A Fast Computation Algorithm for the Decision Feedback Equalizer

Inkyu Lee, *Student Member, IEEE,* and John M. Cioffi, *Senior Member, IEEE*

***Abstract*—A novel fast algorithm for computing the decision feedback equalizer settings is proposed. The equalizer filters are computed indirectly, first by estimating the channel, and then by computing the coefficients in the frequency domain with the discrete Fourier transform (DFT). Approximating the correlation matrices by circulant matrices facilitates the whole computation with very small performance loss. The fractionally spaced equalizer settings are derived. The performance of the fast algorithm is evaluated through simulation. The effects of the channel estimation error and finite precision arithmetic are briefly analyzed. Results of simulation show the superiority of the proposed scheme.**

I. Introduction

THE DECISION feedback equalizer (DFE) is a well-known receiver structure that is used to mitigate intersymbol interference (ISI) in communication channels. The DFE receiver structure has received considerable attention from many researchers because its performance is superior to the linear equalizer. It decodes channel inputs on a symbol-by-symbol basis and uses past decisions to remove trailing ISI. The minimum-mean-square-error decision feedback equalizer (MMSE-DFE) optimizes the feedforward and feedback filter to minimize the mean-square error [1]–[3].

Adaptive equalization is essential when transmitting data over an unknown channel. The equalizer coefficients of the adaptive DFE can be adjusted recursively [4]. However, the convergence of the time-domain LMS adaptation algorithm is slow, due to the high eigenvalue spread inherent in the DFE structure. The recursive LMS adaptation of the DFE may take millions of iterations to converge to the optimal solution on a typical twisted copper wire loop, for example [5]. Fig. 1 shows a learning curve of the DFE on 400 m 26 gauge copper loop channel using the LMS algorithm with 64 feedforward taps and 25 feedback taps. The solid line at the bottom of the figure indicates the minimum mean square error (MMSE). Even after 10 000 iterations, the mean square error of the LMS algorithm is still 3 dB away from the MMSE value. This problem has motivated an interest in non-recursive adaptive methods based on channel-estimates, in contrast to the traditional recursive adaptive methods for computing the equalizer settings [6], [7].

Non-recursive adaptive equalization can be decomposed into two stages: first estimate the channel, then compute the non-recursive DFE coefficients based on the channel estimates. This non-recursive approach has been shown to have a performance advantage over the direct adaptive equalization method [7], [8]. Also, in a time-varying channel, the channel estimate based DFE is reported to have better tolerance in tracking to time variation than RLS adaptive DFE [9].

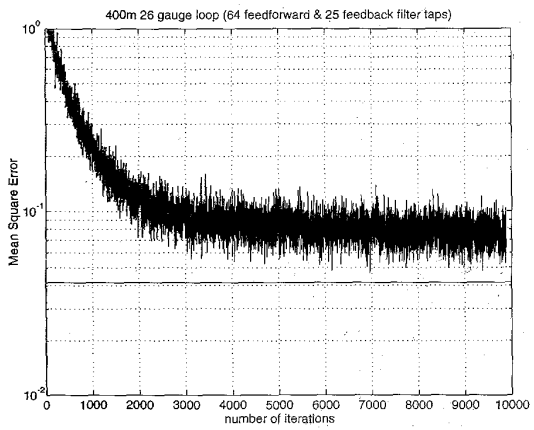

Fig. 1. Learning curve of the LMS algorithm.

This paper presents a novel technique to compute the MMSE-DFE coefficients very efficiently, based on channel estimates. The proposed algorithm requires knowledge of the channel pulse response, along with the noise variance. It is shown in the simulation, however, that the exact value of the noise variance is not important. The channel identification problem, on which the proposed fast algorithm is based, will not be pursued in this paper. Simple methods include cross-correlating channel output samples with a known broadband training sequence [10]. Very accurate and efficient channel identification can also be carried out in the frequency domain using the fast Fourier transform (FFT) with special periodic training sequences [11]. Designs of other training sequences for channel estimates are discussed in [7], [12], [13].

Fast computation of the MMSE-DFE coefficients exploiting structured matrices was derived in [6]. The authors in [6] described an algorithm for computing the feedback filter with $O(M\nu)$ operations where M is the length of the feedforward filter and ν is the length of the channel pulse response. However, $O(M^2)$ operations are still required for obtaining the feedforward filter, and this method involves the fast Cholesky factorization algorithm which makes practical implementation challenging. Also, the number of feedback taps is restricted to ν, which is not a practical assumption.

Paper approved by E. Eleftheriou, the Editor for Equalization and Coding of the IEEE Communications Society. Manuscript received April 27, 1994; revised November 1, 1994 and February 1, 1995. This work was supported by a gift from Samsung, Korea and by NSF under Contract NCR-9203131.

The authors are with the Information Systems Laboratory, the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA.

IEEE Log Number 9414709.

This paper proposes a fast algorithm to compute the MMSE-DFE settings using only the discrete Fourier transform (DFT) and the inverse discrete Fourier transform (IDFT), so that the whole computation can be carried out with $O(M \log_2 M)$ operations, when the fast Fourier transform (FFT) is used. Direct matrix inversion can be avoided by approximating the correlation matrix by a circulant matrix. The proposed algorithm parallels a similar "cyclic equalization" technique applied to the linear equalizer, which uses a periodic psuedo-random training sequence [4], [13]–[17]. The proposed fast algorithm can be implemented very efficiently in practice. Simulation shows that the proposed method comes to within a few tenths of a dB in terms of minimum mean square error performance. Thus, the proposed non-recursive algorithm exhibits superiority in computational efficiency over the conventional recursive adaptive method.

In the following section, the optimum settings for the MMSE-DFE are derived. In Section III, we describe the proposed algorithm for the fractionally spaced equalizer. Then, the performance of the fast algorithm is examined in the symbol spaced equalizer and the effects of channel estimation error and finite precision arithmetic are discussed in Section IV. Section V contains a concluding discussion.

II. Decision Feedback Equalization

The structure of the DFE is shown in Fig. 2. We assume that the pulse response $h(t)$ extends over a finite interval $0 \le t \le \nu T$, where T denotes the symbol period. The standard additive white Gaussian noise channel model is used throughout this paper, where the input/output relation is given by:

$$y(t) = \sum_m x_m h(t - mT) + n(t)$$

where $y(t)$ is the channel output, $\{x_m\}$ is the channel input sequence with signal power $\bar{\mathcal{E}}_x$, $h(t)$ is the channel pulse response, T is the symbol period and $n(t)$ is the additive Gaussian noise with variance σ_n^2.

With a fractionally spaced equalizer sampling at time instants $t = kT - iT/l, i = 0, 1, \cdots, l-1$, the input/output relation for the discrete time equivalent channel has the form:

$$\underline{\boldsymbol{y}}_k = \sum_m \underline{\boldsymbol{h}}_m x_{k-m} + \underline{\boldsymbol{n}}_k \tag{1}$$

where l is the oversampling factor and

$$\underline{\boldsymbol{y}}_k = \left[y\left(kT + \frac{l-1}{l}T\right) y\left(kT + \frac{l-2}{l}T\right) \cdots y(kT) \right]^T,$$

$$\underline{\boldsymbol{h}}_k = \left[h\left(kT + \frac{l-1}{l}T\right) h\left(kT + \frac{l-2}{l}T\right) \cdots h(kT) \right]^T,$$

$$\underline{\boldsymbol{n}}_k = \left[n\left(kT + \frac{l-1}{l}T\right) n\left(kT + \frac{l-2}{l}T\right) \cdots n(kT) \right]^T.$$

We assume that the channel input sequence $\{x_k\}$ and the noise sequence $\{\underline{\boldsymbol{n}}_k\}$ are uncorrelated with each other.

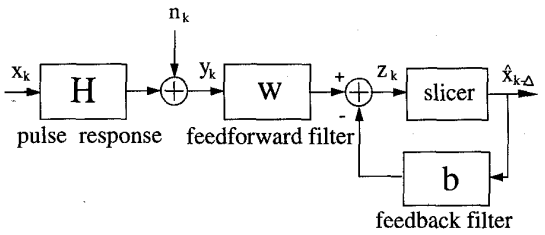

Fig. 2. Structure of DFE.

Using (1) for M successive l-tuples of samples of $y(t)$, we can form the following relation:

$$\begin{aligned} \boldsymbol{y}_k &= \boldsymbol{H}\boldsymbol{x}_k + \boldsymbol{n}_k \\ &= \tilde{\boldsymbol{y}}_k + \boldsymbol{n}_k \end{aligned} \tag{2}$$

where $\tilde{\boldsymbol{y}}_k = \boldsymbol{H}\boldsymbol{x}_k$ represents the noiseless channel output vector,

$$\boldsymbol{y}_k = \begin{bmatrix} \underline{\boldsymbol{y}}_k \\ \underline{\boldsymbol{y}}_{k-1} \\ \vdots \\ \underline{\boldsymbol{y}}_{k-M+1} \end{bmatrix}, \quad \boldsymbol{x}_k = \begin{bmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-M-\nu+1} \end{bmatrix},$$

$$\boldsymbol{n}_k = \begin{bmatrix} \underline{\boldsymbol{n}}_k \\ \underline{\boldsymbol{n}}_{k-1} \\ \vdots \\ \underline{\boldsymbol{n}}_{k-M+1} \end{bmatrix}, \quad \tilde{\boldsymbol{y}}_k = \begin{bmatrix} \tilde{\underline{\boldsymbol{y}}}_k \\ \tilde{\underline{\boldsymbol{y}}}_{k-1} \\ \vdots \\ \tilde{\underline{\boldsymbol{y}}}_{k-M+1} \end{bmatrix},$$

and

$$\boldsymbol{H} = \begin{bmatrix} \underline{\boldsymbol{h}}_0 & \underline{\boldsymbol{h}}_1 & \cdots & \underline{\boldsymbol{h}}_\nu & 0 & \cdots & 0 \\ 0 & \underline{\boldsymbol{h}}_0 & \underline{\boldsymbol{h}}_1 & \cdots & \underline{\boldsymbol{h}}_\nu & \cdots & \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \underline{\boldsymbol{h}}_0 & \underline{\boldsymbol{h}}_1 & \cdots & \underline{\boldsymbol{h}}_\nu \end{bmatrix}.$$

As shown in Fig. 2, the DFE consists of two filters. A fractionally spaced feedforward filter has $M \cdot l$ taps and a symbol-spaced feedback filter has $N+1$ taps. We will denote the feedforward filter and the feedback filter by $\boldsymbol{w} = [\underline{\boldsymbol{w}}_0^T, \underline{\boldsymbol{w}}_1^T, \cdots, \underline{\boldsymbol{w}}_{M-1}^T]^T$ and $\boldsymbol{b} = [1, b_1, b_2, \cdots, b_N]^T$, where $\underline{\boldsymbol{w}}_i = [w_{0,i}, w_{1,i}, \cdots, w_{l-1,i}]^T$.

From Fig. 2, assuming that the decisions are correct, the error signal is defined as

$$\begin{aligned} e_k &= x_{k-\Delta} - z_k \\ &= x_{k-\Delta} - \left(\sum_{m=0}^{M-1} \underline{\boldsymbol{w}}_m^* \underline{\boldsymbol{y}}_{k-m} - \sum_{m=1}^{N} b_m^* x_{k-\Delta-m} \right) \\ &= \boldsymbol{b}^* \boldsymbol{x}_{k-\Delta} - \boldsymbol{w}^* \boldsymbol{y}_k \end{aligned}$$

where Δ represents the decision delay, $\boldsymbol{x}_{k-\Delta} = [x_{k-\Delta}, x_{k-\Delta-1}, \cdots, x_{k-\Delta-N}]^T$, and $*$ indicates the complex conjugate transpose.

The effect of decision delay Δ has been studied in [18]. With a sufficiently long feedback filter, the optimal Δ becomes $M-1$. The choice of Δ can significantly affect performance, especially with short filter lengths. There is no closed-form solution for the optimum Δ in the DFE, while the linear equalizer has a closed-form optimum decision delay.

It is useful to define $\boldsymbol{R}_{xx} = E[\boldsymbol{x}_{k-\Delta}\boldsymbol{x}_{k-\Delta}^*]$, $\boldsymbol{R}_{xy} = \boldsymbol{R}_{yx}^* = E[\boldsymbol{x}_{k-\Delta}\boldsymbol{y}_k^*]$ and $\boldsymbol{R}_{yy} = E[\boldsymbol{y}_k\boldsymbol{y}_k^*]$. We note that these correlation matrices are Toeplitz assuming a stationary process.

From the orthogonality principle, we can derive the minimum mean square error as

$$\sigma^2_{MSE} = \boldsymbol{b}^* \boldsymbol{R}_{x|y} \boldsymbol{b}$$

where

$$\boldsymbol{R}_{x|y} = \boldsymbol{R}_{xx} - \boldsymbol{R}_{xy}\boldsymbol{R}^{-1}_{yy}\boldsymbol{R}_{yx}. \tag{3}$$

Then, it can be shown [19], [20] that the solutions of the optimum DFE are

$$\boldsymbol{b} = \frac{1}{k}\boldsymbol{R}^{-1}_{x|y}\boldsymbol{e}_0 \tag{4}$$

$$\boldsymbol{w} = \boldsymbol{R}^{-1}_{yy}\boldsymbol{R}_{yx}\boldsymbol{b} \tag{5}$$

where k is the (1, 1) element of $\boldsymbol{R}^{-1}_{x|y}$ and $\boldsymbol{e}_0 = [1\ 0 \cdots 0]^T$.

III. Fast Equalization Algorithm

We begin with a brief notational description. A plain lowercase variable denotes a scalar quantity, while a bold lowercase variable denotes a vector. A DFT column vector is represented by a plain uppercase variable, while a matrix is represented by a bold uppercase variable. The symbol $\boldsymbol{I}_n$ represents the n by n identity matrix, and the symbol $\boldsymbol{1}_n$ indicates an all one column vector of length n.

First we describe a circulant matrix. A circulant matrix has the discrete Fourier transform basis vectors as its eigenvectors, and the discrete Fourier transform of its first column as its eigenvalues. Defining $M' = M \cdot l$, an M' by M' circulant matrix can be decomposed as

$$\begin{bmatrix} c_0 & c_{M'-1} & c_{M'-2} & \cdots & c_1 \\ c_1 & c_0 & c_{M'-1} & & \\ c_2 & c_1 & c_0 & & \vdots \\ \vdots & & & \ddots & \\ c_{M'-1} & & \cdots & & c_0 \end{bmatrix} = \frac{1}{M'}\boldsymbol{P}^*\Lambda_C\boldsymbol{P}$$

where $\boldsymbol{P}$ is the discrete Fourier transform (DFT) matrix with

$$p_{m,n} = e^{-j2\pi mn/M'}, \qquad 0 \le m, n \le M' - 1,$$

and Λ_C is an M' by M' diagonal matrix whose diagonal elements are M'-point DFT of $[c_0, c_1, \cdots, c_{M'-1}]$, the first column of the circulant matrix.

We will define $\boldsymbol{P}_M$ as a M' by M submatrix of $\boldsymbol{P}$, consisting of every lth columns of $\boldsymbol{P}$: $\boldsymbol{P}_M = [\boldsymbol{p}_0\ \ \boldsymbol{p}_l\ \ \boldsymbol{p}_{2l} \cdots \boldsymbol{p}_{(M-1)l}]$ where $\boldsymbol{p}_i$ is the ith column of $\boldsymbol{P}$. Similarly, we define $\boldsymbol{P}_N$ as a M' by $N+1$ submatrix of $\boldsymbol{P}$, consisting of every lth columns of $\boldsymbol{P}$: $\boldsymbol{P}_N = [\boldsymbol{p}_0\ \ \boldsymbol{p}_l\ \ \boldsymbol{p}_{2l} \cdots \boldsymbol{p}_{Nl}]$. We note that premultiplying a column vector $\boldsymbol{y}$ of length M' by $\boldsymbol{P}$ yields its DFT column vector, $Y = \boldsymbol{P}\boldsymbol{y}$, and premultiplying a column vector $\boldsymbol{x}$ of length $N+1$ ($\le M$) by $\boldsymbol{P}_N$ produces its zero-padded M'-point DFT column vector, since

$$\begin{aligned}\boldsymbol{P}_N\boldsymbol{x} &= \boldsymbol{P}[x_0\ \ 0\ \ \cdots\ \ 0\ \ x_1\ \ 0\ \ \cdots x_N\ \ 0\ \ \cdots\ \ 0]^T \\ &= [X^T\ \ \cdots\ \ X^T]^T\end{aligned}$$

where the column vector X is the M-point DFT of $[x_0\ \ x_1 \cdots x_N\ \ 0\ \ \cdots\ \ 0]^T$. Similarly, $\boldsymbol{P}^*_N Y$ represents the truncated IDFT vector $\boldsymbol{y}$ since $\boldsymbol{P}^*_N Y = [y_0\ \ y_l\ \ y_{2l}\ \ \cdots\ \ y_{Nl}]^T$.

Note that from the orthogonal property of the DFT basis function $\boldsymbol{P}^*\boldsymbol{P} = \boldsymbol{P}\boldsymbol{P}^* = M'\boldsymbol{I}_{M'}$ and $\boldsymbol{P}^*_N\boldsymbol{P}_N = M'\boldsymbol{I}_{N+1}$, but $\boldsymbol{P}_N\boldsymbol{P}^*_N \ne M'\boldsymbol{I}_N$. Also it can be shown that

$$\boldsymbol{P}_M\boldsymbol{P}^*_M = M[\boldsymbol{I}_M\ \ \cdots\ \ \boldsymbol{I}_M]^T[\boldsymbol{I}_M\ \ \cdots\ \ \boldsymbol{I}_M]. \tag{6}$$

It is convenient to denote Λ_Z as the diagonal matrix whose diagonal element consists of the elements of a column vector Z. Here we assume that Z is a DFT column vector. Using $\boldsymbol{P}^{-1} = (\frac{1}{M'})\boldsymbol{P}^*$, the inverse of the above circulant matrix is $(\frac{1}{M'})\boldsymbol{P}^*\Lambda^{-1}_C\boldsymbol{P}$.

The main innovation of the fast start-up equalization in this paper arises from the approximation of a Toeplitz correlation matrix by a circulant matrix. A circulant matrix is asymptotically equivalent to a Toeplitz matrix [21]. By making this approximation, most computations can be implemented with the discrete Fourier transform and inverse discrete Fourier transform very efficiently.

Our derivation assumes that the length of the feedforward filter, $\boldsymbol{w}$, exceeds that of the feedback filter $\boldsymbol{b}(M' \ge N+1)$. This restriction may not be applicable in some channels such as the subscriber loop, where the feedback filter can be longer than the feedforward filter. However, a long feedforward filter can be accurately approximated by a pole-zero filter with fewer coefficients using a computationally-efficient algorithm described in [22] with little performance loss.

We denote $\overline{\boldsymbol{R}}_{yy}, \overline{\boldsymbol{R}}_{yx}, \overline{\boldsymbol{R}}_{xx}$ and $\overline{\boldsymbol{R}}_{x|y}$ as the approximation to the matrices $\boldsymbol{R}_{yy}, \boldsymbol{R}_{yx}, \boldsymbol{R}_{xx}$, and $\boldsymbol{R}_{x|y}$, respectively. The autocorrelation matrix $\boldsymbol{R}_{yy}$ is computed as $\boldsymbol{R}_{yy} = E[\tilde{\boldsymbol{y}}_k\tilde{\boldsymbol{y}}^*_k] + l \cdot \sigma^2_n\boldsymbol{I}_{M'}$ from (2). To approximate $\boldsymbol{R}_{yy}$ as a circulant matrix, we assume that $\{\tilde{\boldsymbol{y}}_k\}$ is cyclic. Using (6), $E[\tilde{\boldsymbol{y}}_k\tilde{\boldsymbol{y}}^*_k]$ is computed as a time-averaged autocorrelation function:

$$\begin{aligned}E[\tilde{\boldsymbol{y}}_k\tilde{\boldsymbol{y}}^*_k] &= \frac{1}{M}\sum_{k=0}^{M-1}\tilde{\boldsymbol{y}}_k\tilde{\boldsymbol{y}}^*_k \\ &= \frac{1}{M}\boldsymbol{C}_Y\boldsymbol{C}^*_Y \\ &= \frac{1}{M}\left(\frac{1}{M'}\boldsymbol{P}^*\Lambda_{\tilde{Y}}\boldsymbol{P}_M\right)\left(\frac{1}{M'}\boldsymbol{P}^*_M\Lambda_{\tilde{Y}^*}\boldsymbol{P}\right) \\ &= \frac{1}{M'^2}\boldsymbol{P}^*\Lambda_{\tilde{Y}}\begin{bmatrix}\boldsymbol{I}_M \\ \vdots \\ \boldsymbol{I}_M\end{bmatrix}[\boldsymbol{I}_M\ \ \cdots\ \ \boldsymbol{I}_M]\Lambda_{\tilde{Y}^*}\boldsymbol{P} \\ &= \frac{1}{M'^2}\boldsymbol{P}^*\begin{bmatrix}\Lambda_{\tilde{Y}^1} \\ \vdots \\ \Lambda_{\tilde{Y}^l}\end{bmatrix}[\Lambda_{\tilde{Y}^{1*}}\ \ \cdots\ \ \Lambda_{\tilde{Y}^{l*}}]\boldsymbol{P}\end{aligned}$$

where the column vector $\tilde{Y}$ is the M' point DFT of $[\tilde{\boldsymbol{y}}^T_{M-1}\ \tilde{\boldsymbol{y}}^T_{M-2}\ \cdots\ \tilde{\boldsymbol{y}}^T_0]$, the length M column vector $\tilde{Y}^i$ denotes the ith sub-vector of $\tilde{Y}$: $\tilde{Y} = [\tilde{Y}^{1T}\ \ \tilde{Y}^{2T}\ \ \cdots\ \ \tilde{Y}^{lT}]^T$ and

$$\boldsymbol{C}_Y = \begin{bmatrix}\tilde{\boldsymbol{y}}_{M-1} & \tilde{\boldsymbol{y}}_0 & \tilde{\boldsymbol{y}}_1 & \cdots & \tilde{\boldsymbol{y}}_{M-2} \\ \tilde{\boldsymbol{y}}_{M-2} & \tilde{\boldsymbol{y}}_{M-1} & \tilde{\boldsymbol{y}}_0 & & \\ \tilde{\boldsymbol{y}}_{M-3} & \tilde{\boldsymbol{y}}_{M-2} & \tilde{\boldsymbol{y}}_{M-1} & & \vdots \\ \vdots & & & \ddots & \\ \tilde{\boldsymbol{y}}_0 & & \cdots & & \tilde{\boldsymbol{y}}_{M-1}\end{bmatrix}.$$

Similarly, $\boldsymbol{R}_{yx} = E[\boldsymbol{y}_k \boldsymbol{x}^*_{k-\Delta}]$ is equal to $E[\tilde{\boldsymbol{y}}_k \boldsymbol{x}^*_{k-\Delta}]$ since $\boldsymbol{x}_{k-\Delta}$ and $\boldsymbol{n}_k$ are assumed to be uncorrelated with each other, thus $\boldsymbol{R}_{yx}$ is approximated by

$$\begin{aligned}\overline{\boldsymbol{R}}_{yx} &= \frac{1}{M}\sum_{k=0}^{M-1} \tilde{\boldsymbol{y}}_k \boldsymbol{x}^*_{k-\Delta} \\ &= \frac{1}{M}\boldsymbol{C}_Y \boldsymbol{C}^*_X \\ &= \frac{1}{M}\left(\frac{1}{M'}\boldsymbol{P}^* \Lambda_{\tilde{Y}} \boldsymbol{P}_M\right)\left(\frac{1}{M'}\boldsymbol{P}^*_M \Lambda_{\tilde{X}^*} \boldsymbol{P}_N\right) \\ &= \frac{1}{M'^2}\boldsymbol{P}^* \Lambda_{\tilde{Y}} \begin{bmatrix} \boldsymbol{I}_M \\ \vdots \\ \boldsymbol{I}_M \end{bmatrix} [\boldsymbol{I}_M \quad \cdots \quad \boldsymbol{I}_M] \Lambda_{\tilde{X}^*} \boldsymbol{P}_N \\ &= \frac{1}{M'^2}\boldsymbol{P}^* \begin{bmatrix} \Lambda_{\tilde{Y}^1} \\ \vdots \\ \Lambda_{\tilde{Y}^l} \end{bmatrix} [\Lambda_{X^*} \quad \cdots \quad \Lambda_{X^*}] \boldsymbol{P}_N\end{aligned}$$

where

$$\boldsymbol{C}^*_X = \begin{bmatrix} x^*_{M-\Delta-1} & x^*_{M-\Delta-2} & x^*_{M-\Delta-3} & \cdots & x^*_{M-\Delta-N-1} \\ x^*_{M-\Delta} & x^*_{M-\Delta-1} & x^*_{M-\Delta-2} & & \\ x^*_{M-\Delta+1} & x^*_{M-\Delta} & x^*_{M-\Delta-1} & & \vdots \\ \vdots & & & \ddots & \\ x^*_{M-\Delta-2} & & \cdots & & x_{M-\Delta-1} \end{bmatrix}$$

and the column vector $\tilde{X}$ is the M' point DFT of $[x_{M'-\Delta-1} \; 0 \; \cdots \; 0 \; x_{M'-\Delta-2} \; 0 \; \cdots \; x_{M'-\Delta} \; 0 \; \cdots \; 0]$. Thus $\tilde{X} = [X^T \quad \cdots \quad X^T]^T$ where X represents the DFT vector of $[x_{M-\Delta-1} \quad x_{M-\Delta-2} \cdots x_0 \quad x_{M-1} \cdots x_{M-\Delta}]$.

Defining $\otimes$ and $\oslash$ as the element-wise multiplication and division respectively, the inverse of $\overline{\boldsymbol{R}}_{yy}$, along with the matrix inversion lemma,[1] is then

$$\begin{aligned}\overline{\boldsymbol{R}}^{-1}_{yy} &= (E[\tilde{\boldsymbol{y}}_k \tilde{\boldsymbol{y}}^*_k] + l \cdot \sigma_n^2 \boldsymbol{I}_{M'})^{-1} \\ &= \left\{\frac{1}{M'^2}\boldsymbol{P}^*(\boldsymbol{\Gamma}_Y \boldsymbol{\Gamma}^*_Y + M' l \sigma_n^2 \boldsymbol{I}_{M'})\boldsymbol{P}\right\}^{-1} \\ &= \boldsymbol{P}^*(\boldsymbol{\Gamma}_Y \boldsymbol{\Gamma}^*_Y + M' l \sigma_n^2 \boldsymbol{I}_{M'})^{-1}\boldsymbol{P} \\ &= \boldsymbol{P}^*\left(\frac{1}{M' l \sigma_n^2}\left(\boldsymbol{I}_{M'} - \frac{1}{M' l \sigma_n^2}\boldsymbol{\Gamma}_Y \right.\right. \\ &\quad \left.\left. \cdot (\boldsymbol{\Gamma}^*_Y \boldsymbol{\Gamma}_Y + \boldsymbol{I}_M)^{-1}\boldsymbol{\Gamma}^*_Y\right)\right)\boldsymbol{P} \\ &= \frac{1}{M' l \sigma_n^2}\boldsymbol{P}^*\left(\boldsymbol{I}_{M'} - \boldsymbol{\Gamma}_Y \Lambda^{-1}_{\Theta} \boldsymbol{\Gamma}^*_Y\right)\boldsymbol{P}\end{aligned}$$

where $\boldsymbol{\Gamma}_Y = [\Lambda_{\tilde{Y}^1} \quad \cdots \quad \Lambda_{\tilde{Y}^l}]^T, \Theta = \overline{\Theta} + M' l \sigma_n^2 \mathbf{1}_M$, and $\overline{\Theta} = \Sigma_{i=1}^{l} ||\tilde{Y}^i||^2$. Here $||\cdot||^2$ is defined as the element-wise norm square:

$$||[a_0 \quad a_1 \cdots a_{M-1}]^T||^2 = [|a_0|^2 \quad |a_1|^2 \cdots |a_{M-1}|^2]^T.$$

Also, note that $\boldsymbol{\Gamma}^*_Y \boldsymbol{\Gamma}_Y = \Lambda_{\overline{\Theta}}$.

For simplicity, we assume that the input sequence $\{x_k\}$ is a white sequence: $\boldsymbol{R}_{xx} = \overline{\mathcal{E}}_x \boldsymbol{I}_{N+1}$. This assumption can be satisfied by setting $|X_i|^2 = \overline{\mathcal{E}}_x M$ for all i, which implies a constant amplitude zero autocorrelation (CAZAC) sequence.

Substituting the above equations into (5) yields

$$\begin{aligned}\overline{\boldsymbol{R}}^{-1}_{yy}\overline{\boldsymbol{R}}_{yx} &= \frac{1}{M'^2 l \sigma_n^2}\boldsymbol{P}^*(\boldsymbol{\Gamma}_Y \boldsymbol{\Gamma}^*_X - \boldsymbol{\Gamma}_Y \Lambda_{\overline{\Theta}\oslash\Theta}\boldsymbol{\Gamma}^*_X)\boldsymbol{P}_N \\ &= \frac{1}{M'^2 l \sigma_n^2}\boldsymbol{P}^*(\boldsymbol{\Gamma}_Y \Lambda_{(\Theta-\overline{\Theta})\oslash\Theta}\boldsymbol{\Gamma}^*_X)\boldsymbol{P}_N \\ &= \frac{1}{M'}\boldsymbol{P}^*(\boldsymbol{\Gamma}_Y \Lambda^{-1}_{\Theta}\boldsymbol{\Gamma}^*_X)\boldsymbol{P}_N.\end{aligned}$$

Also, using the above results, matrices in (3) become

$$\begin{aligned}\overline{\boldsymbol{R}}_{x|y} &= \overline{\mathcal{E}}_x \boldsymbol{I}_{N+1} - \frac{1}{M'^2}\boldsymbol{P}^*_N \boldsymbol{\Gamma}_X \boldsymbol{\Gamma}^*_Y \boldsymbol{\Gamma}_Y \Lambda^{-1}_{\Theta}\boldsymbol{\Gamma}^*_X \boldsymbol{P}_N \\ &= \frac{1}{M'^2}\boldsymbol{P}^*_N(M'\overline{\mathcal{E}}_x \boldsymbol{I}_{M'} - \boldsymbol{\Gamma}_X \Lambda_{\overline{\Theta}\oslash\Theta}\boldsymbol{\Gamma}^*_X)\boldsymbol{P}_N \\ &= \frac{\overline{\mathcal{E}}_x}{M'}\boldsymbol{P}^*_N(\boldsymbol{I}_{M'} - \boldsymbol{\Gamma}_I \Lambda_{\overline{\Theta}\oslash l\Theta}\boldsymbol{\Gamma}^*_I)\boldsymbol{P}_N \qquad (7)\end{aligned}$$

where $\boldsymbol{\Gamma}_X = [\Lambda_X \cdots \Lambda_X]^T$ and $\boldsymbol{\Gamma}_I = [\boldsymbol{I}_M \cdots \boldsymbol{I}_M]^T$.

From (4), the feedback filter $\boldsymbol{b}$ can be computed directly. Since $\overline{\boldsymbol{R}}_{x|y}$ is block Toeplitz, its inversion can be carried out efficiently using the Levinson–Trench algorithm using $O(l^3 N^2)$ recursions as in [8]. However, implementing the Levinson–Trench algorithm is still not attractive in terms of complexity and cost in real time applications. Thus we make a further approximation in inversion of matrix in (4), so that the whole computation can be done with only the DFT. Here the inverse of $\overline{\boldsymbol{R}}_{x|y}$ is approximated by

$$\begin{aligned}\overline{\boldsymbol{R}}^{-1}_{x|y} &= \frac{1}{\overline{\mathcal{E}}_x M'}\boldsymbol{P}^*_N(\boldsymbol{I}_{M'} - \boldsymbol{\Gamma}_I \Lambda_{\overline{\Theta}\oslash l\Theta}\boldsymbol{\Gamma}^*_I)^{-1}\boldsymbol{P}_N \\ &= \frac{1}{\overline{\mathcal{E}}_x M'}\boldsymbol{P}^*_N(\boldsymbol{I}_{M'} - \boldsymbol{\Gamma}_I(\boldsymbol{\Gamma}^*_I\boldsymbol{\Gamma}_I - \Lambda_{l\Theta\oslash\overline{\Theta}})^{-1}\boldsymbol{\Gamma}^*_I)\boldsymbol{P}_N \\ &= \frac{1}{\overline{\mathcal{E}}_x M'}\boldsymbol{P}^*_N\left(\boldsymbol{I}_{M'} + \frac{1}{M' l^2 \sigma_n^2}\boldsymbol{\Gamma}_I \Lambda_{\overline{\Theta}}\boldsymbol{\Gamma}^*_I\right)\boldsymbol{P}_N.\end{aligned}$$

This approximation turns out to be quite accurate from the simulation as will be shown later. Note that a scalar constant of $\overline{\boldsymbol{R}}^{-1}_{x|y}$ is not important, since scaling takes place in (4).

Plugging the above equation into (4) yields

$$\begin{aligned}\boldsymbol{b} &= \frac{1}{k}\boldsymbol{P}^*_N\left(\boldsymbol{I}_{M'} + \frac{1}{M' l^2 \sigma_n^2}\boldsymbol{\Gamma}_I \Lambda_{\overline{\Theta}}\boldsymbol{\Gamma}^*_I\right)\mathbf{1}_{M'} \\ &= \frac{1}{k}\boldsymbol{P}^*_N\left(\boldsymbol{\Gamma}_I \mathbf{1}_M + \frac{1}{M' l \sigma_n^2}\boldsymbol{\Gamma}_I \Lambda_{\overline{\Theta}}\mathbf{1}_M\right) \\ &= \frac{1}{k'}\boldsymbol{P}^*_N \boldsymbol{\Gamma}_I \Lambda_{\Theta}\mathbf{1}_M \\ &= \frac{1}{k'}\boldsymbol{P}^*_N \boldsymbol{\Gamma}_I \Theta\end{aligned}$$

where k' is a scaling constant to make $\boldsymbol{b}$ monic ($b_0 = 1$).

[1] If $\boldsymbol{A}$ and $\boldsymbol{C}$ are nonsingular and $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ and $\boldsymbol{D}$ have consistent dimensions, then

$$(\boldsymbol{A} + \boldsymbol{BCD})^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{B}(\boldsymbol{DA}^{-1}\boldsymbol{B} + \boldsymbol{C}^{-1})^{-1}\boldsymbol{DA}^{-1}.$$

Once $\boldsymbol{b}$ is obtained, we can compute the feedforward filter $\boldsymbol{w}$ from (5):

$$\begin{aligned}\boldsymbol{w} &= \frac{1}{M'}\boldsymbol{P}^*\boldsymbol{\Gamma}_Y\Lambda_\Theta^{-1}\boldsymbol{\Gamma}_X^*\begin{bmatrix}B\\ \vdots \\ B\end{bmatrix}\\ &= \frac{1}{M}\boldsymbol{P}^*\boldsymbol{\Gamma}_Y X^*\otimes B\oslash\Theta\\ &= \frac{1}{M}\boldsymbol{P}^*\begin{bmatrix}\tilde{Y}^1\otimes X^*\otimes B\oslash\Theta\\ \vdots \\ \tilde{Y}^l\otimes X^*\otimes B\oslash\Theta\end{bmatrix}\\ &= \frac{1}{M}\boldsymbol{P}^*\tilde{Y}\otimes\tilde{X}^*\otimes\begin{bmatrix}B\oslash\Theta\\ \vdots \\ B\oslash\Theta\end{bmatrix}\end{aligned}$$

where the column vector B is the M-point DFT of $[1 \quad b_0 \quad b_1 \quad \cdots \quad b_N \quad 0 \quad \cdots \quad 0]$.

Without loss of generality, we can represent the noiseless channel output sequence $\{\tilde{\boldsymbol{y}}_k\}$ in the discrete frequency domain as a column vector form:

$$\tilde{Y} = H^*\otimes P_\Delta\otimes\tilde{X} \tag{8}$$

where the column vector H is the M'-point DFT of the channel response sequence $\{\underline{\boldsymbol{h}}_k\}$, $P_\Delta = [\tilde{P}_\Delta^T \cdots \tilde{P}_\Delta^T]^T$ and $\tilde{P}_\Delta = [1\ e^{-j2\pi\Delta/M}\ e^{-j2\pi 2\Delta/M}\ \cdots\ e^{-j2\pi(M-1)\Delta/M}]^T$.

Substituting $\tilde{Y}$ into the above equations yields

$$\boldsymbol{b} = \frac{1}{k'}\boldsymbol{P}_N^*\begin{bmatrix}\Theta\\ \vdots \\ \Theta\end{bmatrix} \tag{9}$$

and

$$\boldsymbol{w} = \boldsymbol{P}^*\overline{\mathcal{E}}_x H^*\otimes P_\Delta\otimes\begin{bmatrix}B\oslash\Theta\\ \vdots \\ B\oslash\Theta\end{bmatrix} \tag{10}$$

where $\Theta = \overline{\mathcal{E}}_x M\,\Sigma_{i=1}^{l}(||H^i||^2) + M'l\sigma_n^2\mathbf{1}_M$ and the length M column vector H^i denotes the ith sub-vector of H: $H = [H^{1T} \quad H^{2T} \quad \cdots \quad H^{lT}]^T$.

Defining $\overline{B}_i$ as

$$\overline{B}_i = \frac{1}{k'}\Theta_{i \bmod M}, \qquad i = 0, 1\cdots M'-1 \tag{11}$$

where Θ_k indicates the kth element of the vector Θ, the feedback filter can be obtained from the IDFT operation in (9):

$$b_k = \frac{1}{M'}\sum_{i=0}^{M'-1}\overline{B}_i e^{j2\pi ik/M}, \qquad k = 0, 1, \cdots N. \tag{12}$$

To compute $\boldsymbol{w}$, we multiply both sides of (10) by $\boldsymbol{P}$:

$$W_i = \frac{\overline{\mathcal{E}}_x M H_i^* P_{\Delta,i} B_{i \bmod M}}{\Theta_{i \bmod M}}, \qquad i = 0, 1, \cdots M'-1 \tag{13}$$

where H_i and B_i are the ith element of column vectors H and B.

From this equation, we can obtain w_k using the IDFT:

$$w_k = \frac{1}{M'}\sum_{i=0}^{M'-1} W_i e^{j2\pi ik/M'}, \qquad k = 0, 1, \cdots M'-1.$$

We notice that the whole computation of $\boldsymbol{b}$ and $\boldsymbol{w}$ can be carried out using only the DFT and the IDFT with no matrix inversion and/or multiplication.

It is interesting that $\boldsymbol{b}$ is independent of the decision delay, Δ, which makes sense because $\overline{\boldsymbol{R}}_{x|y}$ is now Toeplitz. Also, we do not need to compute $\boldsymbol{w}$ for every Δ. Once $\boldsymbol{w}$ is computed for $\Delta = 0, \boldsymbol{w}$ with other values for Δ can be obtained by a circular shift by the amount Δ.

We note that cyclic equalization for the linear equalizer is a special case of the proposed algorithm. With $\boldsymbol{b} = [1 \quad 0 \quad \cdots \quad 0]$, (13) becomes the similar equation derived in [4], [15].

Another special case is the symbol-spaced equalizer for $l = 1$. In this case, plugging $l = 1$ into the above equations yields

$$\overline{B}_i = \frac{1}{k'}(\overline{\mathcal{E}}_x|H_i|^2 + \sigma_n^2), \qquad i = 0, 1\cdots M-1, \tag{14}$$

and

$$W_i = \frac{\overline{\mathcal{E}}_x H_i^* P_{\Delta,i}\cdot B_i}{\overline{\mathcal{E}}_x|H_i|^2 + \sigma_n^2}, \qquad i = 0, 1, \cdots M-1. \tag{15}$$

When M' is an integer power of 2, the DFT's are carried out by the fast Fourier transform to speed up the computation. An FFT can be computed very efficiently in $O(M'\log_2 M')$ operations.

Because the asymptotic behavior of eigenvalues of Toeplitz matrices approaches that of circulant matrices as the size increases, the proposed algorithm is expected to converge to the optimal solution as M' increases. This observation will be confirmed in the following section through simulations.

One choice of the training sequence in (8) is a psuedo-random noise sequence. Another attractive method is a chirp sequence since this has a flat power spectrum and low peak-to-average power ratio [23]. Thus we can describe a chirp sequence for channel identification by

$$x_k = e^{j(2\pi/M)k^2} \qquad k = 0, 1, \cdots M-1.$$

For multi-rate systems such as V.34, a chirp sequence is used for probing to also determine the maximum baud rate [24]. With a chirp sequence, the baud rate and the DFE coefficients can be estimated simultaneously.

Simulation results show that a better solution can be obtained by introducing a scaling factor α to calibrate $\boldsymbol{b}$. After $\boldsymbol{b}$ is computed from the previous procedures, $\boldsymbol{b}$ scaled by scalar α can be used instead to compensate for the approximation. The mean square error σ_{DFE}^2 taking into account the approximations after calibration is then

$$\sigma_{\text{DFE}}^2 = [1 \quad \alpha\tilde{\boldsymbol{b}}^*]\tilde{\boldsymbol{R}}_{x|y}\begin{bmatrix}1\\ \alpha\tilde{\boldsymbol{b}}\end{bmatrix} \tag{16}$$

where $\tilde{\boldsymbol{b}} = [b_1\ b_2\ \cdots\ b_N]^T$ and $\tilde{\boldsymbol{R}}_{x|y} = \boldsymbol{R}_{xx} - (2\boldsymbol{R}_{xy} - \overline{\boldsymbol{R}}_{xy}\overline{\boldsymbol{R}}_{yy}^{-1}\boldsymbol{R}_{yy})\overline{\boldsymbol{R}}_{yy}^{-1}\overline{\boldsymbol{R}}_{yx}$. Here $\boldsymbol{R}_{yy}$ and $\boldsymbol{R}_{xy}$ are computed from the input/output relationship in (2).

If we partition $\tilde{\boldsymbol{R}}_{x|y}$ into

$$\tilde{\boldsymbol{R}}_{x|y} = \begin{bmatrix}c & \boldsymbol{e}^*\\ \boldsymbol{d} & A\end{bmatrix}. \tag{17}$$

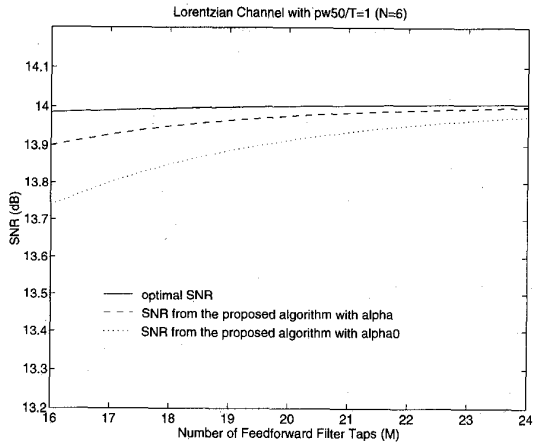

Fig. 3. Performance of the fast algorithm in Lorentzian channel with $N = 6$.

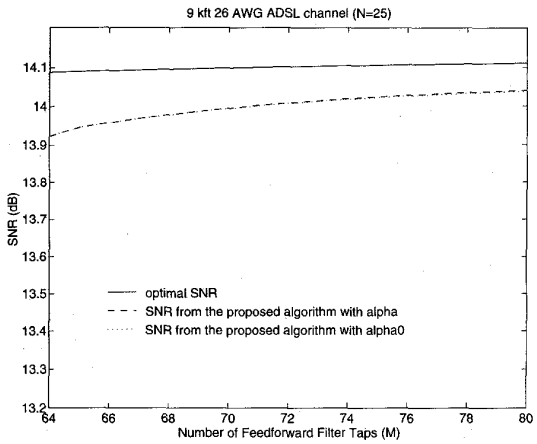

Fig. 4. Performance of the fast algorithm in ADSL channel with $N = 25$.

The optimal α that minimizes the mean square error can be found by differentiating equation (16) with respect to α:

$$\alpha = -\frac{\tilde{\boldsymbol{b}}^* \boldsymbol{d} + \boldsymbol{e}^* \tilde{\boldsymbol{b}}}{2\tilde{\boldsymbol{b}}^* A \tilde{\boldsymbol{b}}}.$$

Still we do not need to invert a matrix to compute α. Then $[1 \quad \alpha\tilde{\boldsymbol{b}}^*]$ is used instead of $\boldsymbol{b}^*$. A simpler scaling constant α_0 can be derived if $\overline{\boldsymbol{R}}_{x|y}$ of (7) is used instead of $\tilde{\boldsymbol{R}}_{x|y}$ in (16). Using the same partition of $\overline{\boldsymbol{R}}_{x|y}$ as in (17), it can be shown that

$$\alpha_0 = -\frac{\tilde{\boldsymbol{b}}^* \boldsymbol{d}}{\tilde{\boldsymbol{b}}^* A \tilde{\boldsymbol{b}}} = -\frac{\tilde{\boldsymbol{b}}^* \boldsymbol{d}}{\sum_{i=0}^{M-1} (|\tilde{B}_i|^2 / \overline{B}_i)}$$

where

$$\boldsymbol{d}^* = [d_1 \quad d_2 \quad \cdots \quad d_N], \quad d_k = \frac{1}{M'} \sum_{i=0}^{M'-1} \frac{1}{\overline{B}_i} e^{j2\pi ik/M},$$

$\overline{B}_i$ is defined in (11), and $\tilde{B}_i$ is the ith DFT element of $[0 \quad \cdots \quad 0 \quad b_1 \quad 0 \quad \cdots \quad 0 \quad b_2 \quad \cdots \quad b_N \quad 0 \quad \cdots \quad 0]$.

Compared with α, α_0 can be obtained with only DFT operations with little expense of performance. Typical values of α and α_0 range from 1.1 to 1.8. Simple inspection of (14) will reveal that scaling $\boldsymbol{b}$ by α has the same effect as having scaled σ_n^2. This indicates that the perfect knowledge of σ_n^2 is not required. Comparison between using α and α_0 is made in the following section.

In summary, the fast equalization for the symbol-spaced MMSE-DFE is carried out as follows:

1) Compute $\overline{\mathcal{E}}_x |H_i|^2 + \sigma_n^2$ from the channel estimate.
2) Compute $\boldsymbol{b}$ using the IDFT as described in (12). The scaling factor α or α_0 can be used to obtain a better solution.
3) Using $\boldsymbol{b}$ computed in the above step, perform the IDFT of $\overline{\mathcal{E}}_x H_i^* P_{\Delta,i} \cdot B_i / (\overline{\mathcal{E}}_x |H_i|^2 + \sigma_n^2)$ to get $\boldsymbol{w}$ as in (15).

In the following section, simulation results using the fast algorithm are given.

IV. Simulations

In this section, we show the simulation results performed by the fast algorithm described in the previous section. Performance is computed assuming that the channel pulse response and the noise variance are given. Two channels are used in these simulations. The first channel is a magnetic recording channel modelled using a Lorentzian step response

$$s(t) = \frac{1}{1 + (2t/pw_{50})^2}$$

where pw_{50} is the width of the pulse at 50% of its peak value. The pulse response, h_k, in (1) is obtained from $s(t)$ using $h_k = s_k - s_{k-1}$ where $s_k = s(kT)$.

The channel pulse response with $pw_{50}/T = 1$ is used in the simulation. The simulation result is shown in Fig. 3. In this simulation, a feedback filter with 6 taps is used, and the feedforward filter taps are changed as indicated in x-axis. Decision delay, Δ, is optimized to get the best SNR, and matched filter bound (MFB) $SNR_{\text{MFB}} = ||h||^2 \overline{\mathcal{E}}_x / \sigma_n^2$ is set to 15 dB. The solid line represents the SNR of the DFE computed from the optimal $\boldsymbol{b}$ and $\boldsymbol{w}$, and the dashed line indicates the SNR computed from the fast algorithm with $\boldsymbol{b}$ scaled by α, and the dotted line shows the SNR computed from the fast algorithm with $\boldsymbol{b}$ calibrated by α_0.

We next use a channel that models a 9 kft 26 AWG asynchronous digital subscriber loop (ADSL) sampled at 640 kHz. The result is shown in Fig. 4. Similarly, the length of the feedback filter is set to 25 and the number of the feedforward filter taps is represented on the x-axis.

The gap between the optimal SNR and the SNR from the fast algorithm in both plots approaches zero as the number of feedforward filter taps increases, as expected. When $\boldsymbol{b}$ is scaled by α_0, performance is close to that scaled by α. It is clear from both plots that the fast algorithm generates the equalizer settings approaching the optimal solution within a tenth of a dB.

In Fig. 5, simulation result performed on 400 m 26 gauge loop channel is shown with different MFB ranging from 10–30 dB. Still, the proposed algorithm yields SNR very close to the optimal value with different MFB values.

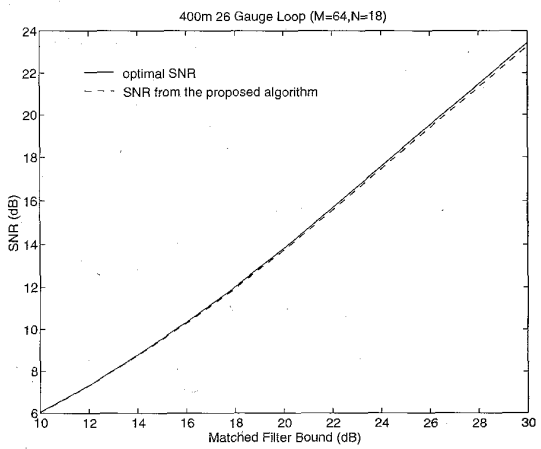

Fig. 5. Performance of the proposed algorithm with different MFB.

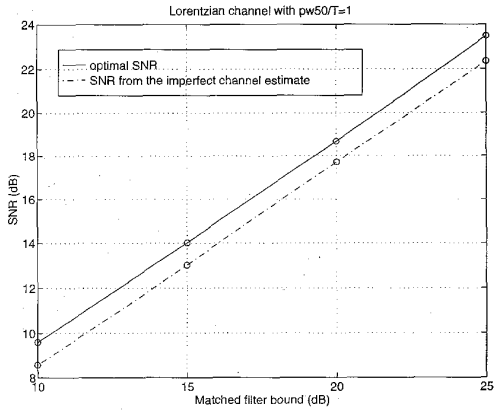

Fig. 6. The effect of the imperfect channel estimation.

While we do not address the channel estimation problem in this paper, the effect of the channel estimation error is briefly analyzed in Fig. 6. Again, the Lorentzian pulse response with $pw_{50}/T = 1$ is used with $M = 24$ and $N = 6$, and the optimized α is used when computing the proposed algorithm. The solid line represents the SNR of the optimal DFE with the perfect pulse response, and the dashed line indicates the SNR of the proposed algorithm with the imperfect channel estimate. In this simulation, the channel response is estimated using the noniterative algorithm in [12] with the averaging factor equal to 10. This requires 80 symbol periods for the channel estimation. The plot shows that the proposed algorithm still yields performance within 1 dB of the optimum solution. As the number of averaging times increases, performance of the fast algorithm approaches the optimal case. A more complete analysis of the channel mis-estimation effect would be an interesting topic.

In Fig. 7, the finite precision effect is investigated. The same channel response as in Fig. 6 is used and the matched filter bound is set to 25 dB. The large eigenvalue spread for the DFE requires high precision arithmetic when direct matrix inversion is used to compute the DFE coefficients in (4) and (5). The plot shows the SNR difference between the infinite precision

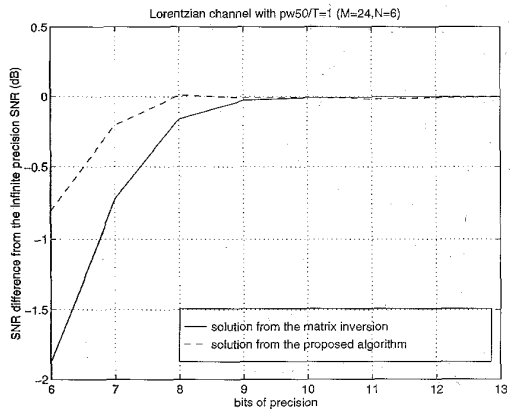

Fig. 7. The effect of finite precision arithmetic.

solution and the finite precision case. The plot indicates that the proposed algorithm is much less sensitive to the finite precision, compared to the solution using matrix inversion. The former starts to show degraded performance at a lower precision than the latter. This is not surprising since the direct matrix inversion is numerically sensitive.

V. Conclusions

We have proposed a novel fast algorithm for the minimum mean square error decision feedback equalizer. Based on channel estimates, the fast algorithm computes the equalizer settings using the DFT and IDFT very efficiently. The overall computation can be carried out without a matrix operation with negligible performance loss as the number of the feedforward filter taps increases. Simulations performed in a magnetic recording channel and ADSL channel show that the fast algorithm yields the near-optimal settings very efficiently. The proposed algorithm is shown to be robust to finite precision arithmetic compared to the direct matrix inversion solution.

Acknowledgment

The authors would like to thank the reviewers for their helpful remarks.

References

[1] J. Cioffi, G. Dudevoir, M. Eyuboglu, and G. D. Forney, "Minimum mean-square-error decision feedback equalization and coding—Part I," *IEEE Trans. Commun.*, vol. 43, no. 10, pp. 2582–2594, Oct. 1995.
[2] C. A. Belfiore and J. H. Park, "Decision feedback equalization," in *Proc. IEEE*, vol. 67, no. 8, pp. 1143–1156, Aug. 1979.
[3] J. Salz, "Optimum mean-square decision feedback equalization," *Bell Syst. Tech. J.*, vol. 52, no. 8, pp. 1341–1373, Oct. 1973.
[4] S. U. H. Qureshi, "Adaptive equalization," in *Proc. IEEE*, vol. 73, pp. 1349–1387, Sept. 1985.
[5] J. S. Chow, private communication, Amati Corporation, 1993.
[6] N. M. W. Al-Dhahir and J. M. Cioffi, "Fast algorithms for the computation of the finite length decision feedback equalizer," in *ICASSP*, Mar. 1992.
[7] R. Ziegler, N. Al-Dhahir, and J. Cioffi, "Nonrecursive adaptive decision feedback equalization from channel estimates," in *IEEE 42nd VTS Conf.*, vol. 2, pp. 600–603, May 1992.
[8] P. Butler and A. Cantoni, "Noniterative automatic equalization," *IEEE Trans. Commun.*, vol. COM-23, pp. 621–633, June 1975.
[9] S. A. Fechtel and H. Meyer, "An investigation of channel estimation and equalization techniques for moderately rapid fading HF-channels," in *Proc. of ICC*, June 1991.

[10] L. Ljung, *System Identification*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
[11] J. M. Cioffi, "Channel identification for high-speed voiceband modems," preprint, Mar. 1991.
[12] K. A. Hamied, M. Rahman, and M. S. El-Hennawey, "A new channel estimator for fast start-up equalization," *IEEE Trans. Commun.*, vol. 39, pp. 177–181, Feb. 1991.
[13] A. Milewski, "Periodic sequences with optimal properties for channel estimation and fast start-up equalization," *IBM J. Res. Develop.*, vol. 27, pp. 426–431, Sept. 1983.
[14] K. H. Mueller and D. A. Spaulding, "Cyclic equalization—A new rapidly converging equalization technique for synchronous data communication," *Bell Syst. Tech. J.*, vol. 54, pp. 369–406, Feb. 1975.
[15] P. R. Chevillat, D. Maiwald, and G. Ungerboeck, "Rapid training of a voiceband data-modem receiver employing an equalizer with fractional-T spaced coefficients," *IEEE Trans. Commun.*, vol. COM-35, pp. 869–876, Sept. 1987.
[16] S. U. H. Qureshi, "Fast start-up equalization with periodic training sequences," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 553–563, Sept. 1977.
[17] H. Sari, "Simplified algorithms for adaptive channel equalization," *Philips J. Res.*, vol. 37, pp. 56–77, 1982.
[18] P. A. Voois, I. Lee, and J. M. Cioffi, "The effect of decision delay in finite-length decision feedback equalization," submitted to *IEEE Trans. Inform. Theory*, Mar. 1993.
[19] N. M. W. Al-Dhahir and J. M. Cioffi, "MMSE-DFE: Finite length results," submitted to *IEEE Trans. Inform. Theory*, Aug. 1993.
[20] M. Abdulrahman and D. D. Falconer, "Cyclostationary crosstalk suppression by decision feedback equalization on digital subscriber loops," *IEEE J. Selected Areas Commun.*, vol. 10, pp. 640–649, Apr. 1992.
[21] R. Gray, "On the asymptotic eigenvalue distribution of toeplitz matrices," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 725–730, Nov. 1972.
[22] N. Al-Dhahir, A. Sayed, and J. Cioffi, "A high performance cost-effective pole-zero MMSE-DFE," in *Allerton Conf. on Commun., Control and Computing*, Sept. 1993, pp. 1166–1175.
[23] J. M. Cioffi and J. A. C. Bingham, "A data-driven multitone echo canceller," *IEEE Trans. Commun.*, vol. 42, pp. 2853–2869, Mar. 1994.
[24] ITU-T (CCITT) Recommendation V.34, A modem operating at data signaling rates of up to 28800 bit/s for use on the general switched telephone network and on leased point-to-point 2-wire telephone-type circuits, 1994.

Inkyu Lee (S'91) was born in Seoul, Korea, on December 26, 1967. He received the B.S. degree in control and instrumentation engineering from Seoul National University, Seoul, Korea, in 1990. He received the M.S. and Ph.D degrees in electrical engineering from Stanford University, Stanford, CA, in 1992 and 1995, respectively.

From 1991 to 1995, he was a research assistant at the Information Systems Laboratory at Stanford University. He is now with AT&T Bell Laboratories. His current research interests include digital communications and signal processing techniques applied to digital transmission and storage systems.

John M. Cioffi (S'77–M'78–SM'91), for photograph and biography, see p. 2594 of the October issue of this TRANSACTIONS.