A Restructured Decision Feedback Equalizer for Facilitating the LMS Algorithm

Cory S. Modlin and John M. Cioffi
Information Systems Laboratory, Stanford University, Stanford CA 94305-4055

Abstract— *The least-mean-square (LMS) algorithm used to adapt the feedforward and feedback filters of a decision feedback equalizer (DFE) is often poorly conditioned. The result is that convergence is slow and misadjustment large. We propose a restructuring of the DFE that will improve the performance of the adaptive algorithm without changing the efficacy of the equalizer. In particular, we propose two similar designs neither of which requires a matrix inversion or matrix multiplication. One takes advantage of some a priori knowledge of the channel and the other draws on simple channel identification as part of the adaptive process to facilitate the algorithm.*

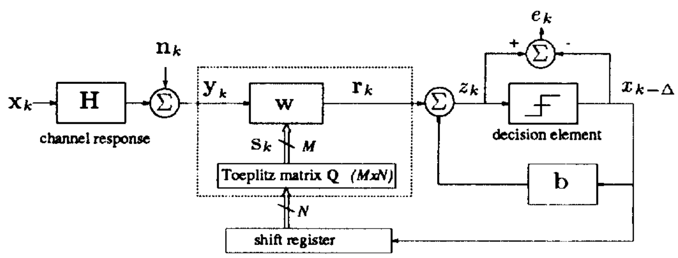

Fig. 1. DFE with addtional feedback loop

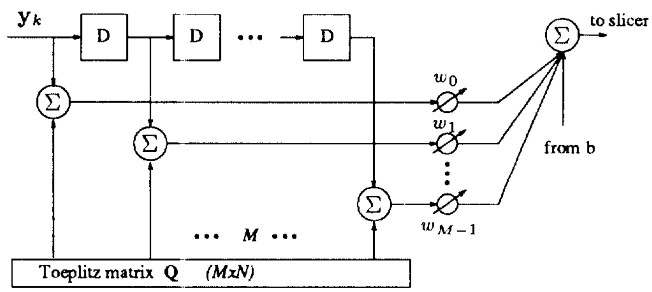

Fig. 2. Detail of addtional feedback loop

I. Introduction

In communications applications favoring simplicity and speed, a decision feedback equalizer (DFE) is an attractive choice. The least-mean-square (LMS) adaptive algorithm is often used in conjunction with a DFE to tune the feedforward and feedback filters, both in cases where there is some previous knowledge of the channel and when the channel is unknown. Because of the structure of the DFE, the LMS algorithm is often slow to converge and suffers from large misadjustment [1]. We propose an alteration to this structure that does not affect the capability of the equalizer itself, adds only slightly to the complexity, and offers greatly improved conditioning of the adaptive process.

For an adaptive DFE, the feedforward and feedback filters are jointly optimized. Although there are often two distinct update expressions for the two filters, these equations are not independent and this is one of the primary reasons for slow convergence and sizable excess mean square error. With the notation in Figure 1, the LMS update equations for the DFE are often written

$$\mathbf{w}_{k+1} = \mathbf{w}_k + 2\mu_1 e_k \mathbf{y}_k, \quad (1)$$

$$\mathbf{b}_{k+1} = \mathbf{b}_k + 2\mu_2 e_k \mathbf{x}_{k-\Delta-1}, \quad (2)$$

where $\mathbf{w}_k$ and $\mathbf{b}_k$ are the feedforward and feedback filters at time k, $\mathbf{y}_k$ is the noisy channel output, $\mathbf{x}_{k-\Delta-1}$ is a vector or previous, assumed correct decisions, and e_k is the equalizer error. The speed and stability are governed by two constants μ_1 and μ_2. In writing two equations, we take advantage of some degree of independence between the two filters, but the factor e_k shared by these two equations causes a complex interaction that degrades performance.

The LMS algorithm is a victim of noisy gradient estimates and unequal sensitivity to this noise among adapting filter coefficients. This disparity in sensitivity can be seen, mathematically, in the spread of the eigenvalues of the autocorrelation matrix, $\mathbf{R}$.

A number of methods have been proposed as a way to reduce the spread of the eigenvalues. Most of these involve a transform to orthogonalize the input signals [2] [3] [4]. A Discrete Fourier Transform (DFT) or

This work is supported by IBM Corporation, Samsung, Philips Semiconductors, and Quantum Corporation.

1058-6393/95 $4.00 © 1995 IEEE

a Discrete Cosine Transform (DCT), for example, may be applied to the input signal and the filters are adapted in the transform domain. The most serious drawback is the added complexity of the transform.

Another way to affect the eigenvalues is to replace the adaptation constant, μ with a multiple of the inverse of the autocorrelation matrix [5]. This way, the effective eigenvalues are equal. Because we frequently do not know what the autocorrelation matrix is in advance and because a full matrix inversion is often too complicated, Mikhael *et. al.* [6] proposed replacing μ with a slowly adapting diagonal matrix.

We propose a way to take advantage of the previous decisions available in a DFE; we add another feedback path to decouple the interaction between the two filters as they adapt [7]. As we will prove, the new feedback path will not change the functionality of the DFE in any way regardless of how it is set; it is there only to help condition the adaptive process. In many practical instances, it can completely replace the original feedback filter, $\mathbf{b}$.

We provide a basis for this claim by investigating how the additional feedback affects the optimal finite length DFE settings. Then we examine two possible implementations, one for an asymmetric digital subscriber loop (ADSL) channel where we presuppose that the input spectrum is nearly flat and that the channel is unknown, and one for the magnetic recording channel where we assume a coded input sequence and initial conditions that closely reflect the actual channel characteristics. Finally we discuss some practical circuit design techniques.

II. Optimal DFE Solution

We derive the optimal settings for a finite length DFE with an additional feedback path. As we will show, the value of the matrix, $\mathbf{Q}$, shown in Figures 1 and 2 will not affect the efficacy of the equalizer. The feedback filter, $\mathbf{b}$, will simply change its value to counteract the effects of $\mathbf{Q}$.

Intuitively, the feedback is now shared by two paths. Neither the optimal feedforward filter, $\mathbf{w}^*$, nor the equalizer output, z_k are altered. There is no change in the influence of error propagation. Our goal is to shift the burden of feedback from $\mathbf{b}$ to $\mathbf{Q}$ and at the same time to improve the conditioning of the LMS algorithm.

Referring again to Figure 1, the symbol spaced, sampled channel output, y_k, is

$$y_k = \sum_m x_m h_{k-m} + n_k$$

and, with the contribution from the new feedback loop, s_k, the FIR filter output, r_k, is

$$r_k = y_k + s_k$$

Then for M successive samples of r, this can be written with matrices as $\mathbf{r}_k = \mathbf{H}\mathbf{x}_k + \mathbf{n}_k + \mathbf{s}_k$, where

$$\mathbf{r}_k = \begin{bmatrix} r_k \\ r_{k-1} \\ \vdots \\ r_{k-M+1} \end{bmatrix}, \mathbf{x}_k = \begin{bmatrix} x_k \\ x_{k-1} \\ \vdots \\ x_{k-M-L+2} \end{bmatrix}, \mathbf{n}_k = \begin{bmatrix} n_k \\ n_{k-1} \\ \vdots \\ n_{k-M+1} \end{bmatrix},$$

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & \cdots & h_{L-1} & 0 & \cdots & 0 \\ 0 & h_0 & h_1 & \cdots & h_{L-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_{L-1} \end{bmatrix}, \quad (3)$$

and $\mathbf{s}_k = \mathbf{Q}\mathbf{x}_{k-\Delta-1}$ where $\mathbf{Q}$ is any constant $M \times N$ matrix and $\mathbf{x}^T_{k-\Delta-1} = [x_{k-\Delta-1}\, x_{k-\Delta-2} \ldots x_{k-\Delta-N}]$.

If the feedforward filter is $\mathbf{w} = [w_0\, w_1 \ldots w_{M-1}]$, and the feedback filter is $\mathbf{b} = [b_0\, b_1 \ldots b_{N-1}]$, the equalizer error is

$$e_k = x_{k-\Delta} - \mathbf{v}\mathbf{u}_k, \quad (4)$$

where

$$\mathbf{v} = [\mathbf{w}\ \ \mathbf{b}], \text{and } \mathbf{u}_k = \begin{bmatrix} \mathbf{y}_k + \mathbf{Q}\mathbf{x}_{k-\Delta-1} \\ \mathbf{x}_{k-\Delta-1} \end{bmatrix}. \quad (5)$$

Setting the square autocorrelation matrix,

$$\mathbf{R} = E\left[\mathbf{u}_k\mathbf{u}_k^T\right] = \left[\begin{array}{c|c} \mathbf{R}_{11} + \mathbf{Q}\mathbf{R}_x\mathbf{Q}^T + \mathbf{R}_{12}\mathbf{Q}^T + \mathbf{Q}\mathbf{R}_{12}^T & \mathbf{R}_{12} + \mathbf{Q}\mathbf{R}_x \\ \hline \mathbf{R}_{12}^T + \mathbf{R}_x\mathbf{Q}^T & \mathbf{R}_x \end{array}\right], \quad (6)$$

where $\mathbf{R}_{12} = E[\mathbf{y}_k\mathbf{x}^T_{k-\Delta-1}]$, $\mathbf{R}_{11} = E[\mathbf{y}_k\mathbf{y}_k^T]$, and $\mathbf{R}_x$ is an $N \times N$ input autocorrelation matrix, and a column vector,

$$\mathbf{P} = E\left[x_{k-\Delta}\mathbf{u}_k\right] = \left[E[x_{k-\Delta}\mathbf{r}_k] \mid E[\mathbf{x}_{k-\Delta-1}x_{k-\Delta}]\right]^T, \quad (7)$$

the familiar solution that minimizes the average squared error is [1]

$$\mathbf{v} = \mathbf{R}^{-1}\mathbf{P}.$$

Now we can explain why the performance of the equalizer and $\mathbf{w}^*$ do not depend on $\mathbf{Q}$. In this structure, trailing intersymbol interference is subtracted by filtering previous decisions by $\mathbf{b}$ and by $\mathbf{w}\mathbf{Q}$; this is evident when we rewrite equation 4 as

$$e_k = x_{k-\Delta} - [\mathbf{w} \mid \mathbf{b} + \mathbf{w}\mathbf{Q}] \left[\frac{\mathbf{H}\mathbf{x}_k + \mathbf{n}_k}{\mathbf{x}_{k-\Delta-1}}\right].$$

Letting $\mathbf{b}^*_{\mathbf{Q}}$ represent the optimal feedback filter when there is no additional feedback ($\mathbf{Q} = \mathbf{0}$), we see in this

	misadjustment		eigenvalue spread	
	Q	no Q	Q	no Q
ADSL	1.32	1.18	6.1e3	1.8e6
Disk	.074	.074	789	2009

TABLE I
Misadjustment and Eigenvalue Spread Results

form of the expression for equalizer error that $\mathbf{b}^*_{\bar{\mathbf{Q}}} = \mathbf{b} + \mathbf{wQ}$ or $\mathbf{b} = \mathbf{b}^*_{\bar{\mathbf{Q}}} - \mathbf{wQ}$. The new filter, $\mathbf{b}$, can always be set to bring the total contribution from feedback back to $\mathbf{b}^*_{\bar{\mathbf{Q}}}$ regardless of $\mathbf{Q}$ or $\mathbf{w}^*$. The contribution from feedback, then, has not changed, so there is no reason that the optimal feedforward filter should change either.

Knowing that $\mathbf{Q}$ will not affect the performance of the equalizer, we are free to choose values that will enhance the celerity and accuracy of the adaptive process. An attempt to find an optimal $\mathbf{Q}$ with respect to the stochastic gradient algorithm would be fraught with difficulty, but the expression for the autocorrelation matrix, $\mathbf{R}$, offers guidance.

We set $\mathbf{Q}$ to

$$\mathbf{Q} = -\mathbf{R}_{12}\mathbf{R}_x^{-1} \tag{8}$$

so that $\mathbf{R}_{12} + \mathbf{QR}_x$ is zero making the autocorrelation matrix, $\mathbf{R}$, block diagonal. The inverse of such a matrix is the inverse of each block [8]. In the top left block, the terms involving $\mathbf{Q}$ will cancel each other leaving a problem that is similar to that of a linear equalizer. The other block is the input autocorrelation matrix, $\mathbf{R}_x$, which is usually nearly equal to the identity.

Some nice features are evident when the input sequence is independent and identically distributed (i.i.d.). The $\mathbf{Q}$ matrix is then toeplitz and is N columns of the channel matrix, $\mathbf{H}$, reversed in sign. Which N columns depends on Δ. Also, the feedback filter, $\mathbf{b}$, diminishes to zero. This is true because the bottom left block of $\mathbf{R}^{-1}$ and the bottom block of $\mathbf{P}$ are zero when $E[x_k x_{k-i}] = 0$ for $i \neq 0$.

These features suggest some possible applications. If the inputs are nearly or completely uncorrelated, the values for $\mathbf{Q}$ can be retrieved by any method of channel identification and the usual update can be applied to $\mathbf{w}$ using the equalizer error (equation 1). If the inputs are more highly correlated and we have an initial channel estimate, we can fix $\mathbf{Q}$ to a close toeplitz matrix and allow both filters to adapt normally.

III. ADSL Example

One way to apply the restructured DFE is by performing channel identification to find the toeplitz $\mathbf{Q}$ matrix, and then using this and the equalizer error to adapt the feedforward filter. We applied the LMS algorithm for channel identification; the relevant equations are

$$\begin{aligned} e_k^c &= y_k - \mathbf{p}_k\mathbf{x}_k \text{ and} \\ \mathbf{p}_{k+1} &= \mathbf{p}_k + 2\mu_c e_k^c \mathbf{x}_k \end{aligned}$$

where y_k is the noisy channel output, $\mathbf{p}_k$ is an estimate of the channel, and e_k^c is the error caused by noise and any error in the estimate of the channel. The $\mathbf{Q}$ matrix at time k is comprised entirely of an $M + N - 1$ segment or window of $\mathbf{p}_k$ as determined by Δ. We can do this windowing in time because the inputs are assumed i.i.d. This adaptation is very fast since the autocorrelation matrix governing this process is the identity and all of the eigenvalues are one. The feedforward filter is still updated as in equation 1 where the equalizer error involves the contribution from $\mathbf{Q}_k$.

To illustrate this implementation, we use a channel that models a 9 Kft copper wire with three bridge taps serving as a carrier for ADSL service.[1] For this example, we selected a 4-PAM input constellation and set the matched filter bound to $35dB$. The feedforward and feedback filters had 20 and 25 taps respectively.

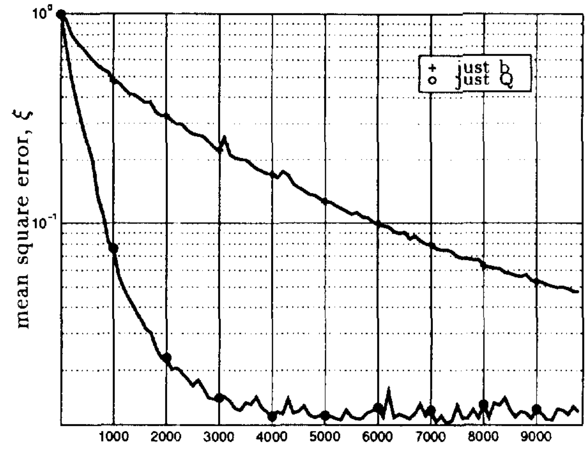

Fig. 3. Convergence Behavior on ADSL Channel

In Figure 3 we compare the restructured DFE with a conventional DFE with simulation assuming that correct decisions are always available and zero initial conditions. The curves each represent the expected squared error at each iteration given the current filter settings. Slightly different than a learning curve, these curves represent what we would expect the squared error to be if at iteration k we were to stop the adaptation process or apply "gear shifting" to continue to adapt much more

[1] We are grateful to Dr. Peter S. Chow of Amati Corporation for providing this data.

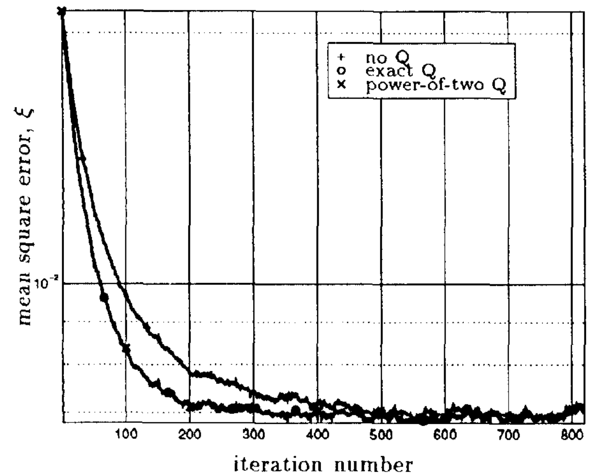

Fig. 4. Convergence Behavior on Magnetic Recording Channel

slowly. We derive these numbers from the relation [1]

$$\xi = \xi_{\text{min}} + (\mathbf{w}_k - \mathbf{w}^*)\mathbf{R}(\mathbf{w}_k - \mathbf{w}^*)^T$$

where ξ is the mean square error and $\mathbf{R}$ is the autocorrelation matrix in equation 6. The filters were adapted using equations 1 and 2 with the traditional structure. The constants for both structures were chosen to maximize convergence speed; they were as large as stability constraints would allow. Table I lists the misadjustment, defined as the excess mean square error divided by the minimum mean square error, for both methods and the eigenvalue spread of $\mathbf{R}$. Even though the new structure converges much more rapidly, it hardly suffers from additional misadjustment. This suggests that the new structure can be set to track more quickly and more accurately than is typically possible.

IV. Magnetic Recording Example

In magnetic recording, run-length codes typically correlate the binary input sequence and full channel identification is often impractical. But it is also true that processing during manufacturing can be done to learn the channel characteristics at a number of different radii or zones on the disk [9] and that these characteristics can be loaded into the detector while the head switches tracks. The adaptive algorithm is then used to track small variations in the head and media.

With this application in mind, we fix the $\mathbf{Q}$ matrix to a toeplitz matrix nearly mirroring the one we find using equation 8 and allow the feedforward and feedback filters to adapt normally. The $\mathbf{Q}$ matrix found using equation 8 will not be toeplitz because of the run-length code. Further simplifying the implementation,

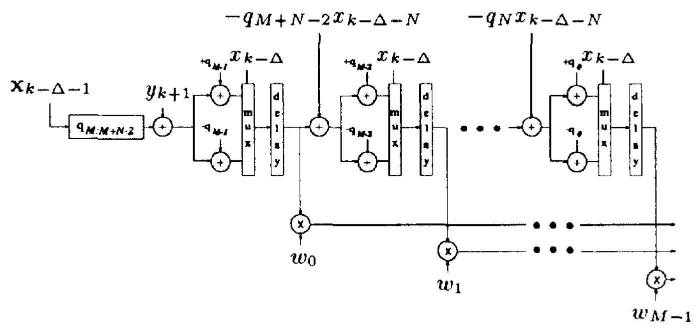

Fig. 5. Circuit Design

we replace each value in $\mathbf{Q}$ with 2^{-n} where n is some integer so that 2^{-n} is close to the original value. For example, if an entry in $\mathbf{Q}$ were .271, we would set $n = 2$ and replace this value with .25. The values for $\mathbf{Q}$, $\mathbf{w}$ and $\mathbf{b}$ are all initialized according to an initial channel estimate.

In Figure 4 we compare three simulations, one with the traditional implementation and two using the new DFE structure. For the traditional structure, $\mathbf{w}$ and $\mathbf{b}$ were initialized according to our initial channel estimate. With additional feedback, one trial evaluated the effect of the toeplitz, power-of-two approximation for $\mathbf{Q}$ comparing it with $\mathbf{Q}$ set as in equation 8. The effect of the power-of-two approximation is not noticeable.

For these trials, we used a linear Lorentzian model of the magnetic recording channel completely determined by its only parameter, the recording density measured in terms of PW_{50}/T where T is the symbol period and PW_{50} is the width of an isolated pulse at half of its peak amplitude. We assumed that the equalizer coefficients were optimized at a density $PW_{50} = 2T$ and that the actual operating density was instead $PW_{50} = 2.3T$. The number of feedforward and feedback taps were 15 and eight respectively and the matched filter bound was $25dB$. A rate 8/9, (0,4) run-length code [10] was applied to the input sequence. Again the curves represent the expected squared error at each iteration and Table I lists the misadjustment and eigenvalue spread. Here, a target misadjustment was chosen and the adaptive update constants changed to meet this constraint.

V. Circuit Design Issues

We offer some circuit design suggestions to emphasize the simplicity of this new structure and to add insight into how it works. A way to handle the dependency of the next decision on the previous decision is also included.

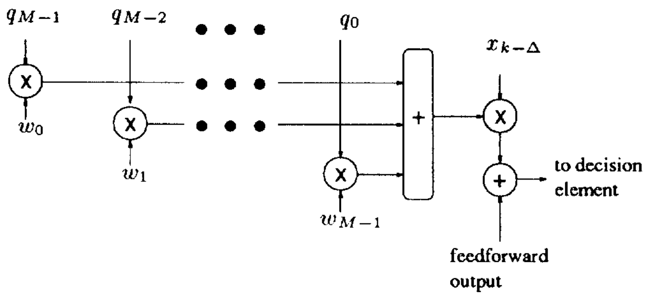

Fig. 6. To Remove Dependency on Most Recent Decision

We begin by writing $\mathbf{Q}$ as

$$\mathbf{Q} = \begin{bmatrix} q_{M-1} & q_M & \cdots & q_{M+N-2} \\ q_{M-2} & q_{M-1} & \ddots & q_{M+N-3} \\ \vdots & \ddots & \ddots & \vdots \\ q_0 & \cdots & q_{N-2} & q_{N-1} \end{bmatrix}.$$

The output of the feedforward filter, expressed in terms of these elements, is

$$\begin{aligned} r_k = & \\ & w_0(y_k + q_{M-1}x_{k-\Delta-1} + \cdots + q_{M+N-2}x_{k-\Delta-N}) + \\ & w_1(y_{k-1} + q_{M-2}x_{k-\Delta-1} + \cdots + q_{M+N-3}x_{k-\Delta-N}) + \\ & \vdots \\ & w_{M-1}(y_{k-M+1} + q_0x_{k-\Delta-1} + \cdots + q_Nx_{k-\Delta-N}) \end{aligned}$$

Defining $d_k = y_k + q_{M-1}x_{k-\Delta-1} + \cdots + q_{M+N-2}x_{k-\Delta-N}$, we can rewrite the feedforward filter output as

$$\begin{aligned} r_k = & w_0(d_k) + \\ & w_1(d_{k-1} + q_{M-2}x_{k-\Delta-1} - q_{M+N-2}x_{k-\Delta-N-1}) + \\ & \vdots \\ & w_{M-1}(d_{k-M+1} + q_0x_{k-\Delta-1} - q_Nx_{k-\Delta-N-1}) \end{aligned} \quad (9)$$

Implementation then involves forming d_k and the correction terms that are needed because there are not $M-1$ zeros in the bottom of the first column or at the end of the first row of $\mathbf{Q}$.

In Figure 5, a circuit design is illustrated assuming a binary input sequence although generalizing to more input levels is straightforward. The most recent decision is used to select one of two precomputed values. Three important factors lead to modest added complexity when compared with a traditional DFE structure. The first is the correction terms that are applied for each filter tap. Second, enhanced numerical precision may be needed since many of the terms in equation 9 are added and later subtracted.

Finally, a third obstacle is that the next decision depends on the current output of the FIR filter. We can remove the dependency on the most recent decision by separately computing the the product of $\mathbf{w}$ and the first column of $\mathbf{Q}$ and then selecting a precomputed output with the most recent decision as shown in Figure 6. This computation is not especially time sensitive and may be simplified further if the power-of-two approximation is used.

VI. Conclusion

We have presented a restructured decision feedback equalizer for enhancing the performance of the LMS algorithm. An additional feedback path can help condition the adaptive algorithm without influencing the efficacy of the equalizer. When the input sequence is independent and identically distributed, the new feedback path can be set using channel identification and can replace the original feedback path. In the case of correlated inputs, a power-of-two, toeplitz matrix can serve as the additional feedback. In either case, the added complexity is modest and the convergence improvement may be substantial.

References

[1] Bernard Widrow and Samuel D. Stearns. *Adaptive Signal Processing*. Prentice-Hall, 1985.

[2] K. R. Rao and P. Yip. *Discrete Cosine Transform*. Academic Press, Inc., San Diego, CA, 1990.

[3] S. S. Narayan, A. M. Peterson, and M. J. Narasimha. Transform domain LMS algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 31(3):609–615, June 1983.

[4] M. Dentino, J. McCool, and B. Widrow. Adaptive fitering in the frequency domain. *Proceeding of the IEEE*, December 1978.

[5] K. H. Mueller. A new, fast converging MS algorithm for adaptive equalizers with partial response signaling. *The Bell Systems Technical Journal*, January 1975.

[6] W. B. Mikhael, F. H. Wu, L. G. Kazovsky, G. S. Kang, and L. J. Fransen. Adaptive filters with individual adaptation of parameters. *IEEE Transactions on Circuits and Systems*, pages 677–686, July 1986.

[7] Cory Modlin and John M. Cioffi. A fast decision feedback LMS algorithm using multiple step sizes. In *ICC94*, 1994.

[8] Thomas Kailath. *Linear Systems*. Prentice-Hall, 1980.

[9] Steve Moore. High performance digital read channel for hard disk drives using PRML techniques. In *Custom Integrated Circuits Conference*, 1993.

[10] Kees A. Schouhamer Immink. *Coding Techniques for Digital Recorders*. Prentice-Hall, 1990.