Pseudo-Maximum-Likelihood Data Estimation Algorithm and Its Application over Band-Limited Channels

Hamid R. Sadjadpour, *Senior Member, IEEE,* and Charles L. Weber, *Fellow, IEEE*

***Abstract*—A pseudo-maximum-likelihood data estimation (PML) algorithm for discrete channels with finite memory in additive white Gaussian noise environment is developed. Unlike the traditional methods that utilize the Viterbi algorithm (VA) for data sequence estimation, the PML algorithm offers an alternative solution to the problem. The simplified PML algorithm is introduced to reduce the computational complexity of the PML algorithm for channels with long impulse response. The adaptive version of the PML algorithm suitable for time-varying channels such as frequency-selective Rayleigh fading channels is also introduced. Computer simulation results demonstrate the performance of these algorithms and compare them to the VA-based techniques for different types of channels. The performance design criterion for the PML algorithm is derived in the Appendix.**

***Index Terms*—Fading channels, maximum-likelihood decoding, MLSE.**

I. Introduction

THE TERM blind equalization or deconvolution in communication and signal processing applications refers to recovering the input data sequence which is transmitted through a linear channel by knowing its output only. If the channel is known or a training sequence is transmitted at the beginning of the transmission, the equalizer utilizes this information to compensate for the intersymbol interference (ISI). There are several different criteria to design an equalizer. Equalization techniques developed from the maximum-likelihood (ML) criterion (see example in [1]) generally utilize maximum-likelihood sequence estimation (MLSE) at the receiver by using the Viterbi algorithm (VA) [2] for data sequence estimation. VA-based techniques estimate a *sequence* of data with a delay proportional to the length of the channel impulse response (CIR).

In time-varying environments such as mobile communications systems, adaptive MLSE receivers are required for adaptive channel identification and tracking. In traditional adaptive MLSE, adaptive channel estimation algorithms are based on the estimated data sequence derived from the VA with significant decision delay. For time-varying fading channels, the channel estimation performance degrades significantly due to this delay in the data estimation inherent in the VA. Adaptive channel estimation in the VA can be improved using tentative decision based on the best survivor path with small delay [15]. In this approach, the channel estimation is more accurate due to the small delay in the channel update. This technique will be referred to as “conventional” adaptive MLSE in this paper.

A new algorithm based on MLSE for an unknown channel is proposed in [10]. The joint channel and data estimation is based on per-survivor processing (PSP). Each state in the VA has a seperate CIR estimate that is based on the survivor path leading to that state. The CIR update in the PSP for each state is carried without any decision delay. This algorithm exhibits considerable improvement compared to the conventional adaptive MLSE while its computational complexity increases significantly.

Another approach is proposed based on the MLSE using a block of N symbols at a time. The metric computation for each block is similar to the VA. The survivor paths for all the states are traced back to detect a merge. If a merge occurs within the block, a decision is made based on the merged data sequence up to that time k. Then a new window of N data symbols is selected starting from time k. If a merge is not detected within the block, the N data symbols with the best survivor path is selected. The states in the VA are initialized based on the results from the previous block. The channel is updated once at the beginning of each block based on the survivor path from the previous block. This technique will be referred to as block sequence estimation (BSE) [12]. The variable f_{ex} in Table I is a function of merging delay inside each block.

We suggest a new algorithm (pseudo-maximum-likelihood [PML]) based on the ML criterion that does not utilize the VA. The PML algorithm is a symbol-by-symbol estimation algorithm with some delay and the CIR is estimated within the algorithm. The PML is suitable for time-invariant channels. Adaptive version of the PML algorithm (APML) is introduced for time-varying environments such as frequency-selective Rayleigh fading channels.

Throughout this paper, the CIR is modeled as a discrete finite impulse response (FIR) filter with length L. The discrete CIR can be thought of as the convolution of the transmitter filter, the physical channel, and the receiver filter. The received signal $(\{r_k\})$ is the convolution of the data sequence $(\{a_k\})$ and the channel coefficients $(\{h_k\})$ corrupted by a sequence of additive white Gaussian noise samples $(\{n_k\})$. The data is

Paper approved by E. Eleftheriou, the Editor for Equalization and Coding of the IEEE Communications Society. Manuscript received August 31, 1998; revised November 5, 1999 and May 25, 2000. This paper was presented in part at the Asilomar Conference on Signals, Systems, and Computers, November 1994, 1996, and 1998, and at the IEEE Military Communications Conference, San Diego, CA, November 1995.

H. R. Sadjadpour is with AT&T Research, Shannon Laboratory, Florham Park, NJ 07932 USA (e-mail: sadjadpour@research.att.com).

C. L. Weber is with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-2565 USA.

Publisher Item Identifier S 0090-6778(01)00263-X.

TABLE I
COMPARISON OF COMPLEX MULTIPLICATION PER DATA SYMBOL BETWEEN VA-BASED METHODS AND THE APML

		BSE	PSP	APML(RLS(0)) L=l=3,i=0	APML(RLS(2)) L=l=3,i=2
Metric update	N_{mul}	$L \times M^L \times f_{ex}$	$L \times M^L$	$\approx l \times M^l$	$L \times M^L$
CIR update (RLS)	N_{mul}	$8 \times L$	$8 \times L \times M^{L-1}$	$8 \times L \times M^i$	$8 \times L \times M^i$

assumed an independent identically distributed (i.i.d.) M-ary signal. The transmitted data sequence and noise are independent. The channel length L is either known or upper bounded. According to this model, r_k can be written as

$$r_k = \sum_{i=1}^{L} h_i a_{k-i+1} + n_k. \tag{1}$$

The above equation for discrete time model is based on a whitened matched filter (WMF) at the front-end receiver [13]. The WMF consists of a filter matched to the continuous time channel impulse response cascaded with a noise whitening filter. The match filter can be realized at the receiver when the continuous time channel $(h(t))$ is known. If $h(t)$ is unknown, the front-end WMF can be derived based on the assumption that the channel is characterized by a finite number of parameters such as the tapped-delay line model [14]. In this case, the front-end processor consists of a fractionally spaced processing filter matched to the known data pulse followed by a whitening filter.

There are no other restrictions on the nature of the channel, i.e., with or without zeros in its frequency response, linear or nonlinear phase. The transmitter and the receiver are assumed synchronous.

The paper is organized as follows. Section II describes the mathematical derivation of the PML algorithm briefly. Section III describes the SPML and APML algorithms. Computer simulation results are given in Section IV. The performance design criterion for the PML algorithm is presented in the Appendix. A summary and some suggestions for future work conclude the paper.

II. PML ALGORITHM

A. Derivation of the PML Algorithm

In vector form (1) becomes

$$\underline{\mathbf{r}}_N = A_N \underline{\mathbf{h}} + \underline{\mathbf{n}}_N \tag{2}$$

where $\underline{\mathbf{r}}_N = [\, r_{k+1} \ \cdots \ r_{k+N} \,]^t$, $\underline{\mathbf{n}}_N = [\, n_{k+1} \ \cdots \ n_{k+N} \,]^t$, $\underline{\mathbf{h}} = [\, h_1 \ \cdots \ h_L \,]^t$, and

$$A_N = \begin{bmatrix} a_{k+1} & a_k & \cdots & a_{k-L+2} \\ a_{k+2} & a_{k+1} & \cdots & a_{k-L+3} \\ \vdots & \vdots & \vdots & \\ a_{k+N} & a_{k+N-1} & \cdots & a_{k+N-L+1} \end{bmatrix}.$$

From (2), the conditional probability density function (pdf) of the received signal given A_N and $\underline{\mathbf{h}}$ is

$$f(\underline{\mathbf{r}}_N / A_N, \underline{\mathbf{h}}) = \frac{1}{(\sigma_n \sqrt{2\pi})^N} \exp\left[\frac{|\underline{\mathbf{r}}_N - A_N \underline{\mathbf{h}}|^2}{-2\sigma_n^2} \right] \tag{3}$$

where σ_n^2 is the variance of the noise. Let

$$C(A_N, h) = |\underline{\mathbf{r}}_N - A_N \underline{\mathbf{h}}|^2 \tag{4}$$

be the likelihood function. The minimization of this likelihood function with respect to $\underline{\mathbf{h}}$ given the data matrix A_N is a straightforward least squared estimation problem.

$$\frac{\partial C(A_N, \underline{\mathbf{h}})}{\partial \underline{\mathbf{h}}} = 2A_N^t \underline{\mathbf{r}}_N - 2A_N^t A_N \underline{\mathbf{h}} \tag{5}$$

where A_N^t is the transpose of A_N. If (5) is set to zero, the ML estimate of $\underline{\mathbf{h}}$ will be obtained.

$$\hat{\underline{\mathbf{h}}}_{\mathrm{ML}} = (A_N^t A_N)^{-1} A_N^t \underline{\mathbf{r}}_N \tag{6}$$

$(A_N^t A_N)^{-1} A_N^t$ is the pseudoinverse of A_N and even when the $A_N^t A_N$ is not full rank, the singular value decomposition (SVD) technique can be used to compute the pseudoinverse of A_N [9, ch. 11].

The new likelihood function is obtained by substituting $\hat{\underline{\mathbf{h}}}_{ML}$ into (4).

$$C(A_N, \hat{\underline{\mathbf{h}}}_{\mathrm{ML}}) = \left| \underline{\mathbf{r}}_N - A_N \left(A_N^t A_N \right)^{-1} A_N^t \underline{\mathbf{r}}_N \right|^2. \tag{7}$$

It can be shown [3] that minimization of (7) is equivalent to maximization of the following likelihood function:

$$C'(A_N) = \underline{\mathbf{r}}_N^t A_N \left(A_N^t A_N \right)^{-1} A_N^t \underline{\mathbf{r}}_N \tag{8}$$

$C'(A_N)$ is a function of the received signal (r_k) and the transmitted data sequence (a_k).

The following assumptions have been made for optimization of the likelihood function. The data sequence is known (or estimated) prior to time $k+N+1$ and our objective is to estimate the data from $k+N+1$ to $k+N+l$. At time k, the known (or estimated) data will be shown as $\hat{a}_k$ and the transmitted data as a_k. Matrix A at $k+N+l$ can be written as

$$A_{N+l} = \begin{bmatrix} A_N \\ \text{- - - - - - - - - - - -} \\ \underline{\mathbf{a}}_{N+l}^t \end{bmatrix}$$

where A_N is the same as before with all its elements are known (or estimated) and $\mathbf{a}_{N+l}^t$ is shown in the equation at the bottom of the page. If $\underline{\mathbf{r}}_{N+l}$ is defined as

$$\underline{\mathbf{r}}_{N+l} = \begin{bmatrix} \underline{\mathbf{r}}_N \\ --- \\ \underline{\mathbf{r}}_l \end{bmatrix}$$

where $\underline{\mathbf{r}}_N$ is the same as before and $\underline{\mathbf{r}}_l = [r_{k+N+1} \quad \cdots \quad r_{k+N+l}]^t$, then $C'(A_{N+l})$ is given by

$$C'(A_{N+l}) = \underline{\mathbf{r}}_{N+l}^t A_{N+l} (A_{N+l}^t A_{N+l})^{-1} A_{N+l}^t \underline{\mathbf{r}}_{N+l}. \tag{9}$$

Our objective is to maximize $C'(A_{N+l})$. $C'(A_{N+l})$ can be expressed recursively. The details of this derivation is in [3] and the results are given here.

$$C'(A_{N+l}) = C'(A_N) + \Delta(\underline{\mathbf{k}}_1, \mathbf{k}_2) \tag{10}$$

where $\underline{\mathbf{k}}_1$ and $\mathbf{k}_2$ are defined as

$$\underline{\mathbf{k}}_1 = \mathbf{a}_{N+l}^t (A_N^t A_N)^{-1} A_N^t \underline{\mathbf{r}}_N \tag{11}$$

$$\mathbf{k}_2 = \mathbf{a}_{N+l}^t (A_N^t A_N)^{-1} \mathbf{a}_{N+l} \tag{12}$$

and $\Delta(\underline{\mathbf{k}}_1, \mathbf{k}_2)$ is given by

$$\begin{aligned} \Delta(\underline{\mathbf{k}}_1, \mathbf{k}_2) = & -\underline{\mathbf{k}}_1^t (\mathbf{k}_2 + I_l)^{-1} \underline{\mathbf{k}}_1 + 2\underline{\mathbf{r}}_l^t \underline{\mathbf{k}}_1 + \underline{\mathbf{r}}_l^t \mathbf{k}_2 \underline{\mathbf{r}}_l \\ & - \underline{\mathbf{r}}_l^t \mathbf{k}_2 (\mathbf{k}_2 + I_l)^{-1} \mathbf{k}_2 \underline{\mathbf{r}}_l - 2\underline{\mathbf{r}}_l^t \mathbf{k}_2 (\mathbf{k}_2 + I_l)^{-1} \underline{\mathbf{k}}_1 \end{aligned} \tag{13}$$

where I_l is the $l \times l$ identity matrix. Equation (10) is a recursive equation and in order to maximize $C'(A_{N+l})$ at each time period, $\Delta(\underline{\mathbf{k}}_1, \mathbf{k}_2)$ should be maximized. It is shown in [4] that $\Delta(\underline{\mathbf{k}}_1, \mathbf{k}_2)$ is maximized when

$$\underline{\mathbf{k}}_1 = \underline{\mathbf{r}}_l. \tag{14}$$

It can be proved that (14) provides the global maximum[1] of $\Delta(\underline{\mathbf{k}}_1, \mathbf{k}_2)$ assuming that $\underline{\mathbf{k}}_1$ can take on an arbitrary continuum of values through $\mathbf{a}_{N+l}$ [4]. $\underline{\mathbf{k}}_1$ can be also written as

$$\underline{\mathbf{k}}_1 = \mathbf{a}_{N+l}^t \hat{\underline{\mathbf{h}}}_{\mathrm{ML}} \tag{15}$$

where $\hat{\underline{\mathbf{h}}}_{\mathrm{ML}}$ is defined in (6).

B. PML Algorithm

1) At the beginning, create the matrix A_N. To create this matrix, $N+L-1$ data is needed. This data is obtained from either a training sequence or a blind initialization algorithm (see Section II-D). Then compute the $\hat{\underline{\mathbf{h}}}_{\mathrm{ML}}$ vector.
2) There are M^l possible values for $\mathbf{a}_{N+l}^t$ matrix or equivalently $\underline{\mathbf{k}}_1$. Multiply each candidate $\mathbf{a}_{N+l}^t$ by $\hat{\underline{\mathbf{h}}}_{\mathrm{ML}}$ and find the minimum Euclidean distance (D) between the vectors $\underline{\mathbf{k}}_1$ and $\underline{\mathbf{r}}_l$. This distance is defined as
$$D = |\underline{\mathbf{r}}_l - \underline{\mathbf{k}}_1|^2 = (\underline{\mathbf{r}}_l - \underline{\mathbf{k}}_1)^t (\underline{\mathbf{r}}_l - \underline{\mathbf{k}}_1). \tag{16}$$
3) If the best data sequence of length l which has the minimum distance D between $\underline{\mathbf{k}}_1$ and $\underline{\mathbf{r}}_l$ is defined as $(a'_{k+N+1} \quad \cdots \quad a'_{k+N+l})$, then $\hat{a}_{k+N+1} = a'_{k+N+1}$. The remaining estimated data sequence $(a'_{k+N+2} \quad \cdots \quad a'_{k+N+l})$ will be discarded. The minimum Euclidean distance computations in Step 3) can be used for the next iteration (see Notes below).
4) Set $k = k+1$. Return to Step 2) and continue the algorithm to estimate the next data.

Notes:

1) The PML algorithm is defined for time-invariant channels. Therefore, when the ML estimate of the CIR ($\hat{\underline{\mathbf{h}}}_{\mathrm{ML}}$) is computed, it can be used for the entire data sequence. If the ML estimate of the CIR needs to be updated, we can use Greville's theorem to recursively compute the pseudoinverse of A_N [16]. For time-varying channels, the APML algorithm recommends more practical and computationally efficient algorithms.
2) This algorithm is general and can be applied to both binary signal set or M-ary complex data. In the case that the data sequence is complex, $\hat{\underline{\mathbf{h}}}_{\mathrm{ML}}$ is a complex vector.
3) In order to minimize the computational complexity for the PML algorithm, the following procedure can be utilized for Steps 2) and 3). Each ith row of $\mathbf{a}_{N+l}^t$ is multiplied by $\hat{\underline{\mathbf{h}}}_{\mathrm{ML}}$ and its squared distance from the ith element in $\underline{\mathbf{r}}_l$ is computed and stored. There will be a total of $\sum_{i=1}^{l} M^i = (M(M^l - 1))/(M-1)$ positive real numbers. Adding these values in the right order will provide all possible values of D. After a_{k+N+1} is estimated, only $M(M^{l-1} - 1)/(M-1)$ possible values of these real numbers are required to be stored for the next time interval. In the next time interval, we only need to compute the vector multiplications of the last row of the new $\mathbf{a}_{N+l}^t$ matrix by $\hat{\underline{\mathbf{h}}}_{\mathrm{ML}}$ and compute its squared distance from the last element of the new vector $\underline{\mathbf{r}}_l$. When $l \leq L$, the computational complexity is proportional to M^l and if $l > L$, the computational complexity is proportional to M^L. The memory requirements is always proportional to $M(M^l - 1)/(M-1)$.
4) The performance of the PML algorithm depends on the accurate channel estimate and correct choice for l. $\hat{\underline{\mathbf{h}}}_{\mathrm{ML}}$ estimation can be improved by choosing a large value for N. In order to evaluate the performance of the PML, a lower bound for the mean value of D is derived in the Appendix. If the average minimum D computed in practice is much larger than this bound, then this is an indication

$$\mathbf{a}_{N+l}^t = \begin{bmatrix} a_{k+N+1} & \hat{a}_{k+N} & \cdots & & \cdots & \hat{a}_{k+N-L+2} \\ \vdots & \cdots & & & \cdots & \vdots \\ a_{k+N+l} & \cdots & a_{k+N+1} & \hat{a}_{k+N} & \cdots & \hat{a}_{k+N+l-L+1} \end{bmatrix}$$

[1] We cannot infer from this development that the ML demodulation will result, since the transmitted data sequence must be constrained to discrete values.

that the PML does not perform well. A new CIR estimate and/or an increase in l is required to improve the performance of the PML algorithm.

5) The PML algorithm as described above is similar to tree search algorithms. In [18], a tree search algorithm similar to the PML is proposed.

C. Intuitive Discussion

The above condition (14) has a significant physical interpretation. When each row of $\mathbf{a}_{N+l}^t$ is multiplied by $\hat{\underline{\mathbf{h}}}_{\mathrm{ML}}$, this is equivalent to the convolution of the transmitted data sequence with the estimated channel coefficients. Obviously, we expect this value to be equal to the received signal at that time period. The received signals are the elements of $\underline{r}_l$ vector. If $l = 1$, then PML is equivalent to a decision-feedback equalizer (DFE) [5]. DFE is an example of symbol-by-symbol detection, where in each time interval, the ISI is subtracted from the received signal and the output is fed back to a slicer which makes the decision about the current symbol. This is exactly the case for the PML algorithm when $l = 1$. When $l \gg L$, then performance of the PML is equivalent to that of the VA. Obviously, any value of l in the middle of these two values will provide a performance of the PML which will be between the performance of the DFE and the VA. Section III will discuss some techniques to reduce the computational complexity of the PML for channels with long impulse response and also application of this algorithm to time-variant systems.

D. Blind Initialization

To start the PML algorithm, $N + L - 1$ data is needed (N is the number of rows in A_N and L is the length of the CIR). If a global search is conducted, there are M^{N+L-1} possible values. The global search will find the best data sequence of length $N + L - 1$ that optimizes (7) or (8). Instead of a global search, we can divide this data sequence into S subsets and try to optimize (7) or (8) by utilizing alternating minimization procedure [6] with some modifications to the original technique. By using this approach, the computational complexity will be reduced from M^{N+L-1} in the global search to $M^{(N+L-1)/S}$. The alternating minimization procedure is a numerical solution for extremum problems arising in information theory and statistics.

Let $d(a, b)$ be a real valued function of two variables $a \in A$ and $b \in B$ where $d(a, b)$ is the Euclidean distance between a and b, and A and B are two convex sets. Let's choose an arbitrary element $b_1 \in B$, and find $a_1 \in A$ which has the closest distance to b_1. Then fix $a = a_1$, and find $b_2 \in B$ which has the closest distance to a_1. If this procedure continues alternatively for a_i and b_i ($i = 1, 2, \cdots$), it is proven in [6] that this method converges to the minimum distance in these two sets, provided that the sets are convex and the distance satisfies certain conditions.

In our problem, it is unknown whether the condition for reaching the global optimum is satisfied. We still choose to apply the alternating minimization procedure as follows.

1) Start with an initial value for the data sequence of length $N + L - 1$.

2) Randomly divide the data sequence of length $N + L - 1$ into S subsets. For example, suppose $N + L - 1 = 10$ and $S = 2$. In this step, we can have two subsets as $\{a_1, a_2, a_3, a_4, a_5\}$ and $\{a_6, a_7, a_8, a_9, a_{10}\}$.

3) Apply the alternating minimization procedure for the jth set while keeping the remaining $(S - 1)$ sets fixed ($j = 1, 2, \cdots, S$). Equation (7) or (8) is used for optimization of the likelihood function. Continue this method until the data sequence of length $N + L - 1$ converges.

4) Based on the converged data sequence in Step 3) above, regroup (in an arbitrary manner) the converged data sequence of length $N + L - 1$ into S, randomly indexed, new data sets again of the same length. Continuing on the example in Step 2), we can have the new two subsets as $\{a_1, a_3, a_5, a_7, a_9\}$ and $\{a_2, a_4, a_6, a_8, a_{10}\}$.

5) Return to Step 3) and continue this procedure until the optimum result for the new S data sets is the same as the previous data sets.

When the alternating minimization procedure is applied to each data set, it does not necessarily converge to the global optimum. However, after it converges to a local optimum, by rearranging the data sequence into a new subsets of data in Step 4, the alternating minimization procedure will converge to a new local optimum which is closer to the global optimum than the previous one if the new iteration has different result from the previous one. The reason is because the previous local optimum is the starting point for the new iteration.

In general, $N + L - 1$ data is needed to start the algorithm. This can be provided by using the blind initialization procedure or using a training sequence.

III. Application of the PML Algorithm

This section describes two versions of the PML algorithm. The SPML algorithm is a suboptimal approach with less computational complexity than the PML algorithm. This algorithm does not search globally among M^l different possible values of $\mathbf{a}_{N+l}^t$ and it is suitable for systems with long CIR and/or large alphabet size of the signal set. It has been shown by simulation that this algorithm can perform close to the PML algorithm with less computational complexity.

The APML algorithm is designed for time-variant channels. This algorithm addresses different issues related to the effective estimation of the CIR specially for time-variant channels.

A. SPML Algorithm

The purpose of the SPML algorithm is to reduce the computational complexity of the PML algorithm. Three techniques are introduced here.

1) Using the Previous Estimated Data Symbols: The PML algorithm performs a global search to find the best data symbols of length l, selects only the first data and discards the remaining $l - 1$ data symbols. In the SPML algorithm, the remaining $l - 1$ estimated data symbols will be used in the next iteration. Suppose at time $k + N + 1$, l data are estimated in the time interval $[k+N+1, k+N+l]$. The first data is selected as the estimated data symbol at time $k + N + 1$, i.e., $\hat{a}_{k+N+1}$. The last J data in the interval $[k+N+l-J+1, k+N+l]$ will be transferred to the next time period. By transferring the last J data to the next time period, the number of global search is reduced from M^l to M^{l-J} and consequently, the number of operations per symbol is now proportional to M^{l-J}.

The rational to use the last J data is as follows. The Euclidean distance (D) between $\mathbf{a}^t_{k+N+l}\hat{\underline{\mathbf{h}}}_{\mathrm{ML}}$ and $\underline{\mathbf{r}}'_l$ is defined as

$$\begin{aligned} D &= |\mathbf{a}^t_{k+N+l}\hat{\underline{\mathbf{h}}}_{\mathrm{ML}} - \underline{\mathbf{r}}'_l|^2 \\ &= (\mathbf{a}^t_{k+N+l}\hat{\underline{\mathbf{h}}}_{\mathrm{ML}} - \underline{\mathbf{r}}'_l)^t(\mathbf{a}^t_{k+N+l}\hat{\underline{\mathbf{h}}}_{\mathrm{ML}} - \underline{\mathbf{r}}'_l). \end{aligned} \tag{17}$$

In (17), a_{k+N+l} is just in the last row of $\mathbf{a}^t_{N+l}$. $a_{k+N+l-1}$ is in the last two rows of $\mathbf{a}^t_{k+N+l}$ and finally, a_{k+N+1} is in all the rows of $\mathbf{a}^t_{k+N+l}$. Obviously, the effect of a_{k+N+l} is less than the effect of a_{k+N+1} in the computation of D. When the last J estimated data symbols are transferred to the next iteration, poor estimation of these J data symbols will not have significant effect in the computation of D and consequently, will not degrade the performance considerably. On the other hand, any mistake in estimating a_{k+N+1} has a significant effect on the value of D. This phenomenon is very similar to weighting the data symbols between time $k+N+1$ to $k+N+l$. The recent data (a_{k+N+1}) is weighted more heavily than the future data (a_{k+N+2} to a_{k+N+l}). In general, if the power of the CIR for the first $J+1$ taps is smaller than the total power of the CIR $((\sum_{i=1}^{J+1} h_i^2/\sum_{i=1}^{L} h_i^2) \ll 1)$, then this technique will not degrade the performance of the PML algorithm considerably. This technique is not very effective for channels with short length of memory. Because if L is small, a_{k+N+l} has almost the same effect as a_{k+N+1} and poor estimation of a_{k+N+l} will have a significant effect on the value of D and consequently, on the symbol-error rate (SER) performance. Notice that after each symbol period, one of the J estimated data symbols will be updated. Therefore, after J symbol period, the previous estimated J data symbols are updated with a new set. On the other hand, the data in the time interval $[k+N+2, k+N+l-J]$ and $k+N+l+1$ are updated at each symbol period while the remaining last J data from time $k+N+l-J+1$ to $k+N+l$ are updated between every two symbol period ($2T_s$, T_s is the symbol period) for $a_{k+N+l-J+1}$ to every J symbol period for a_{k+N+l} depending on their effect on the value of D.

We can also compare this technique to delayed decision-feedback sequence estimator (DDFSE) algorithm. In DDFSE [8], part of the past data is fed back and subtracted from the received signal as the DFE section of the equalizer. In SPML, the future decoded data are transferred to the next time interval to avoid global search.

2) Alternating Minimization Procedure Technique: After transferring J data to the next time interval, there are still $l-J-1$ decoded data symbols available that can be utilized in the next time interval. Instead of searching among M^{l-J} different data symbols, this $l-J$ data can be divided into S sets. Then the alternating minimization procedure [6] will be applied to these sets. $l-J-1$ estimated data from the previous time interval will be used as the initial value to start the alternating minimization procedure. Since this procedure does not start from an arbitrary value for the data symbols of length $l-J-1$, this method can converge very fast. The number of operations per symbol is now proportional to $M^{l-J/S}$. It is important that after $l-J$ data are divided into S subsets, the first search is applied to the most future data subset and the last search is applied to the most recent data subset which includes a_{k+N+1}.

3) Estimating More Than One Data: When l data are estimated in the PML algorithm, we can select more than one data at each time.

The vector multiplications in the PML and SPML algorithms can be computed simultaneously by utilizing parallel processing.

B. APML Algorithm

In some practical applications such as mobile radio communications environment, the CIR varies with time. Mobile radio communications usually occur between a fixed base station (BS) and several mobile stations (MSs). When the BS transmits a signal, the MS does not necessarily receive the signal from one path. The original signal from the BS is reflected and diffracted by buildings and other materials in urban or rural areas. Each received signal by the MS has a different path and, consequently, a different time-delay, amplitude, phase, and Doppler shift. If the symbol rate is high, the spreading of each symbol into the future symbols can cause intersymbol interference (ISI). This section will introduce an adaptive version of the PML algorithm (APML) which can reduce the effect of the ISI for time-variant channels.

In the APML algorithm, the CIR estimation is no longer restricted to ML estimate and any adaptive algorithm such as least mean square (LMS) or recursive least-square (RLS) [9] for channel estimation can be utilized. When the CIR is estimated by some adaptive algorithm, then the rows of $\mathbf{a}^t_{k+l}$ are multiplied by the CIR($\hat{\underline{\mathbf{h}}}(k)$). This multiplication is equivalent to the convolution of the transmitted data sequence in the time interval $[k+1,\ k+l]$ with the estimated CIR ($\hat{\underline{\mathbf{h}}}(k)$) at time k. If the CIR is time-variant, after each time interval T_s, the CIR can change considerably. Therefore, convolving the same CIR for the entire time interval $[k+1,\ k+l]$ is not an accurate assumption. On the other hand, if an update of the CIR is used at each time interval T_s, then the performance of the APML algorithm will improve. Suppose the CIR is known at time k. In order to update the CIR for the time interval $[k+1,\ k+l]$, we need to know the transmitted data sequence in this interval. At time k, however, the transmitted data sequence is unknown for the interval $[k+1,\ k+l]$. Since the data sequence is discrete, there are M possibilities for each transmitted data symbol.

In the APML algorithm, the estimated CIR at time k, i.e., $\hat{\underline{\mathbf{h}}}(k)$, is only multiplied by the first row of the data matrix, $\mathbf{a}^t_{k+l}$. At time k, a_k in unknown and for a M-ary signal set, there are M possible values for a_k. Each possible value of a_k corresponds to a unique CIR estimate at time $k+1$ ($\hat{\underline{\mathbf{h}}}(k+1)$). Each $\hat{\underline{\mathbf{h}}}(k+1)$ is multiplied by some of the candidates for the second row of the data matrix $\mathbf{a}^t_{k+l}$ depending on the value of the data symbol a_k which corresponds to that CIR estimate at time $k+1$.

This procedure can continue for all the rows of the data matrix $\mathbf{a}^t_{k+l}$ by estimating the CIR in the time interval $[k+1,\ k+l]$, namely, $\hat{\underline{\mathbf{h}}}(k+1), \cdots, \hat{\underline{\mathbf{h}}}(k+l-1)$. Notice that the CIR estimate

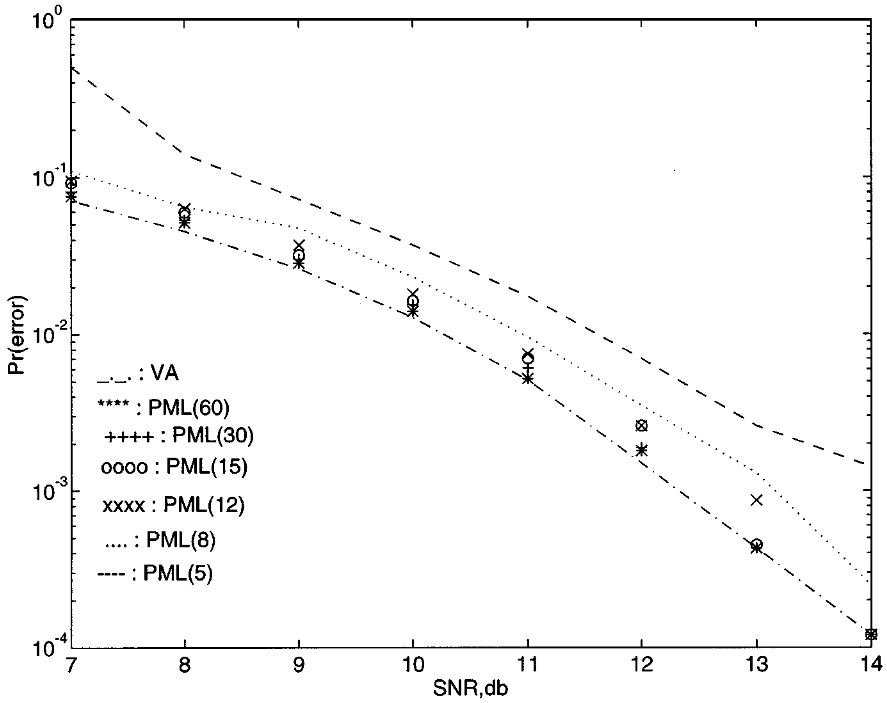

Fig. 1. Performance of PML for channel a and $l = 9$ and $\alpha = 5, 8, 12, 15, 30,$ and 60.

$\hat{\underline{\mathbf{h}}}(k+i)$ is multiplied by the $(i+1)$th row of the data matrix $\mathbf{a}_{k+l}^t$.

The CIR estimation can be calculated for $\hat{\underline{\mathbf{h}}}(k), \hat{\underline{\mathbf{h}}}(k+1), \cdots, \hat{\underline{\mathbf{h}}}(k+i)$ where $0 \le i \le l-1$. Selecting a larger number for i will improve the performance as well as increase the computational complexity of the algorithm. In general, $\hat{\underline{\mathbf{h}}}(k+i)$ has M^i different values. Suppose i is chosen smaller than $l-1$. Then $\hat{\underline{\mathbf{h}}}(k+i)$ is not only multiplied by the $(i+1)$th row of $\mathbf{a}_{N+l}^t$, but also by the remaining rows of $\mathbf{a}_{N+l}^t$, namely, $i+2, \cdots, l$.

This technique is similar to per-survivor processing (PSP) [10] algorithm for using more than one CIR estimate. For a CIR with length L in a M-ary signal set, PSP estimates M^{L-1} CIR's corresponding to all of the states in the VA and their survivor paths leading to those states. In the APML algorithm, there are M^i CIR estimates.

The APML algorithm can be computed similar to the PML (Section II-B, Note 3). Each nth row of $\mathbf{a}_{N+l}^t$ is multiplied by the appropriate CIR estimate ($\hat{\underline{\mathbf{h}}}(k+n-1)$) and its squared distance from the n-th element of $\underline{\mathbf{r}}_l$ is computed. For any value of i, all these computations up to the $(i+1)$th row can be transferred to the next time interval. Specifically for $l = L$ and $i = l-1$, at each time interval, we only need to compute all M^{L-1} possible channel estimates for $\hat{\underline{\mathbf{h}}}(k+i)$. Then multiply these CIR estimates by the last row of the $\mathbf{a}_{N+l}^t$ matrix and compute their squared distance from the last element of the vector $\underline{\mathbf{r}}_l$. In this special case, the computational complexity of the APML is equivalent to that of the PSP (M^{L-1} channel estimation and M^L squared distance computations).

Suppose the CIR is estimated in the APML by utilizing the LMS algorithm and $i = 3$. Then APML[LMS(3)] is the notation for expressing that the LMS algorithm is used for the CIR estimation and for all possible values of $\hat{\underline{\mathbf{h}}}(k+1), \hat{\underline{\mathbf{h}}}(k+2)$, and $\hat{\underline{\mathbf{h}}}(k+3)$ based on $\hat{\underline{\mathbf{h}}}(k)$.

IV. Experimental Results

The PML and SPML algorithms were simulated for the binary signal set and their BER performance are derived for two channels. The first channel, $\underline{\mathbf{h}}_a = [0.407, 0.815, 0.407]$, has a deep null in its frequency response but no phase distortion. The CIR for the second channel is $\underline{\mathbf{h}}_b = [0.04, -0.05, 0.07, -0.21, -0.5, 0.72, 0.36, 0.0, 0.21, -0.5, 0.72, 0.36, 0.0, 0.21, 0.03, 0.07]$. This channel has a relatively flat amplitude spectrum but it has phase distortion [11]. α is defined as $\alpha = (N/L)$. Fig. 1 compares the performance of the optimum receiver (VA with known CIR) to the performance of the PML algorithm for channel "a" and different values of α and constant $l(l = 9)$. If α increases, it results a more accurate CIR estimation and consequently, the performance of the PML algorithm improves. Fig. 2 compares the performance of the optimum receiver to the performance of the PML algorithm for channel "a" and different values of l and constant $\alpha(\alpha = 15)$. From Figs. 1 and 2 it can be concluded that increasing α and/or l will improve the performance of the PML. The SPML algorithm is simulated for the second channel when $l = L = 11$, $J = 3, S = 2$ and alternating minimization procedure is applied only once for all sets in each time interval. Fig. 3 compares BER performance of the SPML to that of the PML and the VA under the assumption that the channel "b" is known. The number of operations per bit in the SPML for this example is approximately 680 compared to 22 528 operations per bit for the PML while its BER performance is approximately 0.5 dB worse than the PML for this example. Memory requirements for the SPML is proportional to 176 while for the PML, it is proportional to approximately 4096. The result demonstrates the advantage of the SPML over the PML in terms of computational complexity and memory requirements.

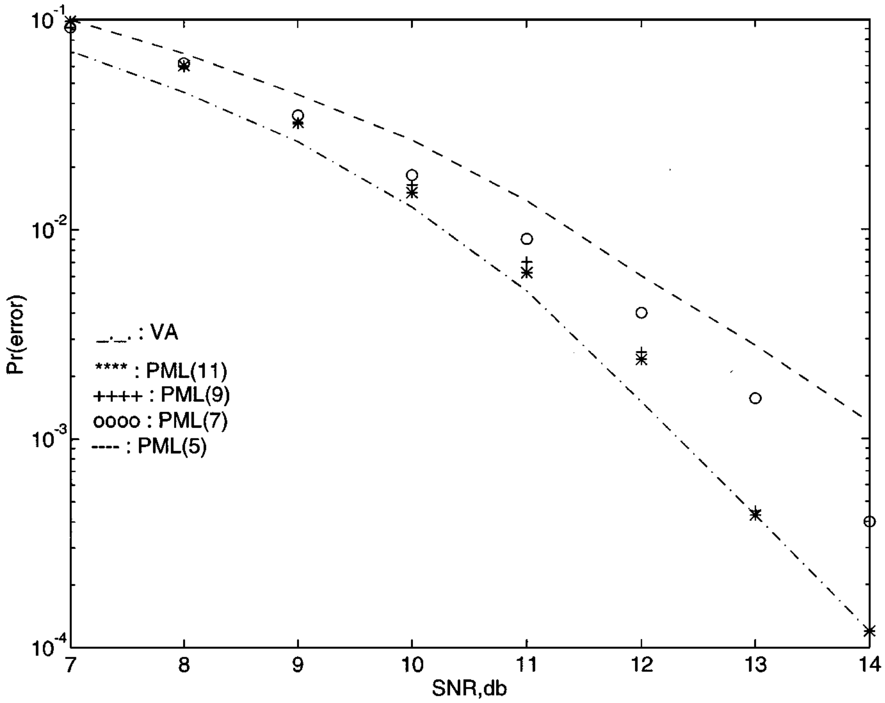

Fig. 2. Performance of PML for channel a and $\alpha = 15$ and $l = 5, 7, 9,$ and 11.

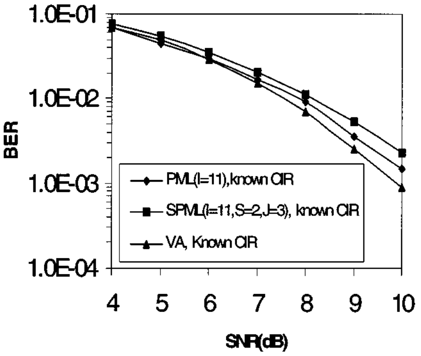

Fig. 3. Comparison between the PML, SPML, and the VA.

The APML algorithm is simulated for a mobile communication channel in a Rayleigh fading environment. The time-division multiple-access (TDMA) scheme based on the North American Dual-Mode Cellular System (IS-54) is applied to a three-tap CIR (L=3). The elements of the CIR ($h_1(k), h_2(k), h_3(k)$) have equal power $\sigma^2_{h_i} = 1/3$ for $i = 1, 2, 3$. This corresponds to a fading environment where there is no line of sight. Each tap of the CIR is generated by passing a white complex Gaussian noise generator through a Doppler filter. The Doppler filter is a second order Butterworth digital filter whose cut-off frequency f_d is determined by

$$f_d = T_s f_c \frac{v}{v_c} \tag{18}$$

where T_s, f_c, v, and v_c are the symbol period, the carrier frequency of the transmitted data sequence, the speed of the MS with respect to the BS, and the speed of light, respectively.

The definition of the signal-to-noise ratio (SNR) for time-variant channels is

$$\mathrm{SNR} = 10 \times \log\left(\frac{\sigma_a^2}{\sigma_n^2} \sum_{i=1}^{L} \mathcal{E}(|h_i|^2)\right) \tag{19}$$

where σ_a^2 is the variance of the input data. For time-invariant channels, the expected value in the summation will not be used.

For this channel, a vehicle speed of v=150 mi/h is selected. The normalized Doppler frequency (f_d) corresponding to this speed is 0.0085 if the carrier frequency is 900 MHz. The equivalent Doppler frequency is 206.5 Hz

The APML algorithm is compared with three different VA-based algorithms. These techniques are PSP [10], BSE [12], and "conventional" MLSE [15]. The computer simulation results of these three VA-based algorithms are obtained from [12].

The modulation scheme of QPSK is chosen for the data. The input alphabet is $\{1 + j, -1 + j, -1 - j, +1 - j\}$, where j denotes the imaginary unit.

The simulation results for PSP, BSE, and conventional MLSE are based on a decision delay of 5 L. The optimum block length for BSE was found to be 11 via simulations. The optimum delay in the CIR estimation for conventional MSLE to achieve the best performance is 4 for these examples. The weighting factor in RLS is optimized for all these techniques in each example to achieve the best possible performance.

When $v = 150$ mi/h, the APML(RLS(0)) and APML(RLS(2)) are simulated for $l = 3$. For all the simulations, the SER was computed over a time interval of 7200 consecutive TDMA time slots, each with the structure of IS-54 time slot for the BS to the MS. For simplicity, only the training sequence of length 28 symbols and the user information of

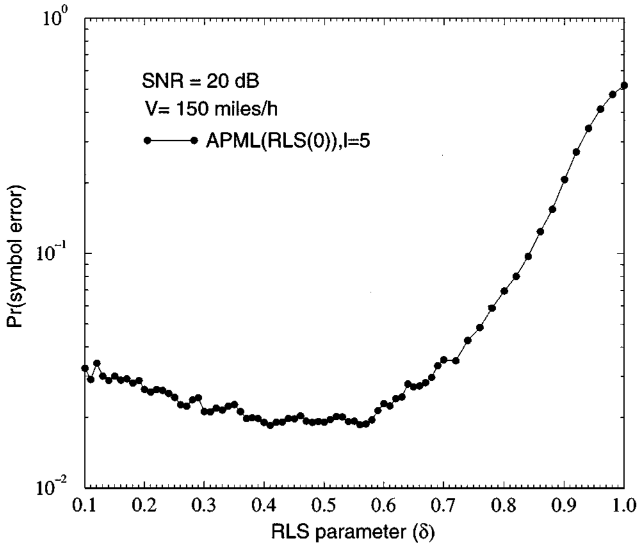

Fig. 4. APML(RLS(0)) versus δ on mobile channel ($f_d = 0.0085$).

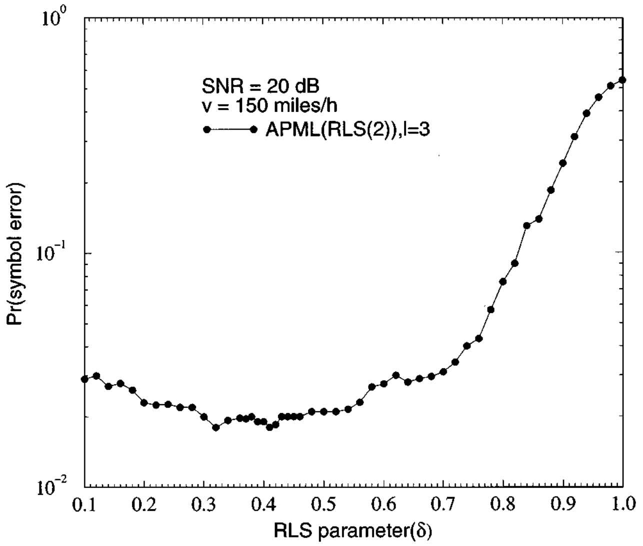

Fig. 5. APML(RLS(2)) versus δ on mobile channel ($f_d = 0.0085$).

260 symbols inside each time slot are considered [19]. At the beginning of each time slot, an initial vector of $[1\ \ 0\ \ 0]^t$ is chosen for the CIR coefficients. Then the training sequence is utilized to estimate the CIR using the RLS algorithm.

Figs. 4 and 5 illustrate the SER performance of the APML(RLS(0)) and APML(RLS(2)) when the RLS weighing factor (δ) varies between 0.1 and 1.0 at SNR = 20 dB. These figures demonstrate that the APML algorithm is not very sensitive to variations of the weighting factor and there is an acceptable range for choosing δ in order to obtain a reasonable SER performance. Fig. 6 compares the SER performance of the APML(RLS(0)) and APML(RLS(2)) algorithms to that of the PSP, BSE, and "conventional" MLSE algorithms. APML(RLS(2)) performs better than the VA-based algorithms. This figure also illustrates the importance of the right choice for l and i in the APML algorithm. If the channel is varying rapidly with time, then i should be selected close to l, i.e., $i = l - 1$.

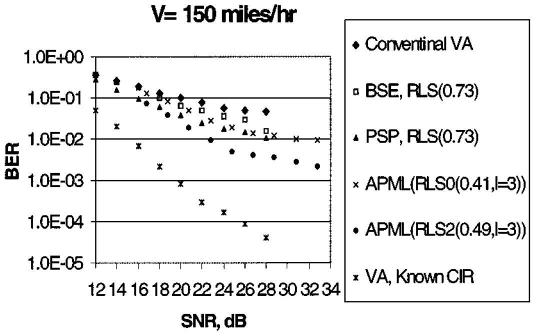

Fig. 6. Simulation performance of APML on mobile channel ($f_d = 0.0085$).

Otherwise, the performance of the APML algorithm degrades considerably. The reason is that the CIR estimate used in the APML does not represent the accurate channel estimation due to small choice for i compared to l.

Table I illustrates the computational complexity of these algorithms in terms of the number of multiplications (N_{mul}). The complexity is computed for a CIR of length L and a symbol alphabet size of M. In general, the APML algorithm is as complex as the PSP algorithm when $L = l$ and $i = l - 1$. In this particular example, APML(RLS(2)) has the same computational complexity as the PSP algorithm for $L = l = 3$ and $i = 2$.

V. Summary

This paper described briefly the PML algorithm for data estimation in an ISI environment using the ML design criterion. Unlike the conventional ML methods which apply the VA, the new algorithm (PML) does not apply the VA. A theoretical lower bound for the mean of D in the PML algorithm is derived. This threshold can reduce the computational complexity of the PML algorithm as well as checking its performance.

The SPML algorithm is a suboptimal approach which is suitable for channels with long impulse responses and has much less computational complexity than the PML while its performance is similar to that of the PML. The APML algorithm is introduced for time-variant channels which can outperform all of the VA-based techniques for some examples.

The PML-based algorithms should be applied to more time-invariant channels as well as time-variant channels to have a better understanding of these algorithms. We do not have an analytical performance analysis of the SPML algorithm. In general, PML-based algorithms can have other applications beside what we have mentioned here that can be explored in the future.

Appendix
Performance Design Criterion for the PML Algorithm

In this section, the error covariance matrix between the channel coefficients and its estimate will be given. Also a lower bound for the mean (expected value) of D in the PML algorithm is developed. This lower bound can be used in the PML to choose the appropriate value for l. For example, if the minimum D in each iteration is much larger than this bound, then l should be increased to improve the performance of the PML algorithm.

A. Cramer–Rao Lower Bound for the Covariance of the Channel Coefficients Error

$\hat{\underline{\mathbf{h}}}_{\mathrm{ML}}$ is an unbiased estimate of the CIR [7]. The lower bound of the covariance matrix can be derived using Cramer–Rao lower bound.

$$\mathrm{covar}\left(\hat{\underline{\mathbf{h}}}_{\mathrm{ML}} - \underline{\mathbf{h}}\right) \geq J^{-1} \tag{20}$$

where J is the Fisher information matrix given by

$$J = -\mathcal{E}\left\{\frac{\partial^2}{\partial \underline{\mathbf{h}}^2}\ln \Pr(\underline{\mathbf{r}}_N/\underline{\mathbf{h}})\right\} = \frac{N\sigma_a^2}{\sigma_n^2} I_L \tag{21}$$

The lower bound of the covariance matrix can be written as

$$\mathrm{covar}\left(\hat{\underline{\mathbf{h}}}_{\mathrm{ML}} - \underline{\mathbf{h}}\right) \geq \frac{\sigma_n^2}{N\sigma_a^2} I_L \tag{22}$$

where σ_a^2 is the variance of the data sequence and I_L is the $L \times L$ identity matrix.

B. A Lower Bound for the Mean of D in PML

The PML algorithm is based on finding the minimum Euclidean distance (D) between $\underline{\mathbf{k}}_1$ and $\underline{\mathbf{r}}_l$. The minimum of D is obtained through a global search of M^l different possible values for the M-ary transmitted data sequence of length l. D is a random variable with a mean and a variance. In this section, a lower bound for the mean of D is derived. This lower bound will be used as an internal check on the performance of the PML algorithm. When D is computed for each candidate of $\underline{\mathbf{k}}_1$, this value of D is compared to the lower bound. If D is smaller than the lower bound, the global search will be stopped and that particular data sequence of length l will be picked as the best data sequence. In this case, the receiver does not need to search through all M^l different values of $\underline{\mathbf{k}}_1$ and consequently, the computational complexity of the PML algorithm is reduced [17]. However, if the minimum D that is obtained by the global search is much larger than the lower bound for all M^l values of $\underline{\mathbf{k}}_l$, then the receiver should increase l and/or N to improve the performance. It will be shown later that the lower bound is a function of l, N, L and σ_n^2. It is assumed that the receiver estimates σ_n^2 by some method and it is beyond the scope of this paper to discuss the details of these methods.

The following notation is defined.

$$\hat{\underline{\mathbf{k}}}_1 \triangleq \hat{\mathbf{a}}^t_{N+l}\hat{\underline{\mathbf{h}}}_{\mathrm{ML}}$$

$\hat{\underline{\mathbf{k}}}_1$ is the estimated value of $\underline{\mathbf{k}}_1$ based on the estimated data sequence of length l in $\hat{\mathbf{a}}_{N+l}$. In order to find a lower bound for the mean of D, we assume that the data sequence of length l is estimated accurately and $\underline{\mathbf{k}}_1$ is used for derivation of the mean of D. From the above assumption, D can be written as

$$\begin{aligned} D &= \left|\mathbf{a}^t_{N+l}\left(\underline{\mathbf{h}} - \hat{\underline{\mathbf{h}}}_{\mathrm{ML}}\right) + \underline{\mathbf{n}}_l\right|^2 \\ &= \sum_{i=1}^{l}\left(\underline{\mathbf{a}}^t_{(N+l)_i}\left(\underline{\mathbf{h}} - \hat{\underline{\mathbf{h}}}_{\mathrm{ML}}\right) + n_{k+N+i}\right)^2 \end{aligned} \tag{23}$$

where $\underline{\mathbf{a}}^t_{(N+l)_i} = [a_{k+N+i} \cdots a_{k+N+i-L+1}]$ and

$$\mathbf{a}^t_{N+l} = \begin{bmatrix} \underline{\mathbf{a}}^t_{(N+l)_1} \\ \cdots \\ \underline{\mathbf{a}}^t_{(N+l)_l} \end{bmatrix}.$$

The mean of D can be calculated as follows:

$$\begin{aligned} \mathcal{E}(D) = \mathcal{E}\Bigg[\sum_{i=1}^{l}\Bigg(&\left(\underline{\mathbf{a}}^t_{(N+l)_i}\left(\underline{\mathbf{h}} - \hat{\underline{\mathbf{h}}}_{\mathrm{ML}}\right)\right)^2 + n^2_{k+N+i} \\ &+ 2n_{k+N+i}\underline{\mathbf{a}}^t_{(N+l)_i}\left(\underline{\mathbf{h}} - \hat{\underline{\mathbf{h}}}_{\mathrm{ML}}\right)\Bigg)\Bigg] \end{aligned} \tag{24}$$

$\mathcal{E}$ represents the expected value of a random variable. The order of summation and expectation can be interchanged in (24).

$\hat{\underline{\mathbf{h}}}_{\mathrm{ML}}$ is the estimated CIR based on the received data sequence prior to $k + N + 1$ and therefore, independent of the data sequence after $k + N + 1$. In (24), $\underline{\mathbf{a}}^t_{(N+l)_i}$ can be partitioned into two parts.

$$\begin{aligned} \underline{\mathbf{a}}^t_{(N+l)_i} &= [a_{k+N+i} \cdots a_{k+N+1} 0 \cdots 0] \\ &\quad + [0 \cdots 0 a_{k+N} \cdots a_{k+N+i-L+1}] \\ &= \underline{\mathbf{a}}^t_{(N+l)_i}(1) + \underline{\mathbf{a}}^t_{(N+l)_i}(2). \end{aligned} \tag{25}$$

The elements of $\underline{\mathbf{a}}^t_{(N+l)_i}(1)$ and $\underline{\mathbf{a}}^t_{(N+l)_i}(2)$ are the data sequence after $k + N$ and prior to $k + N + 1$, respectively. Therefore, $\underline{\mathbf{a}}^t_{(N+l)_i}(1)$ is independent of $\hat{\underline{\mathbf{h}}}_{\mathrm{ML}}$ and $\underline{\mathbf{a}}^t_{(N+l)_i}(2)$. The first term in (24) can hence be partitioned as follows:

$$\begin{aligned} \left(\underline{\mathbf{a}}^t_{(N+l)_i}\left(\underline{\mathbf{h}} - \hat{\underline{\mathbf{h}}}_{\mathrm{ML}}\right)\right)^2 &= \underline{\mathbf{a}}^t_{(N+l)_i}(1) W \underline{\mathbf{a}}_{(N+l)_i}(1) \\ &\quad + 2\underline{\mathbf{a}}^t_{(N+l)_i}(1) W \underline{\mathbf{a}}_{(N+l)_i}(2) \\ &\quad + \underline{\mathbf{a}}^t_{(N+l)_i}(2) W \underline{\mathbf{a}}_{(N+l)_i}(2) \end{aligned} \tag{26}$$

where $W \triangleq (\underline{\mathbf{h}} - \hat{\underline{\mathbf{h}}}_{\mathrm{ML}})(\underline{\mathbf{h}} - \hat{\underline{\mathbf{h}}}_{\mathrm{ML}})^t$. The expected value of the first term of (24) is determined by calculating the expected value of (26).

$$\begin{aligned} &\mathcal{E}\left(\underline{\mathbf{a}}^t_{(N+l)_i}(1) W \underline{\mathbf{a}}_{(N+l)_i}(1)\right) \\ &\quad = \mathcal{E}\left(\mathrm{trace}\left(W \underline{\mathbf{a}}_{(N+l)_i}(1)\, \underline{\mathbf{a}}^t{}_{(N+l)_i}(1)\right)\right) \\ &\quad = \mathrm{trace}(\mathcal{E}(W)\mathcal{E}\left(\underline{\mathbf{a}}_{(N+l)_i}(1)\, \underline{\mathbf{a}}^t_{(N+l)_i}(1)\right) \end{aligned} \tag{27}$$

W and $(\underline{\mathbf{a}}_{(N+l)_i}(1)\underline{\mathbf{a}}^t_{(N+l)_i}(1))$ are statistically independent. $\mathcal{E}(W)$ is obtained from (22).

$$\mathcal{E}(W) \geq \frac{\sigma_n^2}{\sigma_a^2 N} I_L \tag{28}$$

(27) can be lower bounded as

$$(27) \geq \mathrm{trace}\left(\frac{\sigma_n^2}{\sigma_a^2 N} I_L \times \mathrm{diag}(\sigma_a^2 \cdots \sigma_a^2\, 0 \cdots 0)\right) = \frac{\sigma_n^2 i}{N} \tag{29}$$

where $\mathrm{diag}(\sigma_a^2 \cdots \sigma_a^2\, 0 \cdots 0)$ is a diagonal matrix that its first ith diagonal elements are σ_a^2 and the remaining diagonal elements are zero.

$$\mathcal{E}\left(\underline{\mathbf{a}}^t_{(N+l)_i}(1) W \underline{\mathbf{a}}_{(N+l)_i}(2)\right) = 0. \tag{30}$$

Notice that $(\underline{\mathbf{a}}^t_{(N+l)_i}(1))$ is independent of $(W\underline{\mathbf{a}}_{(N+l)_i}(2))$ and the data sequence is zero mean.

$$\mathcal{E}\left(\underline{\mathbf{a}}^t_{(N+l)_i}(2)W\underline{\mathbf{a}}_{(N+l)_i}(2)\right) = \mathcal{E}\left(\underline{\mathbf{a}}^t_{(N+l)_i}(2)\left(\underline{\mathbf{h}} - \hat{\underline{\mathbf{h}}}_{\mathrm{ML}}\right)\right)^2 \geq 0. \tag{31}$$

The first term in (24) can be lower bounded using (29). The second term in (24) is equivalent to the variance of the noise. The third term in (24) is

$$\mathcal{E}\left(2n_{k+N+i}\underline{\mathbf{a}}^t_{(N+l)_i}\left(\underline{\mathbf{h}} - \hat{\underline{\mathbf{h}}}_{\mathrm{ML}}\right)\right) = 0. \tag{32}$$

Equation (24) will be lower bounded as

$$\mathcal{E}(D) \geq \sum_{i=1}^{l}\left(\frac{\sigma_n^2 i}{N} + \sigma_n^2\right) = (l \times \sigma_n^2)\left(1 + \frac{l+1}{2N}\right). \tag{33}$$

The above lower bound for the mean of D is valid when $l \leq L$. If $l > L$, the same steps can be followed and the lower bound will be

$$\mathcal{E}(D) \geq \sigma_n^2 \times \left(l + \frac{L(L+1)}{2N} + \frac{L(l-L)}{N}\right). \tag{34}$$

If the CIR and data sequence are estimated accurately, then $\underline{r}_l - \underline{\mathbf{k}}_1 = \underline{n}_l$. Therefore the mean of D would be $l \times \sigma_n^2$. This is the first term in (33) [or (34)]. The remaining terms are due to the fact that the channel estimation is not perfect and there is some error between $\underline{\mathbf{h}}$ and $\hat{\underline{\mathbf{h}}}_{\mathrm{ML}}$. The above inequalities give a lower bound for the mean of D. Equation (33) [or (34)] will provide a fixed value for the lower bound of the mean once these parameters (l, N, L, and σ_n^2) are measured. However, D is a random variable and for each data, the minimum of D that is obtained through the global search can be smaller or larger than this lower bound. In order to apply the lower bound of the mean of D, we multiply this bound by a constant (c). c is usually chosen smaller than unity to assure that the global search will not stop unless the best data sequence of length l is selected. Computer simulation is demonstrated the results for different values of c in [17].

Acknowledgment

The authors would like to thank Dr. E. Eleftheriou, Editor, and the anonymous reviewers for their helpful comments to improve the paper.

References

[1] M. Ghosh, "An optimal approach to blind equalization," Ph.D. dissertation, Dept. Elec. Eng., Univ. Southern California, Dec. 1991.

[2] G. D. Forney Jr., "The viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.

[3] H. R. Sadjadpour and C. L. Weber, "Pseudo-maximum likelihood estimation of digital sequences in the presence of intersymbol interference," in *Proc. 28th Annu. Asilomar Conf. Signals, Systems, and Computers*, vol. 2, Nov. 1994, pp. 1174–1178.

[4] H. R. Sadjadpour, "Blind equalization techniques using maximum likelihood criterion," Ph.D. dissertation, Dept. Elec. Eng., Univ. Southern California, 1996.

[5] C. A. Belfiore and J. H. Park Jr., "Decision feedback equalization," *Proc. IEEE*, vol. 67, pp. 1143–1156, Aug. 1979.

[6] I. Csiszar and G. Tusnady, "Information geometry and alternating minimizing procedures," *Statistics and Decisions, Supplement Issue*, vol. 1, pp. 205–237, 1984.

[7] J. M. Mendel, *Lessons in Digital Estimation Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[8] A. Duel-Hallen and C. Heegard, "Delayed decision-feedback sequence estimation," *IEEE Trans. Commun.*, vol. 37, pp. 428–436, May 1989.

[9] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[10] R. Raheli, A. Polydoros, and C. Tzou, "Per-survivor processing: A general approach to MLSE in uncertain environment," *IEEE Trans. Commun.*, vol. 43, Feb./Mar./Apr. 1995.

[11] *J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 1989.

[12] T. Vaidis and C. L. Weber, "Block adaptive techniques for channel identification and data demodulation over band-limited channels," *IEEE Trans. Commun.*, vol. 46, Feb. 1998.

[13] G. D. Forney, "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.

[14] K. M. Chugg and A. Polydoros, "MLSE for an unknown channel—Part I: Optimality considerations," *IEEE Trans. Commun.*, vol. 44, pp. 836–846, July 1996.

[15] G. Ungerboeck, "Adaptive maximum likelihood receivers for carrier modulated data transmission systems," *IEEE Trans. Commun.*, vol. COM-22, pp. 624–636, May 1974.

[16] A. Ben-Israel and T. N. Greville, *Generalized Inverses: Theory and Applications*. New York: Wiley, 1974.

[17] H. R. Sadjadpour and C. L. Weber, "Pseudo-maximum likelihood data estimation algorithm and its performance analysis," in *Proc. MILCOM'95*, vol. 1, Nov. 1995, pp. 160–164.

[18] J. Moon and L. R. Carley, "Performance comparison of detection methods in magnetic recording," *IEEE Trans. Magn.*, vol. 26, pp. 3155–3172, Nov. 1990.

[19] D. J. Goodman, "Second generation wireless information networks," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 366–374, May 1991.

Hamid R. Sadjadpour (S'90–M'95–SM'00) received the Ph.D. degree in electrical engineering from the University of Southern California (USC), Los Angeles, in 1995.

During the 1994–1995 period, he was also with LinCom Corporation, Los Angeles, CA, as a Member of Technical Staff. Since 1995, he has been with AT&T Research, Florham Park, NJ, as a Principle Technical Staff Member. During 1999, he was also an Adjunct Professor at Lehigh University, Bethlehem, PA. His research interests include equalization techniques for wireless systems, and DSL modems and error control coding and their applications.

Charles L. Weber (S'57–M'65–SM'77–F86) received the Ph.D. degree from the University of California, Los Angeles, in 1964.

Since then, he has been a member of the faculty of the University of Southern California, Los Angeles, where he is a Professor of Electrical Engineering and the Founding Director of the Communications Sciences Institute. His research activities have explored various aspects of communication systems, including the development of the theory of transmitter optimization of coherent and noncoherent digital communication systems, spread-spectrum and multiple-access communication systems, modulation characterization, and blind equalization. He is the author of Elements of Detection and Signal Design (New York: McGraw-Hill, 1968), which presents the development of signal design for digital systems.

Dr. Weber is a member of Sigma Xi, Eta Kappa Nu, Pi Mu Epsilon, and Tau Beta Pi. One of the publications (with R. A. Scholtz) from his research was nominated by the IEEE AdCom for the annual Browder J. Thompson IEEE Paper of the Year Award.