An LMS-Based Decision Feedback Equalizer for IS-136 Receivers

Wen-Rong Wu and Yih-Ming Tsuie

***Abstract*—In digital mobile communication systems, intersymbol interference is one of the main causes of degrading system performance. Decision feedback equalization (DFE) is the commonly used remedy for this problem. Since the channel is fast-varying, an adaptive algorithm possessing a fast convergence property is then required. The least mean square (LMS) algorithm is well known for its simplicity and robustness; however, its convergence is slow. As a consequence, the LMS algorithm is rarely considered in this application. In this paper, we consider an LMS-based DFE for the North American IS-136 system. We propose an extended multiple-training LMS algorithm accelerating the convergence process. The convergence properties of the multiple-training LMS algorithm are also analyzed. We prove that the multiple-training LMS algorithm can converge regardless of its initial value and derive closed-form expressions for the weight error vector power. We further take advantage of the IS-136 downlink slot format and divide a slot into two subslots. Bidirectional processing is then applied to each individual subslot. The proposed LMS-based DFE has a low computational complexity and is suitable for real-world implementation. Simulations with a 900- MHz carrier show that our algorithm can meet the 3% bit error rate requirement for mobile speeds up to 100 km/hr.**

***Index Terms*—Decision feedback equalizer (DFE), equalizer, least mean square (LMS), IS-136.**

I. Introduction

DIGITAL mobile communications can provide higher capacity and other advantages over its analog counterpart. The increasing demand for personal mobile communication services has driven the successful development of many systems. In North America, the IS-136 has been the standard for time-division multiple access (TDMA) digital cellular systems. One problem associated with digital wireless transmission is intersymbol interference (ISI). This problem is due to the multipath transmission channel between the base and mobile stations. A general remedy to this problem is the use of channel equalization [1]. When the relative speed between the base and mobile stations is high, the fast fading effect arises. This greatly complicates the equalizer design. Many equalization algorithms have been developed. There are mainly two approaches: maximum likelihood sequence estimators (MLSEs) [2]–[9] and decision feedback equalizers (DFEs) [11]–[22]. Generally speaking, the MLSE can yield better performance. However, its computational complexity is higher. Some research reported that the performances of the MLSE and DFE are similar in fast fading environments [10]. In this paper, we consider the DFE for the IS-136 system.

The DFE is a nonlinear equalization algorithm. Basically, it consists of feedforward and feedback filters. The input symbol for the feedback portion is the output from the decision device. Since the channel is time-varying, the coefficients of the DFE are usually trained by some adaptive algorithm. To operate in a highly variant mobile environment, the adaptive algorithm must have a fast convergence property. Two well-known types of adaptive algorithms are least mean square (LMS) and recursive least squares (RLS). The RLS algorithm has a fast convergence property but requires high computational complexity. By contrast, the LMS algorithm converges slowly, but its computational complexity is low. Due to the fast varying characteristics of the wireless channel, the RLS algorithm is the commonly used algorithm [11]–[17]. As a consequence, the equalizer consumes a large amount of the computational power in the receiver.

Besides the choice of the adaptive algorithm, there are other ways to improve the DFE performance. Nakai *et al.* [13] proposed the use of CDVCC codes, which are located in the middle of the IS-136 data slot, as an additional training sequence. The CDVCC codes are used to distinguish the cochannel users and are known before the digital transmission. Since the training symbols are increased, the DFE performance can be improved. The use of CDVCC codes as a training sequence can also be found in [4] and [14] (for MLSE). Liu *et al.* [15] proposed a bidirectional equalization scheme. The idea is to conduct equalization both in the forward and the reverse time direction and to choose one of them as the output using some criterion.

Due to the slow convergence property, the LMS-based DFE is rarely considered in the literature. The only work we can find is that by Wei and Guo [19], in which they proposed a bidirectional DFE with a multiple training LMS algorithm and showed that the DFE can operate for mobile speeds up to 60 km/h. The multiple training algorithm in [19] is similar to that proposed in [18]; however, the purpose is different. The algorithm in [18] is to prevent the stability problem arising from the least squares method.

In this paper, we propose an improved LMS-based DFE for the IS-136 system [20], [22]. This DFE, which can work at higher vehicular speeds, has a simple structure and requires a low computational complexity. Our approach is to develop better adaptive algorithms and training strategies. We proposed an extended multiple training LMS algorithm accelerating the convergence. We also theoretically analyze the convergence properties of the multiple-training LMS algorithm. We prove that if the iteration number is large enough, the multiple-training

Manuscript received October 19, 1999; revised September 13, 2001.

The authors are with the Department of Communication Engineering, National Chiao Tung University, Hsinchu, 300 Taiwan, R.O.C.

Publisher Item Identifier S 0018-9545(02)00443-7.

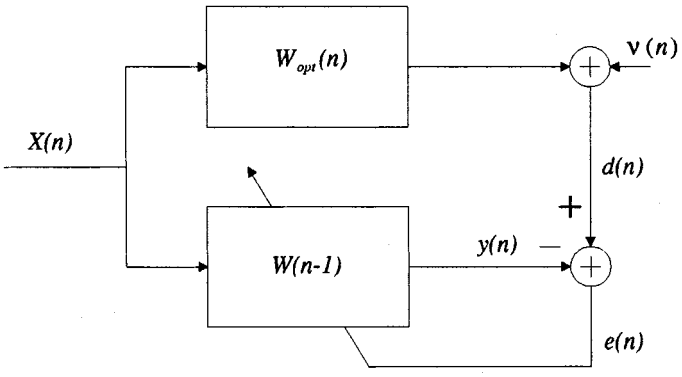

Fig. 1. The system identification model.

LMS algorithm can converge regardless of its initial value, and derive closed-form expressions for the weight error vector power. To further improve the performance, we take advantage of the IS-136 slot format and divide a slot into two subslots. Bidirectional equalization is then applied to the individual subslots. Here, the CDVCC codes are used. This approach is different from those in [15] and [19], where bidirectional processing is applied to the whole slot. Simulations with a 900-MHz carrier show that while the proposed algorithm can achieve the required 3% bit error rate (BER) for mobile speeds up to 100 km/h, its computational complexity is still low.

This paper is organized as follows. In Section II, we describe the the multiple training LMS algorithm and analyze its convergence behavior. In Section III, we discuss in detail the proposed DFE algorithms and make some comparisons with the existing algorithms. In Section IV, we report some simulation results demonstrating the adequacy of our theoretical analysis and the effectiveness of the proposed algorithm. We draw conclusions in Section V.

II. The Multiple-Training LMS Algorithm

A. Formulation

In this scheme, the input signal of an adaptive filter is first divided into blocks. In each block, data are repeatedly trained by the LMS algorithm until a preassigned iteration number is reached. Then the filter outputs the data in the last iteration and continues to process the next block with the previously converged tap weights as its initials. Consider a system identification application depicted in Fig. 1, where the unknown system characterized by $W_{\rm opt}(n)$ is identified by an adaptive filter $W(n)$. Let n be the time index, M the block size, and $n=(l-1)M+m, m=1,2\ldots,M$. Then, the nth symbol in the original input signal can be transferred into the mth symbol in the lth block. Let the iteration number index be k and the total number of iterations per block be K. We use the notation $W_{l,k}(m)$ to denote the filter weight vector corresponding to the mth symbol in the lth block at the kth iteration. Similar definitions are used for the output $y_{l,k}(m)$ and the error signal $e_{l,k}(m)$. Since in the same block the input vector and the desired signals are identical for every iteration, we use $X_l(m)$ and $d_l(m)$ to denote them. The multiple training LMS algorithm (in the lth block of data) can be mathematically described as follows.

For $m=1,2,\ldots,M$, and $k=1,2,\ldots,K$

$$y_{l,k}(m)=W_{l,k}^H(m-1)X_l(m) \tag{1}$$
$$e_{l,k}(m)=d_l(m)-y_{l,k}(m) \tag{2}$$
$$W_{l,k}(m)=W_{l,k}(m-1)+\mu\frac{X_{l,k}(m)}{X_{l,k}^H(m)X_{l,k}(m)}e_{l,k}^*(m) \tag{3}$$

where the superscript "$*$" denotes the complex conjugate operation and "H" the Hermitian operation. The initial weight vector for each block is obtained as

$$W_{l,1}(1)=W_{l-1,K}(M) \tag{4}$$
$$W_{l,k}(1)=W_{l,k-1}(M). \tag{5}$$

The final output for the lth block is then $y_{l,K}(m)$. Note that in (3), the input vector is normalized with respect to its power. For this reason, it is generally referred to as the normalized LMS algorithm. We will drop the word "normalized" in the sequel for reference simplicity. This algorithm was originally proposed in [18] and later used by [19]. We call the algorithm as the multiple-training LMS algorithm (MLMS).

B. Convergence Analysis

In this section, we analyze the convergence behavior of the MLMS algorithm. This was not done previously in the literature. To obtain a trackable result, we consider a system identification application as shown in Fig. 1. The update equations for the standard LMS algorithm are given as

$$y(n)=W^H(n-1)X(n) \tag{6}$$
$$d(n)=W_{\rm opt}^H(n)X(n)+\nu(n) \tag{7}$$
$$e(n)=d(n)-y(n) \tag{8}$$
$$W(n)=W(n-1)+\mu\frac{X(n)}{X^H(n)X(n)}e^*(n) \tag{9}$$

where $\nu(n)$ denotes the white observation noise. Now, we consider the iteration of the LMS algorithm in a single data block with size M. Here, we drop the subscript l, which denotes the block number in (1)–(5). Let $W_k(n)$ denote the filter tap weights at the nth time instant of the kth iteration. For $n=1$, from (9), we have

$$W_1(1)=W_1(0)-\mu\frac{X(1)}{X^H(1)X(1)}e^*(1). \tag{10}$$

Substituting (7) and (8) into (10), we obtain

$$\begin{aligned}W_1(1)&=W_1(0)+\mu\frac{X(1)}{X^H(1)X(1)}\\&\quad\times\{[W_{\rm opt}(1)-W_1(0)]^HX(1)+\nu(n)\}^*\\&=W_1(0)+\mu\frac{X(1)X^H(1)}{X^H(1)X(1)}\\&\quad\times[W_{\rm opt}(1)-W_1(0)]+\mu\frac{X(1)}{X^H(1)X(1)}\nu^*(1).\end{aligned} \tag{11}$$

We then subtract $W_{\rm opt}(2)$ from both sides of (11) and manipulate the right-hand side by adding and subtracting $W_{\rm opt}(1)$. This leads to the following result:

$$\begin{aligned} W_1(1)-W_{\rm opt}(2) &= W_1(0)-W_{\rm opt}(1)+W_{\rm opt}(1)-W_{\rm opt}(2) \\ &\quad +\mu\frac{X(1)X^H(1)}{X^H(1)X(1)}[W_{\rm opt}(1)-W_1(0)] \\ &\quad +\mu\frac{X(1)}{X^H(1)X(1)}\nu^*(1) \qquad (12) \\ \Delta W_1(1) &= \left[I-\mu\frac{X(1)X^H(1)}{X^H(1)X(1)}\right]\Delta W_1(0) \\ &\quad +T(1)+V(1) \\ &= P(1)\Delta W_1(0)+T(1)+V(1) \qquad (13) \end{aligned}$$

where

$$P(1)=\left[I-\mu\frac{X(1)X^H(1)}{X^H(1)X(1)}\right] \qquad (14)$$

$$V(1)=\mu\frac{X(1)}{X^H(1)X(1)}\nu^*(1) \qquad (15)$$

$$T(1)=W_{\rm opt}(1)-W_{\rm opt}(2) \qquad (16)$$

$$\Delta W_1(1)=W_1(1)-W_{\rm opt}(2). \qquad (17)$$

Note that (17) corresponds to the a priori weight error vector, and later we will convert it to the a posteriori one. The vector in (16) is the lag error vector due to the time-varying characteristics of the unknown system. Let $A_i^j=P(j)P(j-1)\dots P(i)$, where $P(k)$ is defined as that in (14) with the input vector replaced by $X(k)$. Note that for the notation A_i^j, j must be greater or equal to i. However, we let $A_{i+1}^i=I$. This will simplify the expressions derived below. Using the recursive relation in (13), we can obtain

$$\Delta W_1(i)=A_1^i\Delta W_1(0)+\sum_{k=2}^{i+1}A_k^i[T(k-1)+V(k-1)]. \qquad (18)$$

Note that for the MLMS algorithm, $\Delta W_{i+1}(0)=\Delta W_i(M)$. Let K be the iteration number. Using the same recursive formula, we can obtain $\Delta W_K(i)$ as

$$\begin{aligned} \Delta W_K(i) &= A_1^i\left(A_1^M\right)^{K-1}\Delta W_1(0)+A_1^i\sum_{k=0}^{K-2}\left(A_1^M\right)^k \\ &\quad \times\sum_{k=2}^{M+1}A_k^M[T(k-1)+V(k-1)] \\ &\quad +\sum_{k=2}^{i+1}A_k^i[T(k-1)+V(k-1)]. \qquad (19) \end{aligned}$$

Now, we convert the a priori weight error vector to the a posteriori one. Using (9), we have

$$\begin{aligned} W(n)-W_{\rm opt}(n) &= \left[I-\mu\frac{X(n)X^H(n)}{X^H(n)X(n)}\right] \\ &\quad \times[W(n-1)-W_{\rm opt}(n)] \\ &\quad +\mu\frac{X(n)}{X^H(n)X(n)}\nu^*(n). \qquad (20) \end{aligned}$$

Denote the a posteriori weight error vector $W(n)-W_{\rm opt}(n)$ as $\Delta\mathcal{W}(n)$. Then, the relation between the a posteriori and a priori weight error vectors at $n=M$ of the Kth iteration can be expressed as

$$\Delta\mathcal{W}_K(M)=P(M)\Delta W_K(M-1)+V(M). \qquad (21)$$

Inserting (19) into (21), we then obtain

$$\begin{aligned} \Delta\mathcal{W}_K(M) &= \left(A_1^M\right)^K\Delta W_1(0)+\sum_{k=0}^{K-1}\left(A_1^M\right)^k \\ &\quad \times\sum_{k=2}^{M+1}A_k^M[T(k-1)+V(k-1)]-T(M). \end{aligned} \qquad (22)$$

Before our further derivation, we consider some special properties of A_1^M. As we previously defined, $A_1^M=P(M)P(M-1)\dots P(1)$. Note that $P(i)=I-\mu Q(i)$, where $Q(i)$ is the projection matrix that projects vectors to the subspace spanned by $X(i)$. Let a vector X be transformed by $P(i)$ and $Y=P(i)X$. Then, $Y=X-\mu Y_p$, where Y_p is the vector projected onto $X(i)$. Thus, if $\mu=1,\ P(i)$ corresponds to a projection matrix that projects vectors to the orthogonal subspace of $X(i)$. Note that if X is not in the orthogonal subspace of $X(i),\ \|Y\|$ is always smaller than $\|X\|$. Using this property, one can show that the eigenvalues of $P(i)$ are all ones except for one. The magnitude of the nonzero eigenvalue is always smaller than one. Similarly, the eigenvalues of $P(i)P(j)$ are all ones except for two. The magnitudes of these two nonzero eigenvalues are smaller than one. Thus, we can conclude that if $X(1), X(2),\dots,$ and $X(M)$ span R^N, where N is the filter length, magnitudes of all eigenvalues of A_1^M will be smaller than one. As a result, $(A_1^M)^K\to 0$ when $K\to\infty$. Using the relation $T(M)=W_{\rm opt}(M)-W_{\rm opt}(1)=-\sum_{i=1}^{M-1}T(i)$ in (22) and letting $K\to\infty$, we finally obtain

$$\begin{aligned} \Delta\mathcal{W}_K(M) &= \left[I-A_1^M\right]^{-1}\sum_{k=2}^{M}\left(A_k^M-A_1^M\right)T(k-1) \\ &\quad +\left[I-A_1^M\right]^{-1}\sum_{k=2}^{M+1}A_k^MV(k-1). \qquad (23) \end{aligned}$$

As (23) shows, the MLMS algorithm can converge in one block regardless of its initial value. Also, the tap weight error vector consists of two parts. The first term on the right-hand side of (23) is due to the time variation of $W_{\rm opt}(n)$, and we call it the lag error vector (LEV); it is denoted as $\Delta\mathcal{W}_{l,K}(M)$. The second term is due to the observation noise $\nu(n)$, and we call it the fluctuation error vector (FEV); it is denoted as $\Delta\mathcal{W}_{f,K}(M)$.

The result in (23) corresponds to a particular input sequence. Next, we analyze the mean-squared convergence of the weight error vector $\Delta\mathcal{W}_M(n)$. We calculate the mean value of the error vector power, which is $\mathbf{E}\{\Delta\mathcal{W}_K^H(n)\Delta\mathcal{W}_K(n)\}$. Let $W_{\rm opt}(n)$ be independent of $\nu(n)$. Then, we have

$$\begin{aligned} \mathbf{E}\left\{\Delta\mathcal{W}_K^H(n)\Delta\mathcal{W}_K(n)\right\} &= \mathbf{E}\left\{\Delta\mathcal{W}_{l,K}^H(n)\Delta\mathcal{W}_{l,K}(n)\right\} \\ &\quad +\mathbf{E}\left\{\Delta\mathcal{W}_{f,K}^H(n)\Delta\mathcal{W}_{f,K}(n)\right\}. \end{aligned} \qquad (24)$$

For simplicity, we denote $\mathbf{E}\{\Delta\mathcal{W}_{f,K}^H(n)\Delta\mathcal{W}_{f,K}(n)\}$ as $\mathcal{E}_f$ and $\mathbf{E}\{\Delta\mathcal{W}_{l,K}^H(n)\Delta\mathcal{W}_{l,K}(n)\}$ as $\mathcal{E}_l$. We first consider the FEV power. As we can see from (23), exact evaluation of this value is almost impossible. Some approximations are

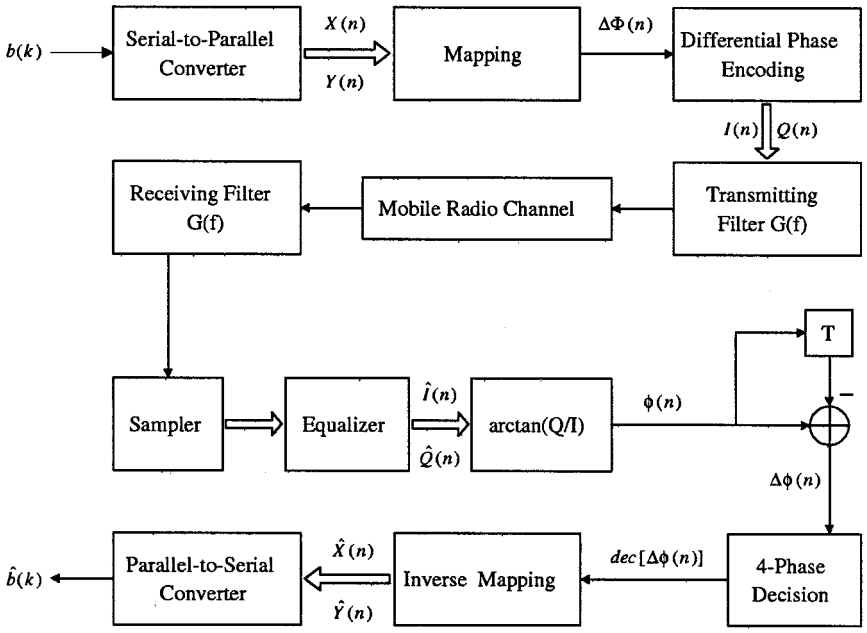

Fig. 2. The baseband model for the $\pi/4$ shifted DQPSK modulation.

necessary. Here, we use the direct averaging method and the independence theory [28]. Let the correlation matrix of $X(n)$ be R_x, $\mathbf{E}\{1/X^H(n)X(n)\}$ be ρ, and the observation noise variance be σ_ν^2. Then, we have

$$\begin{aligned}\mathcal{E}_f &= \mathbf{E}\left\{\Delta\mathcal{W}_{f,K}^H(n)\Delta\mathcal{W}_{f,K}(n)\right\}\\ &= \mathbf{Tr}\left\{\mathbf{E}\left\{\Delta\mathcal{W}_{f,K}(n)\Delta\mathcal{W}_{f,K}^H(n)\right\}\right\}\\ &\approx \mathbf{Tr}\left\{(I-\mathcal{P}^M)^{-1}\left(\sum_{k=1}^{M}\mathcal{P}^{M-k}R_v\mathcal{P}^{M-k}\right)\right.\\ &\qquad\left.\times\,(I-\mathcal{P}^M)^{-1}\right\}\\ &= \mathbf{Tr}\left\{R_v\left[\sum_{k=1}^{M}\mathcal{P}^{M-k}(I-\mathcal{P}^M)^{-2}\mathcal{P}^{M-k}\right]\right\} \quad (25)\end{aligned}$$

where $\mathbf{Tr}[\cdot]$ denotes the trace operation

$$R_v = \mathbf{E}\{V(n)V^H(n)\} \approx \mu^2\rho^2\sigma_\nu^2 R_x$$

and $\mathcal{P} = \mathbf{E}\{P(i)\} \approx I - \mu\rho R_x$. Express R_x as $U\Lambda U^H$, where U is a unitary matrix and Λ is the diagonal matrix containing eigenvalues of R_x. We can rewrite (25) as

$$\begin{aligned}\mathcal{E}_f = \mathbf{Tr}\left\{\mu^2\rho^2\sigma_\nu^2\Lambda\sum_{k=1}^{M}(I-\mu\rho\Lambda)^{M-k}\right.\\ \left.\times\,[I-(I-\mu\rho\Lambda)^M]^{-2}(I-\mu\rho\Lambda)^{M-k}\right\}. \quad (26)\end{aligned}$$

Let $\eta_{f,i}$ denote the ith diagonal element of the matrix inside the trace operator. Then

$$\begin{aligned}\eta_{f,i} &= \frac{\mu^2\rho^2\sigma_\nu^2\lambda_i}{[1-(1-\mu\rho\lambda_i)^M]^2}\sum_{k=1}^{M}(1-\mu\rho\lambda_i)^{2(k-1)}\\ &= \frac{\mu^2\rho^2\sigma_\nu^2}{1-(1-\mu\rho\lambda_i)^2}\,\frac{1+(1-\mu\rho\lambda_i)^M}{1-(1-\mu\rho\lambda_i)^M} \quad (27)\end{aligned}$$

where λ_i is the ith component of Λ. Then, we have the FEV power as

$$\mathcal{E}_f = \sum_{i=1}^{N}\frac{\mu^2\rho^2\sigma_\nu^2}{1-(1-\mu\rho\lambda_i)^2}\,\frac{1+(1-\mu\rho\lambda_i)^M}{1-(1-\mu\rho\lambda_i)^M}. \quad (28)$$

Calculation of the LEV power is similar to that of the FEV power. However, we need a model for the time-varying system. A commonly used model is the random walk, i.e.,

$$W_{\text{opt}}(n+1) = W_{\text{opt}}(n) + G(n+1) \quad (29)$$

where $G(n)$ is the driving noise vector. We assume that each component of $G(n)$ is independent of one another and denote the power of the ith component of $G(n)$ as σ_i^2. From (23), we have

$$\begin{aligned}\mathcal{E}_l &= \mathbf{E}\left\{\Delta\mathcal{W}_{l,K}^H(n)\Delta\mathcal{W}_{l,K}(n)\right\}\\ &= \mathbf{Tr}\left\{\mathbf{E}\left\{\Delta\mathcal{W}_{l,K}(n)\Delta\mathcal{W}_{l,K}^H(n)\right\}\right\}\\ &= \mathbf{Tr}\left\{(I-\mathcal{P}^M)^{-1}\left[\sum_{k=1}^{L-1}\mathcal{P}^{M-k}(I-\mathcal{P}^k)\right.\right.\\ &\qquad\left.\left.\times\,\Lambda_T(I-\mathcal{P}^k)\mathcal{P}^{M-k}\right](I-\mathcal{P}^M)^{-1}\right\}\end{aligned} \quad (30)$$

where $\Lambda_T = \mathbf{E}\{T(n)T^H(n)\}$ is a diagonal matrix consisting of $\{\sigma_1^2,\sigma_2^2,\ldots,\sigma_N^2\}$. Let $\eta_{l,i}$ denote the ith diagonal element of the matrix inside the trace operator. After some manipulations, we can obtain

$$\begin{aligned}\eta_{l,i} &= \frac{\sigma_i^2}{[1-(1-\mu\rho\lambda_i)^M]^2}\\ &\quad\times\sum_{k=1}^{M-1}(1-\mu\rho\lambda_i)^{2(M-k)}[1-(1-\mu\rho\lambda_i)^k]^2. \quad (31)\end{aligned}$$

TABLE I
SYSTEM PARAMETERS OF THE NORTH AMERICAN IS-136 STANDARD

Access Method	TDMA
Transmission Rate	48.6kbps
Modulation	$\pi/4$ shifted DQPSK
Transmitting Filter	SRRC roll-off=0.35
Receiving Filter	SRRC roll-off=0.35

TABLE II
THE MAPPING TABLE OF $\pi/4$ SHIFTED DQPSK MODULATION

$X(n)$	$Y(n)$	$\Delta\phi(n)$
1	1	$-3\pi/4$
0	1	$3\pi/4$
0	0	$\pi/4$
1	0	$-\pi/4$

Using the above expression, we obtain

$$\mathcal{E}_l = \sum_{i=1}^{N} \frac{(1-\mu\rho\lambda_i)^2\sigma_i^2}{[1-(1-\mu\rho\lambda_i)^M]^2} \times \left\{ \frac{1}{1-(1-\mu\rho\lambda_i)^2} \times [1-2(1-\mu\rho\lambda_i)^{M-1}-2(1-\mu\rho\lambda_i)^M + (1-\mu\rho\lambda_i)^{2M-2}+2(1-\mu\rho\lambda_i)^{2M-1}] + (M-1)(1-\mu\rho\lambda_i)^{2(M-1)} \right\} \quad (32)$$

From (28), we can see that if the other parameters (μ, ρ, and λ_i) are fixed, the FEV power is a decreasing function of the block size M. This gives the smoothing property of the MLMS algorithm. The larger the block size, the more noise we can suppress. From (32), we can see that the LEV power is an increasing function of M. This gives the tracking property of the MLMS algorithm. For a large block size, the variation of the system in the block is large. Since all data in the block will affect the final output at a particular time instant, the tracking capability of the MLMS algorithm is then degraded. From the above analysis, we can conclude that the choice of the block size M should be a compromise between the smoothing and tracking capabilities of the MLMS algorithm.

III. THE PROPOSED LMS-BASED DFE

A. The IS-136 System

In this section, we briefly describe the IS-136 system. The system parameters for the IS-136 are summarized in Table I. As the table shows, the access method for IS-136 is TDMA. Each carrier is divided into three time slots. The baseband model for the $\pi/4$ shifted DQPSK modulator and demodulator is shown in Fig. 2. Bits from the data source are first passed through the serial-to-parallel converter; the resulting pair $(X(n), Y(n))$ then maps a phase difference $\Delta\Phi(n)$ according to the rule shown in Table II. The transmission signal phase is then differentially encoded, i.e., $\Phi(n) = \Phi(n-1) + \Delta\Phi(n)$. Thus, the resulting signal constellation of $I(n) + jQ(n)$ is $\{\pm 1, \pm j\}$ when the symbol index n is odd, and is $\{\pm(1/\sqrt{2})(1 \pm j)\}$ when n is even.

The receiver reverses the operations conducted in the transmitter. The received signal is first passed through the receiving filter and sampler. Then, an equalizer is used to eliminate the ISI caused by the channel, the phase $\phi(n)$ of the equalizer output is extracted, and the phase difference $\Delta\phi(n) = \phi(n) - \phi(n-1)$ is calculated. The resulting phase difference is fed to a four-phase decision circuit, producing $\text{dec}[\Delta\phi(n)]$, where $\text{dec}[\cdot]$ denotes the phase decision operation. Using Table II, $\text{dec}[\Delta\phi(n)]$ is inversely mapped into $(\hat{X}(n), \hat{Y}(n))$. Then, the parallel-to-serial converter converts $\hat{X}(n)$ and $\hat{Y}(n)$ into a bitstream.

The downlink (from base station to mobile unit) data slot format of the IS-136 system is shown in Fig. 3. It consists of 14 SYNC symbols, six SACCH symbols, 65 DATA symbols, six CDVCC symbols, 65 DATA symbols, and six RSVD symbols. The 14 SYNC symbols are designed for synchronization. The equalizer can also use these symbols for training. The six CDVCC symbols are known to the receiver before the digital transmission. They are used to distinguish the cochannel users.

B. The Extended Multiple Training Algorithm

The original MLMS algorithm has been used only in the training mode [19]. Also, all 14 SYNC symbols are taken as a training block. There are some problems of this approach. According to our analysis shown in Section II, the block size of the MLMS algorithm should be chosen as a compromise between the smoothing and tracking capabilities. Thus, the optimal block size may not be 14. Since multiple training is not applied in the tracking mode, the decision information is not fully explored. Based on the above observations, we make some extension to the MLMS algorithm. First, we extend the MLMS algorithm to the tracking mode. Second, we let the block size be a design parameter. For different signal-to-noise ratios (SNRs) or mobile speeds, we may use different block sizes. Also, blocks can even be overlapped. The proposed MLMS algorithm is now described as follows. The received slot is first divided into blocks. In each block, data are repeatedly equalized by the DFE using the MLMS algorithm until a preassigned iteration number is reached. Then the DFE outputs the equalized data in the last iteration and continues to process the next block of data. If the DFE is in the training mode, the SYNC sequence is used as the reference signal. If the DFE is in the tracking mode, the output from the decision device is used as the reference signal. For ease of description, we let the slot be divided into L blocks, and each block has M symbols ($LM = 162$). The weight update equations are identical to (1)–(5), except that the desired signal becomes

$$d_{l,k}(m) = \begin{cases} a_l(m) & \text{in training mode} \\ \text{dec}[y_{l,k}(m)] & \text{in tracking mode} \end{cases} \quad (33)$$

where $a_l(m)$ is the original transmitted symbol and $\text{dec}[\cdot]$ denotes a decision operation. The final DFE output is then $y_{l,K_l}(m)$. For reference convenience, we call the algorithm here the EMLMS algorithm. The total number of iterations can be different from block to block, denoted by K_l. For simplicity, we only use two iteration numbers: one for the training mode, denoted as K_1, and the other for the tracking mode, denoted as K_2. These two parameters control the convergence of the EMLMS DFE. The DFE performs better for larger (K_1, K_2);

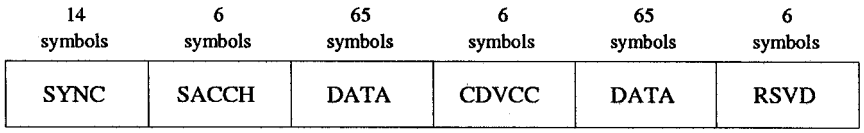

Fig. 3. The downlink slot format of the IS-136 system.

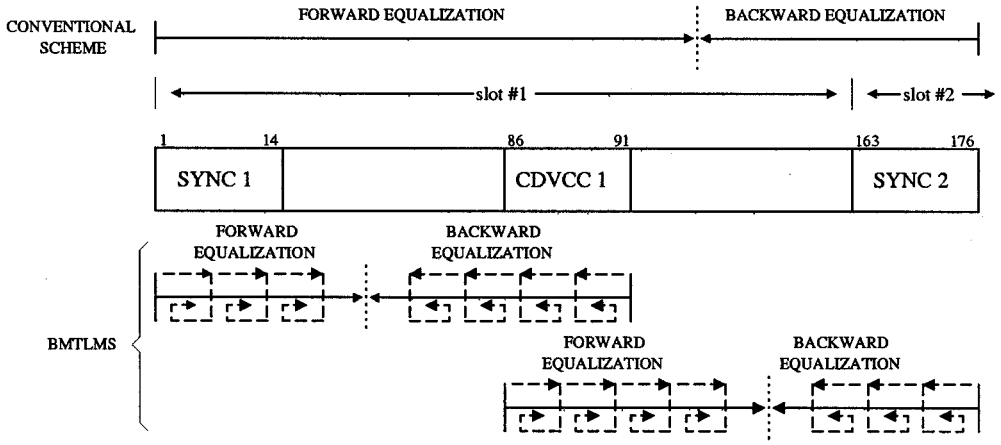

Fig. 4. The proposed bidirectional equalization scheme.

TABLE III
COMPARISON OF DFE ALGORITHMS DESIGNED FOR THE IS-136 SYSTEM

	type	use CDVCC ?	convergence speed	decoding delay
RLS [13]	RLS	Yes	fast	small
BRLS [15]	RLS	No	fast	small ↓ large
BMLMS [19]	LMS	No	medium	small ↓ large
EMLMS	LMS	Yes	medium	small
Threshold BEMLMS	LMS	Yes	Fast	small ↓ medium
Comparison BEMLMS	LMS	Yes	Fast	medium

TABLE IV
COMPUTATIONAL COMPLEXITY COMPARISON OF THE DFE ALGORITHMS DESIGNED FOR THE IS-136

Algorithm	Complex Multiplications	Divisions
RLS [13]	$162(2.5N^2+4.5N)$	$(2*162)$
BRLS [15]	$176(2.5N^2+4.5N)$	$(2*176)+176$
BMLMS [19]	$(28K_1+148)(2N+1)$	$(28K_1+148)+176$
EMLMS	$(20K_1+142K_2)(2N+1)$	$(20K_1+142K_2)$
Threshold BEMLMS	$(40K_1+142K_2)(2N+1)$	$(40K_1+142K_2)+L$
Comparison BEMLMS	$(40K_1+142K_2)(2N+1)$	$(40K_1+142K_2)$

however, the computational load also increases linearly with these numbers. The other parameter influencing the convergence rate is the step size μ. As we can see from Section II, a large μ increases the convergence speed of the DFE, but it also causes a larger fluctuation error.

As we mentioned, the LMS requires a reference signal to adapt the filter weight vector. In the training mode, this signal is provided by the encoded SYNC signal. In the tracking mode, the reference signal is extracted from the received signal [15], [23]. We first partition the symbol constellation of shift $\pi/4$ DQPSK into two sets: Set #0 is $\{\pm(1/\sqrt{2})(1\pm j)\}$ and Set #1 is $\{\pm 1, \pm j\}$. From the above description, we can see that the shift $\pi/4$ DQPSK coded symbol will be in Set #0 when the symbol index n is even, and in Set #1 when n is odd. Thus, if n is even, the phase of the encoded symbol will be an odd multiple of $\pi/4$ (Set #0). In other words, $r(n) = e^{j\,\mathrm{dec}[\phi(n)]} = e^{jk\pi/4}$, where k is the argument minimizing $|\phi(n) - k\pi/4|, k = \pm 1, \pm 3$. If n is odd, the phase of the encoded symbol will be a multiple of $\pi/2$ (Set # 1). In this case, $r(n) = e^{j\,\mathrm{dec}[\phi(n)]} = e^{j(k+1)\pi/4}$, where k is the argument minimizing $|\phi(n) - (k+1)\pi/4|, k = \pm 1, \pm 3$. Note that there is no guarantee that the extracted reference signal is correct all the time.

C. The Bidirectional Equalization

Since the channel is fast-varying, the longer the DFE is in the tracking mode, the higher the probability it may lose track of the channel variations. Thus, the BER is generally higher around the end of a slot. Note that we can have another SYNC sequence if we continuously process the next slot, which may not belong to the current user. Thus, if we can process the current slot backward and use the SYNC sequence of the next slot, the BER around the end of the current slot can be reduced. This is the basic idea of bidirectional equalization. The proposed bidirectional equalization scheme is shown in Fig. 4. In the conventional bidirectional schemes [15], [19], forward equalization starts from the beginning of the current slot and backward equalization starts from the last symbol (symbol 176) in SYNC 2. Our bidirectional processing is different from the conventional technique. Making use of CDVCC, we divide the 176 symbols into

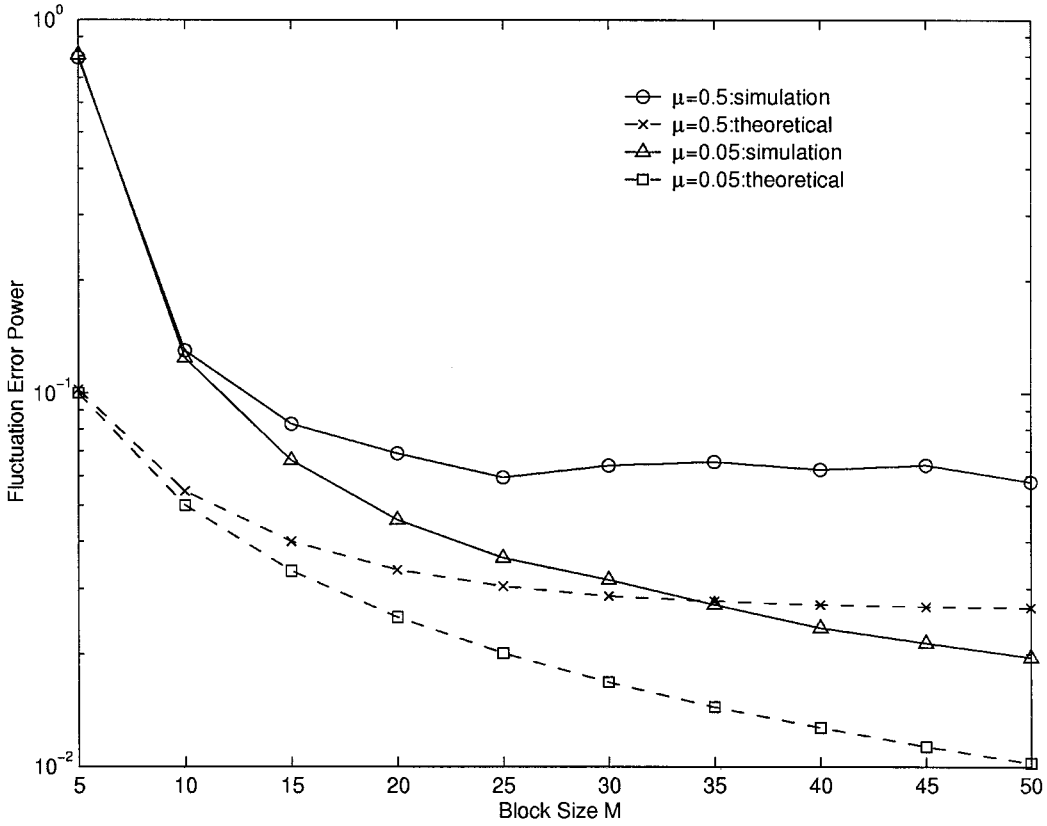

Fig. 5. The relation between the fluctuation error power ε_f and the block size M; $N = 5$, $\sigma_v^2 = 0.1$.

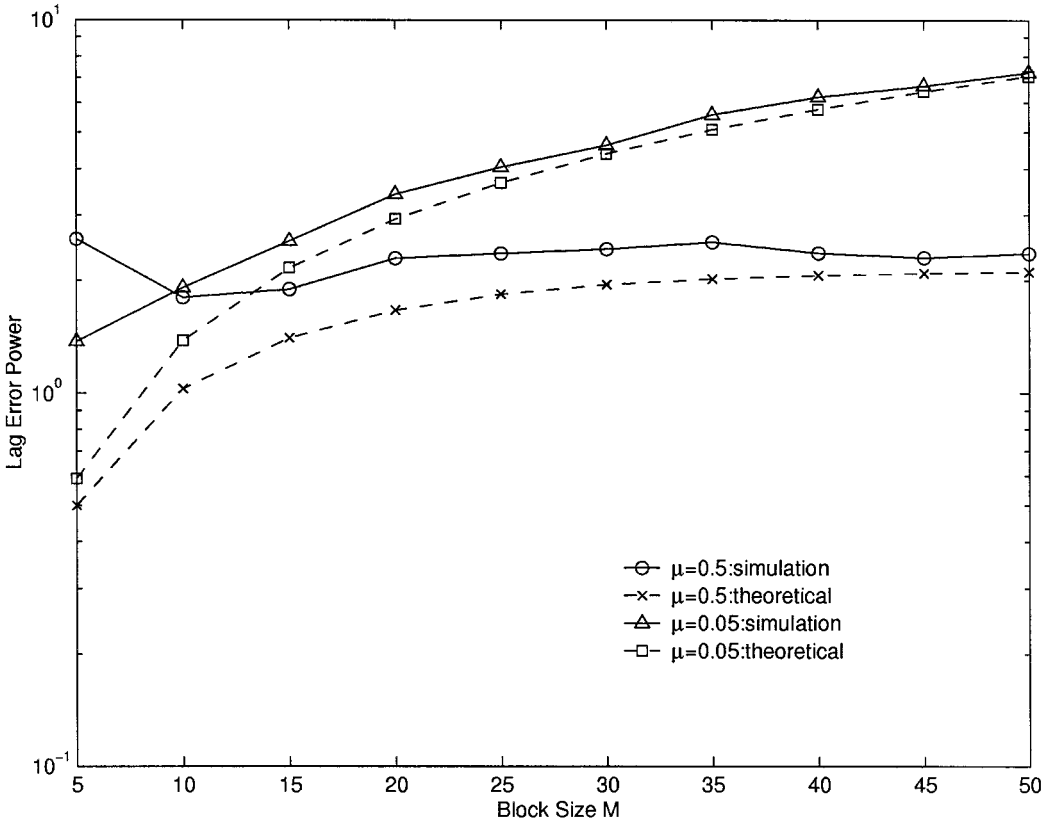

Fig. 6. The relation between the lag error power ε_l and the block size M; $N = 5$, $\sigma^2 = 0.1$.

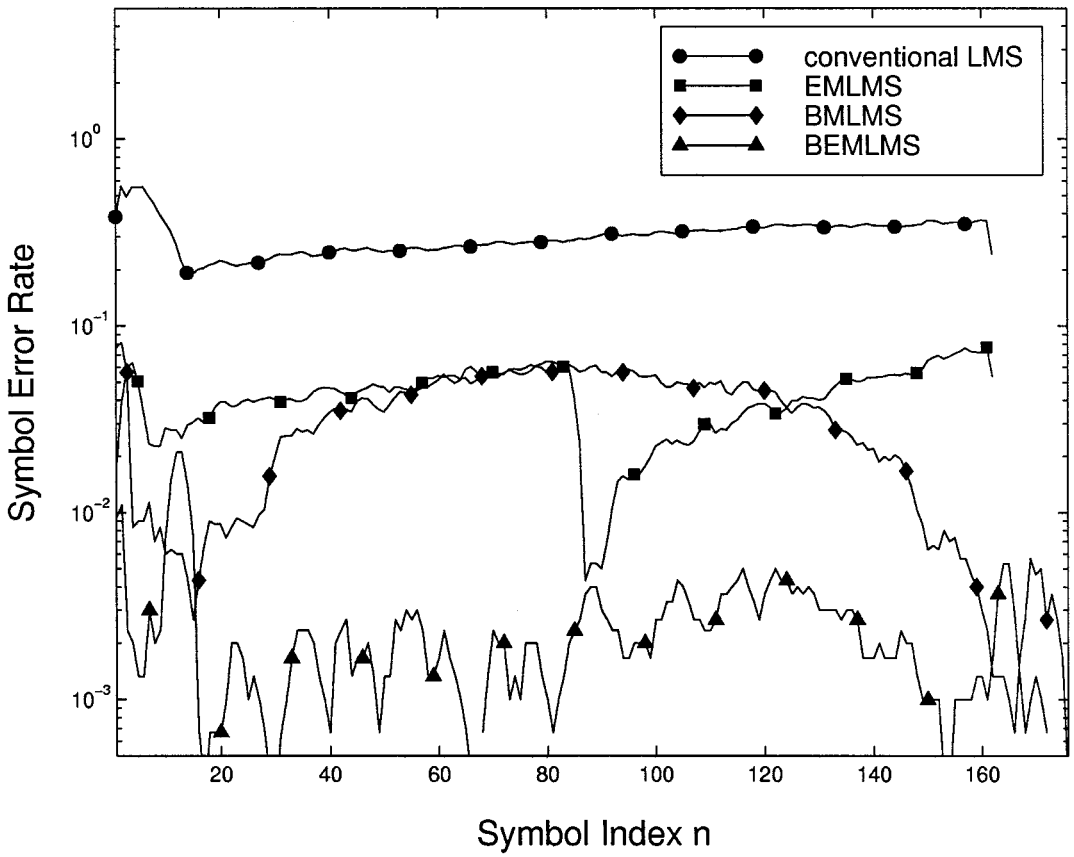

Fig. 7. Tracking capabilities of the conventional LMS DFE, EMLMS DFE, BMLMS DFE, and BEMLMS DFE ($V = 60$ km/h, $\tau = T$).

two subslots with some overlap (see Fig. 4). Since the CDVCC symbols are known, we can use them as an additional training sequence. In the first subslot, forward equalization starts from the first symbol of the current slot and backward equalization starts from the last symbol in the CDVCC field (symbol 91). In the second subslot, forward equalization begins from the first symbol in the CDVCC field (symbol 86) and backward equalization begins from the last symbol of SYNC 2 (symbol 176). The advantage of this bidirectional equalization scheme is apparent. Since the CDVCC field is at the middle of the data slot, the time that DFE operates in the tracking mode is shortened by half. The output delay due to bidirectional processing may also be reduced by half.

To reduce the computational requirement per unit time, the backward and forward equalization are not performed simultaneously. In other words, at a particular time instant, only one directional equalization is allowed. Thus, we require a control mechanism to determine in which direction the equalization should proceed. We designed two control mechanisms: a threshold-based and a comparison-based mechanism. The threshold-based control mechanism is similar to those used in [15] and [19]. This control mechanism requires a monitoring function and an empirical threshold value β. Let L' be the number of blocks in a subslot. Define the monitoring function as

$$f(l) = \frac{1}{M} \sum_{i=(l-1)M+1}^{lM} |e(i)|^2 \tag{34}$$

where l is the block index. As we can see, this monitoring function is block-based. This is in contrast to that used in [15] and [19], where the monitoring function is symbol-based. The monitoring function can be used to indicate the tracking status of the DFE. When $f(p)$ is smaller than a threshold β, we can consider that the DFE keeps good track of the channel variations. When $f(p)$ is larger than β, we consider that the DFE starts to lose track of the channel variations. Using this threshold-based control mechanism, the proposed bidirectional equalization can be summarized as follows.

1) Start the forward equalization using the EMLMS algorithm and calculate the associated monitoring function. If $f(P) > \beta$ at block P and $P < L'$, pause the forward equalization. Otherwise, proceed with the forward equalization to the end of the subslot.
2) Start the backward equalization (from block L') using the EMLMS algorithm and calculate the associated monitoring function. If $f(Q) > \beta$ at block Q and $Q > P+1$, pause the backward equalization. Otherwise, proceed with the backward equalization to block $P+1$.
3) If $Q > P+1$, set a block index as

$$S = \left\lfloor \frac{P+Q}{2} \right\rfloor \tag{35}$$

where $\lfloor x \rfloor$ denotes the integer part of x.
4) Continue the forward equalization from block $P+1$ to block S and backward equalization from block $Q-1$ back to block $S+1$.

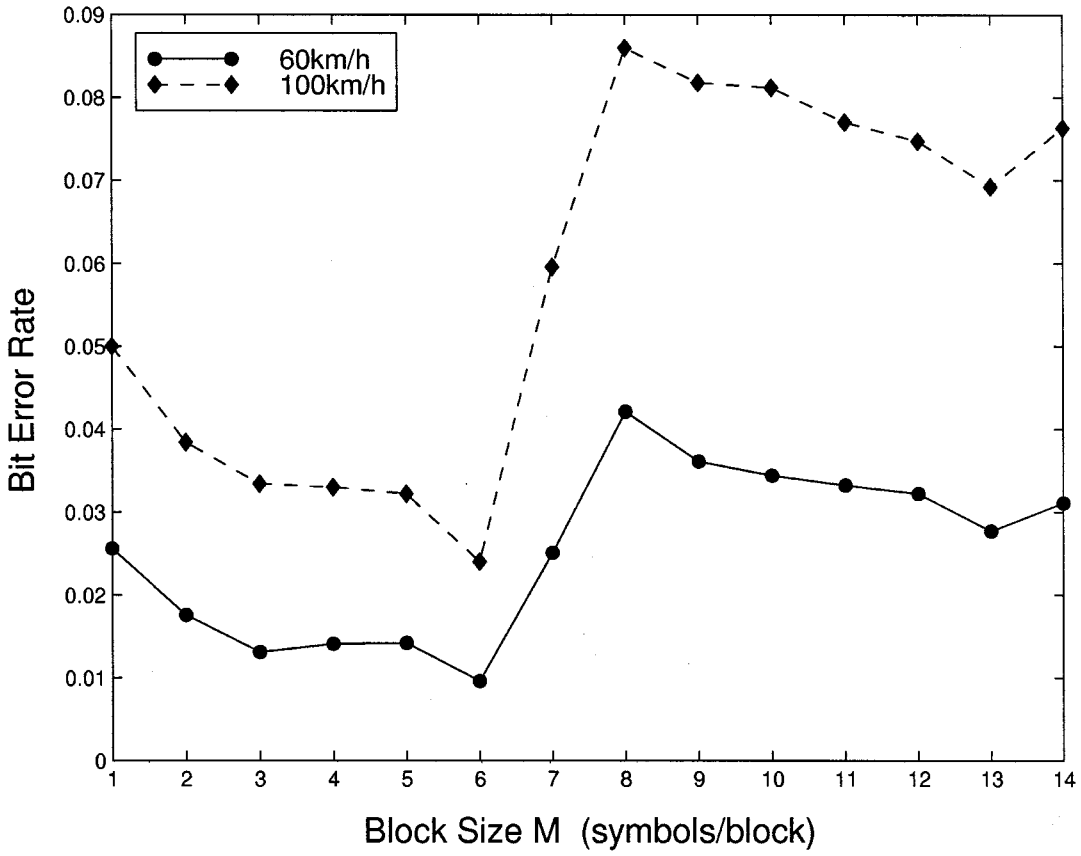

Fig. 8. Effect of the block size M on BEMLMS DFE performance ($\tau = T$, $(K_1, K_2) = (7, 1)$).

With a proper choose of β, the block index S can approach the location of the mobile channel deep fade [15].

The drawback of the threshold-based algorithm is that we need to determine the threshold β empirically. This may not be easy in many cases. To avoid that, we designed another control mechanism that compares the values of the forward and backward monitoring functions. It is reasonable to assume that the smaller the monitoring function, the better the tracking condition of the DFE. Thus, we can compare the values of the forward and backward monitoring functions and determine in which direction the DFE should proceed. If the forward monitoring function is smaller, the DFE proceeds in the forward direction and vice versa. Using this comparison-based control mechanism, the proposed bidirectional equalization can be summarized as follows.

1) Initially, perform forward equalization to block one using the EMLMS algorithm and calculate the monitoring function $f(1)$. Similarly, perform backward equalization to block L' and calculate $f(L')$.
2) Let block p be the latest block that has been forwardly equalized and block q be the latest block that has been backwardly equalized $(q > p + 1)$. If $f(p) < f(q)$, then proceed to equalize block $p+1$ forwardly; otherwise proceed to equalize the block $q - 1$ backwardly.
3) Repeat Step 2) until $q = p + 1$.

Using this scheme, we do not have to determine the threshold value β. Also, the normalization factor $1/M$ is not required here. We call this a bidirectional EMLMS (BEMLMS) algorithm. A similar bidirectional equalization structure was also independently developed in [17]. However, in [17], no detailed description was reported. The other main difference is that the RLS algorithm was used in [17]. A final remark is that the revised IS-136 standard does not guarantee the availability of the SYNC word in the next slot. Adjacent slots may be transmitted at different power levels or not at all.

D. Comparisons of DFE Algorithms

In this section, we compare the merits of various adaptive DFE algorithms designed for the IS-136 system. The algorithms considered include the RLS algorithms proposed by Nakai [13] and by Liu [15], the LMS-type algorithms proposed by Wei [19], the EMLMS algorithm, and the BEMLMS algorithm. The properties of these algorithms are summarized in Table III. For reference convenience, we use the abbreviation BRLS to denote the algorithm in [15] and BMLMS the algorithm in [19]. First, we discuss the processing delay. For the bidirectional algorithms in [15] and [19], the processing delay can be large when the forward DFE meets a deep fade in its early stage. This is because the backward equalization will not be initiated until the entire slot is received. Since we divide the slot into two subslots, the processing delay is inherently smaller than that in [15] and [19]. The processing delay of the comparison-based BEMLMS is larger than that of the threshold-based BEMLMS because the comparison-based control mechanism needs to compute both

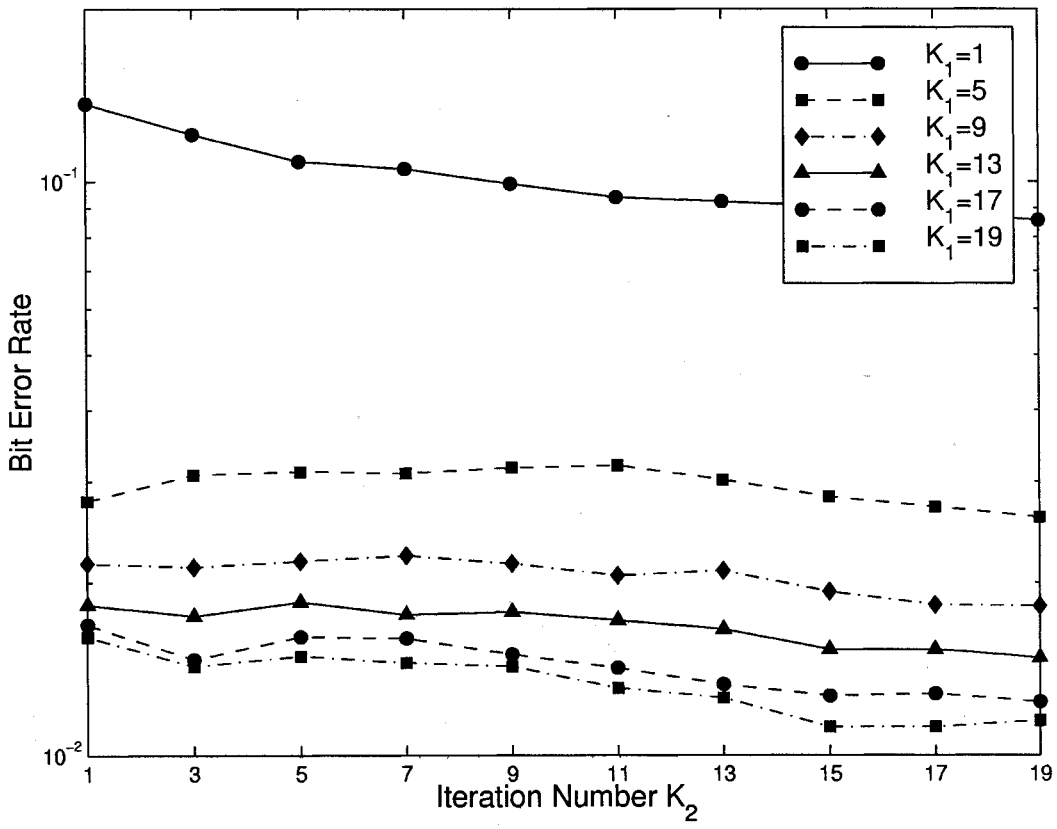

Fig. 9. Effect of iteration numbers (K_1, K_2) on BEMLMS DFE performance ($\mu = 2^{-1}$, $M = 6$, $V = 100$ km/h, $\tau = T$).

monitoring functions for the forward and backward equalization. Thus, the DFE has to wait until all of the subslot symbols are received.

Next, we compare the computational complexity of each algorithm (for processing one slot of symbols). This is shown in Table IV. In the table, N represents the tap length of the DFE, and K_1 and K_2 denote the iteration numbers in the training and tracking modes, respectively. Here, we assume that the standard RLS algorithm and LMS algorithms are used. In the bidirectional processing, additional divisions may be required for the calculation of the monitor function. Note that the bidirectional algorithms of [15] and [19] require 176 additional divisions for the control mechanism. The threshold-based BEMLMS algorithm requires L (number of blocks) additional divisions, while the comparison-based BEMLMS requires no extra divisions. It is clear that the computational complexity of the proposed algorithm greatly depends on the iteration numbers (K_1, K_2).

IV. Simulations

In this section, we report some simulation results demonstrating the effectiveness of the proposed algorithm. We first consider the adequacy of the theoretical results derived in Section II. The input signal $\{x(n)\}$ in Fig. 1 was set to be white Gaussian with unit power and the filter length N to be five. As we have shown, the filter weight vector error has two components: the fluctuation error caused by the observation noise and the lag error caused by the time-varying system. To see the individual effect, we designed two experiments. For the first one, we let the identified system be time-invariant and noisy; for the second one, we let the system be time-varying but noise free. Fig. 5 shows the simulation results for the first case. Each of the results corresponds to the average of 200 simulation trials. This figure gives the relation between the FEV power and the block size M. The variance of $\nu(n)$ was 0.1. The theoretical curves indicate that the larger the M, the smaller the FEV power we can obtain. The simulation curves also show this trend. We also see that the simulation results are always larger than the theoretical one. This is due to the approximations (the direct averaging method and the independence theory [28]) we used in the derivation. We assume that the input vectors are independent, which is not true in practice. Thus, many values are underestimated. The other thing we can observe is that the smaller the step size, the smaller the FEV power we can obtain. This is similar to the standard LMS algorithm. Next, we consider the second case. Each weight of $W_{\text{opt}}(n)$ is modeled as a random walk process. We let all the driving noises have the same power σ^2, which was 0.1. The simulation result is shown in Fig. 6. As we can see, the larger the block size, the higher the LEV power. We also see that the larger the step size μ, the better results we can obtain. The is again similar to the standard LMS algorithm. We also note that the theoretical results match the simulated ones quite well.

We then consider the performance of the proposed algorithm. We adopted the so-called two-ray Rayleigh-fading channel model according to the IS-136 standard recommendation. Each of the independent Rayleigh faders was generated by Jakes fading model [26], [27]. The carrier frequency of the simulation system was assumed to be 900 MHz. The DFE

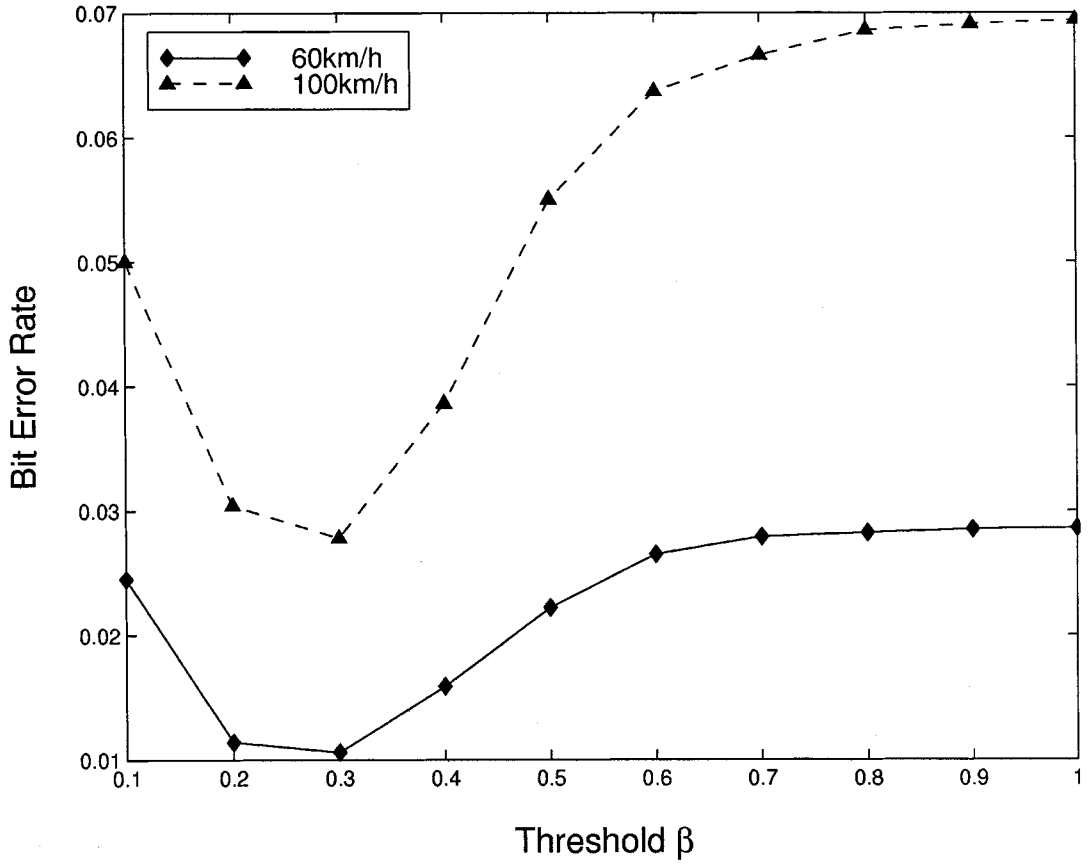

Fig. 10. Effect of the threshold value β on the BEMLMS DFE performance ($\mu = 2^{-1}$, $M = 6$, $(K_1, K_2) = (7, 1)$, $\tau = T$).

used in simulations is fractionally spaced. The feedforward filter consists of four taps with $T/2$ time space, where T is the symbol period and the feedback filter is one tap. Each of the simulation results is obtained from 1000 data slots. Fig. 7 demonstrates the tracking capabilities of the DFE adapted by the conventional LMS, EMLMS, BMLMS [19], and BEMLMS algorithms for 60 km/h mobile speed (denoted as V in the figure caption). Here, the delay for the second path of the two-ray model, denoted as τ, was set to T. In Fig. 7, the horizontal axis represents the symbol index and the vertical axis gives the symbol error rates (SERs). A high SER indicates that the probability of losing track of the channel is high. It is obvious that the conventional LMS algorithm cannot cope with the fast varying characteristics of the mobile channel. The EMLMS algorithm improves the convergence rate of the DFE and the SER in the second half of the slot. The BMLMS algorithm has a low SER around both ends of the slot. However, in the middle of the slot, the SER is still high. Only the BEMLMS algorithm has a low SER for the entire slot, indicating its superior tracking capability.

Since the BEMLMS is a block-iterated algorithm, the block size may have great influence on the final performance. Fig. 8 shows the BER versus the block size M. Here, we let $(K_1, K_2) = (7, 1)$. From Fig. 8, we find that $M = 6$ gives an optimal result. In the next experiment, we explored the effect of the iteration number pair (K_1, K_2). The result is shown in Fig. 9. It is seen that the equalization performance is improved as the values of (K_1, K_2) increase. However, their relationship is not linear. When K_1 is increased from one to five, a large improvement is obtained. However, the improvement is soon saturated when we further increase K_1. The increase of K_2 also helps, but the performance improvement is insignificant. Thus, we can conclude that the influence of K_1, which corresponds to the iteration number in the training mode, is much larger than that of K_2, which is the iteration number in the tracking mode. This is different from the EMLMS algorithm, where K_2 has a larger influence on the final result [20]. This result may be due to the proposed bidirectional equalization scheme, where the tracking period is effectively shortened. Thus, the iteration in the tracking mode does not produce a further advantage. Since the computational complexity is proportional to (K_1, K_2), some tradeoff must be made. Based on the above discussion, we suggest that K_1 be set to a value between five and nine and K_2 be set to one. As we mentioned, the threshold-based control mechanism requires a parameter β. Fig. 10 shows the BER results versus β. It is clear that β should be chosen around 0.3.

Fig. 11 shows the simulation results for the BEMLMS algorithm with the two control mechanisms at mobile speeds of 60 and 100 km/h (SNR = 22 dB, $\beta = 0.3, M = 6$). From this figure, we see that the comparison-based control mechanism performs better than the threshold-based one. However, the consequence is a longer processing delay. Finally, we compared the performance of various LMS-based DFE algorithms. This is shown in Fig. 12, which includes the BMLMS, the EMLMS, and the BEMLMS. We can see that the performance of the BEMLMS DFE with $(K_1, K_2) = (7, 1)$ at 100 km/h is similar

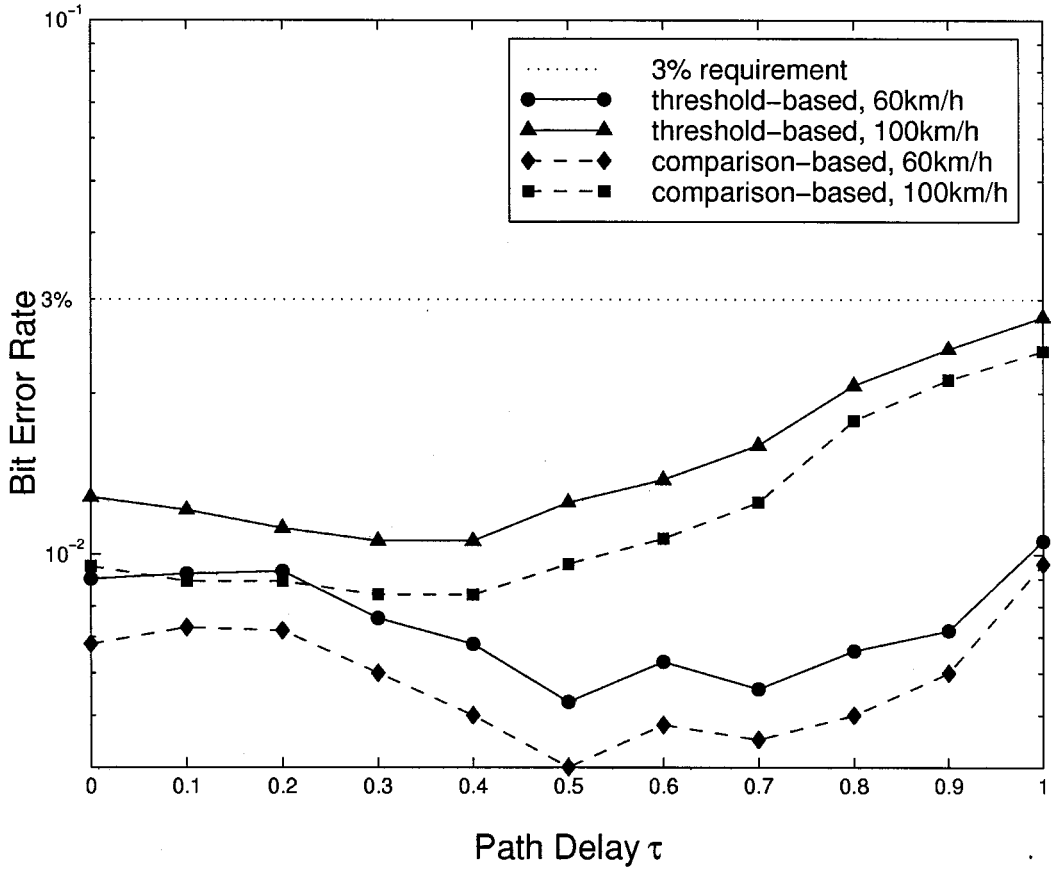

Fig. 11. Performance comparison of the threshold-based and comparison-based control mechanisms (BEMLMS, $\mu = 1/2$, $(K_1, K_2) = (7, 1)$).

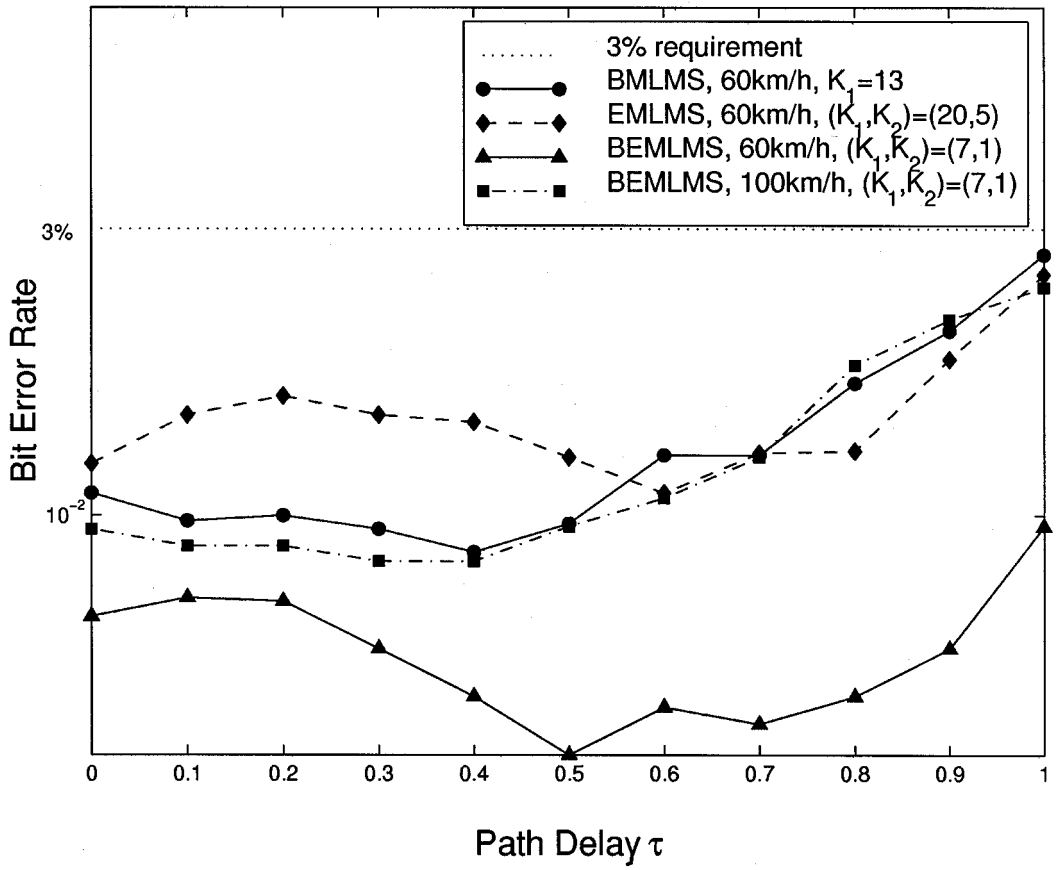

Fig. 12. Performance comparison of the BMLMS DFE, EMLMS DFE, and BEMLMS (with the comparison-based control algorithm) DFE ($\mu = 1/2$, SNR $= 22$ dB).

TABLE V
COMPUTATIONAL COMPLEXITY COMPARISON OF THE DFE ALGORITHMS DESIGNED FOR THE IS-136 SYSTEM ($N = 4$)

Algorithm	Complex Multiplications	R	Remarks
RLS [13]	13770	100%	
BRLS [15]	14960	109%	
BMLMS [19]	5632	41%	$K_1 = 13$
EMLMS	12210	89%	$(K_1, K_2) = (20, 5)$
BEMLMS	4642	34%	$(K_1, K_2) = (7, 1)$

to those of BMLMS DFE with $K_1 = 13$ and the EMLMS DFE with $(K_1, K_2) = (20, 5)$ at 60 km/h. The BER of BEMLMS DFE at 60 km/h is much lower than the others. We also compared the computational complexity of various DFE algorithms. The parameters of the LMS-based DFEs are those used in Fig. 12. Since the conventional adaptive algorithm uses the RLS algorithm, we used its complexity as a reference. Only complex multiplications were compared. We define a complexity ratio R as

$$R = \frac{\mathrm{MUL_{ALG}}}{\mathrm{MUL_{RLS}}} \tag{36}$$

where $\mathrm{MUL_{RLS}}$ denotes the complex multiplications required by the standard RLS DFE (to process one slot) and $\mathrm{MUL_{ALG}}$ denotes the complex multiplications required by the algorithm to be compared. The complexity ratio for various adaptive algorithms can be obtained using Table IV with the proper parameters substituted. The results are shown in Table V, from which we can see that the computational complexity of the BEMLMS algorithm is the smallest among all and is only one-third of the standard RLS algorithm.

V. CONCLUSION

In this paper, we propose a bidirectional LMS-based DFE for the North American IS-136 cellular radio system. The proposed algorithm combines an extended multiple training and bidirectional processing techniques. This combination enables the DFE to effectively equalize symbols distorted by the channel and at the same time maintains the low computational complexity of the LMS algorithm. The convergence properties of the MLMS algorithm are also analyzed. Simulation results show that the proposed DFE can be applied at vehicle speeds as high as 100 km/h and that the required computational complexity is only 34% of that for the standard RLS DFE. To achieve good performance at a high mobile speed, the conventional DFE usually has to use the RLS algorithm. For the first time, to the best of our knowledge, the LMS algorithm can be used to achieve a similar performance as the RLS algorithm. Due to the simple structure and low computational complexity, the proposed algorithm is very suitable for real-world implementation. As a matter of fact, a highly efficient application-specific IC design has been developed in [25].

REFERENCES

[1] J. G. Proakis, "Adaptive equalization for TDMA digital mobile radio," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 333–341, May 1991.

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

[3] G. Larsson, B. Gudmundson, and K. Raith, "Receiver performance for the north american digital cellular system," in *41st IEEE Vehicular Technology Conf.*, 1991, pp. 1–6.

[4] R. D. Koilpillai, S. Chennakesshu, and R. L. Toy, "Low complexity equalizers for U.S. digital cellular system," in *42nd IEEE Vehicular Technology Society Conf.*, vol. 2, 1992, pp. 744–747.

[5] H. Shiino, N. Yamaguchi, and Y. Shoji, "Performance of an adaptive maximum-likelihood receiver for fast fading multipath channel," in *42nd IEEE Vehicular Technology Society Conf.*, vol. 1, 1992, pp. 380–383.

[6] K. Okanoue, A. Ushirokawa, H. Tomita, and Y. Furuya, "Fast tracking adaptive MLSE for TDMA digital cellular systems," *IEICE Trans. Commun.*, vol. E77-B, no. 5, pp. 557–564, May 1994.

[7] W. P. Chou and P. J. Mclane, "16-state nonlinear equalizer for IS-54 digital cellular channels," *IEEE Trans. Veh. Technol.*, vol. 45, pp. 12–25, Feb. 1996.

[8] T. Larsson, A. Lozano, and G. Peponides, "Soft-output MLSE for IS-136 TDMA," in *Conf. Rec. ICUPC'97*, vol. 1, Oct. 1997, pp. 53–57.

[9] A. Lozano and G. Peponides, "Adaptive MLSE receiver for dual-band IS-136 TDMA," in *8th IEEE Int. Symp. Personal, Indoor and Mobile Radio Communications*, vol. 3, Sept. 1997, pp. 816–820.

[10] D. D. Falconer, A. U. H. Sheike, E. Eleftheriou, and M. Tobis, "Comparison of DFE and MLSE receiver performance of HF channels," *IEEE Trans. Commun.*, vol. COM-33, pp. 484–494, May 1985.

[11] S. Chennakeshu, A. Narasimhan, and J. B. Anderson, "Decision feedback equalization for digital cellular radio," in *IEEE Int. Conf. Communications ICC'90*, vol. 4, 1990, pp. 1492–1496.

[12] A. Narasimhan, S. Chennakeshu, and J. B. Anderson, "An adaptive lattice decision feedback equalizer for digital cellular radio," in *40th IEEE Vehicular Technology Conference*, 1990, pp. 662–667.

[13] T. Nakai, S. Ono, Y. Shimazaki, and N. Kondoh, "Adaptive equalizer for digital cellular radio," in *IEEE Vehicular Technology Conf.*, 1991, pp. 13–16.

[14] K. Jamal, G. Brismark, and B. Gudmundson, "Adaptive MLSE performance on the D-AMPS 1900 channel," *IEEE Trans. Veh. Technol.*, pp. 73–75, Aug. 1997.

[15] Y. J. Liu, M. Wallace, and J. W. Ketchum, "A soft-output bidirectional decision feedback equalization technique for TDMA cellular radio," *IEEE J. Select. Areas Commun.*, vol. 11, pp. 1034–1045, Sept. 1993.

[16] R. Sharma, W. D. Grover, and W. A. Krzymien, "Forward-error-control (FEC)-assisted adaptive equalization for digital cellular mobile radio," *IEEE Trans. Veh. Technol.*, vol. 42, pp. 94–102, Feb. 1993.

[17] M. Rupp and A. Bahai, "Training and tracking of adaptive DFE algorithms under IS-136," in *1st IEEE Signal Processing Workshop Signal Processing Advances in Wireless Communications*, 1997, pp. 341–344.

[18] J. F. Doherty and R. J. Mammone, "An adaptive algorithm for stable decision-feedback filtering," *IEEE Trans. Circuits Syst. II*, vol. 40, no. 1, pp. 1–9, Jan. 1993.

[19] C. H. Wei and B. N. Guo, "A bi-directional equalizer using multiple training LMS algorithm for TDMA digital cellular radio," in *Proc. Int. Symp. Communications*, Dec. 1995, pp. 1081–1085.

[20] W. R. Wu and Y. M. Tsuie, "An LMS-based decision feedback equalizer for digital cellular radio," in *Proc. 7th IEEE Int. Symp. PIMRC*, vol. 3, Oct. 1996, pp. 883–887.

[21] S. S. Lin and W. R. Wu, "The ASIC design of an LMS-based decision feedback equalizer for TDMA digital cellular radio," in *Proc. 7th IEEE Int. Symp. PIMRC*, vol. 3, Oct. 1996, pp. 218–221.

[22] W. R. Wu and Y. M. Tsuie, "A bi-directional LMS-based decision feedback equalizer for digital mobile radio," in *Proc. 1997 Int. Symp. Communications*, Hsinchu, Taiwan, 1997.

[23] Y. Li, J. H. Winters, and N. R. Sollenberger, "Spatial-temporal equalization fro IS-136 TDMA systems with rapid dispersive fading and cochannel interference," *IEEE Trans. Veh. Technol.*, pp. 1182–1194, July 1999.

[24] *TIA/EIA/Interim Standard IS 136*, Dec. 1994.

[25] C.-C. Chiang, "Digital Receiver Implementation for IS-54 Cellular Phone," Master Thesis, Institute of Electronic, National Chiao Tung University, Hsinchu, Taiwan, 1997.

[26] W. C. Jakes, *Microwave Mobile Communications*. New York: Wiley, 1974.

[27] P. Dent, G. E. Bottomley, and T. Croft, "Jakes fading model revisited," *Electron. Lett.*, vol. 29, no. 13, pp. 1612–1613, June 1993.

[28] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice Hall, 1996.

Wen-Rong Wu received the M.S. and Ph.D. degrees in electrical engineering from State University of New York at Buffalo in 1986 and 1989, respectively.

Since August 1898, he has been a Faculty Member in the Department of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C. His research interests include statistical signal processing and digital communications.

Yih-Ming Tsuie received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1993. Since 1995, he has been pursuing the Ph.D. degree in the Department of Communication Engineering, National Chiao Tung University, Hsinchu.

His research interests include adaptive signal processing and digital communications.