Pole-Zero Decision Feedback Equalization with a Rapidly Converging Adaptive IIR Algorithm

Pedro M. Crespo, *Member, IEEE*, and Michael L. Honig, *Member, IEEE*

***Abstract*—A decision feedback equalizer (DFE) containing a feedback filter with both poles and zeroes is proposed for high-speed digital communications over the subscriber loop. The feedback filter is composed of two sections: a relatively short finite impulse response (FIR) filter that cancels the initial part of the channel impulse response, which may contain rapid variations due to bridge taps; and a pole-zero, or IIR, filter that cancels the smoothly decaying tail of the impulse response. Modifications of an existing adaptive IIR algorithm, based on the Steiglitz–McBride identification scheme, are proposed to adapt the feedback filter. These new algorithms have comparable complexity to gradient-based adaptive IIR algorithms when the number of poles is small, but converge significantly faster. A measured subscriber loop impulse response is used to compare the performance of the adaptive pole-zero DFE, assuming a two-pole feedback filter, with a conventional DFE having the same number of coefficients. Results show that the pole-zero DFE offers a significant improvement in mean squared error (i.e., 4 dB at a signal-to-noise ratio of 25 dB) relative to the conventional DFE. Furthermore, the speed of convergence of the adaptive pole-zero DFE is comparable with that of the conventional DFE using the standard LMS adaptive algorithm.**

I. Introduction

Decision feedback equalization has been widely proposed as an effective technique for suppressing intersymbol interference (ISI) in the context of high-speed digital communications over dispersive channels [1]–[3]. A DFE, shown in Fig. 1, consists of a feedforward filter $P(z)$ follwed by a feedback loop containing a decision element in the forward path, and a filter $F(z)$ in the feedback path. The prefilter $P(z)$ compensates for precursor ISI, that is, ISI from symbols that have not yet been detected at the current symbol interval, and $F(z)$ cancels postcursor ISI, that is, ISI from previously detected symbols. Typically, $P(z)$ and $F(z)$ are FIR filters.

A particularly attractive application for decision feedback equalization is in high-speed digital subscriber lines (HDSL's) [2], [3]. Channel dispersion in this case causes severe ISI at high data rates (i.e., 800 kb/s), which means that the DFE must contain many taps, assuming a conventional implementation. As an example, a measured subscriber loop impulse response (IR) corresponding to 12 kft of 24 gauge twisted-pair wire is shown in Fig. 2. The lead time before the leading edge of the IR is the group delay of the channel. Assuming that this IR includes the effect of filtering at the transmitter and receiver, then precursor ISI is caused by the portion of the IR preceding the cursor, and postcursor ISI is caused by the tail of the IR. In general, for the subscriber loop application the prefilter $P(z)$ typically requires few taps (i.e., five or less) to adequately suppress precursor ISI. To minimize mean squared error (MSE), however, the number of taps in the feedback filter $F(z)$ must span almost the entire tail of the IR to cancel postcursor ISI. The number of taps in $F(z)$ is therefore approximately τ/T, where τ is the length of the IR tail and T is the time between samples. For the IR in Fig. 2, τ is approximately 200 μs, so that a symbol rate of 500 kbauds implies that $F(z)$ should have approximately 100 taps.

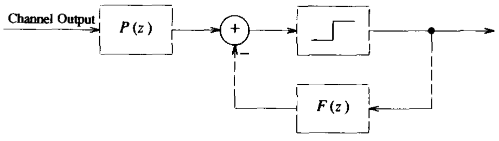


Fig. 1. Decision feedback equalizer.

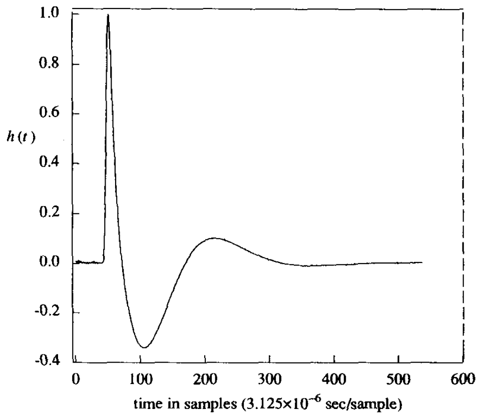


Fig. 2. Normalized measured channel impulse response for 12 kft of 24-gauge twisted-pair cable.

One way to reduce the number of taps in $F(z)$ is to shorten the tail of the IR by adding taps to the prefilter $P(z)$. This however, increases equalizer noise enhancement. Another alternative is to make $F(z)$ an IIR, or pole-zero, filter. The primary problem with this approach is that, in general, IIR filters are notoriously difficult to adapt when the channel IR is initially unknown. Specifically,

Manuscript received October 31, 1990; revised May 1, 1991.
P. M. Crespo is with Telefónica I+D, 28403 Madrid, Spain.
M. L. Honig is with Bellcore, Morristown, NJ 07960.
IEEE Log Number 9101629.

the MSE, when viewed as a function of the filter coefficients, may contain local optima so that the adaptive algorithm may converge to a solution in which significant ISI is present. Furthermore, the filter may becomes unstable during the adaptation. These problems become much less troublesome, however, when the number of poles to be identified, M, is small (i.e., $M \leq 2$), and the number of poles in the adaptive filter is at least M. In this case, stability is easily monitored and local optima are not as likely to appear [4].

In the case of the subscriber loop, the IR typically decays smoothly to zero and can be accurately modeled as the IR of a one- or two-pole transfer function. $F(z)$ can therefore be replaced by a filter consisting of two sections: an FIR filter that compensates for the initial part of the postcursor IR, which may be relatively difficult to model with an IIR filter, and a one- or two-pole filter that compensates for the tail of the postcursor IR. For example, the FIR filter needed to cancel the initial part of the postcursor IR in Fig. 2 is approximately one-third the length of the original FIR feedback filter $F(z)$, and only two or three additional taps are needed to cancel the remaining tail. Numerical results show that for the IR in Fig. 2, a 4 dB improvement in MSE can be obtained by using the proposed IIR feedback filter instead of a conventional DFE of the same complexity, assuming a signal-to-noise ratio (SNR) of 25 dB.

Similar types of structures have been proposed in the context of echo cancellation [5]–[7]. The structures proposed in [5] and [6] are not adaptive, and the adaptive algorithms proposed in [7] are different from those proposed here. References [8]–[10] propose different techniques for cancelling the tail of the echo, which can also be used in a DFE (see also [11]). A summary of these techniques is given in [10]. In this paper, however, we study only IIR feedback filters for tail cancellation and do not attempt to quantitatively compare the various techniques proposed in [5]–[11]. We add that adaptive IIR filtering has also been studied in the context of echo cancellation of speech signals [12]–[14].

Currently, there are a number of adaptive IIR algorithms that can be used to adjust the taps of the proposed IIR feedback filter when the channel is initially unknown [15], [16]. The equation error method is one such algorithm that has been used in the context of voice echo cancellation [12], [13], and has the advantage that the error surface is unimodal so that the tap weights converge to a unique solution. In the presence of noise, however, this solution gives a biased estimate of the poles of the channel transfer function, resulting in some residual ISI. We observe that when the equation error method is used to optimize the pole-zero DFE, the denominator polynomial in the pole-zero feedback filter appears in cascade with the channel transfer function and the prefilter $P(z)$. Adding poles to the pole-zero DFE when the equation error method is used to optimize the feedback filter is therefore analogous to adding taps to the prefilter of a conventional DFE. In the former case, a biased channel estimate causes residual ISI, and in the latter case, a longer prefilter causes additional noise enhancement. Our numerical results indicate that this tradeoff generally favors the conventional DFE. That is, the output MSE for the pole-zero structure, when adapted via the equation error method, is typically larger than the MSE of a conventional DFE of the same complexity.

The IIR feedback filter can also be adapted to minimize the output MSE directly via a gradient algorithm [15], [16]. Although in general the MSE cost function can be multimodal, it is shown in [4] that this is not the case when the number of poles to be estimated, M, is less than or equal to two, and the number of poles in the adaptive filter is at least M. Since the tail of a typical subscriber loop IR (such as the one in Fig. 2) can be accurately modeled with two poles, a two-pole feedback filter in the HDSL application should be sufficient to guarantee the absence of local minima in the error surface. The asymptotic performance of a gradient algorithm, when used to minimize the MSE, will then be superior to that of the equation error method.

Although a two-pole feedback filter adapted via a gradient algorithm achieves the minimum MSE for the cases of interest, it exhibits extremely slow convergence. Our results show that the pole-zero DFE with an IIR gradient algorithm takes more than 10 times as many iterations to converge to the asymptotic MSE as a conventional DFE using the standard LMS transversal algorithm [17]. Adapative IIR algorithms that converge faster than the gradient algorithm can be obtained by modifying the algorithm proposed in [18], which is based on the Steiglitz–McBride identification algorithm. One modification consists of switching the order of the preprocessing filter ($1/D(z)$, where $D(z)$ is obtained from the current estimate of the system poles) and the filter $C(z)$ that estimates the system zeroes. The input to $C(z)$ then becomes the uncorrelated transmitted symbols. When adapted via the LMS algorithm, $C(z)$ therefore converges faster than when its input sequence is first filtered by $1/D(z)$.

A second modification to the algorithm proposed in [18] consists of using a recursive least squares (RLS) algorithm, rather than the LMS algorithm, to update the poles and/or zeroes of the IIR adaptive filter. The convergence speed of this algorithm is found to be comparable with that of a conventional DFE using the LMS algorithm. Since only two poles are considered, this RLS algorithm requires little additional complexity relative to the LMS algorithm.

The next section presents the pole-zero DFE, and Section III discusses adaptive IIR algorithms. Section IV presents numerical results comparing the performance of the adaptive pole-zero DFE with a conventional DFE.

II. An IIR Feedback Filter

A block diagram of the proposed DFE is shown in Fig. 3. The filters $P(z)$, $A(z)$, $C(z)$, and $D(z)$ are finite-length (FIR) transversal filters. The purpose of $A(z)$ is to cancel

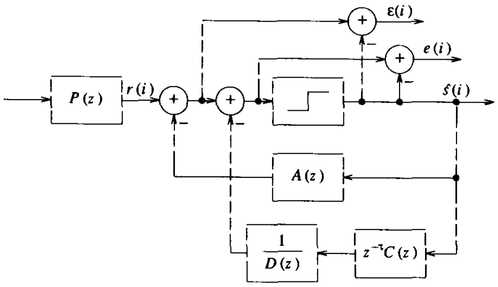


Fig. 3. DFE with pole-zero feedback filter to cancel the tail of the postcursor IR.

postcursor ISI due to the initial part of the IR of the combined transmitter filter, channel, receiver filter, and prefilter $P(z)$. The pole-zero filter $C(z)/D(z)$ then cancels postcursor ISI due to the tail of the IR. It is assumed that $A(z)$ spans $\tau - 1$ symbols, so that the filter $z^{-\tau}C(z)/D(z)$ cancels the residual postcursor IR. Note that the relatively complicated high frequency behavior of some subscriber loop channels, such as those with bridge taps, typically affects only the first part of the IR, which in this case is modeled by $A(z)$.

Throughout this paper it will be assumed that $C(z)/D(z)$ has at most two poles; that is,

$$\frac{C(z)}{D(z)} = \frac{c_0 + c_1 z^{-1}}{1 - d_1 z^{-1} - d_2 z^{-2}}. \tag{1}$$

Stability monitoring then becomes quite easy. That is, the zeroes of $D(z)$ are within the unit circle provided that [15]

$$1 + d_1 - d_2 > 0, \qquad 1 - d_1 - d_2 > 0,$$
$$\text{and } d_2 > -1. \tag{2}$$

Rather than impose the specific feedback filter structure shown in Fig. 3, the filters $A(z)$ and $z^{-\tau}C(z)/D(z)$ can be combined into the single pole-zero filter $C'(z)/D'(z)$ shown in Fig. 4. The two feedback filters in Figs. 3 and 4 are the same when

$$C'(z) = A(z)D(z) + z^{-\tau}C(z)$$

and $D'(z) = D(z)$. If the order of $C'(z)$ is $\tau + 1$, then any feedback filter shown in Fig. 3 can be synthesized as $C'(z)/D'(z)$. The structure shown in Fig. 3, however, has the following advantages over the structure in Fig. 4.

The primary advantage of the structure in Fig. 3 is that it is easier to adapt than the structure in Fig. 4. To see why this is true, note that the coefficients of $A(z)$ in Fig. 3 can be adapted independently of $D(z)$ (with a small enough step-size), and converge to the first $\tau - 1$ values of the postcursor IR. The coefficients of $D(z)$ can subsequently be adapted to cancel the remaining postcursor ISI. In this way, the adaptation problem is partitioned so that the filter poles are adapted to match a smoothly decaying IR, which is relatively easy. In contrast, the structure in Fig. 4 must select zeroes and poles to match the *overall* postcursor IR, which is a significantly harder task. There is no longer any transparent relation between the IR coefficients and the coefficients of $C'(z)$ [or $D'(z)$], and the coefficients of $C'(z)$ cannot be adapted independently of $D'(z)$. Furthermore, for the cases considered in Section IV, the MSE (that is, $E[e^2(i)]$, where $e(i)$ is shown in Fig. 3), is much more sensitive to variations in $C'(z)$ in Fig. 4 than to variations in $A(z)$ in Fig. 3. Consequently, the step-size needed in a gradient algorithm to adapt the structure in Fig. 4 to the global minimum must be extremely small, resulting in very poor convergence properties.

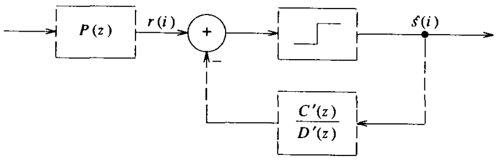


Fig. 4. DFE with pole-zero feedback filter.

Another advantage of the structure in Fig. 3 is that if the filter $P(z)$ is fixed, then this DFE structure can be easily combined with the timing recovery scheme described in [19]. This scheme relies on estimates of the channel IR, which can be obtained from $A(z)$, to determine the optimal sampling phase of the received signal. For transmission at moderate data rates over twisted pairs, such as the current ISDN standard of 160 kb/s, precursor ISI is typically very small, so that $P(z)$ can be replaced by a constant gain. In this case, the pole-zero DFE proposed here combined with the timing recovery scheme in [19] may be attractive.

III. Adaptive Algorithms

We first show how the equation error method and a simplified gradient algorithm can be used to update the coefficients of the IIR feedback filter in the proposed DFE. We then describe some different sequentially adaptive IIR algorithms which give unbiased estimates of the channel poles in the presence of additive white noise, and converge faster than the previous algorithms. It will be seen that these proposed IIR algorithms are closely related to the IIR gradient algorithm. Although the algorithms are explicitly stated assuming that $C(z)$ and $D(z)$ are given by (1), generalizations to higher-order polynomials are straightforward.

A. The Equation Error Method

Fig. 5 shows a block diagram of an adaptive version of the DFE in Fig. 3 using the equation error method [15], [16], [20]. The output of the prefilter $P(z)$ in Fig. 5 at time iT, where $1/T$ is the symbol rate, is

$$r(i) = \sum_{k=-\infty}^{\infty} s(k)h(i-k) + n(i) \tag{3}$$

where $h(i)$ is the ith sample of the equivalent discrete-time IR of the channel including filtering at the transmitter, receiver, and $P(z)$, $s(k)$ is the kth transmitted symbol, and $\{n(i)\}$ is a white noise sequence. We will assume

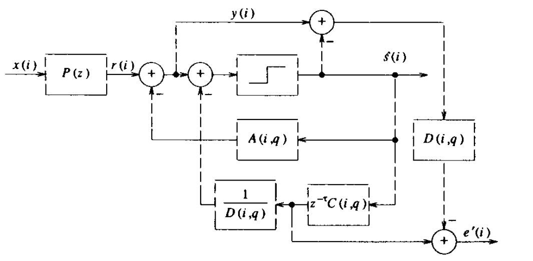


Fig. 5. DFE with pole-zero feedback filter adapted via the equation error method.

throughout the rest of the paper that $h(0) = 1$. That is, an automatic gain control ensures that the cursor is set to one.

In HDSL applications, the additive noise $n(i)$ may contain crosstalk and is therefore likely to be colored. However, a fractionally-spaced feedforward filter, $P(z)$, can be used to both whiten wide-sense stationary noise and suppress cyclostationary interference (i.e., crosstalk) [21]. Consequently, if the prefilter in the DFE is adaptive, then the assumption of additive white noise seems appropriate for initial comparisons between the proposed pole-zero DFE and the conventional DFE.

Except for $P(z)$, all of the filters shown in Fig. 5 are assumed to be adaptive and are therefore time-varying. An adaptive prefilter will be considered later in this section. In order to distinguish a transfer function that is being adapted from the standard z-transform which assumes a time-invariant filter, we use the arguments (i, q) to denote a time-varying transfer function, where q replaces z as the delay operator. That is,

$$A(i, q) = \sum_{k=1}^{\tau-1} a_k(i) q^{-k}$$

where $a_k(i)$, $k = 1, \cdots, \tau - 1$, are the filter coefficients at time i. The output of $A(i, q)$ at time i in response to the input sequence $\{\hat{s}(k)\}$ is denoted as

$$A(i, q)\,[\hat{s}(i)] = \sum_{k=1}^{\tau-1} a_k(i)\hat{s}(i - k).$$

Let $H(z)$ denote the z-transform of the impulse response sequence $\{h(i)\}$. Then it is easily verified that the equation error shown in Fig. 5 satisfies $e'(i) = D(i, q)[n(i)]$ if $H(z) = 1 + A(z) + z^{-\tau}\, C(z)/D(z)$ and the symbol decisions $\hat{s}(i) = s(i)$ for every i.

A stochastic gradient algorithm can be used to adapt the filters $A(i, q)$, $D(i, q)$, and $C(i, q)$ to minimize the mean squared equation error $E[e'^2(i)]$ where $e'(i)$ is shown in Fig. 5. Specifically,

$$e'(i) = D(i, q)\,[y(i) - \hat{s}(i)] - C(i, q)\,[\hat{s}(i - \tau)] \tag{4a}$$

$$a_k(i + 1) = a_k(i) + \beta_1\hat{s}(i - k)e'(i), \quad k = 1, \cdots, \tau - 1 \tag{4b}$$

$$c_k(i + 1) = c_k(i) + \beta_1\hat{s}(i - \tau - k)e'(i), \quad k = 0, 1 \tag{4c}$$

$$d_k(i + 1) = d_k(i) + \beta_2[y(i - k) - \hat{s}(i - k)]e'(i), \quad k = 1, 2 \tag{4d}$$

where the coefficients of $D(z)$ and $C(z)$ are given by (1), and β_1 and β_2 are step-sizes. Two copies of the filter $D(i, q)$ are shown in Fig. 5. One is used to generate the equation error, and $1/D(i, q)$ is used to estimate postcursor ISI.

Since $E[e'^2(i)]$ is a quadratic function of the filter coefficients of $A(z)$, $C(z)$, and $D(z)$, it has a unique minimum. Specifically, let $S(z)$, $R(z)$, and $N(z)$ be the z-transforms of the sequences $\{s(i)\}$, $\{r(i)\}$, and $\{n(i)\}$, respectively, where $r(i)$ is given by (3). Assuming that $\hat{s}(i) = s(i)$ for every i, then the z-transform of the equation error sequence is

$$\begin{aligned} E(z) &= [R(z) - A(z)S(z) - S(z)]D(z) - z^{-\tau}C(z)S(z) \\ &= [H(z)S(z) + N(z) - A(z)S(z) - S(z)]D(z) \\ &\quad - z^{-\tau}C(z)S(z) \\ &= \{[H(z) - A(z) - 1]D(z) - z^{-\tau}C(z)\}S(z) \\ &\quad + D(z)N(z). \end{aligned} \tag{5}$$

Assuming the transmitted symbols are independent and that the noise is independent of the symbols, then to minimize $E[e'^2(i)]$ the coefficients of $A(z)$ and $C(z)$ should be selected to cancel as many terms (powers of z) as possible in the braces. It follows that $a_k = h(k)$, $k = 1, \cdots, \tau - 1$, $c_0 = h(\tau)$, and $c_1 = h(\tau + 1) - d_1h(\tau)$.

Suppose now that

$$H(z) = 1 + \tilde{A}(z) + \frac{z^{-\tau}C(z)}{\tilde{D}(z)}$$

for fixed $\tilde{A}(z)$ and $\tilde{D}(z)$, where the orders of $\tilde{A}(z)$ and $A(z)$ are the same. If $A(z) = \tilde{A}(z)$, then (5) implies that

$$E(z) = \left(\frac{D(z)}{\tilde{D}(z)} - 1\right) z^{-\tau}C(z)S(z) + D(z)N(z),$$

so that the $D(z)$ that minimizes $E[e'^2(i)]$ is

$$D(z) = \frac{\sigma^2 C(z)C(z^{-1})\tilde{D}(z)}{S_n(z)\tilde{D}(z)\tilde{D}(z^{-1}) + \sigma^2 C(z)C(z^{-1})} \tag{6}$$

where $S_n(z) = E[N(z)N(z^{-1})]$ is the noise spectrum, and the transmitted symbols are assumed to be independent and identically distributed with variance σ^2. In the absence of noise, $D(z) = \tilde{D}(z)$ so that minimizing the mean squared equation error also minimizes the MSE, $E[e^2(i)]$. However, in the presence of noise, the estimate of $\tilde{D}(z)$ is biased, resulting in residual ISI.

After some inspection, it becomes apparent that the DFE structure in Fig. 5 is closely related to the conventional DFE in Fig. 1. Specifically, in Fig. 5 the denominator polynomial $D(i, q)$ is in casade with a "modified" channel $\tilde{H}(z)$. That is, if $A(z)$ is selected optimally, then $\tilde{H}(z)$ differs from $H(z)$ only in that the first $\tau - 1$ coefficients of $\tilde{H}(z)$ are zero. The effect of the filter $D(i, q)$ in Fig. 5 is therefore similar to that of a prefilter in a conventional DFE. The structure in Fig. 5 differs from the conventional DFE, however, in that the equation error which is being minimized is not the true performance criterion. In addition, the convergence properties of the structure in Fig. 5 may differ from those of the conventional DFE since the input to $D(i, q)$ in Fig. 5 is different from the input to $P(z)$ in Fig. 1.

Assuming that $D(z)$ and $C(z)$ are given by (1), it is of interest to compare the output MSE of a conventional DFE with $\tau + 1$ feedback coefficients and a second-order prefilter with that of the pole-zero DFE in Fig. 5 where $P(z) = 1$ and the mean squared equation error is minimized. Stated another way, the comparison is between the structure in Fig. 5 and the conventional DFE that can be derived from Fig. 5 by moving the filter $D(i, q)$ before $P(z)$ and setting $1/D(i, q)$ in Fig. 5 to one. This comparison can be interpreted as trading residual ISI, caused by a biased channel estimate, for noise enhancement caused by a longer prefilter. The numerical results in Section IV indicate that the MSE, or mean squared output error, in Fig. 5 is typically larger than the MSE corresponding to the conventional DFE. There are some cases, however, in which the structure in Fig. 5 performs marginally better than the conventional DFE.

It has been assumed so far that $P(z)$ is a fixed filter that compensates for precursor ISI. Ideally, it is desirable to be able to adapt $P(z)$ simultaneously with the feedback filter to minimize the mean squared equation error. However, because $P(z)$ is in cascade with $D(i, q)$ in Fig. 5, $E[e'^2(i)]$ is not a quadratic function of the coefficients of these filters. The equation error method, therefore, cannot be readily applied to simultaneously adapt the coefficients of both $P(i, q)$ and $D(i, q)$, although it may be possible to update $P(i, q)$ by using the output error $e(i)$ instead of the equation error. This latter scheme was not tried, however, since the numerical results in Section IV show that the MSE produced by the equation error method when precursor ISI is negligible is often worse than that produced by a conventional DFE of the same complexity.

B. Simplified Gradient Algorithm

The filters $A(i, q)$, $C(i, q)$, $D(i, q)$, and $P(i, q)$ in Fig. 3 can also be adapted to minimize the MSE directly, but at the risk of converging to a local, but not global, optimum. One can at least reduce the MSE resulting from the equation error method by switching to a gradient algorithm using the MSE cost function after the equation error algorithm has converged. Furthermore, for fixed $P(z)$, if the postcursor channel IR can be modeled with the feedback filter in Fig. 3, where $D(z)$ has one or two poles, then the MSE cost function does not contain local optima [4].

Referring to Fig. 3, $A(i, q)$ can be adapted to minimize either $E[\epsilon^2(i)]$ or the MSE, $E[e^2(i)]$. In either case, optimality occurs when $a_k = h(k)$, $k = 1, \cdots, \tau - 1$. Adapting $A(i, q)$ with the error $\epsilon(i)$ has the advantage that $A(i, q)$ becomes independent of $D(i, q)$. That is, the convergence behavior of $A(i, q)$ is the same as if $D(i, q)$ were not present. However, adapting $A(i, q)$ with $\epsilon(i)$ has the disadvantage that $\epsilon(i)$ contains residual ISI that cannot be cancelled by $A(i, q)$ alone. This residual ISI acts as a noise source which increases coefficient variance due to adaptation. Of course, this variance can be reduced by decreasing the gradient step-size, but at the expense of increasing the convergence time. The update equation for the coefficients of $A(i, q)$ remains the same as (4b), where $e'(i)$ is replaced by either $\epsilon(i)$ or $e(i)$. (The error $e(i)$ was used to generate the results in Section IV.)

A simplified gradient algorithm can be used to adapt $C(i, q)$, and $D(i, q)$ to minimize the MSE [15]:

$$g^c(i) = \frac{1}{D(i, q)} [\hat{s}(i - \tau)] \tag{7a}$$

$$v(i) = \frac{C(i, q)}{D(i, q)} [\hat{s}(i - \tau)] \tag{7b}$$

$$g^d(i) = \frac{1}{D(i, q)} [v(i - 1)] \tag{7c}$$

$$e(i) = y(i) - \hat{s}(i) - v(i) \tag{7d}$$

$$c_k(i + 1) = c_k(i) + \beta g^c(i - k)e(i), \qquad k = 0, 1 \tag{7e}$$

$$d_k(i + 1) = d_k(i) + \beta g^d(i - k + 1)e(i), \qquad k = 1, 2 \tag{7f}$$

where β is the step-size, $g^d(i - k) \approx \partial e(i)/\partial d_k(i)$, and $g^c(i - k) \approx \partial e(i)/\partial c_k(i)$ [15]. Note that

$$g^c(i) = \frac{1}{D(i, q)} [\hat{s}(i - \tau)] = \hat{s}(i - \tau) + \sum_{k=1}^{2} d_k(i) g^c(i - k).$$

The filter $P(i, q)$ can be updated in the conventional way:

$$p_k(i + 1) = p_k(i) + \beta' x(i - k)e(i), \qquad k = 0, 1, \cdots, K \tag{8}$$

where $p_k(i)$, $k = 0, 1, \cdots, K$ are the coefficients of $P(i, q)$, $x(i)$ is the input to $P(i, q)$ at time i, and β' is a step-size that may be different from β in (7). The algorithm can be initialized by setting all variables to zero. The stability conditions (2) can be checked at each iteration. If one of these conditions is violated, then the coefficient updates (7f) are not performed. Note that this algorithm requires somewhat more computation than the equation error method, due to the filtering of the sequence $\{v(i)\}$ by $1/D(i, q)$ to produce the (approximate) gradient signal $g^d(i)$. The gradient algorithm is illustrated in Fig. 6, which shows the intermediate gradient signals $g^c(i)$ and $g^d(i)$.

A sequential gradient algorithm can also be easily obtained for the IIR feedback filter shown in Fig. 4. That is, $C'(i, q)$ and $D'(i, q)$ in Fig. 4 can be adapted according to an appropriate modification of (7). Unlike the preceding gradient algorithm for the structure in Fig. 3, the coef-

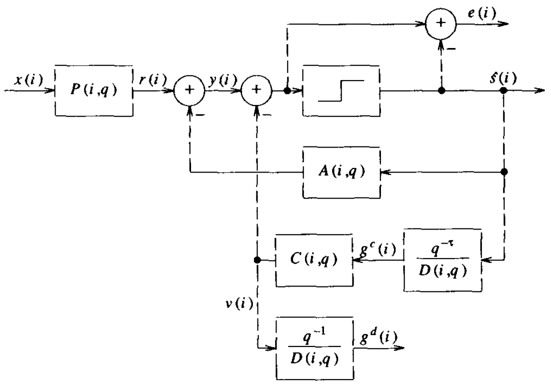


Fig. 6. Illustration of the simplified IIR gradient algorithm.

ficients of $C'(i, q)$ in Fig. 4 depend strongly on the coefficients of $D'(i, q)$ resulting in very slow convergence.

C. Sequential Steiglitz-McBride Algorithms

In order to explain the algorithms that follow, we first give a brief description of the Steiglitz-McBride (SM) identification algorithm [22]. Referring to Fig. 7, suppose that the unknown system with rational transfer function $C(z)/D(z)$ is to be identified with a finite number of input data samples $u(0), \cdots, u(N - 1)$. Suppose that the N' outputs, $y(0), \cdots, y(N' - 1)$, are observed where $N' \geq N$. (For purposes of the following discussion, any reasonable windowing scheme for the input data can be assumed.) Let $\hat{D}(z)$ in Fig. 7 be some fixed filter. Then by solving a set of linear equations, we can compute the filters $X(z)$ and $Y(z)$ that minimize the sum of the squared equation errors $\Sigma_{i=0}^{N'} e'^2(i)$. If $\hat{D}(z) = 1$, this is simply the equation error method, and the resulting estimate of $D(z)$, namely $Y(z)$, is biased. However, if $\hat{D}(z) = D(z)$, then the estimates obtained will be unbiased when the noise is white.

The preceding observation suggests the following iterative estimation scheme [22].

i) Initialize $\hat{D}(z) = 1$.

ii) Compute the $X(z)$ and $Y(z)$ that minimize $\Sigma_{i=0}^{N'} [e'(i)]^2$ in Fig. 7.

iii) Set $\hat{D}(z) = Y(z)$.

iv) Repeat from ii) until $\hat{D}(z)$ and $Y(z)$ are sufficiently close.

1) The Fan-Jenkins Algorithm: It is possible to use the SM algorithm i)–iv) to estimate the feedback filters in Fig. 3 given a block of transmitted symbols and the corresponding channel outputs. However, the least squares (LS) estimate in step ii) is computationally expensive to obtain, and this step must be executed many times. An algorithm that processes the data sequentially is also desirable in this application.

A sequential IIR adaptive algorithm, based on the SM algorithm, has been proposed by Fan and Jenkins [18]. Rather than compute the LS estimate in step ii), for fixed $\hat{D}(z)$ it is possible to adapt $X(i, q)$ and $Y(i, q)$ with the standard LMS algorithm. Furthermore, since $Y(i, q)$ should converge to $D(i, q)$, we can simply replace $\hat{D}(z)$ in Fig. 7 by $Y(i, q)$. Applying this adaptive algorithm to the pole-zero DFE results in the algorithm (7), where (7c) and (7f) are replaced, respectively, by

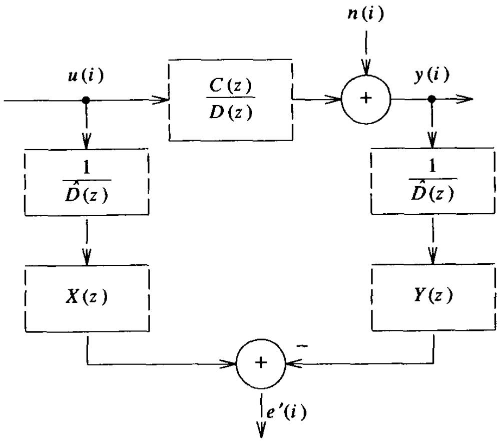


Fig. 7. Block diagram illustrating the Steiglitz-McBride identification algorithm.

$$g^d(i) = \frac{1}{D(i, q)} [y(i) - \hat{s}(i)]$$

and

$$d_k(i + 1) = d_k(i) + \beta g^d(i - k)e(i), \qquad k = 1, 2.$$

This algorithm is illustrated in Fig. 8. Because the cascaded filters $1/D(i, q)$ and $D(i, q)$ in Fig. 8 result in a (time-invariant) unity transfer function, the error $e(i)$ in Fig. 8 is given by (7d).

As pointed out in [18], the only difference between this algorithm and the gradient algorithm (7) is that in the Fan–Jenkins (FJ) algorithm $y(i) - \hat{s}(i)$ is filtered by $1/D(i, q)$ to produce the gradient $g^d(i)$ for updating $D(i, q)$, whereas in the gradient algorithm $v(i)$ is filtered to produce $g^d(i)$. Since $v(i)$ is the current estimate of $y(i) - \hat{s}(i)$, the performance of the FJ algorithm is likely to be similar to that of the gradient algorithm. Results in [18], as well as our own simulation results, indicate that this is indeed the case. Specifically, the pole-zero DFE in Fig. 3 was simulated using both the gradient and FJ algorithms to adapt the pole-zero feedback filter. The speed of convergence for both algorithms was observed to be virtually identical. Consequently, numerical results for the FJ algorithm are not explicitly shown in the next section, since they are essentially the same as those shown for the gradient algorithm.

Although the FJ algorithm does not immediately improve upon the performance of the adaptive IIR algorithms already discussed, Figs. 7 and 8 suggest the following modifications of the FJ algorithm, which do improve performance for the application considered.

2) Reversing the Order of Filtering—the SM-LMS Algorithm: Referring to Fig. 7, the order in which the

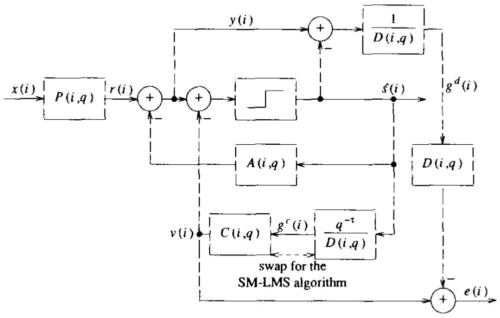


Fig. 8. Illustration of the Fan-Jenkins and SM-LMS algorithms. The order of the filters $C(i, q)$ and $q^{-\tau}/D(i, q)$ is reversed for the SM-LMS algorithm.

filters $1/\hat{D}(z)$ and $X(z)$ appear is intrinsic to the SM algorithm. That is, the output of the filter $1/\hat{D}(z)$ in response to the input sequence $\{u(i)\}$ is used to compute $X(z)$, so that the order of these filters cannot be reversed. If, however, a sequential algorithm such as the LMS algorithm is used to adapt $X(i, q)$ and $Y(i, q)$, then in steady state it should not matter whether $X(i, q)$ comes before or after $1/\hat{D}(i, q)$. This simple interchange, however, can have a dramatic effect on the convergence properties of the algorithm.

For example, in the DFE application the input sequence $\{u(i)\}$ consists of the transmitted symbols, which are typically uncorrelated. Filtering this sequence by $1/\hat{D}(z)$ causes the input to $X(z)$ to Fig. 7 to be correlated. Loosely speaking, the closer the zeroes of $\hat{D}(z)$ are to the unit circle, the more correlated the output of $1/\hat{D}(z)$ will be. It is well known that the LMS algorithm converges much faster in response to uncorrelated data, as opposed to a strongly correlated input. Consequently, faster convergence relative to that of the FJ algorithm should be obtained by using an uncorrelated sequence as the input to $X(i, q)$, and placing $1/\hat{D}(i, q)$ after $X(i, q)$.

The preceding discussion suggests switching the order of $1/D(i, q)$ and $C(i, q)$ shown in Fig. 8. The FIR filters $D(i, q)$ and $C(i, q)$ can be adapted to minimize $E[e^2(i)]$ via the LMS algorithm, assuming that the all-pole filter $1/D(i, q)$ is time-invariant. Of course, $1/D(i, q)$ *is* time-varying and is determined by the current estimate $D(i, q)$. Specifically, the SM-LMS algorithm is

$$g^d(i) = \frac{1}{D(i, q)}[y(i) - \hat{s}(i)] \tag{9a}$$

$$e(i) = y(i) - \hat{s}(i) - \frac{C(i, q)}{D(i, q)}[\hat{s}(i - \tau)] \tag{9b}$$

$$d_k(i + 1) = d_k(i) + \beta g^d(i - k)e(i), \quad k = 1, 2 \tag{9c}$$

$$c_k(i + 1) = c_k(i) + \beta\hat{s}(i - \tau - k)e(i), \quad k = 0, 1. \tag{9d}$$

$A(i, q)$ and $P(i, q)$ can again be adapted via the LMS algorithm.

The input to $C(i, q)$ in this case is the sequence of uncorrelated transmitted symbols. If we assume for the moment that $1/D(i, q)$ is time-invariant, then the statistics of the error $e(i)$ are approximately the same as the statistics of the error in the FJ algorithm. Consequently, $C(i, q)$ should converge faster in this configuration, as compared with the configuration shown in Fig. 8, when the LMS algorithm is used to update the coefficients. The numerical results in Section IV show that this is indeed the case, where the speedup in convergence of (9) relative to the gradient algorithm is roughly a factor of 4–5.

Of course, it is also possible to switch the order of the cascaded filters $1/D(i, q)$ and $D(i, q)$ in Fig. 8. However, because the signal $y(i) - \hat{s}(i)$ is correlated, adapting $D(i, q)$ with this signal instead of $g^d(i)$, as in (9), is unlikely to offer a substantial improvement in performance.

3) A Sequential SM Algorithm with Recursive Least Squares (RLS) Updates: For any fixed $\hat{D}(z)$ in Fig. 7, the least squares (LS) estimate of $X(z)$ and $Y(z)$ in step ii) of the SM algorithm can be obtained sequentially. That is, LS estimates for $Y(z)$ and $X(z)$, given $N + 1$ input and output samples, can be obtained recursively from the LS estimates of $Y(z)$ and $X(z)$ given N input and output samples plus some additional state information. Once the recursive LS (RLS) algorithm has converged for fixed $\hat{D}(z)$, we can then replace $\hat{D}(z)$ by $Y(z)$, reinitialize the RLS algorithm, and recompute $X(z)$ and $Y(z)$.

The RLS algorithm must be reinitialized after updating $\hat{D}(z)$, since state information corresponding to the old $\hat{D}(z)$ would otherwise be retained, thereby corrupting successive estimates. This has the disadvantage, however, that the RLS algorithm must be periodically terminated and restarted with zero state information, so that the resulting estimates of $Y(z)$ and $X(z)$ are quite poor until the algorithm has once again converged. An attractive alternative is to use an RLS algorithm with exponentially fading memory. That is, the weighted sum of squared errors, $\Sigma_{i=0}^{N} w^{N-i}[e'(i)]^2$, is minimized for each N. This way, the algorithm discounts past input data and can track changing input statistics, which is caused by updating $\hat{D}(i, q)$ in Fig. 7. (There are other ways of implementing RLS algorithms with fading or finite memory [23]; however, exponential weighting is relatively simple and is found to perform quite well for the examples in the next section.)

The rate at which $\hat{D}(i, q)$ is updated determines how fast the statistics of the input to the RLS algorithm are changing. For fastest convergence, it seems plausible to update $\hat{D}(i, q)$ as fast as possible while maintaining stability. Algorithm stability also depends critically on the exponential weight w. As w decreases, the RLS algorithm tracks changing statistics faster although the variance of the estimate increases, potentially causing instability. We chose to update $\hat{D}(i, q)$ at each iteration, so that the stability and convergence speed of the algorithm is determined solely by w. For the two-pole example considered in the next section, taking $w = 0.999$ gave satisfactory results.

To show how the adaptive IIR algorithm just described can be applied to the proposed pole-zero DFE, we define the following vectors associated with Fig. 8.

$$\mathbf{g}'(i) = [g^d(i-1), g^d(i-2), g^c(i), g^c(i-1)]$$

$$\mathbf{d}'(i) = [d_1(i), d_2(i), c_0(i), c_1(i)].$$

The coefficients of $D(i, q)$ and $C(i, q)$ are updated as follows.

Initialization:

$$\mathbf{d}(0) = \mathbf{0},\ \mathbf{R}^{-1}(0) = \frac{1}{\delta}\mathbf{I},\ \hat{s}(i) = g^c(i) = g^d(i) = 0,\ i < 0 \tag{10a}$$

For $i = 0, 1, 2, \cdots$ do:

$$g^c(i+1) = \frac{1}{D(i, q)}[\hat{s}(i+1-\tau)] \tag{10b}$$

$$g^d(i+1) = \frac{1}{D(i, q)}[y(i+1) - \hat{s}(i+1)] \tag{10c}$$

$$e(i+1) = y(i+1) - \hat{s}(i+1) - \frac{C(i, q)}{D(i, q)}[\hat{s}(i+1-\tau)] \tag{10d}$$

$$\mathbf{y}(i+1) = \mathbf{R}^{-1}(i)\mathbf{g}(i+1) \tag{10e}$$

$$\mathbf{R}^{-1}(i+1) = \frac{1}{w}\left(\mathbf{R}^{-1}(i) - \frac{1}{w + \mathbf{g}'(i+1)\mathbf{R}^{-1}(i)\mathbf{g}(i+1)} \cdot [\mathbf{y}(i+1)\mathbf{y}'(i+1)]\right) \tag{10f}$$

$$\mathbf{d}(i+1) = \mathbf{d}(i) + [\mathbf{R}^{-1}(i+1)\mathbf{g}(i+1)]e(i+1). \tag{10g}$$

In the initialization, δ is chosen to be some small constant that is large enough to ensure stability. The RLS algorithm (10) uses the matrix inversion lemma [24, Sect. 5.1] to compute the inverse of

$$\mathbf{R}(i) = \sum_{k=0}^{i} w^{i-k}\mathbf{g}(k)\mathbf{g}'(k). \tag{11}$$

(Strictly speaking, $\mathbf{R}^{-1}(i)$ in (10) only approximates the inverse of $\mathbf{R}(i)$ defined by (11) due to the approximate initial condition.)

Of course, there are many alternative RLS algorithms to the one used in (10). Of particular interest are the ARMA lattice algorithms described in [25]. (See also [24, Sect. 10.2]). That is, the filters $C(i, q)$ and $D(i, q)$ in Fig. 8 can be implemented as a single ARMA lattice filter, and the filter $1/D(i, q)$ can also be implemented as an all-pole lattice filter [25], which greatly simplifies stability checking when $D(i, q)$ has more than two zeroes. In addition, lattice structures are generally less sensitive to roundoff errors than direct form implementations. It is also possible to replace the RLS algorithm (10) by one of the computationally efficient RLS transversal algorithms described in [23] and [24, Ch. 5]. However, for the example considered in the next section, $\mathbf{R}$ is a 4 × 4 matrix, so that the potential savings in complexity offered by the "fast" RLS algorithms is minor.

A simple way to reduce the complexity of the SM-RLS algorithm (10), while compromising performance only a little, is to use the RLS algorithm to estimate *only* the poles $[D(i, q)]$ and use the standard LMS algorithm to estimate $C(i, q)$. Specifically, we can redefine the vectors

$$\mathbf{g}'(i) = [g^d(i-1), g^d(i-2)]$$

$$\mathbf{d}'(i) = [d_1(i), d_2(i)]$$

so that $\mathbf{R}(i)$, defined by (11), is now a 2 × 2 matrix. The SM-RLS algorithm (10) can then be used to estimate $D(i, q)$, and $C(i, q)$ can be updated as in the SM-LMS algorithm, i.e., via (9d). The convergence speed of the resulting algorithm will be somewhat slower than when the RLS algorithm is used to update both $D(i, q)$ and $C(i, q)$. However, the results in the next section indicate that the savings in complexity makes this an attractive alternative for the application considered.

There are, of course, many other possible variations on the algorithms proposed in this section. In general, the filters $C(i, q)$ and $D(i, q)$ can be updated via any FIR adaptive algorithm and can be implemented with any FIR filter structure. The resulting estimate for $D(i, q)$ can then be substituted back into the all-pole filter $1/D(i, q)$ in Fig. 8. (Again, depending on the particular algorithm considered, it may be advantageous to change the order of the filters in Fig. 8.) The stability and convergence properties of this class of algorithms is a very interesting topic which is not pursued here.

IV. Numerical Results

Throughout this section, $C(z)$ and $D(z)$ in the pole-zero DFE are assumed to be given by (1). The conventional DFE, to which the pole-zero DFE will be compared, has $\tau + 1$ feedback taps, where $\tau - 1$ is the number of taps in $A(z)$ in Fig. 3, and has a prefilter containing two more zeros than the number of zeroes in the pole-zero DFE prefilter. The number of feedback taps in $F(z)$ in the conventional DFE shown in Fig. 1 is therefore the same as the total number of taps in $A(z)$ and $C(z)$ shown in Fig. 3. Also, the number of taps in $P(z)$ in Fig. 1 is the same as the number of taps in the pole-zero DFE prefilter plus the number of taps in $D(z)$ in Fig. 3. As discussed in Section III-A, this illustrates the design option of adding taps to the prefilter of a DFE, which in the presence of severe ISI tends to shorten the channel IR at the expense of enhancing the noise, or using the additional taps to construct a pole-zero feedback filter.

Before presenting performance results for adaptive DFE's, we first show the potential advantage in MSE offered by the proposed pole-zero DFE relative to a conventional DFE having the same complexity. We will subsequently demonstrate that this performance advantage can be achieved when the channel is initially unknown by using one of the adaptive IIR algorithms described in the last section. All of the results presented here assume that the channel has the measured subscriber loop IR shown in Fig. 2. (Filtering at the transmitter and receiver is assumed to be included in this IR.)

A. Comparisons of Minimum MSE

Two sets of results are shown in Figs. 9 and 10. The first set, shown in Fig. 9, assumes a symbol rate of 400 kbauds, which, because of the sharp leading edge of the IR in Fig. 2, is slow enough so that precursor ISI is quite small. Consequently, the pole-zero DFE in this case has $P(z) = 1$. The second set of results in Fig. 10 corresponds to a symbol rate of 800 kbauds, so that precursor ISI is no longer negligible. Here we assume that the pole-zero DFE has a third-order prefilter $P(z)$. The results at the lower symbol rate are included for two reasons. First, adapting the prefilter $P(i, q)$ in addition to the pole-zero feedback filter is significantly more difficult than adapting only the feedback filter with a fixed prefilter. It is, therefore, of interest to compare the performance of adaptive algorithms in these two situations. The second reason is that assuming a fixed prefilter simplifies the computation of MSE when the mean squared equation is to be minimized. The performance of the equation error method is therefore shown only at the lower symbol rate.

Fig. 9 shows a plot of MSE versus the total number of taps in $A(z)$ and $C(z)$ for the DFE structure in Fig. 3. The signal-to-noise ratio (SNR) is 20 dB, and is defined as $x^2/E[n^2(i)]$, where x is the maximum value of the channel impulse response shown in Fig. 2, and the noise sequence is assumed to be white. Plots are shown corresponding to minimization of both the mean squared equation error and MSE. Also shown are plots of minimum MSE (MMSE) versus number of taps in $F(z)$ for the conventional (FIR) DFE in Fig. 1, assuming that the prefilter $P(z)$ is a second-order FIR filter, and assuming that $P(z) = 1$. In the former case, the total number of filter coefficients corresponding to an x-axis entry in Fig. 9 is the same as the number of filter coefficients for the pole-zero DFE. The results shown in Figs. 9 and 10 assume that the sampling delay (in number of symbols) is optimized for both structures, and that the sampling phase is the same in each case.

Fig. 9 shows that the equation error method performs much worse than a conventional DFE with a second-order prefilter when the number of feedback taps is relatively small (less than 20), and performs about the same as the conventional DFE when the number of feedback taps is large. Consequently, the biased pole estimates obtained via the equation error method in Fig. 5 generally create more MSE in the form of residual ISI than the MSE due to noise enhancement from a second-order prefilter. Fig. 9 also shows, however, that if the MSE is minimized directly, then the pole-zero structure requires only 13 feedback taps to achieve the maximum SNR of 20 dB. The improvement relative to the conventional DFE is nearly 4 dB in this case. Alternatively, to achieve the performance of the pole-zero DFE with 13 feedback taps, a conventional DFE requires nearly 40 feedback taps.

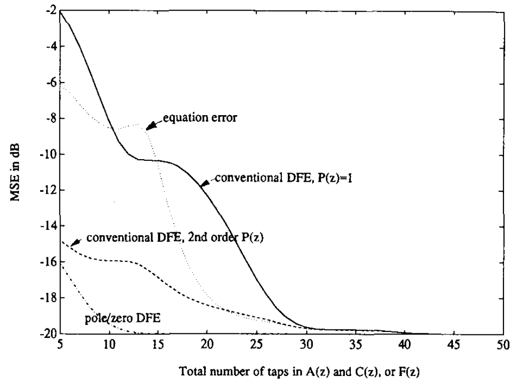


Fig. 9. Minimum MSE versus number of feedback taps for the pole-zero DFE in Fig. 3, and a conventional DFE with an FIR feedback filter. The symbol rate is 400 kbaud and the SNR is 20 dB. Output MSE for the pole-zero DFE assuming that the mean squared equation error is minimized is also shown.

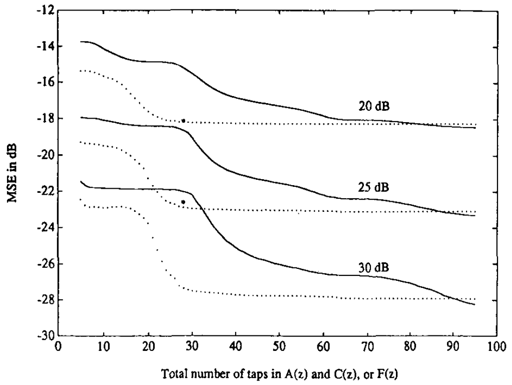


Fig. 10. MMSE versus number of feedback taps for the pole-zero DFE and the conventional DFE, assuming SNR's of 20, 25, and 30 dB. The symbol rate is 800 kbauds. The pole-zero DFE contains a third-order prefilter, and the conventional DFE contains a fifth-order prefilter. The asterisks correspond to asymptotic averaged squared error for the conventional and pole-zero DFE's when adapted via the LMS and SM-RLS algorithms, respectively, and assuming that the SNR is 25 dB in each case.

Fig. 10 compares the MMSE for the pole-zero DFE with that of a conventional DFE at three different SNR's, assuming a symbol rate of 800 kbauds. Third- and fifth-order prefilters are assumed for the pole-zero DFE and the conventional DFE, respectively. Consequently, as the number of feedback taps goes to infinity, the conventional DFE performs slightly better than the pole-zero DFE because of the extra feedforward taps. In all cases, however, the pole-zero DFE requires less than 30 feedback taps to achieve nearly the same MMSE as a conventional DFE with a third-order prefilter and an infinite number of feedback taps. Fig. 10 shows that to achieve this MSE, the conventional DFE requires approximately three times the number of feedback taps as the pole-zero DFE. Alternatively, assuming 30 feedback taps, as the SNR increases from 20 to 30 dB, the difference in MMSE for the two structures increases from approximately 3 dB to 5.5 dB. The reason for this is that the MMSE for the conventional DFE with only 30 feedback taps is due primarily to residual ISI. Consequently, the difference between this MMSE and 1/SNR increases as the SNR increases. In contrast, the pole-zero DFE with 30 feedback taps is able to eliminate ISI, so that the MMSE in this case is approximately 1/SNR. The upper and lower asterisks in Fig. 10 correspond to the averaged squared error obtained by the LMS algorithm and the SM-RLS algorithm, respectively after 50 000 iterations, assuming that the SNR is 25 dB.

We remark that the computation of the MMSE for the pole-zero DFE shown in Fig. 3 is not entirely straightforward. This is because the MSE is a nonlinear function of the feedback taps. Including $P(z)$ in the optimization further complicates the computation. With fixed $P(z)$, the MMSE was computed by emulating the SM algorithm. That is, the algorithm (i)–(iv) in Section III-C was used to compute the filters $C(z)$ and $D(z)$ in Fig. 8, where instead of computing LS estimates of $C(z)$ and $D(z)$ from randomly generated data, MMSE estimates were computed from the channel IR. In general, this procedure is not guaranteed to converge to the globally optimal solution although, as previously pointed out, this is not a problem when two poles are sufficient to model the tail of the channel IR. To compute the MMSE with $P(z)$ present, we used the following simple approach. First compute the MMSE feedback filter with some fixed $P(z)$. Then fix the feedback filter and compute the MMSE prefilter, and so forth until convergence. Initially, $P(z)$ was set to 1. Although local minima may again be a problem with this algorithm, the computed values for the MSE shown in Fig. 10 appear to be global minima.

B. Performance Comparisons of Adaptive IIR Algorithms

We now compare the performance of the adaptive IIR algorithms described in Section III. Fig. 11 shows averaged squared error versus iteration for the pole-zero DFE using the equation error algorithm (4), the simplified gradient algorithm (7), and the SM-LMS algorithm (9) at the lower symbol rate of 400 kbauds and an SNR of 20 dB. The number of feedback taps in $A(i, q)$ is 13 for the SM-LMS algorithm and is 15 for the other two algorithms. All of the simulation results presented here assume binary transmitted symbols, i.e., $s(i) \in \{1, -1\}$. The asymptotic averaged squared error in Fig. 11 is somewhat greater than the corresponding results for MSE in Fig. 9 due to tap fluctuations. This excess MSE is less than 1 dB in all cases. The plots in Fig. 11 were generated by averaging successive blocks of 1250 squared errors.

Fig. 11 shows that the SM-LMS algorithm takes less than 125 000 iterations to converge, which is more than four times faster than the convergence speed of the gradient algorithm, and is comparable with the convergence speed of the equation error algorithm. It was also found that the SM-LMS algorithm is much less sensitive to the choice of step-size than the gradient algorithm.

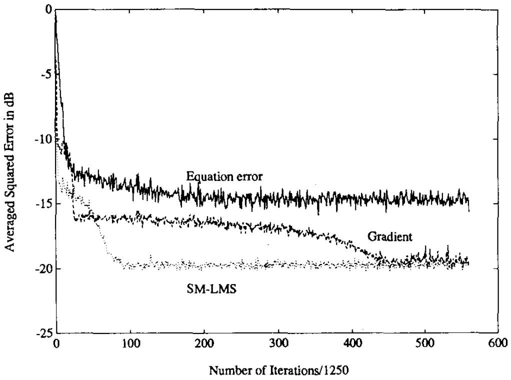


Fig. 11. Averaged squared error versus iteration for an adaptive pole-zero DFE assuming: i) the equation error algorithm (4); ii) the IIR gradient algorithm (7); and iii) the SM-LMS algorithm (9) is used to adapt the pole-zero filter. The symbol rate is 400 kbauds and the SNR is 20 dB. The feedback filter $A(i, q)$ contains 13 taps for the SM-LMS algorithm, and 15 taps for the other two algorithms.

It is not surprising that the equation error algorithm (4) converges faster than the gradient algorithm (7) since the mean squared equation error is a quadratic function of the feedback taps, whereas the output MSE is a nonlinear function of the feedback taps. This is illustrated in Fig. 12, which shows MSE^{-1} versus the feedback taps d_1 and d_2 in the pole-zero DFE. Each point on the surface corresponds to the optimal choice of $A(z)$ and $C(z)$ given d_1 and d_2. There is only one local optimum; however, the cost function exhibits a sharp peak at the optimal values of d_1 and d_2, and is relatively flat elsewhere. The averaged squared error corresponding to the gradient algorithm in Fig. 11 therefore stays relatively constant, while the gradient algorithm traverses the plateau in Fig. 12, and subsequently decreases to its minimum value as the gradient algorithm traverses the spike. Although the convergence properties of both algorithms can be altered somewhat by changing the step-size β, the comparison in Fig. 11 is a representative illustration of the slow convergence of the IIR gradient algorithm.

The SM-RLS algorithm described in Section III-C is compared with the conventional FIR DFE in Fig. 13, assuming the lower baud rate of 400 kbauds, and an SNR of 25 dB. The pole-zero DFE has 13 taps in the feedback filter $A(i, q)$ and has no prefilter. The conventional DFE has 13 feedback taps and a second-order prefilter. Because of prefilter noise enhancement, the asymptotic MSE for the conventional DFE is greater than that for the pole-zero DFE.

Fig. 13 shows that the speedup in convergence offered by the SM-RLS algorithm relative to the other adaptive IIR algorithms considered is dramatic. Each point in Fig. 13 corresponds to an average over a block of 100 succes-

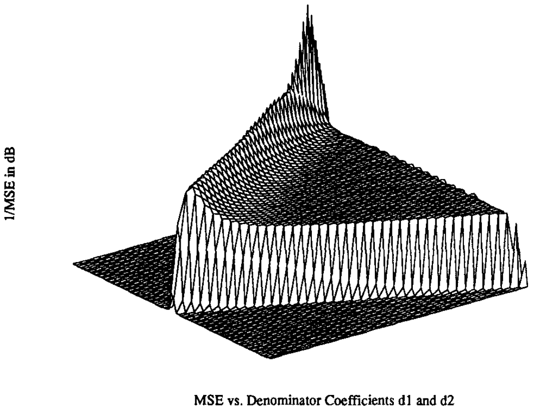


Fig. 12. MMSE surface as a function of tap coefficients d_1 and d_2. The filter $A(i, q)$ contains 15 taps, and the remaining parameters are the same as in Fig. 11.

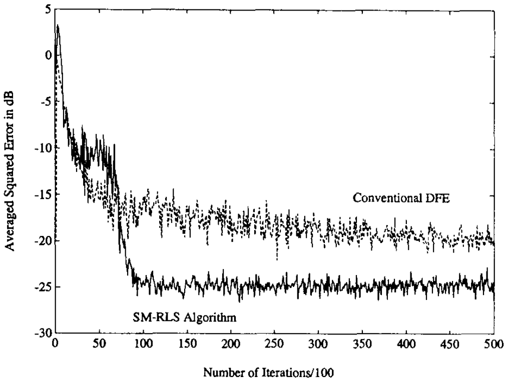


Fig. 13. Averaged squared error versus iteration for: i) the conventional DFE with a second-order prefilter using the LMS algorithm: and ii) the pole-zero DFE with no prefilter using the SM-RLS algorithm. The symbol rate is 400 kbauds, the SNR is 25 dB, and the filter $A(i, q)$ contains 13 taps.

sive squared output errors. The SM-RLS algorithm therefore takes approximately 10 000 iterations to converge, which is more than 10 times faster than the convergence speed of the SM-LMS algorithm shown in Fig. 11 and about 40 times faster than the IIR gradient algorithm. The conventional DFE using the LMS algorithm takes more than 10 000 iterations to converge to the asymptotic MSE.

Averaged squared error versus number of iterations for the SM-RLS algorithm and the conventional DFE are shown in Fig. 14 for the higher symbol rate of 800 kbauds and an SNR of 25 dB. Both DFE's have third-order prefilters; however, the conventional DFE has 100 feedback taps so that the asymptotic MSE is the same for both structures. Each point in Fig. 14 again corresponds to an average over 100 successive squared errors. Both versions of the SM-RLS algorithm described in Section III-C are represented in Fig. 14. That is, one plot assumes an RLS algorithm is used to estimate both $C(z)$ and $D(z)$ in Fig.

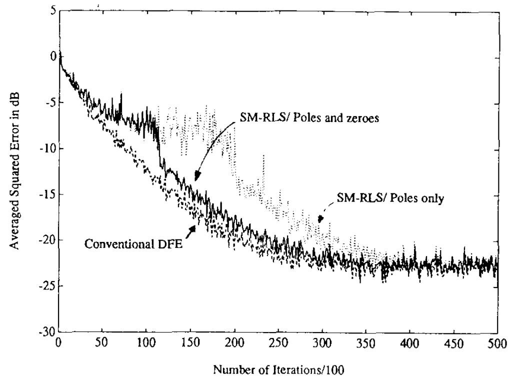


Fig. 14. Averaged squared error versus iteration for: i) the conventional DFE using the LMS algorithm, ii) the pole-zero DFE using the SM-RLS algorithm; and iii) the pole-zero DFE using the simplified SM-RLS algorithm, in which the RLS algorithm is used only to estimate $D(i, q)$. The symbol rate is 800 kbauds, the SNR is 25 dB, both DFE's contain a third-order prefilter, $A(i, q)$ contains 28 taps, and the conventional DFE contains 100 feedback taps.

8, and the other assumes an RLS algorithm is used to estimate only $D(z)$ (the poles). The exponential weight in (10f) for both cases is $w = 0.999$. The SM-RLS algorithm takes approximately 30 000 iterations to converge, which is essentially the same as the convergence time for the conventional DFE. The two plots in Fig. 14 corresponding to the two SM-RLS algorithms show approximately the same convergence time. This is fortunate since using the RLS algorithm (10d–g) to obtain an estimate of only $D(z)$ requires significantly less computation than using this RLS algorithm to estimate both $C(z)$ and $D(z)$. The averaged squared errors obtained by these algorithms after 50 000 iterations are represented by the asterisks in Fig. 10.

V. Conclusions

A pole-zero feedback filter has been proposed to reduce the complexity of decision feedback equalization for channels with severe ISI. Although this pole-zero DFE has been studied only in the context of high-speed digital communications over the subscriber loop, it is potentially advantageous whenever the tail of the channel IR can be accurately modeled with a two-pole filter. For the IR considered in Fig. 2, our results indicate that the proposed pole-zero DFE offers a 3–5 dB improvement in MSE relative to a conventional DFE having the same complexity at moderate SNR's. As shown in the preceding section, this improvement generally increases with SNR.

Important practical issues, such as error propagation and finite precision effects, have not been considered. Because of the recursive nature of IIR filters, finite precision effects may be more of a problem for the pole-zero DFE than for the conventional DFE. This requires further study. However, ignoring this potential problem, the degradation in performance due to error propagation for both DFE structures should be the same, since the IR of the pole-zero feedback filter in Fig. 3 is approximately the same as that of a conventional FIR filter that cancels the entire postcursor IR.

In addition to the practical issues just mentioned, there are other issues which also deserve further investigation. The convergence speed of the adaptive pole-zero DFE can be made faster, relative to the results in Section IV, by using a fast converging FIR adaptive algorithm (such as RLS) to adapt the filters $A(z)$ and $P(z)$ in Fig. 3. It would be of interest to see how much improvement can be obtained, although these faster algorithms are likely to be too complex for practical applications. A related issue is to characterize the convergence properties of the adaptive IIR algorithms given in Section III. At present, little is known about this although some results are available [26], [27]. A rigorous characterization of the MSE versus iteration for any of the adaptive algorithms with random inputs is likely to be quite difficult. However, an approximate treatment, using the types of convergence models described in [28], may provide some insight.

Finally, it is of interest to determine whether or not the adaptive IIR algorithms described in Section III-C may be useful in other contexts, such as echo cancellation of speech or data signals [5]-[14]. These applications may require the estimation of more than two poles, so that the presence of local minima becomes an important problem. In addition, stability checking becomes more difficult as the number of poles increases. The combination of other filter structures, such as the ARMA lattice structure [25], with a sequential SM algorithm may prove useful in this context.

Acknowledgment

The authors thank K. Steiglitz for discussions concerning this work, and S. H. Nam, J. Gibson, and B. Petersen for their comments on the original manuscript.

References

[1] C. A. Belfiore and J. H. Park, "Decision feedback equalization," *Proc. IEEE*, vol. 67, no. 8, Aug. 1979.

[2] D. G. Messerschmitt, "Design issues in the ISDN U-interface transceiver," *IEEE J. Select. Areas Commun.*, vol. SAC-4, no. 8, pp. 1281-1293, Nov. 1986.

[3] D. W. Lin, "Wide-band digital subscriber access with multidimensional block modulation and decision-feedback equalization," *IEEE J. Select. Areas Commun.*, vol. 7, no. 8, pp. 996-1005, Aug. 1989.

[4] H. Fan and M. Nayeri, "On error surfaces of sufficient order adaptive IIR filters: Proofs and counterexamples to a unimodality conjecture," *IEEE Trans. Signal Proces.*, vol. 37, no. 9, pp. 1436-1442, Sept. 1989.

[5] C. Mogavero, G. Nervo, and G. Paschetta, "Mixed recursive echo canceller," in *Proc. 1986 GLOBECOM Conf.*, Dec. 1986, pp. 2.2.1-2.2.5.

[6] M. Ari, M. Yamaguchi, F. Nakagawa, H. Shibata, A. Kanemasa, T. Makabe, and S. Koike, "Design techniques and performance of an LSI-based 2B1Q transceiver," in *Proc. 1988 IEEE GLOBECOM Conf.*, Hollywood, FL, Dec. 1988, paper 25.2.

[7] W. Chen, "An all-pole IIR echo canceller," in *Proc. 1990 Int. Symp. Circ. Syst.*, New Orleans, LA, May 1990.

[8] G. W. Davidson and D. D. Falconer, "Reduced complexity echo cancellation using orthonormal functions," submitted to *IEEE Trans. Commun.*

[9] T. Aboulnasr, A. Abousaada, and W. Steenaart, "Efficient implementation of echo cancellers for ISDN applications," in *Proc. 1989 IEEE ICASSP Conf.*, Glasgow, Scotland, paper 1626-A.

[10] A. Abousaada, "Echo tail canceller based on AIFIR filtering," M.S. thesis, Ottawa-Carleton Institute for Electrical Engineering, Jan. 1990.

[11] P. Mohanraj, V. Joshi, D. D. Falconer, and T. A. Kwasniewski, "Reduced-complexity trellis coding/decoding for high bit rate digital subscriber loop transmission," in *Proc. 1989 Int. Conf. Commun.*, Boston, MA, June 1989, paper 17.2.

[12] R. D. Gitlin and J. S. Thompson, "A new structure for adaptive digital echo cancellation," in *Proc. ICC*, Chicago, IL, June 1985, pp. 1482-1486.

[13] G. Long, D. Shwed, and D. Falconer, "Study of a pole-zero adaptive echo canceller," *IEEE Trans. Cir. Syst.*, vol. 34, no. 7, July 1987.

[14] H. Fan and K. Jenkens, "An investigation of an adaptive IIR echo canceller: Advantages and problems," *IEEE Trans. Signal Proces.*, vol. 36, no. 12, pp. 1819-1834, Dec. 1988.

[15] J. J. Shynk, "Adaptive IIR filtering," *IEEE ASSP Mag.*, vol. 6, no. 2, pp. 4-21, Apr. 1989.

[16] C. R. Johnson, "Adaptive IIR filtering: Current results and open issues," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 2, pp. 237-250, Mar. 1984.

[17] B. Widrow *et al.*, "Adaptive noise canceling: Principles and applications," *Proc. IEEE*, vol. 63, no. 12, pp. 1692-1716, Dec. 1975.

[18] H. Fan and W. K. Jenkins, "A new adaptive IIR filter," *IEEE Trans. Circ. Syst.*, vol. CAS-33, no. 10, pp. 939-947, Oct. 1986.

[19] A. M. Gottlieb, P. M. Crespo, J. L. Dixon, and T. R. Hsing, "The DSP implementation of a new timing recovery technique for high-speed digital data transmission," in *Proc. 1990 ICASSP*, Albuquerque, NM, Apr. 1990, paper 48.D11.11.

[20] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*. Cambridge, MA: M.I.T. Press, 1983.

[21] B. R. Petersen and D. D. Falconer, "Minimum mean-square equalization in cyclostationary and stationary interference—Analysis and subscriber line calculations," *IEEE J. Select. Areas Commun.*, this issue, pp. 931-940.

[22] K. Steiglitz and L. McBride, "A technique for the identification of linear systems," *IEEE Trans. Automatic Control*, vol. AC-10, pp. 461-464, Oct. 1965.

[23] J. Cioffi, "The block-processing FTF adaptive algorithm," *IEEE Trans. Signal Proces.*, vol SP-34, no. 1, pp. 77-90, Feb. 1986.

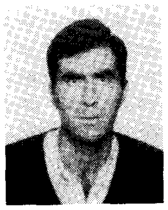
[24] P. Strobach, *Linear Prediction Theory: A Mathematical Basis for Adaptive Systems.* New York: Springer-Verlag, 1990.

[25] B. Friedlander, "Lattice filters for adaptive processing," in *Proc. IEEE*, vol. 70, no. 8, Aug. 1982.

[26] P. Stoica and T. Soderstrom, "The Steiglitz-McBride identification algorithm revisited—Convergence analysis and accuracy aspects," *IEEE Trans. Automatic Control.*, vol. AC-26, no. 3, pp. 712-717, June 1981.

[27] H. Fan, "Application of Benveniste's convergence results in the study of adaptive IIR filtering algorithms," *IEEE Trans. Inform. Theory*, vol. 34, no. 4, pp. 692-709, July 1988.

[28] M. L. Honig, "Convergence models for adaptive gradient and least squares algorithms," *IEEE Trans. Signal Proces.*, vol. 31, no. 2, pp. 415-425, Apr. 1983.



Pedro M. Crespo (S'80-M'84) was born in Barcelona, Spain.in 1955. He received the engineering degree from B. T. Superior de Ingenieros de Telecomunicación, Barcelona, Spain, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1981 and 1984, respectively.

From September 1984 to April 1990, he was a Member of the Technical Staff in the Signal Processing Systems Research Group at Bell Communications Research, Morristown, NJ, where he worked in the areas of data communications and signal processing. At present, he is a Manager at Telefonics Investigación y Desarrollo, Madrid, Spain. His current research interests include diversity systems, digital synchronization, and digital communications over fading channels.

Michael L. Honig (S'80-M'81) was born in Phoenix, AZ in 1955. He received the B.S. degree in electrical engineering from Stanford University, CA, in 1977, and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1978 and 1981, respectively.

From July 1981 to October 1983, he was a Member of the Technical Staff at AT&T Information Systems, formerly part of Bell Laboratories, Holmdel, NJ, where he worked on the design and performance analysis of local area networks and on voiceband data transmission. He subsequently transferred to the Systems Principles Research Division at Bell Communications Research, Morristown, NJ, where he is currently working in the areas of data communications and signal processing.

Dr. Honig is a free-lance trombonist, and is a member of Tau Beta Pi and Phi Beta Kappa.