorithm's MSE learning curves estimated by simulation (1) and analysis (2); (a) Channel 2, (b) Channel 5. (3) and (4) are the MSE learning curves for the L-J_4 and J_4-LMS algorithms, when equalizing the channel which is the combination of the minimum phase channel $\mathbf{B}_6 = [0.80.30.1]$ and Channel 5.

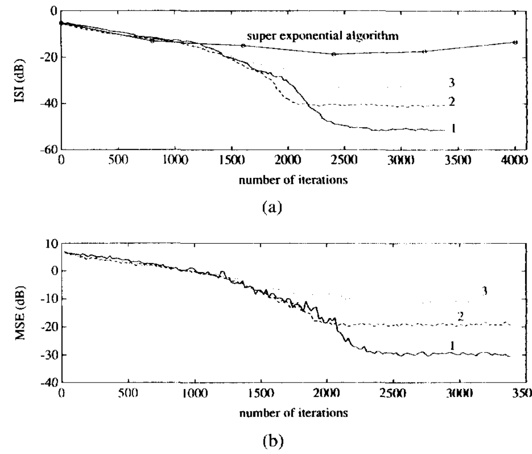


Fig. 7. The influence of the noise power level on the (a) ISI and (b) MSE performance of the $J_{2,\text{hoc}}$-FTF algorithm in equalizing Channel 2; (1) SNR = 30 dB, (2) SNR = 20 dB, (3) SNR = 10 dB. The learning curve of the batch-processed super exponential method at SNR = 30 dB is also shown (circled points).

requires a complexity of $O(9M)$ when its stabilized version is used; the second class needs $O(6M)$ only. As various numerical examples have shown, these new algorithms give very fast convergence rates, robustness against eigenspread variations, low steady-state MSE and ISI, and are suitable for real-time implementation. Simulation results also show that their learning behaviors are consistent with what the analysis had predicted.

Appendix A
Convergence Analysis

A. General Results

This Appendix provides MSE analysis of the proposed algorithms, assuming known signal constellation and channel impulse response. We follow the approach suggested in [19] where a recursive method is used to evaluate the time-evolution of the MSE. The basic assumptions used are[1]:

1) The data symbols, $a(k)$ are zero-mean independent and identically distributed symbols derived from a PAM data constellation.
2) The tap-weight vector $\mathbf{h}(k)$ is independent of the equalizer input vector $\mathbf{y}(k)$.
3) The equalizer output $x(k)$, conditioned on $a(k)$ and $\mathbf{h}(k)$, is zero-mean with variance $\sigma^2(n)$.

In general, the recursive formula governing an equalizer's tap-weight vector update is of the form

$$\mathbf{h}(k) = \mathbf{h}(k-1) - \mu C(k)e(k)\mathbf{y}(k) \qquad \text{(A.1)}$$

where $C(k)$ is an $M \times M$ matrix and $e(k)$ is the error signal at the kth iteration. Since the autocorrelation matrix R_L is positive definite, it can be decomposed into

$$R_L = \mathbf{PDP}^T \qquad \text{(A.2)}$$

where $\mathbf{P}$ is an unitary matrix, $\mathbf{D}$ is a diagonal matrix. Premultiplying (A.1) by $\mathbf{P}$, we obtain

$$\mathbf{W}(k) = \mathbf{W}(k) - \mu C(k)e(k)\mathbf{Z}(k) \qquad \text{(A.3)}$$

where $\mathbf{W}(k) = \mathbf{Ph}(k)$ and $\mathbf{Z}(k) = \mathbf{Py}(k)$ and the equalizer output becomes

$$x(k) = \mathbf{h}^T(k)\mathbf{y}(k) = \mathbf{W}^T(k)\mathbf{Z}(k). \qquad \text{(A.4)}$$

Invoking the assumption

$$E[w_i^{l-1}(k)w_j(k)] = \begin{cases} E\{w_i^{l-1}(k)\}E\{w_j(k)\} & i \neq j \\ \Gamma_i^{(l)}(n) & i = j \end{cases} \qquad \text{(A.5)}$$

where $\mathbf{W}(k) = [w_0(k), w_1(k), \cdots, w_{M-1}(k)]^T$ and using the definition

$$\begin{aligned}\sigma^2(k) &= E\{[x(k) - a(k)]^2\} \\ &= E\{x^2(k)\} + E\{a^2(k)\}\{-2E\{x(k)a(k)\}\} \qquad \text{(A.6)}\end{aligned}$$

[1]In [19] the extra assumption that $x(k)$ is stationary Gaussian is needed.

we arrive at

$$\sigma^2(k) = \mathbf{D}^T\mathbf{\Gamma}^{(2)}(k) + m_a^2 - 2m_a^2\mathbf{M}^T(k)\mathbf{G} \tag{A.7}$$

where $\mathbf{D}$ is the eigenvalue vector of R_L and $\mathbf{\Gamma}^{(2)}(k)$ is the vector formed by the mean square values of the equalizer taps in the transformed domain, $\mathbf{M}(k) = E[\mathbf{W}(k)]$ and $\mathbf{G} = \mathbf{PB}$ where $\mathbf{B}$ is the column vector representing the channel impulse response. Thus, time evolution of the MSE $\sigma^2(k)$ can be derived if we can recursively compute $\mathbf{M}(k)$ and $\mathbf{\Gamma}^{(2)}(k)$ when $C(k)$ and $e(k)$ are known. In the next two sections we derive these related recursive equations for the J_4-LMS algorithm and the L-J_4 algorithm. Convergence analysis for other algorithms can be derived in a similar manner.

B. J_4-LMS Algorithm

Substituting

$$C(k)e(k) = 4x(k)\{E[x^2(k)] - m_a^2\} + 8x^3(k)\{E[x^3(k)] - E[a^4(k)]\} \tag{A.8}$$

into (A.3) and taking expectation on both sides, we obtain

$$\mathbf{M}(k) = \mathbf{M}(k) - \mu\{4E[x(k)\mathbf{Z}(k)](E[x^2(k)] - m_a^2) + 8(E[x^4(k)] - E[a^4(k)])E[x^3(k)\mathbf{Z}(k)]\}. \tag{A.9}$$

Assumption 2) implies

$$E[x(k)\mathbf{Z}(k)] = \mathbf{D}\mathbf{\Gamma}^{(2)}(k) \tag{A.10}$$
$$E[x^4(k)] = \mathbf{D}^2\mathbf{\Gamma}^{(4)}(k) \tag{A.11}$$
$$E[x^3(k)\mathbf{Z}(k)] = \mathbf{D}^2\mathbf{\Gamma}^{(3)}(k) \tag{A.12}$$
$$E[x^2(k)] = \mathbf{D}^T i\mathbf{\Gamma}^{(2)}(k). \tag{A.13}$$

Substituting (A.8) into (A.3), taking square and then expectation, we have

$$\begin{aligned}\Gamma_i^{(2)}(k) = \Gamma_i^{(2)}(k-1) &+ 16\mu^2 E[x^2(k)Z_i^2(k)] \\ &\cdot (E[x^2(k)] - m_a^2)^2 + 64\mu^2\{E[x^4(k)] \\ &- E[a^4(k)]\}E[x^6(k)Z_i^2(k)] \\ &- 8\mu M_i(k)E[x(k)Z_i(k)](E[x^2(k)] - m_a^2) \\ &+ 16M_i(k)\mu\{E[x^4(k)] - E[a^4(k)]\} \\ &\cdot E[x^3(k)Z_i(k)] - 64\mu^2\{E[x^4(k)] - E[a^4(k)] \\ &\cdot E[x^4(k)Z_i^2(k)]E[x^2(k)] - m_a^2\}.\end{aligned} \tag{A.14}$$

Similarly, we can show

$$E[x^4(k)Z_i^2(k)] = D_{ii}^3\Gamma_i^{(4)}(k) \tag{A.15}$$
$$E[x^6(k)Z_i^2(k)] = D_{ii}^4\Gamma_i^{(6)}(k). \tag{A.16}$$

Define

$$w_i(k) = w_i(k-1) + \Delta w_i(k). \tag{A.17}$$

It is reasonable to assume that the ith tap weights at successive iterations are independent and therefore in the transformed domain, $\Delta w_i(k)$ is independent of $w_i(k-1)$. We further assume $\Delta w_i(k) \ll 1$ and define $m_i(k) = E[w_i(k)]$. Taking various powers of (A.17), employing the binomial expansion, and neglecting all terms involving $E[\Delta w_i^n(k)]$, $n \geq 2$, we obtain

$$\Gamma_i^{(6)}(k) = \Gamma_i^{(6)}(k-1) + 6\Gamma_i^{(5)}(k-1)\Delta m_i(k) \tag{A.18}$$
$$\Gamma_i^{(5)}(k) = \Gamma_i^{(5)}(k-1) + 5\Gamma_i^{(4)}(k-1)\Delta m_i(k) \tag{A.19}$$
$$\Gamma_i^{(4)}(k) = \Gamma_i^{(4)}(k-1) + 4\Gamma_i^{(3)}(k-1)\Delta m_i(k) \tag{A.20}$$
$$\Gamma_i^{(3)}(k) = \Gamma_i^{(3)}(k-1) + 3\Gamma_i^{(2)}(k-1)\Delta m_i(k). \tag{A.21}$$

C. L-J_4 Algorithm

The MSE performance of the L-J_4 algorithm can be closely approximated by that of the J_4-LMS algorithm when the latter is presented with an uncorrelated and stationary input sequence; see Fig. 6. For this case, $\mathbf{P}$ becomes an identity matrix and $\mathbf{D}$ a diagonal matrix with a single eigenvalue equal to the equalizer input signal power. $\mathbf{M}(k)$ and $\mathbf{\Gamma}^{(2)}(n)$ are still updated by (A.9) and (A.14) but (A.10)–(A.13) and (A.15)–(A.16) are to be replaced by

$$E[x(k)\mathbf{Z}(k)] = \sigma_p^2\mathbf{\Gamma}^{(2)}(k) \tag{A.22}$$

$$E[x^2(k)] = \sigma_p^2\sum_i\Gamma_i^{(2)}(k) \tag{A.23}$$

$$E[x^4(k)] = \sigma_p^4\sum_i\Gamma_i^{(4)}(k) \tag{A.24}$$

$$E[x^3(k)\mathbf{Z}(k)] = \sigma_p^4\mathbf{\Gamma}^{(3)}(k) \tag{A.25}$$

$$E[x^4(k)Z_i^2(k)] = \sigma_p^6\Gamma_i^{(4)}(k) \tag{A.26}$$

$$E[x^6(k)Z_i^2(k)] = \sigma_p^8\Gamma_i^{(6)}(k) \tag{A.27}$$

where $\sigma_p^n = E[|y(k)|^n]$.

REFERENCES

[1] Y. Sato, "A method of self-recovering equalization for multilevel amplitude-modulation system," *IEEE Trans. Commun.*, vol. COM-23, pp. 679–682, June 1976.

[2] A. Benveniste and M. Goursat, "Blind equalization," *IEEE Trans. Commun.*, vol. COM-32, pp. 871–883, Aug. 1984.

[3] D. N. Godard, "Self-recovering equalization and carrier tracking in two dimensional data communication system," *IEEE Trans. Commun.*, vol. COM-28, pp. 1867–1875, Nov. 1980.

[4] G. B. Giannakis, "Cumulants: A powerful tool in signal processing," *Proc. IEEE*, vol. 75, pp. 1333–1334, Sept. 1987.

[5] G. B. Giannakis and J. M. Mendel, "Identification of nonminimum phase systems using higher order statistics," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 360–377, Mar. 1989.

[6] A. Swami and J. M. Mendel, "Closed-form recursive estimation of MA coefficients using autocorrelations and third order cumulants," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1794–1795, Nov. 1989.

[7] K. S. Tugnait, "Identification of linear stochastic systems via second- and fourth-order cumulant matching," *IEEE Trans. Inform. Theory*, pp. 393–407, May 1987.

[8] D. Hatzinakos and C. L. Nikias, "Blind equalization using a tricepstrum-based algorithm," *IEEE Trans. Commun.*, vol. 39, pp. 669–682, May 1991.

[9] F.-C. Zheng, S. Mclaughlin, and B. Mulgrew, "Blind equalization of nonminimum phase channel: Higher order cumulant based algorithm," *IEEE Trans. Signal Processing*, vol. 41, pp. 681–691, Feb. 1993.

[10] S. A. Alshebeili, A. N. Venetsanopoulos, and A. Enis Çetin, "Cumulant based identification approaches for nonminimum phase FIR systems," *IEEE Trans. Signal Processing,* vol. 41, pp. 1576–1588, Apr. 1993.
[11] B. Porat and B. Friedlander, "Blind equalization of digital communication channels using high-order moments," *IEEE Trans. Signal Processing,* vol. 39, pp. 522–526, Feb. 1991.
[12] O. Shalvi and E. Weinstein, "New criteria for blind deconvolution of nonminimum phase systems (channel)," *IEEE Trans. Inform. Theory,* pp. 312–321, Mar. 1990.
[13] ——, "Super-exponential methods for blind deconvolution," *IEEE Trans. Inform. Theory,* pp. 504–519, Mar. 1993.
[14] A. Benveniste, M. Goursat, and G. Ruget, "Analysis of stochastic approximation schemes with discontinuous and dependent forcing terms with applications to data communication algorithms," *IEEE Trans. Automat. Contr.,* pp. 1042–1058, Dec. 1980.
[15] R. D. Gitlin and F. R. Magee, "Self-orthogonalizing adaptive equalization algorithms," *IEEE Trans. Commun.,* vol. COM-25, pp. 666–672, July 1977.
[16] A. Papoulis, *Probability, Random Variables, and Stochastic Processes,*3rd ed. New York: McGraw-Hill, 1991.
[17] B. Widrow and S. D. Stearns, *Adaptive Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1985.
[18] S. Haykin, *Adaptive Filter Theory.* Englewood Cliffs, NJ: Prentice-Hall, 1991.
[19] V. Weerackody, S. A. Kassam, and K. R. Laker, "Convergence analysis of an algorithm for blind equalization," *IEEE Trans. Commun.,* vol. 39, pp. 856–865, June, 1991.
[20] J. M. Cioffi and T. Kailath, "Fast, recursive-least-squares transversal filters for adaptive filtering," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-32, pp. 304–337, 1984.

Fang-Biau Ueng was born in Chia-Yi, Taiwan, in 1967. He received the B.S.E.E. degree from the National Tsing Hua University, Hsinchu, Taiwan, in 1990, and the M.S. degree in control engineering from the National Chiao Tung University, Hsinchu, Taiwan, in 1992. He has been pursuing the doctoral degree since September 1992 at the Institute of Electronics, National Chiao Tung University.

Yu T. Su received the B.S. degree from Tatung Institute to Technology, Taipei, Taiwan, in 1974, and the M.S. and Ph.D. degrees from the University of Southern California, Los Angeles, in 1983, all in electrical engineering.

From May 1983 to September 1989 he was with LinCom Corporation, Los Angeles, CA, where he was involved in satellite communication system design. He is currently a faculty member in the Microelectronic and Information Systems Research Center and the Department of Communication Engineering of the National Chiao Tung University, Hsinchu, Taiwan, R.O.C. His main research interests are communication systems and statistical signal processing.