A High-Performance Cost-Effective Pole-Zero MMSE-DFE*

N. Al-Dhahir, A. Sayed and J. Cioffi
Information Systems Laboratory
Stanford University, Stanford CA 94305

Abstract

In this paper, we derive a computationally–efficient algorithm for accurately approximating a long FIR filter by a reduced–parameter pole–zero filter. Our derivation successfully extends the "embedding" approach of [1, 2] to the case of an unequal number of poles and zeros.

Our main emphasis is on applying the algorithm to reduce the implementation complexity of the long FIR feedforward and feedback filters of the MMSE–DFE encountered in digital subscriber loops. We also describe several other applications where the algorithm leads to a significant reduction in implementation complexity.

1 Introduction

In a variety of applications, the engineer is faced with the task of implementing a long finite impulse response (FIR) filter on a digital signal processor. For example, this long FIR filter could be the feedforward or the feedback filter of a decision feedback equalizer (DFE), an echo canceller (EC), a simulator of a channel impulse response (IR), etc. Moreover, in some situations the underlying true system response is of infinite length (IIR). Assuming it to be FIR (e.g., by truncating up to the most significant N samples) has some advantages : An ease of computation (e.g., through the use of efficient time–domain algorithms such as the Levinson algorithm or frequency–domain algorithms such as the FFT algorithm) and a lower sensitivity to finite–precision effects. However, to achieve satisfactory performance (a high decision–point SNR for the DFE, large echo suppression for the EC, and a low model approximation error for the channel IR), a large number of FIR filter taps is usually needed. This can lead to a prohibitive implementation cost in terms of the increased memory needed to store the filter taps and the large record of previous input samples, in addition to the high processing power required to compute the filter output samples through sum–of–products calculations. This cost even multiplies for high–speed applications and for time–varying environments where the filter taps are frequently updated.

In this paper, we study the generic problem of approximating a long FIR filter by a pole–zero filter with a much smaller total (numerator and denominator) number of coefficients. This problem has been investigated by many researchers in the contexts of IIR digital filter design (see, e.g., [3, 4, 5]), echo cancellation (see, e.g., [6, 7]), and more recently zero–forcing decision feedback equalization in [8]. Our main focus will be on the DFE application. However, our approach differs significantly from that of [8] in several aspects. First, the tail of the long FIR filter was assumed in [8] to be accurately modeled by two poles only. This assumption is specific to the High–Bit–Rate Digital Subscriber Loop (HDSL) environment considered in that paper. Although we shall also use an HDSL channel in our simulations, the algorithm that we shall present is quite general and does not make any such assumptions. Second, pre–cursor ISI was assumed to be negligible in [8], and hence the feedforward filter was assumed to be a short FIR filter. We do not make this assumption either since for higher data rates and less benign channel characteristics, as in the Asymmetric Digital Subscriber Loop (ADSL) environment, the feedforward filter must be very long to achieve satisfactory performance. Therefore, we shall attempt to approximate both the feedforward and feedback filters by pole–zero models. Finally, the DFE coefficients were computed in [8] using adaptive IIR algorithms. Again, the environment–specific assumption of a 2–pole model made stability monitoring a simple task, which would not be the case in situations where a 2–pole model is not adequate (e.g. in echo cancellation). Instead, we shall compute the DFE coefficients directly from the available channel and noise estimates using the efficient algorithms of [9]. In case of environment changes, straightforward adaptation is performed on the long FIR filter, which is then converted to a pole–zero filter for a reduced–complexity implementation.

*This work was supported by NSF contract number NCR-9203131, JSEP contract number DAAL03-91-C-0010, NASA contract number NAG2-842.

2 Pole-Zero Modeling of a Long FIR Filter

2.1 Motivation

The FIR MMSE–DFE is a widely–used receiver structure that mitigates the effects of severe ISI and noise to restore communication integrity. As shown in Figure 1, it consists of two FIR filters. A feedforward filter $w(D)$, with N_f taps $\{w_{-\Delta}, \cdots, w_{N_f-1-\Delta}\}$ (where Δ is the decision delay) that combats pre–cursor ISI and noise, and a strictly–causal feedback filter $b(D)$ with N_b taps $\{-b_1, \cdots, -b_{N_b}\}$ that suppresses post–cursor ISI.

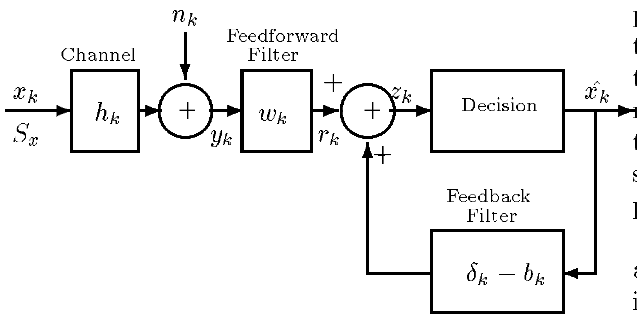

Figure 1: Block diagram of the FIR MMSE–DFE

The channel pulse response $h(D)$ is assumed to be a linear time–invariant FIR filter with a memory of ν, i.e., $h(D) = h_0 + h_1 D + \cdots + h_\nu D^\nu$.

Closed–form expressions for the optimal feedforward and feedback filter settings that minimize the mean square error $MSE \stackrel{def}{=} \mathrm{E}|e_k|^2 = \mathrm{E}|x_{k+N_f-1-\Delta} - z_k|^2$ were derived in [10] for arbitrary choices of N_f and N_b. However, it was shown in [9] that *significant* computational reductions can be achieved when we assume that the number of feedback taps is equal to the channel memory, i.e., $N_b = \nu$. Under this assumption, the feedforward filter becomes strictly anti–causal and $\Delta = N_f - 1$. More specifically, assuming white input with average energy of S_x per complex dimension and correlated–noise with a (non–singular) correlation matrix $\mathbf{R}_{nn}$, then the optimal FIR MMSE–DFE can be computed efficiently from the Cholesky factorization :

$$\frac{1}{S_x}\mathbf{I}_{N_f+\nu} + \mathbf{H}^*\mathbf{R}_{nn}^{-1}\mathbf{H} \stackrel{def}{=} \mathbf{LDL}^* \qquad (1)$$

as follows :

$$\mathbf{b}_{opt.} = \mathbf{L}\mathbf{e}_{N_f} \qquad (2)$$

$$\mathbf{w}^*_{opt.} = d^{-1}_{N_f-1}(\mathbf{e}^*_{N_f}\mathbf{L}^{-1})\mathbf{H}^*, \qquad (3)$$

where $\mathbf{I}_{N_f+\nu}$ is the identity matrix of size $N_f+\nu$, $(.)^*$ denotes the conjugate transpose, $\mathbf{e}_{N_f}$ is the N_f^{th} unit vector, and $\mathbf{H}$ is the $N_f \times (N_f + \nu)$ fully–windowed Toeplitz matrix given by

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & \cdots & h_\nu & 0 & \cdots & 0 \\ 0 & h_0 & h_1 & \cdots & h_\nu & 0 & \cdots \\ \vdots & & & & & & \vdots \\ 0 & \cdots & 0 & h_0 & h_1 & \cdots & h_\nu \end{bmatrix}. \qquad (4)$$

In many applications (such as high–speed data transmission on twisted copper lines that will be discussed in more detail in Section 3), ν is very large, which entails the use of very long feedforward and feedback filters (a total of more than a hundred taps) to achieve satisfactory performance. Although these long filters can be computed very efficiently using (1)-(3), implementing them in real–time is very costly. This consideration has motivated us to develop an efficient algorithm to convert long FIR filters to pole–zero filters with much fewer parameters, without loosing stability, and while still maintaining satisfactory performance.

Another requirement on this algorithm is the ability to handle mixed–phase FIR filters. This is due to the fact shown in [10] that the optimal feedforward (feedback) filters of the FIR MMSE–DFE are not necessarily maximum phase (minimum phase), *unless* N_f *is infinite.*

2.2 The Generalized ARMA-Levinson Algorithm

In this section, we shall derive a *new* algorithm for approximating a long FIR filter by a pole–zero *stable* filter with much fewer taps. The algorithm is a generalization of the ARMA–Levinson algorithm derived in [2] using the "embedding" technique of [1]. The novelty in our algorithm is *its ability to relax the restriction of an equal number of poles and zeros* that was assumed in [2, 1]. This flexibility will prove to be useful in obtaining better fits, as it will become clear from the simulation results of Section 3.

An ARMA model with q zeros and p poles, denoted in this paper by ARMA(p, q), is described by the following difference equation : [1]

$$y_k = -a_1 y_{k-1} - \cdots - a_p y_{k-p} + n_0 x_k + \cdots + n_q x_{k-q}\,. \qquad (5)$$

Assume first a *proper* transfer function so that $p \geq q$. The case of $q > p$ requires some modifications and will be described later.

Our objective is to estimate the $(p + q + 1)$ ARMA parameters $\{a_1, \cdots, a_p, n_0, \cdots, n_q\}$ based on knowledge of the second–order statistics of the

[1] In the sequel, the numerator polynomial coefficients $\{n_0, \cdots, n_q\}$ are not to be confused with the noise samples n_k. Distinction should be clear from the context.

input and output sequences. If we denote these estimates at the j^{th} recursion $(1 \le j \le p)$ by $\{a_1^j, \cdots, a_j^j, n_0^j, \cdots, n_{j-\delta}^j\}$ where $\delta \stackrel{def}{=} p - q \ge 0$, then the output of this ARMA$(j, j-\delta)$ filter can be calculated as follows :

$$\begin{aligned} y_k &= -a_1^j y_{k-1} - \cdots - a_j^j y_{k-j} + n_0^j x_k + \cdots \\ &+ n_{j-\delta}^j x_{k-j+\delta} + e_k^j \end{aligned} \quad (6)$$

where e_k^j is the j^{th} order residual error sequence.

If we define the augmented vector $\mathbf{z}_k \stackrel{def}{=} \begin{bmatrix} y_k \\ x_{k+\delta} \end{bmatrix}$, then the ARMA model of (6) can be converted to the 2–channel AR model :

$$\begin{bmatrix} y_k \\ x_{k+\delta} \end{bmatrix} = \begin{bmatrix} -a_1^j & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_{k-1} \\ x_{k+\delta-1} \end{bmatrix} + \cdots$$
$$+ \begin{bmatrix} -a_\delta^j & n_0^j \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_{k-\delta} \\ x_k \end{bmatrix} + \begin{bmatrix} -a_{\delta+1}^j & n_1^j \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_{k-\delta-1} \\ x_{k-1} \end{bmatrix}$$
$$+ \cdots + \begin{bmatrix} -a_j^j & n_{j-\delta}^j \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_{k-j} \\ x_{k-j+\delta} \end{bmatrix} + \begin{bmatrix} e_k^j \\ x_{k+\delta} \end{bmatrix}, \quad (7)$$

or more compactly,

$$\mathbf{z}_k = -\mathbf{\Theta}_1^j \mathbf{z}_{k-1} - \cdots - \mathbf{\Theta}_j^j \mathbf{z}_{k-j} + \begin{bmatrix} e_k^j \\ x_{k+\delta} \end{bmatrix}, \quad (8)$$

where we have defined

$$-\mathbf{\Theta}_i^j \stackrel{def}{=} \begin{cases} \begin{bmatrix} -a_i^j & 0 \\ 0 & 0 \end{bmatrix} & : 1 \le i \le \delta - 1 \\ \begin{bmatrix} -a_i^j & n_{i-\delta}^j \\ 0 & 0 \end{bmatrix} & : \delta \le i \le j \end{cases} . \quad (9)$$

If we multiply both sides of (8) by $\mathbf{z}_{k-i}^*$ $(0 \le i \le j)$ and take expectations, we arrive at

$$\mathbf{R}(0) + \mathbf{\Theta}_1^j \mathbf{R}(-1) + \cdots + \mathbf{\Theta}_j^j \mathbf{R}(-j) = \mathbf{\Sigma}_j^f \ : \ i = 0 \quad (10)$$

$$\mathbf{R}(i) + \mathbf{\Theta}_1^j \mathbf{R}(i-1) + \cdots + \mathbf{\Theta}_j^j \mathbf{R}(i-j) = \mathbf{0} \ : \ 1 \le i \le j \quad (11)$$

where

$$\begin{aligned} \mathbf{R}(i) &\stackrel{def}{=} \mathrm{E}[\mathbf{z}_k \mathbf{z}_{k-i}^*] = \begin{bmatrix} R_{yy}(i) & R_{yx}(i-\delta) \\ R_{xy}(i+\delta) & R_{xx}(i) \end{bmatrix} \\ &= \mathbf{R}^*(-i) . \end{aligned} \quad (12)$$

Alternatively, (10) and (11) can be written in matrix form as follows :

$$\mathbf{\Theta}^j \mathbf{R}^j = \begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{\Sigma}_j^f \end{bmatrix}, \quad (13)$$

where

$$\mathbf{\Theta}^j = \begin{bmatrix} \mathbf{\Theta}_j^j & \cdots & \mathbf{\Theta}_1^j & \mathbf{I} \end{bmatrix}$$

$$\mathbf{R}^j = \begin{bmatrix} \mathbf{R}(0) & \cdots & \mathbf{R}(-(j-1)) & \mathbf{R}(-j) \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{R}(j-1) & \cdots & \mathbf{R}(0) & \mathbf{R}(-1) \\ \mathbf{R}(j) & \cdots & \mathbf{R}(1) & \mathbf{R}(0) \end{bmatrix} .$$

Equation (13) describes a j^{th}–order AR model whose vector parameters $\mathbf{\Theta}_i^j$ can be calculated by solving a block–Toeplitz Hermitian system of linear equations. This can be done efficiently using the following multichannel form of the scalar Levinson Algorithm, sometimes known as the Levinson–Wiggins–Robinson (LWR) Algorithm[11, 12].

Algorithm 1 (Generalized ARMA-Levinson)
Given $\{\mathbf{R}(0), \cdots, \mathbf{R}(p)\}$

Initial Conditions :

$$\begin{aligned} \mathbf{\Theta}_1^1 &= \mathbf{K}_1^f = -\mathbf{R}^{-1}(0)\mathbf{R}(1) . \\ \mathbf{\Phi}_1^1 &= \mathbf{K}_1^b = -\mathbf{R}^{-1}(0)\mathbf{R}(-1) . \\ \mathbf{\Sigma}_1^f &= \mathbf{R}(0)(\mathbf{I} - \mathbf{K}_1^b \mathbf{K}_1^f) . \\ \mathbf{\Sigma}_1^b &= \mathbf{R}(0)(\mathbf{I} - \mathbf{K}_1^f \mathbf{K}_1^b) . \end{aligned}$$

Recursions :
for $1 \le j \le p-1$

$$\begin{aligned} \mathbf{\Delta}_{j+1}^f &= \mathbf{R}(j+1) + \mathbf{R}(j)\mathbf{\Theta}_1^j + \cdots + \mathbf{R}(1)\mathbf{\Theta}_j^j . \\ \mathbf{K}_{j+1}^f &= -(\mathbf{\Sigma}_j^b)^{-1} \mathbf{\Delta}_{j+1}^f . \\ \mathbf{K}_{j+1}^b &= -(\mathbf{\Sigma}_j^f)^{-1} \mathbf{\Delta}_{j+1}^{*f} . \\ \mathbf{\Sigma}_{j+1}^f &= \mathbf{\Sigma}_j^f (\mathbf{I} - \mathbf{K}_{j+1}^b \mathbf{K}_{j+1}^f) . \\ \mathbf{\Sigma}_{j+1}^b &= \mathbf{\Sigma}_j^b (\mathbf{I} - \mathbf{K}_{j+1}^f \mathbf{K}_{j+1}^b) . \\ \mathbf{\Theta}_i^{j+1} &= \mathbf{\Theta}_i^j + \mathbf{\Phi}_{j-i+1}^j \mathbf{K}_{j+1}^f \ : \ 1 \le i \le j . \\ \mathbf{\Theta}_{j+1}^{j+1} &= \mathbf{K}_{j+1}^f . \\ \mathbf{\Phi}_i^{j+1} &= \mathbf{\Phi}_i^j + \mathbf{\Theta}_{j-i+1}^j \mathbf{K}_{j+1}^b \ : \ 1 \le i \le j . \\ \mathbf{\Phi}_{j+1}^{j+1} &= \mathbf{K}_{j+1}^b , \end{aligned} \quad (14)$$

where the backward prediction vector $\mathbf{\Phi}^j \stackrel{def}{=} \begin{bmatrix} \mathbf{I} & \mathbf{\Phi}_1^j & \cdots & \mathbf{\Phi}_j^j \end{bmatrix}$ satisfies the following auxiliary block–Toeplitz system of equations :

$$\mathbf{\Phi}^j \mathbf{R}^j = \begin{bmatrix} \mathbf{\Sigma}_j^b & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} .$$

Assuming the input sequence to be *white* (i.e., $R_{xx}(l) = S_x \delta_l$), then the output auto–correlation and input–output cross–correlation sequences needed to form (12) can be computed using knowledge of the FIR filter taps as follows

$$R_{yy}(l) = S_x \sum_{m=0}^{\nu} h_m h_{m-l}^* . \quad (15)$$

$$R_{yx}(l) = S_x h_l = R_{xy}^*(-l) . \quad (16)$$

The parameters $\{a_1^p, \cdots, a_p^p, n_0^p, n_1^p, \cdots, n_q^p\}$ are read off directly from $\mathbf{\Theta}_i^p$ $(1 \le i \le p)$.

Remarks :

1. For the special case of an equal number of poles and zeros, we have $\delta = 0$. Hence, the first δ terms of (7) disappear, i.e.,

$$\begin{bmatrix} y_k \\ x_k \end{bmatrix} = \begin{bmatrix} -a_1^j & n_1^j \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_{k-1} \\ x_{k-1} \end{bmatrix} + \cdots$$

$$+ \begin{bmatrix} -a_j^j & n_j^j \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_{k-j} \\ x_{k-j} \end{bmatrix} + \begin{bmatrix} 1 & n_0^j \\ 0 & 1 \end{bmatrix} \begin{bmatrix} e_k^{'j} \\ x_k \end{bmatrix} . \quad (17)$$

Algorithm 1 can still be applied (with $\delta = 0$) to compute $\{a_1^p, \cdots a_p^p, n_1^p, \cdots, n_p^p\}$ and $n_0^p = \mathbf{\Sigma}_p^f(1,2)$.

2. Although our algorithm uses only second–order statistics, it can generate non–minimum phase approximations (i.e., the zeros of the numerator polynomial could lie outside the unit circle) because of the use of embedding. However, it can be shown that the pole polynomial is always guaranteed to be stable.

3. The matrices $\mathbf{\Sigma}_{j+1}^f$ and $\mathbf{\Sigma}_{j+1}^b$ are called the forward and backward prediction residual error matrices of order $(j+1)$, respectively.

4. In the presence of additive noise that is *independent of the input* with a correlation sequence $R_{nn}(l)$, Equation (15) is modified to : $R_{yy}(l) = S_x \sum_{m=0}^{\nu} h_m h_{m-l}^* + R_{nn}(l)$.

5. The case $q > p$ can be handled by interchanging the roles of the input and output sequences in (6). More specifically, consider the following ARMA(q,p) model of the *inverse filter*:

$$\begin{aligned} x_k &= -n_1^j x_{k-1} - \cdots - n_j^j x_{k-j} + a_0^j y_k + \cdots \\ &\quad + a_{j-\delta}^j y_{k-j+\delta} + e_k^{''j} , \end{aligned} \quad (18)$$

where $\delta \stackrel{def}{=} q - p$ and $1 \le j \le q$. Then, the embedding relation of (7) is modified to

$$\begin{bmatrix} x_k \\ y_{k+\delta} \end{bmatrix} = \begin{bmatrix} -n_1^j & 0 \\ \times & \times \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k+\delta-1} \end{bmatrix} + \cdots$$

$$+ \begin{bmatrix} -n_\delta^j & a_0^j \\ \times & \times \end{bmatrix} \begin{bmatrix} x_{k-\delta} \\ y_k \end{bmatrix}$$

$$+ \begin{bmatrix} -n_{\delta+1}^j & a_1^j \\ \times & \times \end{bmatrix} \begin{bmatrix} x_{k-\delta-1} \\ y_{k-1} \end{bmatrix}$$

$$+ \cdots + \begin{bmatrix} -n_j^j & a_{j-\delta}^j \\ \times & \times \end{bmatrix} \begin{bmatrix} x_{k-j} \\ y_{k-j+\delta} \end{bmatrix} + \begin{bmatrix} e_k^{''j} \\ y_{k+\delta} \end{bmatrix} . \quad (19)$$

Define $\mathbf{R}(i) \stackrel{def}{=} \begin{bmatrix} R_{xx}(i) & R_{xy}(i-\delta) \\ R_{yx}(i+\delta) & R_{yy}(i) \end{bmatrix}$, then Algorithm 1 can be applied to compute the parameters of the ARMA model $h(D) \approx \frac{1 + n_1^q D + \cdots + n_q^q D^q}{a_0^q + a_1^q D + \cdots + a_{q-\delta}^q D^{q-\delta}}$.

6. A useful way to interpret Algorithm 1 is as follows : we apply the $\mathbf{\Theta}_i^{j+1}$ recursion in (14) only for $\delta \le i \le j$. The parameters $\{\mathbf{\Theta}_1^{\delta-1}, \cdots, \mathbf{\Theta}_{\delta-1}^{\delta-1}\}$, or equivalently the all–pole prediction coefficients $\{a_1^{\delta-1}, \cdots, a_{\delta-1}^{\delta-1}\}$ can be computed by applying the classical (scalar) Levinson Algorithm. That is, we first find the best δ^{th} order AR model and then use it for the initialization of Algorithm 1 that will successively generate the best ARMA$(i+\delta, i)$ models $(1 \le i \le q)$, until it finally arrives at the desired ARMA$(q+\delta, q)$ model.

7. The formulation of (7) accommodates the more general case where the input sequence is also modeled as an ARMA process. Our assumption of a white input sequence is a special case of that general formulation.

8. It was brought to our attention recently [2] that a multichannel least squares algorithm with a different number of parameters to be estimated in each channel was previously derived in [13]. However, the algorithm of [13] is distinct from ours in that it is of lattice type, time–recursive, and was not derived using the "embedding" approach that we follows here. In addition, the intended application of this algorithm being investigated in this paper, namely pole–zero modeling of long FIR filters, is new, to the best of our knowledge.

2.3 Model Order Selection

A critical component in generating an accurate ARMA(p,q) approximation is the choice of p and q. Several tests have been proposed in the literature for determining p and q from the output data record or from the output auto–correlation sequence [11] (which can be computed from (15) using knowledge of the FIR filter taps and assuming a white input sequence). These tests include an information–theoretic criterion (AIC) by Akaike [14] and singularity tests of an output auto–correlation matrix [15]. More recently, Pillai *et al.* proposed an order–determining method based on a degree–reducing procedure from passive network theory [16].

For the purposes of this paper, we shall choose p and q according to the following two considerations. The first consideration is cost–driven in that we assume a maximum allowable implementation complexity, which in turn sets an upper bound on the values of p and q. Therefore, we seek to find the best, in terms of low normalized

[2] Thanks to Professor H. Lev-Ari of Northeastern University.

norm tap error NNTE $\stackrel{def}{=} \frac{\sum_{i=0}^{N-1}|h_i-\hat{h}_i|^2}{\sum_{i=0}^{N-1}|h_i|^2}$ (where N is the number of FIR filter taps), pole–zero approximation, subject to this complexity constraint. It is worth emphasizing that increasing p and/or q could result in a worse approximation depending on the FIR filter response, as it will be shown in Section 3. The second consideration is performance–driven and requires an exhaustive search over different choices of p and q, within the allowable range determined by the first consideration above, until we arrive at an acceptable fit. Depending on the application, complexity might be partially traded for performance (as in echo cancellation) or vice versa, although in most cases a compromise between the two is more desirable.

3 Simulation Results

In this section, we shall evaluate the performance of Algorithm 1 by applying it to a representative example from the HDSL environment to implement the FIR MMSE–DFE filters in a pole–zero form that reduces complexity while still retaining satisfactory performance.

Approximating the feedforward filter :

As mentioned previously in Section 2.1, when $N_b = \nu$, the optimal feedforward filter of the FIR MMSE–DFE becomes strictly anti–causal and can be realized with a delay of $(N_f - 1)$ symbol periods, i.e.,

$$w^*(D^{-1}) = w^*_{-(N_f-1)}D^{-(N_f-1)} + \cdots + w^*_{-1}D^{-1} + w^*_0 \ . \quad (20)$$

We have found through extensive computer simulations that a direct reduced–parameter pole–zero approximation of $D^{N_f}w^*(D^{-1})$ is very difficult to obtain. This is due to the fact that the initial part of the impulse response (IR) prior to the peak could be very long. Pole–zero models can more easily model the decaying tail of the IR following the peak. With this observation in mind, we propose the following procedure for deriving pole–zero models of (20). Denote the peak sample of the feedforward filter IR by $w^*_{-\tau}$ $(0 \leq \tau \leq N_f - 1)$. Shift the feedforward IR by τ samples to center the peak sample at the origin. The resulting IR, namely $D^\tau w^*(D^{-1})$, will have a causal and a strictly anti–causal components that we shall denote by $w_1(D)$ and $w_2^*(D^{-1})$, respectively.

$$D^\tau w^*(D^{-1}) = w_1(D) + w_2^*(D^{-1}) \ . \quad (21)$$

Now, each of $w_1(D)$ and $w_2(D)$ has the shape of a decaying IR tail that can be easily modeled as a pole–zero filter with few parameters. Therefore, (21) becomes

$$D^\tau w^*(D^{-1}) \approx \frac{n^{w_1}(D)}{a^{w_1}(D)} + \frac{n^{*w_2}(D^{-1})}{a^{*w_2}(D^{-1})} \ ,$$

and the causal feedforward filter IR is approximated by

$$D^{N_f-1}w^*(D^{-1}) \approx D^{N_f-1-\tau}\left(\frac{n^{w_1}(D)}{a^{w_1}(D)} + \frac{n^{*w_2}(D^{-1})}{a^{*w_2}(D^{-1})}\right) \quad (22)$$

$$= D^{N_f-1-\tau}\left(\frac{n^{w_1}(D)a^{*w_2}(D^{-1}) + a^{w_1}(D)n^{*w_2}(D^{-1})}{a^{w_1}(D)a^{*w_2}(D^{-1})}\right) \ . \quad (23)$$

Realizing the feedforward as in (22) is preferable over the form of (23) since the former requires a smaller number of taps. This further explains the difficulty encountered when attempting a direct reduced–parameter pole–zero approximation of the overall IR of the feedforward filter.

Example 1 :

For the HDSL environment we shall assume the worst case 9 kft 26 AWG DSL. The input power level is 17 dBm evenly–distributed over the transmission bandwidth and the 2–sided AWGN power spectral density (psd) is taken to be -113 dBm/Hz. The standard $|H_x(f)|^2 = k_{NEXT}f^{\frac{3}{2}}$ near–end crosstalk (NEXT) model is adopted with $k_{NEXT} = 10^{-13}$. The target bit rate is set at 800 kbps and the input signal constellation is 16 QAM which is near–optimum [17]. A fixed probability of error $P_e = 10^{-7}$ is assumed and a 4.2 coding gain is included. The finite–length MMSE–DFE is assumed to have 96 feedforward taps and 64 feedback taps. This choice results in an operational margin of around 4.9 dB [17].

In Tables 1-3 we present some of the best pole–zero approximations of the 64–tap feedback filter and the two components of the 96–tap feedforward filter obtained using Algorithm 1 and their corresponding NNTE's. The impulse responses of the those approximations together with that of the desired response are given in Figures 2,4, and 5.

It is evident that the proposed algorithm generates fairly accurate pole–zero approximations with the added advantages of *significant* reduction in the number of filter coefficients, fast computation, and guaranteed stability.

4 Other Applications

Although our main motivation for deriving Algorithm 1 was to reduce the implementational complexity of the finite–length MMSE–DFE, these

(p,q)	$\hat{b}(D)$	NNTE(dB)
$(3,2)$	$\frac{-1+0.5354D+0.5056D^2}{1-1.3501D+0.2757D^2+.1275D^3}$	-27.2946
$(4,4)$	$\frac{-1+0.6731D+0.7541D^2-.2917D^3-.1145D^4}{1-1.4879D+0.1395D^2+0.5757D^3-.1944D^4}$	-31.1634
$(7,3)$	$\frac{-1+.7325D+.572D^2-.2791D^3}{1-1.5472D+.3699D^2+0.3943D^3-.2299D^4+.0636D^5-.0157D^6+.0048D^7}$	-31.384

Table 1: ARMA approximations for $b(D)$ of Example 1 using Algorithm 1 and their achievable NNTE's

(p,q)	$\hat{w_1}(D)$	NNTE(dB)
$(3,2)$	$\frac{47.1549-6.4777D-18.9055D^2}{1-1.0134D-.127D^2+.207D^3}$	-36.5522
$(5,2)$	$\frac{47.1549-4.3601D-24.7598D^2}{1-.9685D-.2905D^2+.3231D^3-.0263D^4+.0166D^5}$	-41.9531
$(6,2)$	$\frac{47.1549-6.5874D-24.4024D^2}{1-1.0157D-.2416D^2+.3091D^3-.0193D^4+.0131D^5+.0028D^6}$	-42.2905

Table 2: ARMA approximations for $w_1(D)$ of Example 1 using Algorithm 1 and their achievable NNTE's

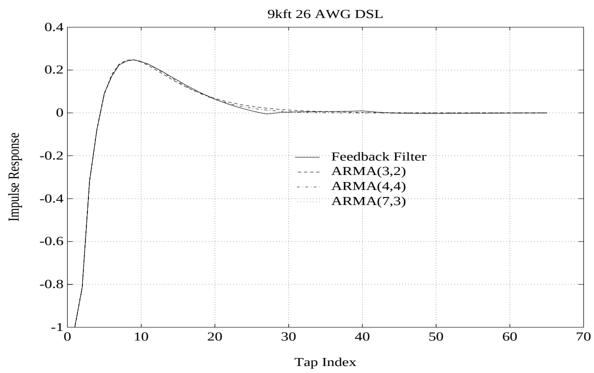

Figure 2: Pole-zero approximations of the feedback filter of Example 1 generated using Algorithm 1.

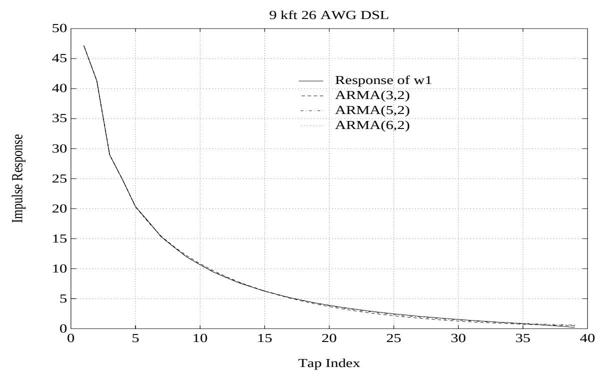

Figure 4: Pole-zero approximations for $w_1(D)$ of Example 1 generated using Algorithm 1.

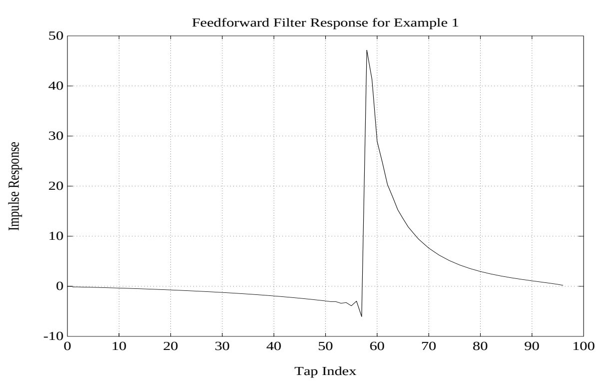

Figure 3: Impulse Response of the feedforward filter of Example 1.

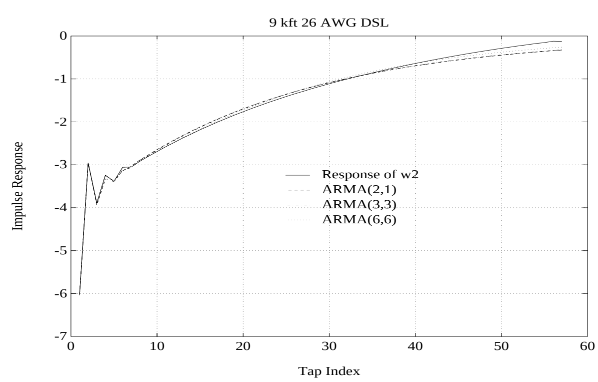

Figure 5: Pole-zero approximations for $w_2(D)$ of Example 1 generated using Algorithm 1.

(p,q)	$\hat{w}_2(D)$	NNTE(dB)
$(2,1)$	$\frac{-6.0337+.4348D}{1-.5621D-.3772D^2}$	-26.0219
$(3,3)$	$\frac{-6.0337+2.7584D+.6662D^2-.3245D^3}{1-.9472D-.2917D^2+.2708D^3}$	-26.4864
$(6,6)$	$\frac{-6.0337+2.5235D+.6947D^2+.7904D^3+.7084D^4+.2405D^5-.1915D^6}{1-.9082D-.3155D^2+.0725D^3-.0224D^4+.0965D^5+.0914D^6}$	-30.4245

Table 3: ARMA approximations for $w_2(D)$ of Example 1 using Algorithm 1 and their achievable NNTE's

algorithms can be used in several other applications. We briefly explore some of these applications next.

4.1 Revisiting the infinite-length MMSE-DFE

It is well known (see e.g. [18, 19]) that the optimum settings of the *infinite-length* MMSE–DFE can be computed from the spectral factorization

$$\frac{1}{SNR} + h(D)h^*(D^{-1}) \stackrel{def}{=} \gamma_0^2 g(D) g^*(D^{-1}) \quad (24)$$

as follows

$$b(D) = g(D) \ : \ \nu \text{ taps} \quad (25)$$

$$w^*(D^{-1}) = \frac{1}{\gamma_0^2}\frac{h^*(D^{-1})}{g^*(D^{-1})} : \mathrm{ARMA}(\nu,\nu) \quad (26)$$

In some applications, such as high–speed data transmission on DSL, the channel impulse response is very long (large ν), which makes the spectral factorization of (24) very costly to compute. However, if we use Algorithm 1 to convert $h(D)$ to a pole–zero model, $h(D) \approx \frac{n^h(D)}{a^h(D)} \stackrel{def}{=} \frac{\sum_{i=0}^{q} n_i^h D^i}{1+\sum_{i=1}^{p} a_i^h D^i}$, then (24) becomes

$$\frac{1}{SNR} a^h(D) a^{*h}(D^{-1}) + n^h(D) n^{*h}(D^{-1}) \approx$$

$$\gamma_0^2 g(D) g^*(D^{-1}) a^h(D) a^{*h}(D^{-1}) \stackrel{def}{=} \gamma_0^2 r(D) r^*(D^{-1}) \quad (27)$$

since $a^h(D)$ is a *canonical* response. Then, the optimum filter settings of the infinite–length MMSE–DFE can be calculated approximately from :

$$b(D) \approx \frac{r(D)}{a^h(D)} : \mathrm{ARMA}(p, \max(p,q)) \quad (28)$$

$$w^*(D^{-1}) \approx \frac{1}{\gamma_0^2}\frac{n^{*h}(D^{-1})}{r^*(D^{-1})} : \mathrm{ARMA}(\max(p,q), q)\ . \quad (29)$$

The spectral factorization of (27) requires finding the roots of a polynomial of degree=$\max(p,q)$, which is of much lower complexity (even after adding in the complexity of converting $h(D)$ to a pole–zero model) than finding the roots of a ν^{th}–degree polynomial as in (27), since $\nu \gg \max(p,q)$. Nevertheless, we need to take into consideration the degradation of the decision–point SNR when calculating the MMSE–DFE filter settings from (28) and (29) instead of (25) and (26). This issue is addressed in [20].

Another point that deserves attention is the concern that implementing the feedback filter in a pole–zero form might exacerbate the problem of *error propagation*. However, a little reflection shows that when error propagation occurs, the resulting performance degradation should be almost the same whether we implement the feedback filter in an FIR or IIR form, as long as the NNTE is low enough to ensure that the IR's of both implementations are almost the same. Furthermore, error propagation can be avoided either by ensuring a high decision–point SNR that prevents initial errors from occurring or by moving the feedback filter to the transmitter, where no errors can occur, using the **precoding** technique.

4.2 Adaptive IIR filtering

In adaptive filtering applications, the reductions in the implementation cost of pole–zero filters over long FIR filters become more dramatic since a larger memory size and a higher computational power is needed (to store the filter taps and previous input samples and to compute the filter output samples) when the FIR filter coefficients are updated frequently.

The two most common methods for adaptive IIR filtering are the equation–error method and the output–error method [21]. Unlike the well–understood adaptive FIR algorithms, several problems arise in adaptive IIR filtering including bias in equation–error methods and local minima in output–error methods; many other issues (such as stability monitoring) remain open to date [22].

Algorithm 1 could also find useful application in adaptive IIR filtering, as follows. Apply standard adaptive algorithms (such as the robust LMS algorithm) to adapt a long FIR filter, then use Algorithm 1 to convert the updated long FIR filter to a pole–zero filter with much fewer parameters to reduce the real–time implementation cost. Continue the FIR adaptation process in the background and the conversion process in the foreground.

4.3 Echo Cancellation

Echo cancellation for full–duplex transmission on DSL might prove to be one of the most challenging applications for the algorithm of Section 2. The stringent requirement of an NNTE of -60 dB or less in this application makes the pole–zero modeling process more difficult than it is, e.g., in the DFE application. Furthermore, the echo path IR is typically characterized by a rapidly varying initial part followed by a very long tail that could span hundreds of taps, especially as the sampling rate increases.

Additional procedures will be needed to obtain high–quality pole–zero approximations of the echo response. Among those are use of the "splitting" technique (see [7, 8] and Section 4.4) to model the initial part of the echo IR by an FIR filter and only the tail of the IR by a pole–zero filter. We might also need to relax the requirement of $p, q \leq 10$ that was made in the DFE simulations of Section 3, and consider higher–order approximations.

4.4 Channel Identification

Very accurate and efficient channel identification can be carried out in the frequency domain using the Fast Fourier Transform (FFT) algorithm and special periodic training sequences [23]. However, for high spectral resolution, the FFT size N has to be large (typically 512 or 1024) which results in a long N–tap FIR representation of the channel. Algorithm 1 can then be used to convert this long FIR filter to a pole–zero filter.

Direct identification of ARMA(p, q) channel models can also be achieved using Algorithm 1 by computing estimates of the correlation sequences based on known training sequences and measured output samples.

For a variety of other tasks, including transmitter optimization (optimizing the input symbol rate and the transmit power spectrum shape [19, 17]), receiver optimization of the MMSE–DFE (c.f. Section 4.1), and performance evaluation using computer simulations, significant computational savings can be accrued by obtaining a more compact (i.e., has fewer parameters) description of the channel.

Example 2

Consider again the 9 kft 26 AWG DSL of Example 1. In Table 4, we list a number of pole–zero approximations to the channel impulse response, obtained using Algorithm 1, together with their achievable NNTE's.

To illustrate the ability of the algorithm to model non minimum–phase responses such as the channel response of this example (which has 7 non minimum–phase zeros), we have plotted in Figures 6-7 the magnitude and phase responses of the channel and its pole–zero approximations generated using Algorithm 1.

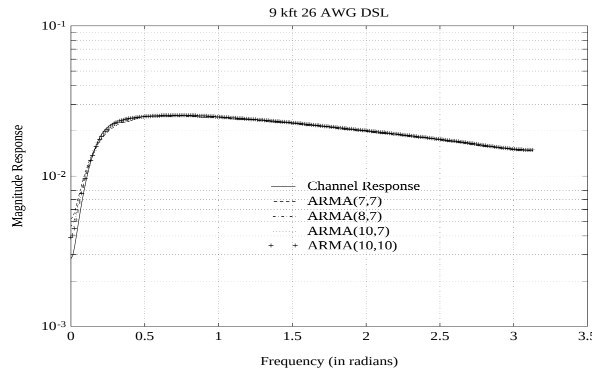

Figure 6: Magnitude response of $h(D)$ of Example 3 and its pole-zero approximations generated using Algorithm 1.

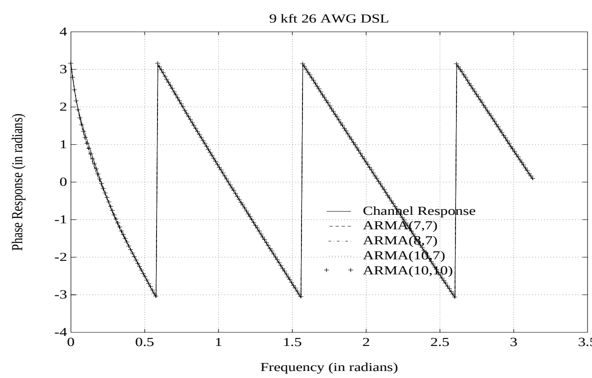

Figure 7: Phase response of $h(D)$ of Example 3 and its pole-zero approximations generated using Algorithm 1.

Nevertheless, we have found that by splitting $\hat{h}(D)$ into a short FIR filter, call it $\hat{h}_1(D)$ (that models the initial part of the channel IR prior to the peak sample), and a pole–zero filter, call it $\hat{h}_2(D)$ (that models the tail of the IR), better fits with lower orders were obtained,[3] as shown in Table 5. The impulse response of those approximations superimposed on the response of $h_2(D)$ are shown in Figure 8. The taps of the FIR filter $\hat{h}_1(D)$ are set equal to the corresponding ones in $h(D)$ and are given by $\hat{h}_1(D) = 10^{-4}(D - D^2 + 2D^3 - 4D^4 + 11D^5)$.

4.5 Digital Filter Design

Recursive digital filters are generally more difficult to design than nonrecursive ones. Traditional methods for designing recursive digital fil-

[3] This agrees with conclusions arrived at in [8, 7].

(p,q)	$\hat{h}(D)$	NNTE(dB)
$(7,7)$	$\frac{10^{-4}(D-D^2+3D^3-5D^4+15D^5+174D^6-196D^7)}{1-.9929D+.2163D^2-.0486D^3+.0263D^4-.0116D^5+.0099D^6-.0031D^7}$	-29.7433
$(8,7)$	$\frac{10^{-4}(D-D^2+3D^3-5D^4+15D^5+176D^6-195D^7)}{1-.9901D+0.2165D^2-.0479D^3+.0271D^4-.0111D^5+.0111D^6-.0038D^7+.006D^8}$	-31.3082
$(10,7)$	$\frac{10^{-4}(D-D^2+3D^3-5D^4+15D^5+174D^6-195D^7)}{1-.9883D+.2166D^2-0.0474D^3+0.0276D^4-.0106D^5+.0117D^6-.0034D^7+.007D^8-.0011D^9+.0046D^{10}}$	-32.5621
$(10,10)$	$\frac{10^{-4}(D-D^2+2D^3-5D^4+15D^5+176D^6-194D^7-66D^8+45D^9+23D^{10})}{1-.9593D-.1493D^2+.1746D^3+0.0709D^4-0.0193D^5+.008D^6-.0025D^7+.0045D^8-.0008D^9+.0029D^{10}}$	-34.5509

Table 4: ARMA approximations for $h(D)$ of Example 3 using Algorithm 1 and their achievable NNTE's

(p,q)	$\hat{h}_2(D)$	NNTE(dB)
$(2,2)$	$\frac{.0186-.0156D-.0046D^2}{1-.7679D-.0171D^2}$	-26.4861
$(4,1)$	$\frac{.0186-.0197D}{1-.9882D+0.2126D^2-.0453D^3+.0233D^4}$	-29.7532
$(4,4)$	$\frac{.0186-.0197D-.0095D^2+.0089D^3+.0012D^4}{1-.9852D-.299D^2+.3965D^3-.0219D^4}$	-31.6322
$(8,8)$	$\frac{10^{-4}(186-186D-45D^2+15D^3-4D^4+18D^5-18D^6+25D^7+6D^8)}{1-.9285D-.0271D^2+.0343D^3-.047D^4+.0645D^5-.1211D^6+.1158D^7+.0013D^8}$	-37.4376

Table 5: ARMA approximations for $h_2(D)$ of Example 3 using Algorithm 1 and their achievable NNTE's

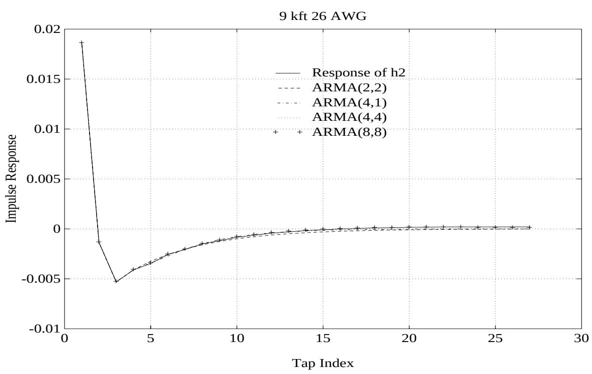

Figure 8: Pole-zero approximations of $h_2(D)$ of Example 3 generated using Algorithm 1.

ters make use of the existing wealth in analog filter design methods by applying a transformation (as in the impulse–invariant or bilinear methods) on an analog filter that meets the given desired frequency response characteristics [24, 25].

Algorithm 1 can be used as a computationally–efficient time–domain recursive digital filter design method, however, the design criterion is now related to *matching second-order statistics*.

As pointed out earlier, recursive digital filters can usually achieve a similar magnitude response to an FIR filter with much fewer parameters, especially in situations where there is a sharp transition between band edges. However, we should point out that the exact amount of reduction in implementation cost depends on the architecture of the signal processor used for filter implementation. Moreover, recursive digital filters have other limitations including their inability to have an exactly constant group delay and their sensitivity to finite–precision effects such as coefficient errors, quantization noise, overflow, and limit cycles. The finite–precision effects can be partially offset by using alternative realizations (other than the direct pole–zero form) such as parallel, cascade, or lattice forms. The latter realization can be constructed using the set of forward and backward reflection coefficients generated by Algorithm 1.

5 Conclusion

We have derived a computationally–efficient stable algorithm that reduces the implementation complexity of the MMSE–DFE by accurately approximating its long FIR feedforward and feedback filters by pole–zero filters with fewer coefficients.

Extensive simulation results on a representative loop from the HDSL environment demonstrated the viability of the proposed algorithm. Finally, a host of other applications where the derived algorithm could be used were described and discussed.

References

[1] D. Lee, B. Friedlander, and M. Morf. Recursive Ladder Algorithms for ARMA Modeling. *IEEE Transactions on Automatic Control*, AC-27(4):753–764, August 1982.

[2] J. Tu. *Theory, Design, and Application of Multi-Channel Modulation for Digital Communications*. PhD thesis, Stanford University, June 1991.

[3] C. Burrus and T. Parks. Time Domain Design of Recursive Digital Filters. *IEEE Transactions on Audio and Electroacoustics*, AU-18(2):137–141, June 1970.

[4] A. Evans and R. Fischl. Optimal Least Squares Time-Domain Synthesis of Recursive Digital Filters. *IEEE Transactions on Audio and Electroacoustics*, AU-21(1):61–65, February 1973.

[5] A. Shaw. An Optimal Method for Identification of Pole-Zero Transfer Functions. In *ISCAS Proceedings, San Diego, CA.*, pages 2409–2412, 1992.

[6] G. Long, D. Shwed, and D. Falconer. Study of a Pole-Zero Adaptice Echo Canceller. *IEEE Transactions on Circuits and Systems*, CAS-34:765–769, July 1987.

[7] G. Davidson and D. Falconer. Reduced Complexity Echo Cancellation Using Orthonormal Functions. *IEEE Transactions on Circuits and Systems*, CAS-38, January 1991.

[8] P. Crespo and M. Honig. Pole-Zero Decision Feedback Equalization with a Rapidly Converging Adaptive IIR Algorithm. *IEEE Journal on Selected Areas in Communications*, 9(6):817–829, August 1991.

[9] N. Al-Dhahir and J. Cioffi. Fast Algorithms for the Computation of the FIR MMSE-DFE. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, March 1992.

[10] N. Al-Dhahir and J. Cioffi. MMSE Decision Feedback Equalizers : Finite-Length Results. Submitted to IEEE Transactions on Information Theory, August 1993.

[11] S. Kay. *Modern Spectral Estimation: Theory and Applications.* Prentice Hall, 1988.

[12] T. Kailath. Linear Estimation for Stationary and Near-Stationary Processes. In *Lecture Notes for Arab School on Science and Technology.* Hemisphere Publishing Co., 1983. Summer School on Signal Processing.

[13] F. Ling and J. Proakis. A Generalized Multichannel Least Squares Lattice Algorithm Based on Sequential Processing Stages. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-32(2):338–343, April 1984.

[14] H. Akaike. A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, AC-19(6):716–723, December 1974.

[15] J. Chow. On Estimating the Orders of an Autoregressive Moving Average Process with Uncertain Observations. *IEEE Transactions on Automatic Control*, AC-17, October 1972.

[16] S. Pillai, T. Shim, and D. Youla. A New Technique for ARMA-System Identification and Rational Approximation. *IEEE Transactions on Signal Processing*, 41(3):1281–1304, March 1993.

[17] N. Al-Dhahir and J. Cioffi. Bandwidth Optimization for Combined Equalization and Coding Transceivers with Emphasis on the MMSE-DFE. In *IEEE Milcom Conference*, October 1993.

[18] J. Salz. Optimum Mean-Square Decision Feedback Equalization. *Bell System Tech. J.*, 52(8), October 1973.

[19] J. Cioffi, G. Dudevoir, M. Eyuboglu, and G.D. Forney. Minimum Mean-Square-Error Decision Feedback Equalization and Coding - Part I: Equalization Results. *IEEE Transactions on Communications*, 1993. To Appear.

[20] N. Al-Dhahir and J. Cioffi. Mismatched MMSE Decision Feedback Equalizers. In preparation, September 1993.

[21] J. Shynk. Adaptive IIR Filtering. *IEEE ASSP Magazine*, pages 4–21, April 1989.

[22] C. Johnson. Adaptive IIR Filtering : Open Issues. *IEEE Transaction on Information Theory*, IT-30:237–250, March 1984.

[23] J. Cioffi. Channel Identification for High-Speed Voiceband Modems. Preprint, March 1991.

[24] L. Jackson. *Digital Filters and Signal Processing.* Kluwer Academic Publishers, 1986.

[25] T. Parks and C. Burrus. *Digital Filter Design.* John Wiley & Sons Inc., 1987.