Adaptive Lattice Decision-Feedback Equalizers—Their Performance and Application to Time-Variant Multipath Channels

FUYUN LING, MEMBER, IEEE, AND JOHN G. PROAKIS, FELLOW, IEEE

Abstract—**This paper presents two types of adaptive lattice decision-feedback equalizers (DFE), the least squares (LS) lattice DFE and the gradient lattice DFE. Their performance has been investigated on both time-invariant and time-variant channels through computer simulations and compared to other kinds of equalizers. An analysis of the self-noise and tracking characteristics of the LS DFE and the DFE employing the Widrow-Hoff least mean square adaptive algorithm (LMS DFE) are also given. The analysis and simulation results show that the LS lattice DFE has the faster initial convergence rate, while the gradient lattice DFE is computationally more efficient. The main advantages of the lattice DFE's are their numerical stability, their computational efficiency, the flexibility to change their length, and their excellent capabilities for tracking rapidly time-variant channels.**

I. INTRODUCTION

IN high speed digital communications, efficient use of available channel bandwidth is often limited by the presence of intersymbol interference caused by the nonideal channel characteristics. Maximum-likelihood sequence estimation (MLSE) is the most effective detection technique (optimum in the sense of minimizing the probability of a sequence error) for digital signals corrupted by intersymbol interference and additive noise [1]. Despite its effectiveness, the computational complexity of the MLSE technique limits its applications. In practice, linear equalizers[1] and decision-feedback equalizers (DFE) are more often used [2]-[4].

The linear equalizer is widely used for equalization of telephone channels. Usually, a transversal (tapped-delay-line) filter structure is employed, with tap weight coefficients that are adjusted adaptively using the gradient-type LMS (least mean square) algorithm due to Widrow and Hoff [5], [6]. On channels that have spectral nulls, the linear equalizer yields very poor error rate performance. Since this kind of channel characteristic is often encountered on time-variant channels, the linear equalizer is inadequate for equalization of such channels [4].

The DFE has a similar computational complexity as the linear equalizer, but it has a better error rate performance, especially on channels having spectral nulls. As a result, the DFE is appropriate for equalization of time-variant multipath channels.

The simplest adaptive DFE consists of two transversal filters, a feedforward filter and a feedback filter, and uses the Widrow-Hoff LMS algorithm [5] to adapt its coefficients. This LMS DFE is computationally efficient and gives good performance in slowly time-variant channels; however, its slow convergence rate limits its performance on rapidly fading multipath channels such as HF, as shown in the paper by Hsu *et al.* [7]. It has also been shown in the same paper that a Kalman DFE, which uses the Kalman algorithm [8] to adjust its coefficients, can be used to track rapidly time-variant multipath channels.

To reduce the computational burden of the Kalman algorithm, which requires a number of operations for each iteration proportional to N^2, where N is the length of the equalizer, more efficient algorithms, including the least squares (LS) and gradient lattice algorithms [9], [10] and the fast Kalman algorithm [11], have been developed. The linear lattice equalizers described in [9] and [10] have many advantages, including insensitivity to roundoff noise and the flexibility to increase or decrease the number of stages.

This paper presents two types of adaptive lattice DFE algorithms, the least squares lattice DFE and the gradient lattice DFE. The former has been derived in our previous work [12], [13], while the latter is briefly described in [13]. The performance of these algorithms is investigated on both time-invariant and time-variant channel models by means of computer simulations. An analysis of the self-noise and tracking characteristics of the DFE algorithms is also given.

The lattice DFE algorithms differ from conventional multichannel least squares lattice algorithms [14], which are restricted to have the same number of stages in each channel, i.e., the same number of feedforward and feedback stages. In some applications the restriction is highly undesirable and many cause numerical instability. The lattice DFE algorithms presented in this paper allow for a different number of feedforward and feedback stages.

II. LATTICE DFE ALGORITHMS

In this section, we present the least squares lattice DFE and the gradient lattice DFE algorithms. First, however, we establish some notation and introduce an equivalent discrete-time channel model.

The following notation is used. A boldface character represents a matrix or a vector. A prime denotes the transpose of a matrix or a vector. A star "*" denotes the complex conjugate of a scalar or the complex conjugate and transpose of a matrix or a vector.

A. Discrete-Time Channel Model

A simplified discrete-time equivalent model is adopted for digital transmission over a time-dispersive channel. The transmitted information sequence $\{x(n)\}$, $n = 0, 1, \cdots, t, \cdots$, is passed through a channel which introduces intersymbol inter-

Paper approved by the Editor for Data Communication Systems of the IEEE Communications Society for publication after presentation at the International Conference on Communications, Boston, MA, June 1983. Manuscript received August 19, 1983; revised August 27, 1984. This work was supported by the National Science Foundation under Grant ECS-8205596.

F. Ling was with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115. He is now with Codex Corporation, Mansfield, MA 02048.

J. G. Proakis is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115.

[1] In general, there are two types of linear equalizers: the nonrecursive and recusive linear equalizer. Since the nonrecursive linear equalizer is primarily used in practice, when we mention the "linear equalizer" in this paper, we mean the nonrecursive linear equalizer. The same holds for the lattice linear equalizer.

ference and additive noise. The model for the channel to be used in the investigation has a tapped delay line (TDL) filter structure with tap spacing equal to the symbol interval. The transfer function of the TDL filter can be written as

$$H(z) = \sum_{i=0}^{M-1} a_i z^{-i}. \tag{2.1}$$

For modeling a time-invariant channel, the a_i's are constant. For modeling a time-variant channel, such as HF, these coefficients are functions of time and are written as $a_i(t)$. We model the $a_i(t)$ as narrow-band Gaussian random processes with zero mean. Each sample function of the random process is generated on a digital computer by passing white Gaussian noise through a low-pass filter of a specified bandwidth.

B. The Least Squares (LS) Criterion

To retrieve $x(t)$, the decision-feedback equalizer (DFE) uses the linear combination of the received signal, $y(t)$, its delays, $y(t-i)$, and the previously detected samples of $x(t)$, denoted as $\tilde{x}(t)$. The estimate of $x(t)$, denoted as $\hat{x}_{DFE}(t)$, is expressed as

$$\hat{x}_{DFE}(t) = \sum_{i=1}^{N_1} c_i(t) y(t-i+1) + \sum_{i=1}^{N_2} c_{i+N_1}(t)\tilde{x}(t-i) = C'(t) Y_{DFE}(t) \tag{2.2}$$

where the N-dimensional coefficient vector for the DFE at time t is

$$C(t) = [c_1(t), c_1(t), \cdots, c_{N_1+N_2}(t)]', \tag{2.3}$$

the data vector is

$$Y_{DFE}(t) = [y(t), y(t-1), \cdots, y(t-N_1+1), \tilde{x}(t-1), \cdots, \tilde{x}(t-N_2)]' \tag{2.4}$$

and $N = N_1 + N_2$.

The coefficient vector $C(t)$ is selected to minimize the time average error $\epsilon_{LS}(t)$, defined as

$$\epsilon_{LS}(t) = \sum_{n=0}^{t} w^{t-n} |x(n) - \hat{x}_{LS}(n)|^2 = \sum_{n=0}^{t} w^{t-n} \cdot |x(n) - C_{LS}^*(t) Y_{DFE}(n)|^2 \tag{2.5}$$

where w is the exponential weighting factor, which is less than but close to 1. It is easy to show that $C_{LS}(t)$ satisfies the equation

$$C_{LS}(t) = R_{LS}^{-1}(t) Q_{LS}(t) \tag{2.6}$$

where

$$R_{LS}(t) = \sum_{n=0}^{t} w^{t-n} Y_{DFE}(n) Y_{DFE}^*(n) \tag{2.7}$$

and

$$Q_{LS}(t) = \sum_{n=0}^{t} w^{t-n} Y_{DFE}(n) x^*(n). \tag{2.8}$$

Equation (2.5) can be solved recursively in time. One algorithm that is applicable is the conventional Kalman algorithm [8]. Another is the square-root Kalman algorithm [15]. A third is the computationally efficient fast Kalman algorithm [11].

Although these algorithms have optimal convergence properties and good tracking ability, they have some drawbacks. The Kalman algorithm is computationally complex. It requries about $2.5N^2 + 4.5N$ operations per iteration. It is also known to be sensitive to roundoff noise. The square-root Kalman algorithm using an L-D-U decomposition [15] exhibits less sensitivity to roundoff noise, but its computational complexity also increases with N^2. Specifically, it requires $1.5N^2 + 6.5N$ operations per iteration.

The fast Kalman algorithm is computationally very efficient. It requires about $20N$ operations per iteration. However, the main problem with this algorithm is its sensitivity to roundoff noise. Our simulation results, as well as those of others, show that after several thousand iterations the coefficients determined by the fast Kalman algorithm diverge from their optimum values due to the buildup of roundoff noise, even when floating point arithmetic is used in computer simulations. Some newly derived algorithms reduce the computational complexity further [16], [17], but the problem of the sensitivity to roundoff noise still exists.

An alternative to Kalman-type algorithms is to employ a lattice filter structure to solve the least squares problem of (2.5). Lattice equalizers are known to exhibit less sensitivity to roundoff noise and also have a computational complexity proportional to N. Another advantage of the lattice structure is the orthogonality property, which makes it possible to increase or decrease the number of lattice stages without affecting the parameters of the previous stages.

C. The Least Squares and Gradient Lattice DFE's

Since the input signals of a DFE are from two scalar sources, the received signal $y(t)$ and the detected signal $\tilde{x}(t)$, the LS lattice DFE can be realized by using a two-channel version of the lattice algorithms. The LS lattice DFE algorithm was first derived in [18] by M. Shensa. Although his algorithm is correct if the DFE has the same number of feedforward and feedback stages, it fails to give the least squares solution when the numbers are not equal. In general, however, it is desirable to have a DFE with an unequal number of feedforward and feedback stages. We have recently derived generalized LS and gradient multichannel adaptive lattice algorithms [13], [19], which have the property that the number of stages in each channel may be different. Lattice DFE's using a modified two-channel version of these algorithms have been simulated and their performance has been evaluated for both time-invariant and time-variant channels. These DFE's are presented below.

The data vector $Y_{DFE}(t)$ defined in (2.4) is an $(N_1 + N_2)$-dimensional vector. Similar to the case of the linear equalizer, a certain shifting property exists for $Y_{DFE}(t)$. Without loss of generality we assume that $N_1 \geqslant N_2$. Also, we define a set of data vectors as follows: for $m = 1, \cdots, N_1$, we let

$$Y_m(t) = [y(t), \cdots, y(t-m+1)]', \qquad (1 \leqslant m \leqslant N_1 - N_2) \tag{2.9}$$

$$Y_m(t) = [y(t), \cdots, y(t-m+1), \tilde{x}(t-1), \cdots, \tilde{x}(t-m+N_1-N_2)]', \qquad (N_1 - N_2 < m < N_1). \tag{2.10}$$

$Y_m(t)$ is the same as the data vector of the linear equalizer for $m \leqslant N_1 - N_2$. For $m > N_1 - N_2$, we can show that

$$Y_{m+1}(t) = T_{m+1}[Y_m'(t), y(t-m), \tilde{x}(t-m+N_1-N_2+1)]' = S_{m+1}[y(t), \tilde{x}(t-1), Y_m'(t-1)]' \tag{2.11}$$

where T_{m+1} and S_{m+1} are permutation matrices which have been used in [11] for deriving the fast Kalman algorithm. Furthermore, we define a set of least squares estimation problems as follows. Let

$$e_m(t) = x(t) - \hat{x}_m(t) = x(t) - C_m'(t)Y_m(t) \quad (1 \leq m \leq N_1) \tag{2.12}$$

where $C_m(t)$ is the coefficient vector that minimizes the sum of the weighted squared errors $e_m(t)$ for each m. By specializing the generalized form of the LS multichannel algorithm given in [13], [19], we can solve for $\hat{x}_m(t)$ recursively in order and in time for $m = 1, \cdots, N_1$. For $m = N_1$ we obtain $\hat{x}_{N_1}(t) = \hat{x}_{LS}(t)$ required by the LS DFE described in (2.5)-(2.8).

The least squares lattice DFE algorithm is given below. The assumption is made that $N_1 \geq N_2$. If $N_2 > N_1$, the generalized LS lattice algorithm given in [13], [19] can be used directly. During equalization, $x(t)$ is not known at the receiver. In that case, the DFE algorithm takes the so-called *a priori* error form, namely, $C_m(t-1)$ is used to estimate $\hat{x}_m(t)$, as proposed in [9], [13]. The derivation of the algorithm is lengthy and will be omitted. The interested reader may refer to [13] and [19] for the derivation of the generalized multichannel lattice algorithm.

Initialization

$$b_0(t) = f_0(t) = y(t), \quad k_M{}^b(0) = 0 \quad (M = N_1 - N_2) \tag{2.13}$$

$$r_0{}^f(t) = r_0{}^b(t) = wr_0{}^f(t-1) + |y(t)|^2, \quad e_0(t) = \tilde{x}(t), \quad \hat{x}_0(t) = 0 \tag{2.14}$$

$$\alpha_m(t) = 1, \quad k_m{}^x(0) = 0, \quad r_m{}^f(0) = r_m{}^b(0) = \delta \quad (m = 1, \cdots, N_1 - N_2 - 1) \tag{2.15}$$

$$k_m(0) = k_m{}^x(0) = 0, \quad \boldsymbol{k}_m(0) = \boldsymbol{0}, \quad \boldsymbol{k}_m{}^x(0) = \boldsymbol{0} \tag{2.16}$$

$$\boldsymbol{r}_m{}^f(0) = \boldsymbol{r}_m{}^b(0) = \delta I \quad (m = N_1 - N_2, \cdots, N_1). \tag{2.17}$$

Scalar Lattice Stages ($0 < m < N_1 - N_2$ unless otherwise specified)

$$f_m(t) = f_{m-1}(t) - k_m{}^*(t-1)b_{m-1}(t-1)/r_{m-1}{}^b(t-2) \tag{2.18}$$

$$b_m(t) = b_{m-1}(t-1) - k_m(t-1)f_{m-1}(t)/r_{m-1}{}^f(t-1) \tag{2.19}$$

$$k_m(t) = wk_m(t-1) + \alpha_{m-1}(t-1)f_{m-1}{}^*(t)b_{m-1}(t-1) \tag{2.20}$$

$$r_m{}^f(t) = r_{m-1}{}^f(t) - |k_m(t)|^2/r_{m-1}{}^b(t-1) \quad (m \leq N_1 - N_2) \tag{2.21}$$

$$r_m{}^b(t) = r_{m-1}{}^b(t-1) - |k_m(t)|^2/r_{m-1}{}^f(t) \quad (m \leq N_1 - N_2) \tag{2.22}$$

$$\hat{x}_m(t) = \hat{x}_{m-1}(t) + k_m{}^{x*}(t-1)b_{m-1}(t)/r_{m-1}{}^b(t-1) \tag{2.23}$$

$$\alpha_m(t) = \alpha_{m-1}(t) - |b_{m-1}(t)\alpha_{m-1}(t)|^2/r_{m-1}{}^b(t) \tag{2.24}$$

$$e_m(t) = \tilde{x}(t) - \hat{x}_m(t) \quad (0 < m \leq N_1 - N_2) \tag{2.25}$$

$$k_m{}^x(t) = wk_m{}^x(t-1) + \alpha_{m-1}(t)e_{m-1}{}^*(t)b_{m-1}(t). \tag{2.26}$$

Transitional Lattice Stage ($M = N_1 - N_2$)

$$\hat{b}_M(t) = e_{M-1}(t-1) - k_M{}^b(t-1)f_{M-1}(t)/r_{M-1}{}^f(t-1) \tag{2.27}$$

$$k_M{}^b(t) = wk_M{}^b(t-1) + \alpha_{M-1}(t-1)f_{M-1}{}^*(t-1)e_{M-1}(t-1) \tag{2.28}$$

$$\boldsymbol{f}_M(t) = [f_M(t), e_M(t-1)]' \tag{2.29}$$

$$\boldsymbol{b}_M(t) = [b_M(t), \hat{b}_M(t)]' \tag{2.30}$$

$$\boldsymbol{r}_M{}^f(t) = w\boldsymbol{r}_M{}^f(t-1) + \alpha_M(t-1)\boldsymbol{f}_M(t)\boldsymbol{f}_M{}^*(t) \tag{2.31}$$

$$\boldsymbol{r}_M{}^b(t) = w\boldsymbol{r}_M{}^b(t-1) + \alpha_M(t)\boldsymbol{b}_M(t)\boldsymbol{b}_M{}^*(t). \tag{2.32}$$

Two-Dimensional Lattice Stages ($N_1 - N_2 < m < N_1$ unless otherwise specified)

$$\boldsymbol{f}_m(t) = \boldsymbol{f}_{m-1}(t) - \boldsymbol{k}_m{}^*(t-1)\boldsymbol{r}_{m-1}{}^{-b}(t-2)\boldsymbol{b}_{m-1}(t-1) \tag{2.33}$$

$$\boldsymbol{b}_m(t) = \boldsymbol{b}_{m-1}(t-1) - \boldsymbol{k}_m(t-1)\boldsymbol{r}_{m-1}{}^{-f}(t-1)\boldsymbol{f}_{m-1}(t) \tag{2.34}$$

$$\boldsymbol{k}_m(t) = w\boldsymbol{k}_m(t-1) + \alpha_{m-1}(t-1)\boldsymbol{b}_{m-1}(t-1)\boldsymbol{f}_{m-1}{}^*(t) \tag{2.35}$$

$$\boldsymbol{r}_m{}^f(t) = w\boldsymbol{r}_m{}^f(t-1) + \alpha_m(t-1)\boldsymbol{f}_m(t)\boldsymbol{f}_m{}^*(t) \tag{2.36}$$

$$\boldsymbol{r}_m{}^b(t) = w\boldsymbol{r}_m{}^b(t-1) + \alpha_m(t)\boldsymbol{b}_m(t)\boldsymbol{b}_m{}^*(t) \tag{2.37}$$

$$\alpha_m(t) = \alpha_{m-1}(t) + \alpha_{m-1}{}^2(t)\boldsymbol{b}_{m-1}{}^*(t)\boldsymbol{r}_{m-1}{}^{-b}(t) \cdot \boldsymbol{b}_{m-1}(t) \tag{2.38}$$

$$\hat{x}_m(t) = \hat{x}_{m-1}(t) + \boldsymbol{k}_m{}^{x*}(t-1)\boldsymbol{r}_{m-1}{}^{-b}(t-1)\boldsymbol{b}_{m-1}(t) \quad (m < N_1) \tag{2.39}$$

$$e_m(t) = \tilde{x}(t) - \hat{x}_m(t) \tag{2.40}$$

$$\boldsymbol{k}_m{}^x(t) = w\boldsymbol{k}_m{}^x(t-1) + \alpha_{m-1}(t)\boldsymbol{b}_{m-1}(t)e_{m-1}{}^*(t) \quad (m < N_1). \tag{2.41}$$

In the above algorithm, all the quantities take complex values. Hence, this algorithm is suitable when M-ary PSK or QAM signals are used. $\boldsymbol{f}_m(t)$, $\boldsymbol{b}_m(t)$, and $\boldsymbol{k}^x(t)$ are 2×1 vectors, and $\boldsymbol{r}_m{}^f(t)$, $\boldsymbol{r}_m{}^b(t)$, and $\boldsymbol{k}_m(t)$ are 2×2 matrices. All other quantities are scalars. We also denote the inverses of $\boldsymbol{r}_m{}^f(t)$ and $\boldsymbol{r}_m{}^b(t)$ by $\boldsymbol{r}_m{}^{-f}(t)$ and $\boldsymbol{r}_m{}^{-b}(t)$, respectively. In (2.15) and (2.17) δ is a small positive number.

The structure of the LS DFE is depicted in Fig. 1. It consists of a multichannel lattice predictor part and a joint estimator part. The multichannel lattice predictor has $M = N_1 - N_2$ single-channel lattice stages followed by $N_2 - 1$ two-channel lattice stages. The joint estimator part is similar to the conventional joint process lattice algorithms. The heavy black lines indicate that the quantities $\boldsymbol{f}_m(t)$ and $\boldsymbol{b}_m(t)$ are 2×1 vectors. The lattice DFE has two inputs. One input signal is the received signal $y(t)$. The second is the detected symbol $\tilde{x}(t)$. During training periods the true transmitted symbol $x(t)$ is used instead of $\tilde{x}(t)$.

Although this algorithm does not allow one to change the number of feedforward and feedback stages individually after it is started, a pair of stages can be added or dropped during operation, while keeping the difference between the number of feedforward and feedback stages unchanged. To determine the optimal length of the DFE, the average output error is monitored. If stage m has a sufficiently small average error, $\hat{x}_m(t)$ is used as the estimate of $x(t)$ and is compared with the threshold to produce $\tilde{x}(t)$.

The gradient lattice DFE can be obtained from the LS lattice DFE by performing some minor modification to (2.13)-(2.41). If we set all $\alpha_m(t)$ equal to 1, the LS lattice DFE will degenerate to the gradient lattice DFE. Since $\alpha_m(t)$ is always equal to unity, (2.24) and (2.38) can be omitted. The remaining equations remain unchanged.

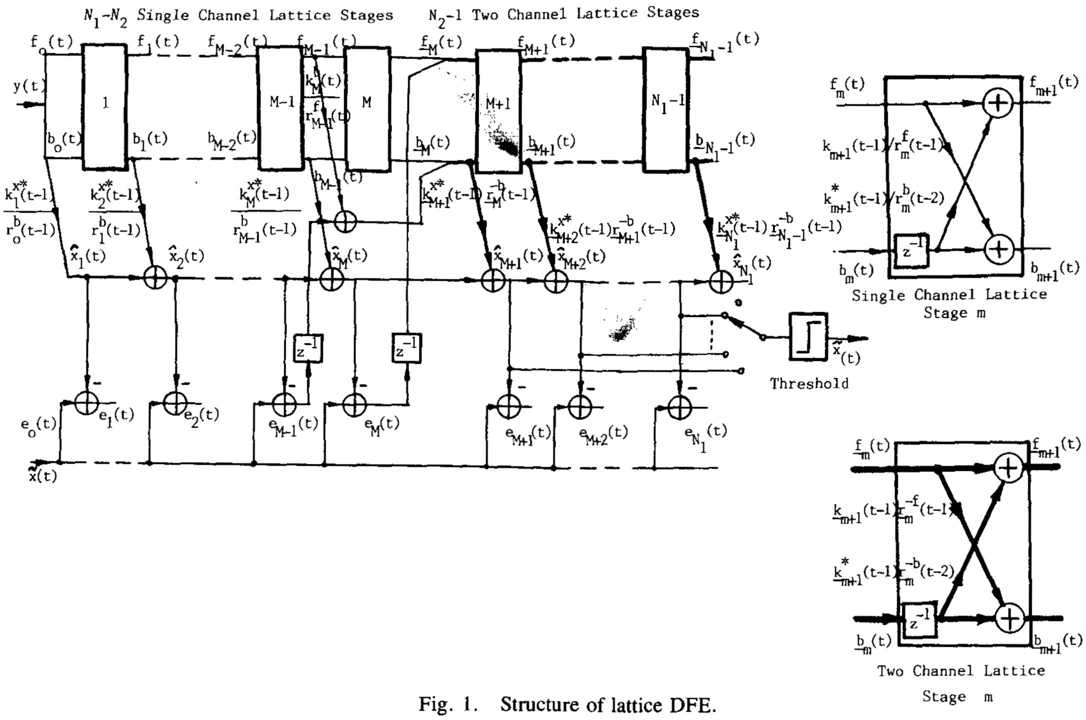

Fig. 1. Structure of lattice DFE.

TABLE I
COMPUTATIONAL COMPLEXITY OF ADAPTIVE DFE ALGORITHMS

Algorithm	Total Number of Operations	Number of Divisions
Gradient transversal DFE	$2N + 1$	0
Fast Kalman DFE	$20N + 5$	3
Kalman DFE	$2.5N^2 + 4.5N$	2
Square-root Kalman	$1.5N^2 + 6.5N$	N
Gradient lattice DFE	$13N_1 + 33N_2 - 36$	$2N_1$
LS lattice DFE	$18N_1 + 39N_2 - 39$	$2N_1$

D. Comparison of Computational Complexity

Table I compares the computational burden of the LS, the gradient lattice, the fast Kalman, the square-root Kalman, and the transversal gradient DFE's. In the table, $N = N_1 + N_2$ is the total number of parameters of a DFE, where N_1 and N_2 are the number of feedforward and feedback stages of a DFE, respectively.

To facilitate the comparisons, the number of operations is plotted in Fig. 2 as a function of N. In the lattices we assume that $N_1 = N_2 = 0.5N$. We observe that the LS lattice is more efficient that the Kalman DFE or the square-root Kalman, when N is greater than 12. We also observe that the computational burden of the gradient lattice DFE is close to the fast Kalman DFE. The numbers of divisions required by the lattice DFE's and the square-root Kalman DFE are also similar.

III. An Analysis of Self-Noise and Tracking Characteristics of the DFE's Using the LMS and LS Algorithms

It is well known that all adaptive filters capable of adapting at real-time rates experience a loss in performance because their adjustments are based on time averages taken with limited sample sizes [6]. The loss in performance is due mainly to two sources of error. First, the adaptive algorithm yields a misadjustment of the estimated coefficients relative to their optimum values. This is basically an estimation error resulting from the additive observation noise corrupting the signal.

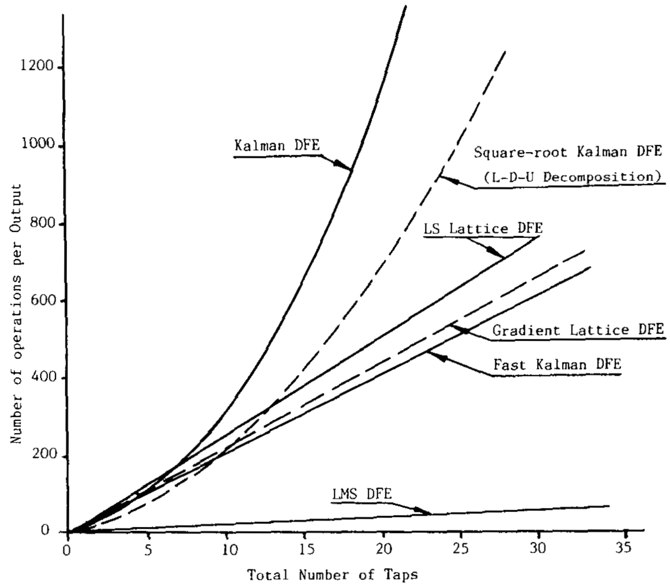

Fig. 2. Computational complexity of adaptive DFE's.

Second, in time-variant channels there is always a "lag" between the estimated values and the optimum values of the DFE's coefficients. Both sources of error add to the minimum mean square error.

The extra error due to the coefficient misadjustment is also called self-noise. An adaptive algorithm capable of tracking a rapidly changing environment will yield a smaller lag, i.e., better tracking performance, but larger self-noise. In this section we compare these two factors for the LMS and LS adaptive DFE's.[2] The effect of the channel characteristics on the tracking performance of the DFE's is also considered. The results explain the poor tracking characteristics of the LMS DFE on time-variant channels.

[2] Different LS DFE algorithms differ from each other in computational complexity and numerical stability. Since their tracking characteristics are similar, we refer to all of them as LS algorithms in this section.

A. The Self-Noise of the LMS and LS DFE's

Detailed analysis of the LMS algorithm has been given in many papers, including [2] and [6]. Here we summarize the results for the LMS DFE.

For an adaptive transversal DFE using the LMS algorithm having a step size Δ, the average self-noise, ϵ_{self}, can be expressed as

$$\epsilon_{\text{self}} = \frac{\Delta}{2} \operatorname{Tr} [\boldsymbol{R}_{\text{DFE}}] \epsilon_{\text{opt}} \tag{3.1}$$

where ϵ_{opt} is the minimum mean square error, $\boldsymbol{R}_{\text{DFE}} = E[\boldsymbol{Y}_{\text{DFE}} \boldsymbol{Y}_{\text{DFE}}^*]$, where $E[\ ]$ denotes the ensemble average operator, is the autocorrelation matrix of $\boldsymbol{Y}_{\text{DFE}}$, and $\operatorname{Tr} [\boldsymbol{R}_{\text{DFE}}]$ is the trace of $\boldsymbol{R}_{\text{DFE}}$. It is easy to show that

$$\operatorname{Tr} [\boldsymbol{R}_{\text{DFE}}] = \sum_{p=1}^{N} \lambda_p = N \,(\text{Average of } \lambda_p) \tag{3.2}$$

where λ_p's are the eigenvalues of $\boldsymbol{R}_{\text{DFE}}$.

If we want to maintain the self-noise equal to a certain percentage of ϵ_{opt}, e.g., $\epsilon_{\text{self}} = h \cdot \epsilon_{\text{opt}}$, it is necessary that

$$\Delta = 2h \Big/ \left(\sum_{p=1}^{N} \lambda_p \right) = 2h/(N_1 E[y^2(n)] + N_2 E[x^2(n)]) \tag{3.3}$$

holds. We can make $\Sigma_{p=1}^{N} \lambda_p = N$ by controlling the power of $y(n)$ and $x(n)$. In such a case

$$\epsilon_{\text{self}} = \frac{1}{2} \Delta N \epsilon_{\text{opt}} \tag{3.4}$$

where $N = N_1 + N_2$ is the dimension of $\boldsymbol{R}_{\text{DFE}}$.

For an exponentially weighted LS DFE, a even simpler formula exists for its self-noise. For the LS DFE having a weighting factor w, we have

$$\epsilon_{\text{self LS}} = \epsilon_{\text{opt}} N(1-w)/(1+w). \tag{3.5}$$

If $w \approx 1$, we obtain

$$\epsilon_{\text{self LS}} \approx \frac{N}{2} (1-w)\, \epsilon_{\text{opt}}. \tag{3.5a}$$

The above formula was first given in [20], for LS linear predictors. We have extended it to any shape windowed LS estimator and filter by using a new derivation, which is given in [21]. The formula is verified by the simulation results also given in [21].

From (3.5) we notice that the level of the self-noise for the LS DFE does not depend on the power of the input signal. This makes it easier to choose w than Δ. If (3.4) is satisfied and w is close to unity, for $\Delta = (1 - w)$ we will have the same self-noise for both the LMS and LS algorithms.

B. The Tracking Characteristics of the LMS and LS Algorithms

When a DFE is operating on a time-variant channel, it is attempting to adapt to a nonstationary environment. Since both the LMS algorithm and the exponentially weighted LS algorithm are geometrically converging processes, it is possible to place exponential envelopes on the adaptive processes. The time constants of the exponential envelopes describe the tracking characteristics of the algorithm.

For the LMS algorithm, as given by Widrow *et al.* in [6], the convergence rate or time constant is different for each coordinate, sometimes called mode, which corresponds to a particular eigenvalue of the autocorrelation matrix of the input vector.[3] For the eigenvalue λ_p, the time constant T_p is

$$T_p = \tfrac{1}{2} \lambda_p. \tag{3.6}$$

Since a smaller λ_p corresponds to a larger T_p, the speed at which the LMS algorithm completely converges from one state to a new state is determined by the time constant corresponding to the smallest eigenvalue. By using (3.1) and (3.2) we conclude that if we maintain $\epsilon_{\text{self}} = h\epsilon_{\text{opt}}$, we have

$$\begin{aligned} T_p &= \operatorname{Tr} [\boldsymbol{R}_{\text{DFE}}]/(4h\lambda_p) \\ &= N \,(\text{average of eigenvalues})/(4h\lambda_p). \end{aligned} \tag{3.7}$$

If some eigenvalue of $\boldsymbol{R}_{\text{DFE}}$ is much smaller than the average of the eigenvalues, it would take a long time for the algorithm to converge to its new optimum state. For the LS algorithm, there exists a unique convergence time constant regardless of the eigenvalue distribution. We have shown this property in [21]. The time constant of the LS algorithm can be expressed as

$$T_{\text{LS}} = 1/[2(1-w)]. \tag{3.8}$$

If we use the same criterion to maintain the self-noise level as in the LMS algorithm, we obtain from (3.8) and (3.5a)

$$T_{\text{LS}} \approx N/4h. \tag{3.9}$$

Comparing (3.7) and (3.9), it is easy to see that $\max\{T_p\} = T_{\text{LS}}$ if and only if all λ_p are equal. At this point we can conclude that the tracking ability of the LS DFE is always better than the LMS DFE. The LMS DFE may approach the performance of the LS DFE only if the DFE's autocorrelation matrix has equal eigenvalues, and the power of the input signal is maintained at a fixed level. The first condition is not satisfied when intersymbol interference exists, and the second condition is usually not satisfied for time-variant channels. Hence, the LS DFE is always a better choice for equalization of such channels.

C. The Eigenvalue Distribution of the DFE's Autocorrelation Matrix

The eigenvalue distribution of the autocorrelation matrix of an equalizer's input data vector, which is termed the equalizer's autocorrelation matrix in the sequel, plays an important role in the tracking ability and performance of the equalizer. It is well known that the eigenvalues of the linear equalizer are determined by the channel spectral characteristics. The ratio of the maximum to the minimum eigenvalues is approximately equal to the ratio of the maximum value to the minimum value of the spectral density of $y(n)$. It is also equal to the ratio of the maximum to minimum magnitude squared of the channel frequency response. Since the convergence rate of the LMS algorithm mainly depends on the minimum eigenvalue, if the channel has nulls in its frequency response, the convergence of the LMS linear equalizer will be very slow. Now, we show that this is also true for the LMS DFE.

The autocorrelation matrix of the DFE is

$$\boldsymbol{R}_{\text{DFE}} = E[\boldsymbol{Y}_{\text{DFE}}(t) \boldsymbol{Y}_{\text{DFE}}^*(t)] \tag{3.10}$$

[3] When the channel is time-variant, we assume that time variations are slow compared to the convergence rate of the LMS algorithm. Thus, even when the eigenvalues change (slowly) with time, (3.6) for the time constants still obtains.

as given in Section II. We partition $Y_{DFE}(t)$ into two parts, the feedforward signal part $Y_1(t)$ and the feedback signal part $Y_2(t)$, i.e.,

$$Y_{DFE}(t) = \begin{bmatrix} Y_1(t) \\ Y_2(t) \end{bmatrix}. \tag{3.11}$$

We can rewrite R_{DFE} as

$$R_{DFE} = \begin{bmatrix} E[Y_1(t)Y_1^*(t)] & E[Y_1(t)Y_2^*(t)] \\ E[Y_2(t)Y_1^*(t)] & E[Y_2(t)Y_2^*(t)] \end{bmatrix} = \begin{bmatrix} R_1 & U \\ U^* & I \end{bmatrix} \tag{3.12}$$

where I is the identity matrix.

From the definition of the vector $Y_1(t)$ we can view R_1 as an autocorrelation matrix of some linear equalizer that has dimension $N_1 < N$. If $N_1 \gg 1$, R_1 will have the property of the linear equalizer mentioned above.

Assume that λ_{min}, μ_{min} and λ_{max}, μ_{max} are the minimum eigenvalues and the maximum eigenvalues of R_{DFE} and R_1, respectively. It is easy to show, cf. [22], that $\lambda_{min} \leqslant \mu_{min}$ and $\lambda_{max} \geqslant \mu_{max}$. Hence, if the length of the feedforward part is not too short, the eigenvalues of R_{DFE} also depend on the channel frequency response. If the channel has deep nulls in its spectrum, the minimum eigenvalue will be small compared to the average of the eigenvalues. When this situation occurs, it takes a long time for all the modes to converge. This is the case that is often encountered on time-variant channels and causes the poor performance of the LMS DFE.

Since the LS algorithm converges with the same time constant for the whole system regardless of the eigenvalue spread, the LS DFE is adequate for equalization of time-variant channels. With regard to the gradient lattice DFE, if the channel changes relatively slowly, the backward prediction errors at the different stages are uncorrelated. Each lattice stage converges independently with the same time constant. Hence, the eigenvalue spread will not affect the convergence rate of the gradient lattice DFE. Its convergence is close to the convergence of the LS DFE.

IV. Simulation Results

The two adaptive lattice DFE's were simulated on a digital computer. Single precision floating point arithmetic is used unless otherwise specified. The mantissa is represented by 22 bits. Simulation results of the LMS DFE and the lattice linear equalizer are also given for the purpose of comparison. All the DFE's used have nine feedforward stages and two feedback stages .

A. The Simulated Discrete-Time Channel Model

Three discrete-time channel models were used in the simulation. Each channel consists of three taps. The discrete-time channel models are described by the following transfer functions:

$$H_1(z) = 0.3 + 0.9z^{-1} + 0.3z^{-2}$$
$$H_2(z) = 0.408 + 0.816z^{-1} + 0.408z^{-2}$$
$$H_3(z) = a_0(t) + a_1(t)z^{-1} + a_2(t)z^{-2}.$$

The first two channels are time-invariant, and the third channel model represents a fading channel with $\{a_i(t)\}$ varying with time. The time-variant coefficients $\{a_i(t)\}$ are generated on a digital computer by passing white Gaussian noise through a low-pass filter of a specified bandwidth. If we assume that we have a nominal 3 kHz HF channel, the signaling rate is 2400 symbols/s, and the low-pass filter is a two-pole Butterworth filter having a 3 dB bandwidth of 0.5 Hz. The curves of the tap values changing with time are depicted in Fig. 3. The first 100 samples were used as a training sequence for the lattice DFE's (for the LMS DFE the first 1000 samples were used as a training sequence).

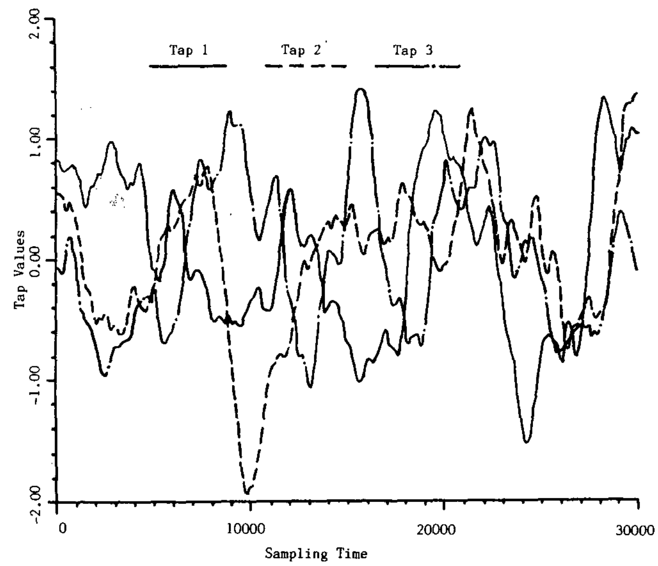

Fig. 3. Tap values of the time-variant channel 3.

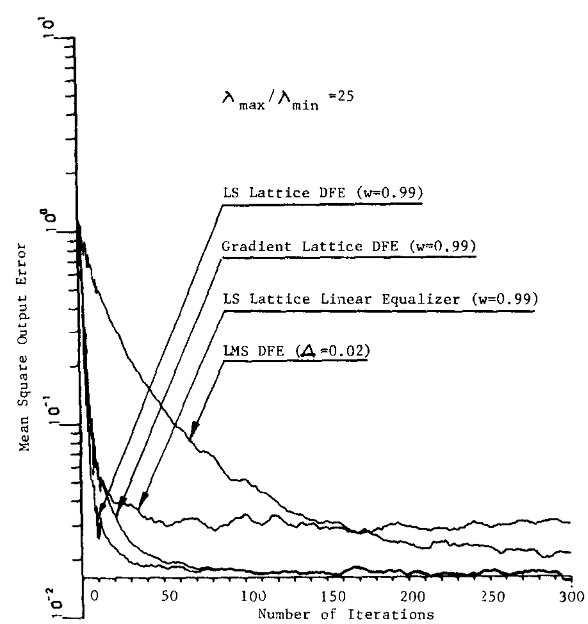

Fig. 4. Convergence rate of equalizers (channel 1).

B. Initial Convergence

The time-invariant channels 1 and 2 were used to evaluate the initial convergence rate of the DFE's. The ratio of the maximum to minimum squared values of the channel frequency response is equal to 25 for channel 1. Since channel 2 has nulls on its spectrum, this ratio may be considered to be infinite.

Figs. 4 and 5 show the simulation results for channels 1 and 2, respectively. The weighting factor w for both lattice algorithms is 0.99. The step size Δ for the gradient transversal equalizer is 0.02. From the plots we observe that the LMS DFE converges slowly on these channels. As expected, the LS lattice DFE has the most rapid convergence rate. It converges in about 30–50 iterations. It is more than ten times

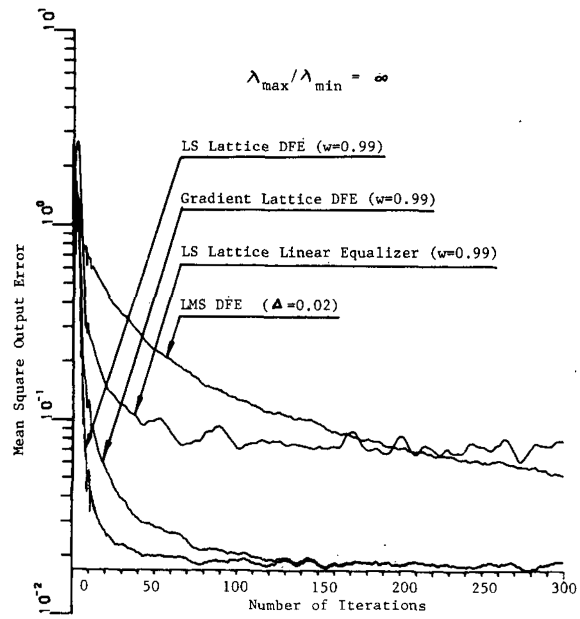

Fig. 5. Convergence rate of equalizers (channel 2).

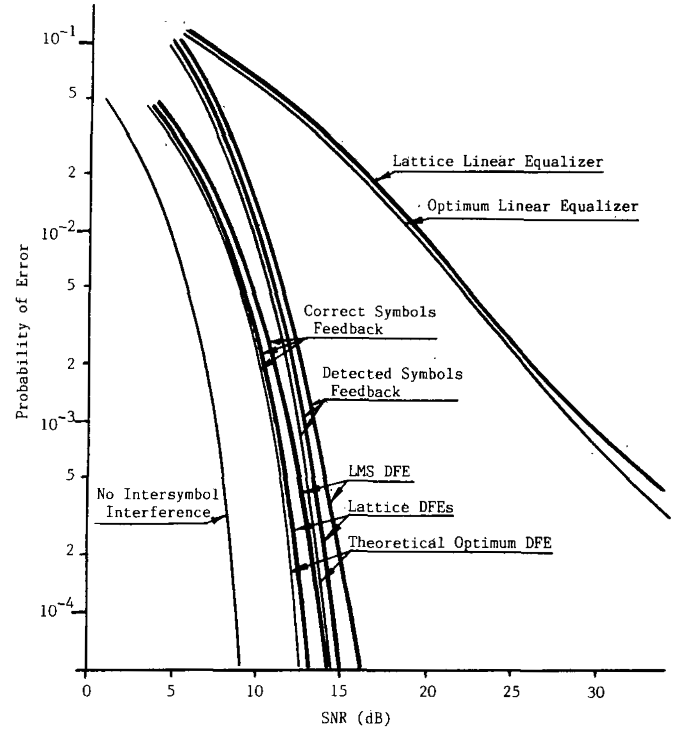

Fig. 6. Error rate performance of equalizers for time-invariant channel 2.

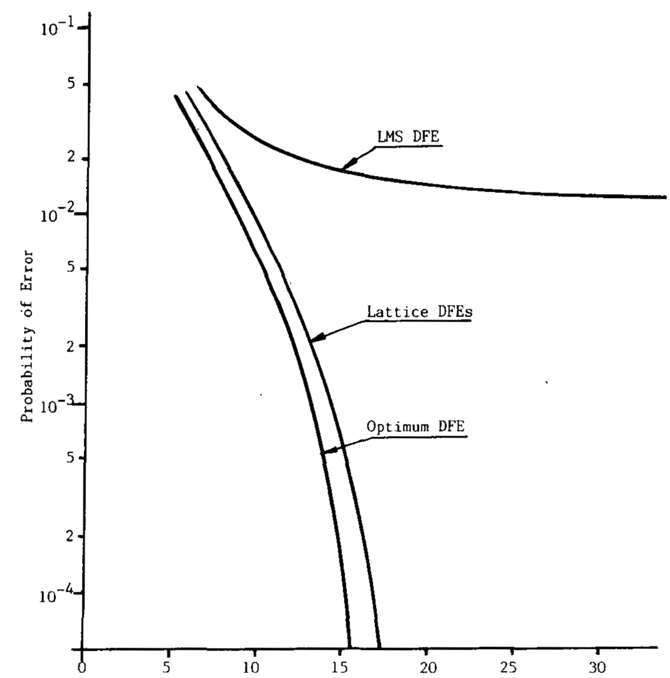

Fig. 7. Error rate performance of DFE's on time-variant channel 3 (correct symbols fed back).

faster than the gradient transversal DFE. The gradient lattice DFE converges in about one-half the rate of the LS DFE, but it is still far superior to the LMS DFE. It is quite interesting to note that these two lattice DFE's yield almost identical output noise after they converge. Since $\Delta > (1 - w)$, the self-noise produced by the LMS DFE is greater than the self-noise of the lattice DFE's. This verifies the analytical results given by (3.4), (3.5), and (3.5a).

Another point that we wish to make is that the initial convergence rate of the LS lattice DFE is almost the same as the Kalman DFE. The simulation results on the convergence of these two DFE's have been given in another paper [12].

C. Error Probability in a Time-Invariant Channel

Fig. 6 shows the error probability versus SNR for channel 2. For comparison we also give the simulation results for the LS linear equalizer. It is apparent from these curves that the DFE gives much better performance than a linear equalizer. The two lattice DFE's have almost identical performance, so we only use one curve to represent both the LS and gradient lattice DFE's. In the same plot we also give the simulation results for the optimum DFE, whose coefficients are set to yield minimum mean square output error, and the LMS DFE. The difference between the lattice and the optimum DFE is less than 0.5 dB. We also observe that the lattice DFE always performs better than the LMS DFE, especially in the high SNR region.

D. Performance on a Fading Channel

The simulated fading channel 3 was used to evaluate the capability of the lattice DFE's to track time-variant dispersive channels. Fig. 7 shows the performance of the DFE's in terms of error probability versus SNR. In the simulation, all the symbols fed back are correct, i.e., we assume that all the symbols are known. As a comparison, the theoretically optimum DFE was also simulated. This is a transversal DFE with coefficients selected according to the exact characteristic of the channel at each time instant, to yield minimum mean squared error. From the simulation we found that a smaller weighting factor w in the lattice DFE's or a larger step size Δ in the LMS DFE yields lower error rates in the high SNR region, but higher error rates in the low SNR region. This means that the tracking ability is better but the self-noise is higher for smaller w. However, when w is less than 0.95 or Δ is greater than 0.05, the performance gets worse for any SNR. The best result is obtained around $w = 0.97$ and $\Delta = 0.03$, for our simulated fading channel.

It is interesting to note that the error rate performances of the LS and the gradient lattice DFE's are almost the same (the difference is within 1 percent). Again we use one curve to represent the performance of both lattice DFE's. We also note that the performance of the lattice DFE's is quite close to the theoretical optimum. In the low SNR region the per-

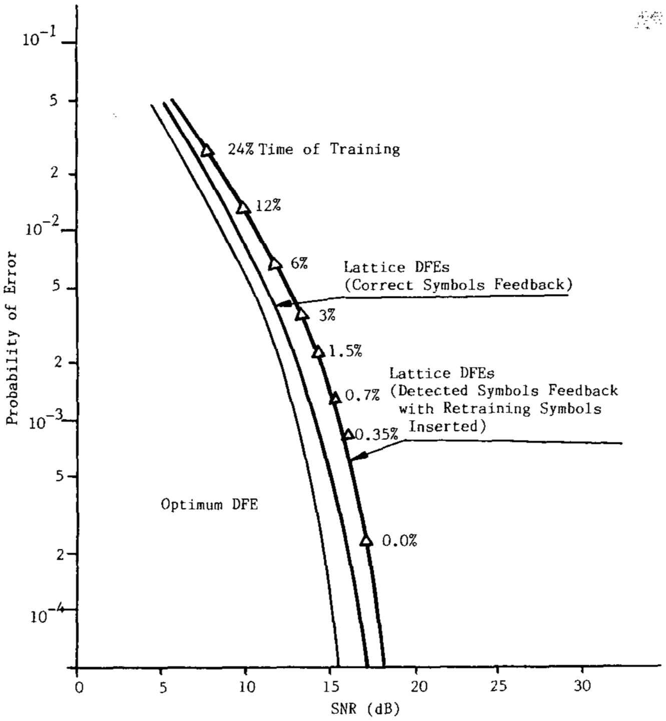

Fig. 8. Error rate performance of lattice DFE's on time-variant channel 3 (detected symbols fed back with training symbols inserted).

formance difference is no more than 1 dB, and about 1.5 dB for SNR's greater than 10 dB. It is observed from the same figure that the LMS DFE gives very poor performance in a fading channel. It saturates at $P_e = 0.01$ when the SNR > 15 dB.

Fig. 8 shows the simulation results of the lattice DFE's using the detected symbols in the feedback section. Because the channel is time-variant, during some time periods the instantaneous SNR may become very small, although the average SNR is still quite high, e.g., when the average SNR is 15 dB, the instantaneous SNR is less than 3 dB at around 5000 and again at about 23 000 samples. In this low SNR region the coefficients of the DFE may diverge from the optimum value. This behavior is caused by the high error rate of the symbols fed back. We used the following method to keep the DFE tracking the channel characteristic in this region. The exponentially weighted output error is monitored throughout the whole equalization process. Once this SNR is below 3 dB, a training sequence is inserted, until the output SNR becomes greater than 3.5 dB. In Fig. 8 we give the error rate versus SNR and the percentage of training time used in the simulation as well. For an input SNR equal to 14 dB, a 3.0 percent training time was needed and the error rate is 0.004. For an SNR greater than 17 dB, no training was required over the time period of the simulation. The above method was proposed originally in [7]. In a real communication system, a feedback link is required to inform the transmitter when to start and to stop inserting the retraining sequence. If it is impossible to have such a feedback link, the lattice DFE's will work by using a periodic training sequence. From simulation we observe that, even after the coefficients of the lattice DFE's have departed from their optimum values due to the high feedback error rate, inserting a training sequence can bring them back to the normal condition again.

E. Sensitivity to Roundoff Noise

From our simulation, the lattice DFE's did not suffer from instability, even when fixed point arithmetic with a word length as short as 8 bits was used. When the word length is longer than 12 bits, very little degradation in performance is observed, compared with floating point arithmetic. When the word length is reduced to fewer than 10 bits, the probability of error increases, due to the effect of roundoff noise. However, the lattice DFE's are still stable, and no coefficient divergence was observed even in a long run of 10^6 samples. The exponential weighting factor used in the simulations was chosen in the range from 0.9 to 0.975. This behavior is better than the square-root Kalman DFE, which requires reinitialization to ensure stable operation when a short word length is used [15].

When a fixed-point implementation is used, the scale factor is also easier to choose for the lattice DFE than for the square-root Kalman DFE. This is due to the fact that the optimum scaling for lattice algorithms does not depend on the signal characteristics, which may change during equalization, but for the square-root Kalman algorithm, it does.

V. Conclusions

This paper has focused on efficient stable lattice algorithms for decision-feedback equalization. We have investigated quantitatively the self-noise behavior and the tracking ability of the DFE's using the LMS and LS adaptive algorithms. Simulation results have been given to show the advantages of the lattice DFE's and to reinforce the analytical results.

We have shown that the DFE employing the LMS algorithm cannot track rapidly time-variant channels well, because its tracking ability is affected by the large eigenvalue spread of its autocorrelation matrix due to the nulls in the channel frequency response. The tracking capability of the LS algorithms and the gradient lattice algorithm is not affected by the channel characteristics. Consequently, the LS DFE's and the gradient lattice DFE, in general, have a much better error rate performance than the LMS DFE in time-variant channels. This has been demonstrated by simulation results.

With regard to the two kinds of lattice DFE's, the LS lattice DFE has a faster initial convergence rate as expected, but requires a little more computation (about 4:3 for LS versus gradient). Both lattice DFE's are computationally more efficient than the Kalman and square-root Kalman algorithms. In particular, the gradient lattice DFE requires only a little more computation than the fast Kalman DFE. The error rates of the LS and the gradient lattice DFE's are almost identical for both time-invariant and time-variant (fading) channels. They are similar in performance to the results obtained from a simulation of the Kalman DFE, and are close to the theoretical optimum. Since the performance of both DFE's is so similar and the computational complexities are also close, the choice of an algorithm depends on the requirements of the application. When the computational burden is critical, the gradient lattice is appropriate. If fast initial convergence rate is highly desirable, the LS lattice would be preferable.

Simulation results also indicate that the lattice DFE's are relatively insensitive to roundoff noise. No reinitialization was required in simulation for both lattice DFE's, even when a very short computer word length was used. This is a better result than the stable square-root Kalman algorithm. These facts suggest that the lattice DFE's are an attractive choice for the equalization of rapidly time-variant multipath channels.

References

[1] G. D. Forney, Jr., "Maximum-likelihood sequence estimation of digital signaling in the presence of intersymbol interference," *IEEE Trans. Inform. Theory,* vol. IT-18, pp. 363–378, May 1972.

[2] J. G. Proakis and J. H. Miller, "Adaptive receiver for digital signaling through channels with intersymbol interference," *IEEE Trans. Inform. Theory,* vol. IT-15, pp. 487–497, July 1979.

[3] M. E. Austin, "Decision feedback equalization for digital communication over dispersive channels," M.I.T. Lincoln Lab., Lexington, MA, Tech. Rep. 437, Aug. 1967.

[4] J. G. Proakis, *Digital Communications.* New York: McGraw-Hill, 1983.

[5] B. Widrow and M. E. Hoff, Jr., "Adaptive switching circuits," in *IRE Wescon Conv. Rec.*, 1960, part 4, pp. 96–104.

[6] B. Widrow *et al.*, "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, pp. 1156–1162, Aug. 1976.

[7] F. M. Hsu *et al.*, "High speed modem techniques for fading dispersive channels," presented at NUSC Workshop Commun. in Fading Dispersive Medium, New London, CT, June 5–6, 1979.

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

[9] E. H. Satorius and J. D. Pack, "Application of least squares lattice algorithms to adaptive equalization," *IEEE Trans. Commun.*, vol. COM-29, pp. 136–142, Feb. 1981.

[10] E. H. Satorius and S. T. Alexander, "Channel equalization using adaptive lattice algorithms," *IEEE Trans. Commun.*, vol. COM-27, pp. 899–905, June 1979.

[11] D. D. Falconer and L. Ljung, "Application of fast Kalman estimation to adaptive equalization," *IEEE Trans. Commun.*, vol. COM-26, pp. 1439–1446, Oct. 1978.

[12] F. Ling and J. G. Proakis, "A generalized least square lattice algorithm and its application to decision-feedback equalization," in *Proc. ICASSP*, Paris, France, May 1982, pp. 1439–1446.

[13] ——, "Lattice decision-feedback equalizers and their application to fading dispersive channels," in *Proc. Int. Conf. Commun.*, Boston, MA, June 1983, pp. C8.2.1–C8.2.5.

[14] D. T. L. Lee *et al.*, "Recursive least squares ladder estimation algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 627–641, June 1981.

[15] F. M. Hsu, "Square-root Kalman filtering for high-speed data received over fading dispersive HF channels," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 753–763, Sept. 1982.

[16] G. Carayannis, D. Manolakis, and N. Kalouptsidis, "Fast Kalman type algorithms for sequential signal processing," in *Proc. IEEE ICASSP*, Boston, MA, Apr. 1983, pp. 186–189.

[17] J. M. Cioffi and T. Kailath, "Fast, fixed-order, least-squares algorithms for adaptive filtering," in *Proc. IEEE ICASSP*, Boston, MA, Apr. 1983, pp. 679–682.

[18] M. J. Shensa, "A least squares lattice decision-feedback equalizer," in *Proc. IEEE Int. Conf. Commun.*, Seattle, WA, June 1980.

[19] F. Ling and J. G. Proakis, "A generalized multichannel least squares lattice algorithm with sequential processing stages," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 381–389, Apr. 1984.

[20] R. S. Medaugh, "A comparison of two fast linear predictors," Ph.D. dissertation, Univ. Colorado, Boulder, 1981.

[21] F. Ling and J. G. Proakis, "Nonlinear learning characteristic of least squares adaptive algorithms," in *Proc. IEEE ICASSP*, San Diego, CA, Mar. 1984, pp. 7.3.1–7.3.4.

[22] A. S. Householder, *The Theory of Matrices in Numerical Analysis.* New York: Blaisdell, 1964, ch. 3.

Fuyun Ling (S'81–M'84) was born in Shanghai, China, on November 1, 1944. He received the B.S. degree in electrical engineering from Qinghua University, Beijing, China, in 1967, and the M.S. and Ph.D degrees in electrical engineering from Northeastern University, Boston, MA, in 1981 and 1984, respectively.

From 1968 to 1978, he was an Electrical Engineer in the Chaoyang Heavy-Machinery Factory, Liaoning, China. From 1981 to 1984 he was with Northeastern University, where he served as a Teaching Assistant and Research Assistant. Since July 1984, he has been employed by Codex Corporation, Mansfield, MA, where he is a Senior Engineer in the Modulation Products Department. His technical interests include communication theory, adaptive filtering, and digital signal processing.

Dr. Ling is a member of Sigma Xi, Tau Beta Pi, and Phi Kappa Phi.

★

John G. Proakis (S'58–M'62–SM'82–F'84) was born in Chios, Greece, on June 10, 1935. He received the E.E. degree from the University of Cincinnati, Cincinnati, OH, in 1959, the S.M. degree in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1961, and the Ph.D. degree in engineering from Harvard University, Cambridge, in 1966.

From June 1959 to September 1963, he was with M.I.T., first as a Research Assistant in the Electronic Systems Laboratory, and later as a Staff Member at the Lincoln Laboratory, Lexington, MA. From 1963 to 1966 he was with Harvard University, where he was a Research Assistant in the Division of Engineering and Applied Physics. In December 1966, he joined the staff of the Communication Systems Laboratories of Sylvania Electronic Systems and later transferred to the Waltham Research Center of General Telephone and Electronics Laboratories, Inc., Waltham, MA. Since September 1969, he has been with Northeastern University, Boston, MA, where he holds the rank of Professor of Electrical Engineering. From July 1982 to June 1984 he was an Associate Dean of the College of Engineering and Director of the Graduate School of Engineering. At present, he is the Chairman of the Department of Electrical and Computer Engineering. His interests have centered in digital communications, adaptive filtering, and digital signal processing. He is the author of the book *Digital Communications* (New York: McGraw-Hill, 1983).

Dr. Proakis served as an Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY from 1974 to 1977, and the IEEE TRANSACTIONS ON COMMUNICATIONS during 1973–1974. He has also served on the Board of Governors of the Information Theory Group, and is a Past Chairman of the Boston Chapter of the Information Theory Group, a Registered Professional Engineer in the State of Ohio, and a member of Eta Kappa Nu, Tau Beta Pi, and Sigma Xi.