Adaptive IIR Filtering Using Parallel-Form Realizations

JOHN J. SHYNK, MEMBER, IEEE

Abstract—**The parallel form in adaptive IIR filtering is an efficient realization that provides robust stability monitoring with less complexity than that of the direct form. Several parallel-form adaptive IIR filters are presented in this paper, including a frequency-domain implementation based on the discrete Fourier transform, and a recursive frequency-sampling structure. The performance of the frequency-domain adaptive IIR filter is investigated in a system identification application, which includes an analysis of its modeling capabilities and a discussion of the MSE performance surface. Computer simulation results are presented to illustrate the robust convergence properties of the adaptive algorithm, and to demonstrate the stability of the filter.**

I. Introduction

OVER the last decade, adaptive infinite-impulse-response (IIR) filtering has been an active area of research [1], [2]. Most adaptive IIR filters discussed in the literature have been direct-form realizations [3]–[9]. Although the direct form is a convenient and simple structure, it is often difficult to ensure stability of the adaptive filter when using, for example, the Gauss–Newton (GN) algorithm [10]. Instability can occur if a particular application requires that the poles be located close to the unit circle. If the GN algorithm adapts too rapidly, one or more poles could accidently move outside the unit circle because of the noisy gradient estimate, leading to a potentially unstable filter. Some method of stability checking, therefore, is necessary to prevent the poles from updating outside the unit circle.[1] Although several methods have been suggested, they are either computationally expensive or nonrobust [1]. To resolve this problem, several alternative implementations including the cascade, parallel, and lattice forms have been considered [5], [11]–[13]. These structures offer simple stability monitoring without the large complexity required by the direct form.

In this paper, we describe several parallel-form adaptive IIR filters that overcome the stability complexity problem and appear to be quite robust. They are comprised of first-order sections that facilitate monitoring of the filter poles during adaptation to ensure stability. Because of the parallel structure, they are also less sensitive to coefficient quantization than direct-form realizations and they are well suited for modular hardware implementation (e.g., VLSI). Section II describes the basic parallel-form adaptive IIR filter [5], [13] and demonstrates that the GN algorithm can have convergence problems if the coefficients of each section are identically initialized. This provides the motivation for alternative parallel configurations that are less sensitive to this initialization and have improved convergence properties.

Section III discusses a parallel-form realization implemented in the frequency domain that uses a discrete Fourier transform (DFT) to preprocess the input signal. Three different forms of the adaptive algorithm are presented, and a frequency-sampling realization that offers a reduction in complexity is described. The analyses in Section IV illustrate several performance characteristics of the frequency-domain adaptive IIR filter (FDAF) in a system identification application. It is demonstrated that the adaptive filter can model any rational system with distinct poles. (It will become clear that parallel-form realizations, in general, cannot exactly model a system with multiple repeated poles.) Some properties of the mean-square-error surface are examined, and a discussion of the asymptotic variance of the coefficient estimates, determined by the Cramer–Rao lower bound, is presented. Several computer simulations illustrate the robust convergence properties of the FDAF. Conclusions are then outlined in Section V.

II. Parallel-Form Realization

A. Filter Description

Consider the parallel-form adaptive IIR filter shown in Fig. 1 that is comprised of N first-order sections, each having a single feedforward coefficient $b_k(n)$ and a single feedback coefficient $a_k(n)$. The output $y_k(n)$ of each section is given by

$$y_k(n) = \left(\frac{b_k^*(n)}{1 - a_k^*(n)q^{-1}}\right)x(n) \qquad (1)$$

where $x(n)$ is the *common* input, the superscript * denotes complex conjugate, and q^{-1} is the delay operator [i.e., $q^{-1}x(n) = x(n-1)$]. (If $x(n)$ is strictly real, this configuration should be comprised of second-order sections as in [5]; this will permit complex poles.) The over-

Manuscript received June 1, 1987; revised July 25, 1988.

The author is with the Department of Electrical & Computer Engineering, University of California, Santa Barbara, CA 93106.

IEEE Log Number 8826097.

[1] It should be noted that not all adaptive algorithms for IIR filtering require stability monitoring, such as those based on pseudolinear regression methods [10]. However, these algorithms generally require that a strictly positive real (SPR) condition be satisfied, which is not always possible in practice. This paper is concerned only with the Gauss–Newton algorithm for which the SPR condition is not necessary.

0096-3518/89/0400-0519$01.00 © 1989 IEEE

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 13,2010 at 21:39:13 UTC from IEEE Xplore. Restrictions apply.

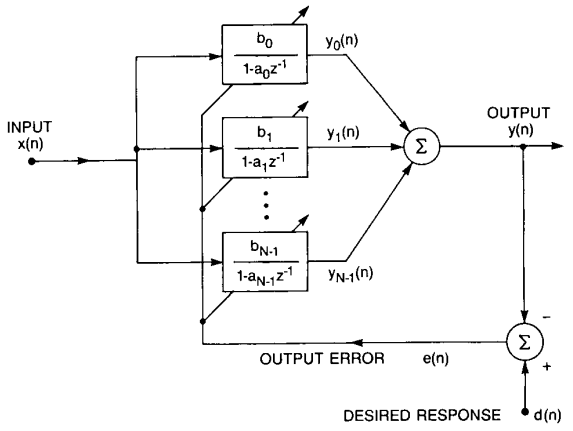


Fig. 1. Parallel-form adaptive IIR filter.

all output $y(n)$ of the filter is

$$y(n) = \sum_{k=0}^{N-1} y_k(n) = \theta^H(n)\,\phi(n) \tag{2}$$

where the superscript H denotes complex conjugate transpose. The coefficient vector $\theta(n)$ and the signal vector $\phi(n)$ each have length $2N$ and are defined as follows:

$$\theta(n) = \left(a_0(n) \cdots a_{N-1}(n)\, b_0(n) \cdots b_{N-1}(n)\right)^T \tag{3a}$$

$$\phi(n) = \left(y_0(n-1) \cdots y_{N-1}(n-1)\, x(n) \cdots x(n)\right)^T. \tag{3b}$$

Finally, the output error $e(n)$ is formed as the difference between the desired response $d(n)$ (which depends on the application of the adaptive filter [14]) and $y(n)$.

The adaptive algorithm that adjusts the coefficients $\{a_k(n), b_k(n)\}$ to minimize the mean-square error (MSE) is a specific form of the (complex) recursive Gauss–Newton (GN) method [15]. It is given by

$$R(n+1) = \lambda R(n) + \alpha\psi(n)\,\psi^H(n) \tag{4a}$$

$$\theta(n+1) = \theta(n) + \alpha R^{-1}(n+1)\,\psi(n)e^*(n) \tag{4b}$$

where $R(n)$ is the (estimated) Hessian matrix, α is the convergence step size, $\lambda = 1 - \alpha$, and $\psi(n)$ is the so-called information vector defined as

$$\psi(n) = \left(y_0^f(n-1) \cdots y_{N-1}^f(n-1)\, x_0^f(n) \cdots x_{N-1}^f(n)\right)^T. \tag{5}$$

The components $x_k^f(n)$ and $y_k^f(n-1)$ are computed by filtering the input $x(n)$ and the intermediate output $y_k(n-1)$, respectively, with the denominator polynomial of the corresponding first-order section as follows:

$$x_k^f(n) = \left(\frac{1}{1 - a_k^*(n)q^{-1}}\right)x(n) \tag{6a}$$

$$y_k^f(n-1) = \left(\frac{1}{1 - a_k^*(n)q^{-1}}\right)y_k(n-1). \tag{6b}$$

Notice that each $x_k^f(n)$ depends on the *same* (and only) input $x(n)$, whereas each $y_k^f(n-1)$ depends on the intermediate output $y_k(n-1)$ of the subfilter in (1).

Instead of computing the inverse of the Hessian as in (4b), $P(n) \equiv R^{-1}(n)$ is generally updated directly using the matrix-inversion lemma [10]. A simpler algorithm can be derived if the update for the Hessian in (4a) is not performed and $R^{-1}(n+1)$ in (4b) is replaced by the identity matrix. The resulting algorithm corresponds to a steepest-descent (SD) method [14] which has less complexity than the GN algorithm. It is well known, however, that an SD algorithm generally converges more slowly than a GN algorithm, so that there is a tradeoff between the computational complexity and the rate of convergence.

The primary advantage of the parallel configuration is that *stability monitoring is trival*. In the event that a pole attempts to move outside the unit circle, the update for that first-order section is simply ignored [i.e., $a_k(n+1) = a_k(n)$]. All stable poles are updated, however, so that the adaptive filter changes state and the algorithm is less likely to lock up [1]; furthermore, all unstable poles can be easily projected back inside the unit circle to some appropriate location. As a consequence, stability monitoring in this configuration is robust and it does not increase the complexity of the algorithm. (If $x(n)$ is real and second-order sections are used in Fig. 1, stability checking in this case is again trivial.) It should be noted that the parallel form does not solve the problem of where to project the unstable poles [10], it only facilitates the process of identifying them.

B. Nonconvergence of the GN Algorithm

Although the parallel form has several advantages, there is a potential drawback with the structure in Fig. 1. If the subfilters are identically initialized, then it is possible that the GN algorithm will not converge. To guarantee convergence, it is necessary that the Hessian matrix in (4) be nonsingular (positive definite) for all n. In the direct form, $R(n)$ will generally be positive definite if the input signal $x(n)$ has persistent excitation [10]. Although this is a necessary condition for $R(n)$ to be positive definite in the parallel configuration, it is not always sufficient. Consider the update expression for the Hessian matrix which can be written in the following nonrecursive form:

$$\begin{aligned} R(n+1) &= \lambda^{n+1}R(0) + \alpha\sum_{k=0}^{n}\lambda^k\psi(n-k)\,\psi^H(n-k) \\ &= \lambda^{n+1}\delta I + \alpha\sum_{k=0}^{n}\lambda^k\psi(n-k)\,\psi^H(n-k), \end{aligned} \tag{7}$$

where δ is the initial estimate of the signal power and I is the identity matrix. Because $\lambda < 1$, (7) can be approxi-

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 13,2010 at 21:39:13 UTC from IEEE Xplore. Restrictions apply.

mated for large n by

$$R(n+1) \approx \alpha \sum_{k=0}^{n} \lambda^k \psi(n-k)\, \psi^H(n-k). \quad (8)$$

If the subfilters in (1) are identical, each $x_k^f(n)$ in (5) will also be identical as will each $y_k^f(n-1)$; the matrix $\psi(n-k)\,\psi^H(n-k)$, therefore, will have a block structure where each block is comprised of identical elements. As a result, $R(n)$ in (7) will eventually become singular even though the initial matrix $R(0) = \delta I$ is nonsingular.

Recent analysis of the MSE performance surface of the parallel form [16], [17] demonstrates that the manifold of equivalent subfilters lies along a gradient line until it reaches a saddle point. This indicates that any numerical differences between the subfilters (including that caused by noise) should lead to satisfactory convergence of a gradient-based algorithm, a result that has been demonstrated in computer simulations [17], [18]. By an appropriate initialization of the algorithm, it may be possible to avoid the singularity problem described above, although there appears to be no guarantee. The simulations in Section IV and [18] illustrate that convergence of the GN algorithm is significantly slower whenever the poles lie close to this manifold, suggesting that the Hessian matrix is ill conditioned. The above result demonstrates that such a manifold can lead to a singular Hessian matrix and cause problems with convergence.

C. Preprocessing of the Input Signal

Consider the configuration in Fig. 2 where the input signal $x(n)$ is *preprocessed* to generate N different signals in parallel. The signal vector in (3b) becomes

$$\phi(n) = \big(y_0(n-1) \cdots y_{N-1}(n-1)\, x_0(n) \cdots x_{N-1}(n)\big)^T \quad (9)$$

and $x(n)$ in (6a) is replaced by $x_k(n)$. By filtering $x(n)$, the subfilter inputs can be modified so that the GN algorithm is less sensitive to the condition of identical subfilters, as demonstrated in computer simulations (see Section IV). Unlike the information vector associated with Fig. 1, the transfer functions from $x(n)$ to $\{x_k^f(n)\}$, and from $x(n)$ to $\{y_k^f(n)\}$, can now be different for any set of subfilter coefficients within the stable region. It should be noted that preprocessing of the input does not guarantee convergence (as this depends on the spectral nature of $x(n)$ [10]), but it appears that certain types of preprocessing can improve the convergence properties of the GN algorithm.

There are several issues concerning the effect of the preprocessing on the properties of the adaptive filter. For example, consider an application in system identification where the adaptive filter models the transfer function of some unknown system. It is important for the preprocessing to be "transparent" in the sense that the adaptive filter be capable of exactly modeling an arbitrary rational system; i.e., if the order of the adaptive filter is chosen large enough, the output error can be driven to zero (in the noise-free case).

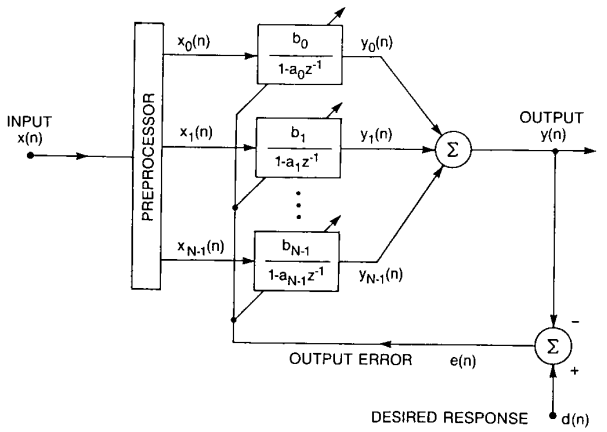


Fig. 2. Parallel-form adaptive IIR filter with preprocessing of the input signal.

It would also be advantageous for $\{x_k(n)\}$ to be approximately uncorrelated so that the Hessian matrix is nearly diagonal. The GN adaptive algorithm could be simplified considerably using a diagonal matrix as an estimate for the Hessian, thereby uncoupling the update for each subfilter without significantly degrading the convergence rate.

III. Frequency-Domain Realization

A. Filter Description

One method of preprocessing that satisfies the above-mentioned properties is the DFT; the resulting frequency-domain adaptive IIR filter (FDAF) [19], [20] is shown in Fig. 3. A tapped delay line (TDL) stores a vector of N values of the input $\{x(n) \cdots x(n-N+1)\}$ and at each instant of time, a DFT of the TDL is computed to separate the input signal $x(n)$ into N parallel signals $\{x_k(n)\}$. The form of the DFT is

$$x_k(n) = \frac{1}{N} \sum_{l=0}^{N-1} x(n-l) W_N^{kl} \quad (10)$$

where $W_N = e^{-j2\pi/N}$. Each $x_k(n)$ is independently filtered by $H_k(n, z)$ to generate the intermediate output signals

$$y_k(n) = \left(\frac{b_k^*(n) + c_k^*(n) q^{-1}}{1 - a_k^*(n) q^{-1}}\right) x_k(n) \quad (11)$$

where $\{a_k(n), b_k(n), c_k(n)\}$ are the adjustable coefficients. These are then summed to generate the overall filter output $y(n)$. It will be evident in Section III and from (18), which corresponds to the transfer function of (10), that this form of preprocessing ensures that each component of the information vector is derived from a different transfer function for any set of coefficients in the stable region.

Frequency-domain realizations for adaptive finite-impulse-response (FIR) filtering have been widely studied

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 13,2010 at 21:39:13 UTC from IEEE Xplore. Restrictions apply.

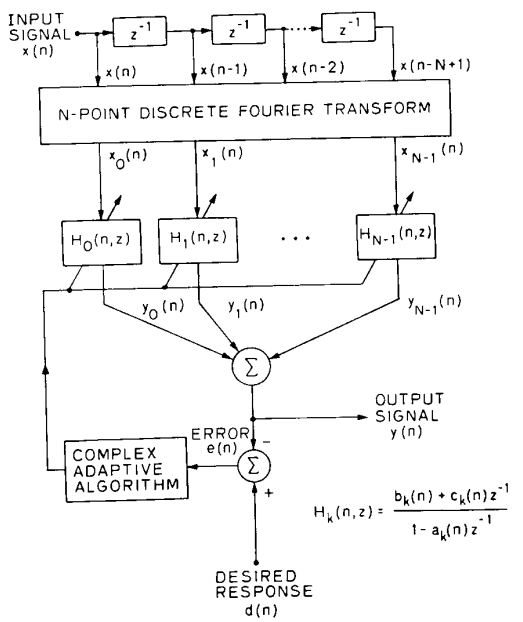


Fig. 3. Frequency-domain adaptive IIR filter.

[21]–[26]. The use of adaptive *poles* in the frequency domain was apparently first investigated by Bershad and Feintuch [27]. To simplify the convergence analysis of an adaptive IIR line enhancer, they examined an equivalent frequency-domain representation. This representation was not considered to be an implementation of the filter, however, and it was developed for a specific application. The FDAF in Fig. 3 has general application and is an extension of the frequency-domain adaptive FIR filter introduced by Narayan [23]. Notice that three coefficients are used in each first-order subfilter instead of two as in Figs. 1 and 2; this additional coefficient enables the FDAF to model an Nth-order system with distinct poles (as shown in Section IV).

If $x(n)$ is real, then x_0 and $x_{N/2}$ (for N even) are also real signals, and the centers of the corresponding DFT frequency bins[2] are at $\omega_0 = 0$ and $\omega_{N/2} = \pi$, respectively. Because the adaptive algorithm used to update a subfilter will not generate complex coefficients if the subfilter input is real [20], the FDAF will require a DFT with an order $> N$ to exactly model an Nth-order system that has no real poles. It is possible to resolve this problem with a *generalized* DFT (GDFT) rather than the standard DFT. In this case, the inputs to the subfilters are

$$x_k(n) = \frac{1}{N} \sum_{l=0}^{N-1} x(n-l) W_N^{l(k+k_o)} \tag{12}$$

where $-1/2 \le k_o \le 1/2$. This expression corresponds to a DFT where the frequency index has been shifted by the noninteger-valued k_o. This form is more desirable because the bin centers can be shifted arbitrarily around the unit circle, and k_o can be chosen such that no bin centers lie on the real axis; instead, they will be at $\omega_k = W_N^{(k+k_o)}$. Typically, k_o should equal one-half.

[2]Frequency bin k refers to the band of frequencies centered at $\omega_k = W_N^k$ with a bandwidth of approximately $2\pi/N$.

B. Frequency-Domain Adaptive Algorithm

The adaptive algorithm is a special case of the GN algorithm in (4); it employs the following $3N$-vectors:

$$\theta(n) = \left(\theta_0^T(n) \cdots \theta_{N-1}^T(n)\right)^T \tag{13a}$$

$$\phi(n) = \left(\phi_0^T(n) \cdots \phi_{N-1}^T(n)\right)^T \tag{13b}$$

$$\psi(n) = \left(\psi_0^T(n) \cdots \psi_{N-1}^T(n)\right)^T \tag{13c}$$

where

$$\theta_k(n) = \left(a_k(n), b_k(n), c_k(n)\right)^T \tag{14a}$$

$$\phi_k(n) = \left(y_k(n-1), x_k(n), x_k(n-1)\right)^T \tag{14b}$$

$$\psi_k(n) = \left(y_k^f(n-1), x_k^f(n), x_k^f(n-1)\right)^T. \tag{14c}$$

The filtered variables $\{x_k^f(n)\}$ and $\{y_k^f(n-1)\}$ are derived from (6) [where $x(n)$ in (6a) is replaced by the DFT output $x_k(n)$]. The complete algorithm is summarized in Table I where the inverse $P(n)$ of the Hessian is updated directly using the matrix-inversion lemma. A generalized fast Fourier transform (GFFT) based on (12) produces the DFT signals $\{x_k(n)\}$.

It is possible to reduce the complexity of the FDAF algorithm from order N^2 to order N because the DFT signals $\{x_k(n)\}$ are approximately orthogonal [28]. If we assume they are exactly orthogonal, the Hessian matrix will have the following *block-diagonal* form:

$$R(n) = \begin{bmatrix} R_0(n) & & 0 \\ & \ddots & \\ 0 & & R_{N-1}(n) \end{bmatrix} \tag{15}$$

where each $R_k(n)$ has rank 3. As a result, the update for the Hessian matrix in (4a) becomes N separate updates as

$$R_k(n+1) = \lambda R_k(n) + \alpha \psi_k(n) \psi_k^H(n), \tag{16}$$

and the corresponding update for $\theta_k(n)$ is

$$\theta_k(n+1) = \theta_k(n) + \alpha R_k^{-1}(n+1) \psi_k(n) e^*(n), \tag{17}$$

which uses the common error $e(n)$ for each value of k. Table II summarizes this simplified adaptive algorithm where the Hessian matrices are updated directly because the matrix-inversion lemma does not reduce the computational complexity for such matrices of small rank. The block-diagonal Hessian will generally result in a slower convergence rate because the $\{x_k(n)\}$ are only approximately uncorrelated. Simulation studies indicate, however, that the performance is not severely degraded (see Section IV).

Further simplification of the adaptive algorithm can be achieved by forcing each $R_k(n)$ in (15) to be diagonal. In this case, all coefficients of the adaptive filter are uncoupled and updated independently. The overall complexity

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 13,2010 at 21:39:13 UTC from IEEE Xplore. Restrictions apply.

TABLE I
FREQUENCY-DOMAIN ADAPTIVE IIR ALGORITHM

INITIALIZATION:

$$b_k(0) = c_k(0) = 0 \ , \ |a_k(0)| < 1$$

$$x_k(-1) = y_k(-1) = x_k^f(-1) = y_k^f(-1) = 0$$

$$P(0) = \delta^{-1} I$$

VECTOR DEFINITIONS:

$$\theta_k(n) = [\, a_k(n) \, , \, b_k(n) \, , \, c_k(n) \,]^T$$

$$\theta(n) = [\, \theta_0^T(n) \, , \, \ldots \, , \, \theta_{N-1}^T(n) \,]^T$$

$$\phi_k(n) = [\, y_k(n-1) \, , \, x_k(n) \, , \, x_k(n-1) \,]^T$$

$$\phi(n) = [\, \phi_0^T(n) \, , \, \ldots \, , \, \phi_{N-1}^T(n) \,]^T$$

$$\psi_k(n) = [\, y_k^f(n-1) \, , \, x_k^f(n) \, , \, x_k^f(n-1) \,]^T$$

$$\psi(n) = [\, \psi_0^T(n) \, , \, \ldots \, , \, \psi_{N-1}^T(n) \,]^T$$

FOR EACH NEW INPUT $x(n)$, $d(n)$; $n \geq 0$:

$$x_k(n) = \text{GFFT}\left[x(n) \, , \, \ldots \, , \, x(n-N+1) \right]$$

FOR $k = 0, 1, \ldots, N-1$:

$$x_k^f(n) = x_k(n) + a_k^*(n)\, x_k^f(n-1)$$

$$y_k(n) = \theta_k^H(n)\, \phi_k(n)$$

$$y_k^f(n) = y_k(n) + a_k^*(n)\, y_k^f(n-1)$$

$$e(n) = d(n) - \sum_{k=0}^{N-1} y_k(n)$$

$$P(n+1) = \frac{1}{\lambda} \left[P(n) - \frac{P(n)\psi(n)\psi^H(n)P(n)}{\lambda / \alpha + \psi^H(n)P(n)\psi(n)} \right]$$

$$\theta(n+1) = \theta(n) + \alpha\, P(n+1)\, \psi(n)\, e^*(n)$$

TABLE II
SIMPLIFIED FREQUENCY-DOMAIN ADAPTIVE IIR ALGORITHM

INITIALIZATION:

$$b_k(0) = c_k(0) = 0 \ , \ |a_k(0)| < 1$$

$$x_k(-1) = y_k(-1) = x_k^f(-1) = y_k^f(-1) = 0$$

$$R_k(0) = \delta\, I$$

VECTOR DEFINITIONS:

$$\theta_k(n) = [\, a_k(n) \, , \, b_k(n) \, , \, c_k(n) \,]^T$$

$$\phi_k(n) = [\, y_k(n-1) \, , \, x_k(n) \, , \, x_k(n-1) \,]^T$$

$$\psi_k(n) = [\, y_k^f(n-1) \, , \, x_k^f(n) \, , \, x_k^f(n-1) \,]^T$$

FOR EACH NEW INPUT $x(n)$, $d(n)$; $n \geq 0$:

$$x_k(n) = \text{GFFT}\left[x(n) \, , \, \ldots \, , \, x(n-N+1) \right]$$

FOR $k = 0, 1, \ldots, N-1$:

$$x_k^f(n) = x_k(n) + a_k^*(n)\, x_k^f(n-1)$$

$$y_k(n) = \theta_k^H(n)\, \phi_k(n)$$

$$y_k^f(n) = y_k(n) + a_k^*(n)\, y_k^f(n-1)$$

$$e(n) = d(n) - \sum_{k=0}^{N-1} y_k(n)$$

FOR $k = 0, 1, \ldots, N-1$:

$$R_k(n+1) = \lambda\, R_k(n) + \alpha\, \psi_k(n)\, \psi_k^H(n)$$

$$\theta_k(n+1) = \theta_k(n) + \alpha\, R_k^{-1}(n+1)\, \psi_k(n)\, e^*(n)$$

is still of order N, but there are no matrix operations. Unfortunately, simulations demonstrate that this additional reduction in complexity occurs at the expense of a severely degraded convergence rate (see Section IV).

The adaptive algorithm requires proper initialization of both $\{R_k(n)\}$ and $\{\theta_k(n)\}$. For satisfactory convergence, each $R_k(0)$ must be positive definite and, for stability, each $a_k(0)$ must have a magnitude <1. Generally, it is sufficient to initialize as $R_k(0) = \delta I$ and $\theta_k(0) = \mathbf{0}$ where the real scalar δ is an estimate of the power of $x_k(n)$ and $\mathbf{0}$ is a column vector of 0's. Alternatively, it may be advantageous in some applications to initialize the pole coefficients to different nonzero values. It has been shown in [29] that the manifold of equivalent subfilters can lead to slower convergence of the FDAF, and that initialization of the subfilters away from this manifold may improve the convergence properties.

Because the poles appear explicitly in the subfilters, it is not only possible to easily initialize them to any value, but it is also straightforward to constrain their adaptation to any subregion of the z-plane. In system identification, for example, if it is known *a priori* that the system poles lie in the right half of the z-plane, the FDAF poles could be constrained to update only in that region. This could increase the convergence rate since the poles would not temporarily "wander" in the left-half plane. Although constrained adaptation would not be useful in all applications, it is still another advantage of a parallel implementation but not for the direct form (when $N > 2$).

C. Frequency-Sampling Realization

It is possible to obtain $\{x_k(n)\}$ using a frequency-sampling (FS) implementation [24], [30] of the DFT instead of the FFT that would reduce the computational complexity of computing the filter output from order $N \log N$ to only order N. This implementation corresponds to the

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 13,2010 at 21:39:13 UTC from IEEE Xplore. Restrictions apply.

transfer function from $x(n)$ to $x_k(n)$, derived from (10) [or (12)], and is expressed as

$$\frac{X_k(z)}{X(z)} = \frac{1}{N}\frac{1 - z^{-N}}{1 - W_N^k z^{-1}} \tag{18}$$

where $X(z)$ and $X_k(z)$ are the z-transforms of $x(n)$ and $x_k(n)$, respectively. The numerator consists of N zeros equally spaced around the unit circle according to $z = W_N^l$, $l = 0, \cdots, N-1$. A pole at $z = W_N^k$ cancels exactly one of these zeros so that (18) actually represents an all-zero filter, as is evident from the nonrecursive nature of (10).

Although this recursive structure is an equivalent representation of the DFT, it is not a practical implementation because the poles at $z = W_N^k$ are only marginally stable. To realize the FS structure, the poles (and zeros) must be placed just inside the unit circle by substituting $z^{-1} = \beta z^{-1}$ in the right-hand side of (18), where β is a real scalar ($0 < \beta < 1$) chosen arbitrarily close to 1. In this case, $x_k(n)$ is computed according to

$$x_k(n) = \frac{1}{N}\left(\frac{1 - \beta^N q^{-N}}{1 - \beta W_N^k q^{-1}}\right)x(n) \tag{19}$$

where the filter has N zeros equally spaced around a circle with a radius of β; a pole at $z = \beta W_N^k$ exactly cancels one of these zeros. Fig. 4 is a diagram of this implementation of the FDAF where, for convenience, $\beta = 1$. It can be seen in (19) that the comb filter $(1 - \beta^N z^{-N})/N$ is common to each $x_k(n)$, and its output can be calculated as follows:

$$w(n) = \frac{1}{N}\left[x(n) - \beta^N x(n - N)\right]. \tag{20a}$$

Each $x_k(n)$ is then obtained from $w(n)$ as

$$x_k(n) = w(n) + \beta W_N^k x_k(n - 1). \tag{20b}$$

These expressions are an efficient method of computing the DFT that is recursive but stable. The corresponding adaptive algorithm is derived from Table I or II by substituting (20) for the FFT. (An FS implementation of the GDFT in (12) is obtained in a similar way.)

When implementating (19), one cannot expect perfect pole-zero cancellation (because of coefficient quantization) so that the impulse response is actually infinite. This could be undesirable because a significantly long impulse response may adversely affect the convergence rate of the adaptive algorithm. If β in the numerator of (19) is replaced by $\beta + \epsilon$ where ϵ represents the pole-zero mismatch, then the impulse response of the kth frequency-sampling filter becomes

$$f_k(n) = \frac{(\beta W_N^k)^n}{N}\left[u(n) - \left(1 + \frac{\epsilon}{\beta}\right)^N u(n - N)\right] \tag{21}$$

where $u(n)$ is the unit step function. If we assume that $\epsilon \ll 1$, then $(1 + (\epsilon/\beta))^N \approx 1 + N(\epsilon/\beta)$ and the magnitude of (21) can be approximated by

$$\left|f_k(n)\right| \approx \frac{\beta^n}{N}\left[u(n) - u(n - N)\right] + \epsilon\beta^{n-1}u(n - N). \tag{22}$$

The first term has finite duration and corresponds to the impulse response of (19) if exact pole-zero cancellation were possible. The second term has infinite duration and is introduced by the pole-zero mismatch; however, it is proportional to ϵ. Because ϵ is generally $\ll 1$, little effect from imperfect pole-zero cancellation is expected, except possibly for large N and for β very close to 1. It may be desirable, therefore, to choose β somewhat less than 1 (e.g., $\beta = 0.9$). The time constant τ for the decay of the infinite part of the impulse response is approximately $\tau = 1/(1 - \beta)$. Choosing $\beta = 0.9$ results in $\tau = 10$ iterations compared to $\tau = 1000$ iterations for $\beta = 0.999$.

D. Additional Realizations

From the FS implementation, the TDL/DFT can be viewed as a means of converting a wide-band adaptive filter into several narrow-band adaptive filters. The DFT operates as a *bank of bandpass filters* with approximately nonoverlapping spectra. As such, other types of bandpass filters (or other orthogonal transformations [28]) might be used so that the FDAF can be generalized to a class of parallel-form adaptive IIR filters [31] as shown in Fig. 5.

For example, bandpass filters with smaller overlap could result in less correlated signals $\{x_k(n)\}$ and cause the adaptive algorithm in Table II to converge more rapidly. These filters would undoubtedly require greater complexity, resulting in a tradeoff between the convergence rate of the algorithm and the complexity of the bandpass filters. A modeling issue may also become a consideration; the bandpass filters should enable the adaptive filter to model an arbitrary rational system. This property is satisfied only by certain types of bandpass filters.

A Lagrange filter [30] is a generalization of the FS filter in (18) that satisfies the system-modeling property. Its transfer function is

$$\frac{X_k(z)}{X(z)} = \frac{1}{N}\frac{\prod_{l=0}^{N-1}(1 - z_l z^{-1})}{(1 - p_k z^{-1})} \tag{23}$$

where $\{z_l\}$ are N *arbitrarily placed* zeros and p_k is a pole that exactly cancels one of these zeros. This structure would be useful, for example, if it is known *a priori* that the spectrum of $d(n)$ is concentrated in some region of ω. Here, it would be more efficient to use bandpass filters based on the Lagrange filter instead of the FS filter.

In this case, the expressions in (20) for the FS implementation are replaced by the following similar equations:

$$w(n) = \frac{1}{N}\sum_{l=0}^{N-1} s_l x(n - l) \tag{24a}$$

$$x_k(n) = w(n) + p_k x_k(n - 1) \tag{24b}$$

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 13,2010 at 21:39:13 UTC from IEEE Xplore. Restrictions apply.

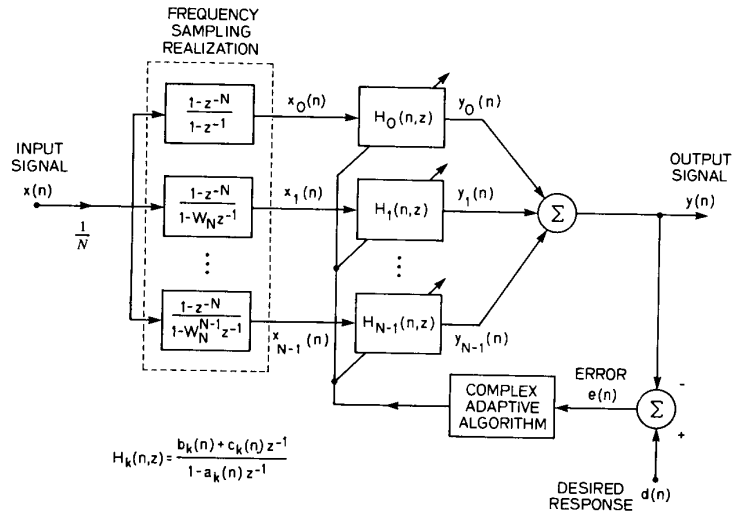


Fig. 4. Frequency-sampling adaptive IIR filter.

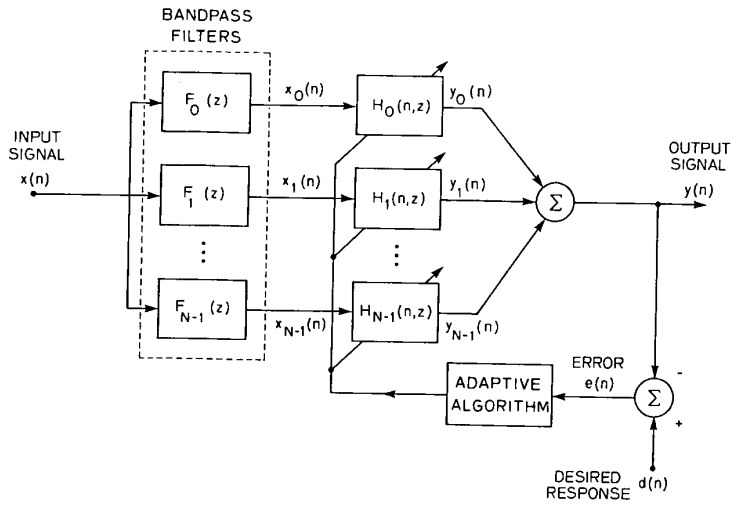


Fig. 5. Parallel-form adaptive IIR filter using bandpass filters.

where $\{s_l\}$ are the coefficients of the polynomial corresponding to the numerator in (23). Assuming perfect pole-zero cancellation, (24) represents an FIR filter. (The previous discussion concerning imperfect pole-zero cancellation also applies here.)

IV. Performance in System Identification

System identification and parameter estimation [10], [32] are fundamental to many applications in automatic control, communications, and signal processing. This section investigates several performance characteristics of the FDAF in a system identification configuration. An understanding of coefficient convergence here can be useful in predicting the FDAF performance in applications where there is no underlying parametric system or where the signals are nonstationary.

In adaptive filtering applications of system identification, the desired response $d(n)$ and adaptive filter input $x(n)$ are often assumed to be generated as

$$d(n) = G(q^{-1})\,x(n) + v(n) \tag{25}$$

where $x(n)$ and $v(n)$ are zero-mean mutually uncorrelated signals, and $G(z)$ is the system to be identified. The additive signal $v(n)$ is usually attributed to measurement noise.

A. System Modeling

Consider an arbitrary system $G(z)$ with denominator order P and numerator order M that has the following transfer function:

$$G(z) = K\frac{\prod_{k=0}^{M-1}(1 - d_k z^{-1})}{\prod_{k=0}^{P-1}(1 - a_k z^{-1})} \tag{26}$$

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 13,2010 at 21:39:13 UTC from IEEE Xplore. Restrictions apply.

where K is a constant, the poles $\{a_k\}$ and zeros $\{d_k\}$ are distinct, and

$$a_k, d_l \neq W_N^m \qquad (27)$$
$$k = 0, \cdots, P-1, \quad l = 0, \cdots, M-1,$$
$$m = 0, \cdots, N-1.$$

Based on the frequency-sampling (FS) structure in (19), (26) can be rewritten as

$$G(z) = \frac{(1 - z^{-N})}{N}\left[\frac{NK\prod_{k=0}^{M-1}(1 - d_k z^{-1})}{(1 - z^{-N})\prod_{k=0}^{P-1}(1 - a_k z^{-1})}\right] \qquad (28)$$

where, without loss of generality, it is assumed that $\beta = 1$. For $M < 2N$, this can be expanded as a partial fraction expansion (PFE)

$$G(z) = \frac{(1 - z^{-N})}{N}\left(\sum_{k=0}^{P-1}\frac{A_k}{1 - a_k z^{-1}} + \sum_{k=0}^{N-1}\frac{B_k}{1 - W_N^k z^{-1}}\right) \qquad (29)$$

where A_k and B_k are the PFE residues and

$$1 - z^{-N} = \prod_{k=0}^{N-1}(1 - W_N^k z^{-1}) \qquad (30)$$

has been substituted. Condition (27) is necessary to prevent pole-zero cancellation in (28). The poles must be distinct so that the PFE results in the form given by (29).

When $P = N$, the order of the FDAF [represented by the FS structure contained in (28)] matches the denominator order of $G(z)$; this case will be referred to as *exact modeling*. By appropriate choice of $\{b_k\}$ and $\{c_k\}$, (29) becomes equivalent to

$$G(z) = \frac{(1 - z^{-N})}{N}\sum_{k=0}^{N-1}\frac{b_k + c_k z^{-1}}{(1 - a_k z^{-1})(1 - W_N^k z^{-1})} = \frac{(1 - z^{-N})}{N}\sum_{k=0}^{N-1}\frac{H_k(z)}{1 - W_N^k z^{-1}}, \qquad (31)$$

which is the transfer function from $x(n)$ to $y(n)$ of the FDAF. (The time index n emphasizing the adaptive nature of the filter has been suppressed.) For the equivalence between (29) and (31) to hold, all three coefficients $\{a_k, b_k, c_k\}$ generally must be nonzero. It should be noted that (29) follows from (28) when $M < 2N$ instead of the more restrictive $M < N$. As a result, (31) can represent an improper system with a numerator order of nearly twice that of the denominator. This is possible because each $H_k(z)$ has three coefficients instead of the two required by the parallel implementation in Fig. 1.

When the above conditions are satisfied, it is clear that (29) is an equivalent expression for (26) and the residues $\{A_k, B_k\}$ are unique. The transition from (29) to (31) is not unique, however, because there are $N!$ pairings of the adaptive filter poles at $z = a_k$ with those of the DFT (FS structure) at $z = W_N^k$ which will produce different values of $\{b_k, c_k\}$. Equation (31), therefore, is not a unique representation for (26). Because of this, the MSE surface is multimodal—having $N!$ minima with zero MSE and, depending on the initial conditions, it is possible for the adaptive algorithm to converge to any one of them. This should not be a problem, however, since each minimum has the same MSE and does not correspond to a local minimum. If the algorithm converges to one of these optimal solutions, the overall transfer function in (31) will be equivalent to $G(z)$.

The overmodeling ($N > P$), undermodeling ($N < P$), and multiple-pole cases are described in the Appendix.

B. Performance Surface—MSE

Consider the second-order system

$$G(z) = \frac{1}{1 - 0.1z^{-1} - 0.2z^{-2}}, \qquad (32)$$

which has poles at $z = 0.5$ and $z = -0.4$. The intermediate output signals of the FDAF for $N = 2$ are derived from (11) as follows:

$$y_0(n) = \left(\frac{b_0 + c_0 q^{-1}}{1 - a_0 q^{-1}}\right)x_0(n) \qquad (33a)$$

$$y_1(n) = \left(\frac{b_1 + c_1 q^{-1}}{1 - a_1 q^{-1}}\right)x_1(n) \qquad (33b)$$

where the (fixed) coefficients are real in this case, and from (12) with $k_o = 0$ (or from (19) with $\beta = 1$)

$$x_0(n) = \tfrac{1}{2}(1 + q^{-1})\,x(n) \qquad (34a)$$

$$x_1(n) = \tfrac{1}{2}(1 - q^{-1})\,x(n). \qquad (34b)$$

From the PFE in (29), the residues are $A_0 = -0.37$, $B_0 = 1.43$, $A_1 = -0.17$, and $B_1 = 1.11$. Because $N = 2$, two sets of $\{b_k, c_k\}$ yield a zero MSE. If the adaptive filter pole at $z = 0.5$ is paired with the DFT pole at $z = -1$ (i.e., $a_0 = -0.4$, $a_1 = 0.5$), the following coefficients are obtained: $b_0 = 0.74$, $c_0 = -0.93$, $b_1 = 1.26$, and $c_1 = 0.74$. Using these coefficients, the FDAF poles a_0 and a_1 were then varied and the MSE was computed. The results are shown in Fig. 6. (Similar results can be shown if the adaptive filter pole at $z = 0.5$ is instead paired with the DFT pole at $z = 1$.)

It can be seen in Fig. 6(a) that the MSE increases toward infinity as $a_0 \to 1$ and $a_1 \to -1$; on the other hand, the MSE remains *finite* as $a_0 \to -1$ and $a_1 \to 1$. This behavior is unlike that of the direct-form adaptive IIR filter where the MSE generally increases toward infinity if the magnitude of any pole approaches 1. The FDAF differs because the DFT has zeros on the unit circle that may cancel $\{a_k\}$. From (33) and (34), we see that the DFT

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 13,2010 at 21:39:13 UTC from IEEE Xplore. Restrictions apply.

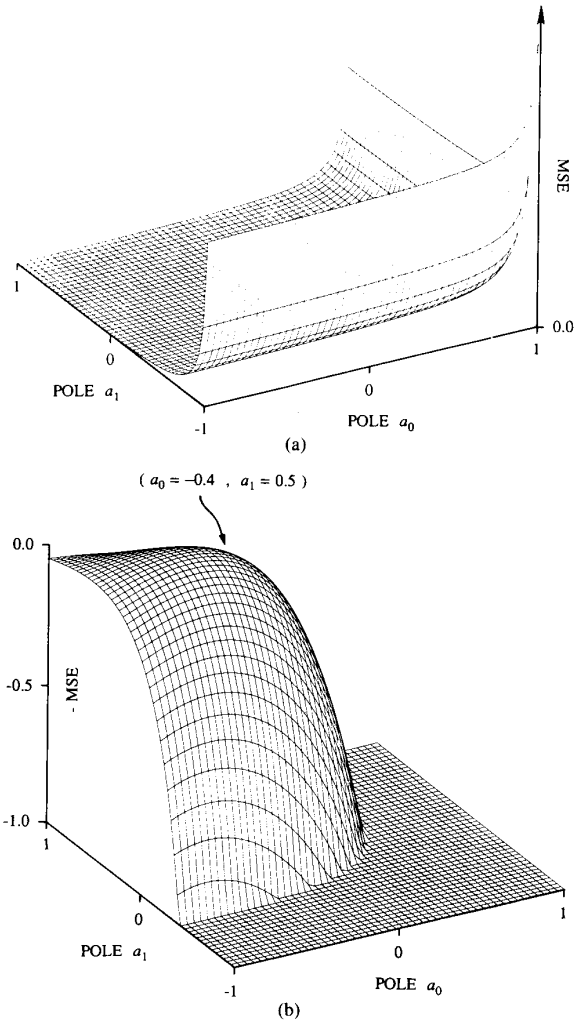


Fig. 6. Performance surface. (a) MSE. (b) Negative of the MSE truncated at -1.

zero at $z = -1$ cancels a pole when $a_0 = -1$; similarly, the DFT zero at $z = 1$ cancels the other pole when $a_1 = 1$. The MSE surface, therefore, does not have an infinite slope at all points along the edge of the stability region.

Fig. 6(b) shows the MSE surface of Fig. 6(a) turned upside down (corresponding to the negative of the MSE) where all values of MSE greater than 1 have been truncated to 1. This enables viewing small values of MSE, particularly such minima as the global minimum at $a_0 = -0.4$ and $a_1 = 0.5$. Additional properties of the MSE surface are described in [29].

C. Asymptotic Variance—CRLB

A useful measure of the asymptotic variance of parameter estimates in system identification is the Cramer–Rao lower bound (CRLB) [10], [33]. Based on certain assumptions [10],

$$\text{cov}\left[\theta(n) - \theta_o\right] \geq \frac{1}{N_s} J^{-1} \tag{35}$$

TABLE III
PER-SAMPLE CRAMER–RAO LOWER BOUND OF A SECOND-ORDER FREQUENCY-DOMAIN ADAPTIVE IIR FILTER

System Poles $\pm p$	Per-Sample CRLB
$\pm$ 0.01	2.65×10^{15}
$\pm$ 0.10	2.52×10^{7}
$\pm$ 0.25	1.73×10^{4}
$\pm$ 0.50	7.03×10^{1}
$\pm$ 0.75	1.82
$\pm$ 0.90	1.24×10^{-1}
$\pm$ 0.99	8.33×10^{-4}

where cov denotes covariance and J is the Fisher information matrix

$$J = E\left[\psi(n, \theta_o)\, \psi^H(n, \theta_o)\right]. \tag{36}$$

The scalar N_s is the number of data samples and θ_o represents the actual system parameters. Note that the expectation must be evaluated at θ_o; this expression, therefore, is contingent on the assumption that $\theta(n)$ converges to θ_o [10].

The CRLB for the coefficients of the FDAF can be determined from (35) using the information vector in (13c). Table III summarizes the *per-sample* ($N_s = 1$) bound for the variance of a_k, which corresponds to a diagonal element of J^{-1}, for the following second-order system:

$$G(z) = \frac{1}{(1 - pz^{-1})(1 + pz^{-1})}. \tag{37}$$

The FDAF coefficients $\{b_k, c_k\}$ were computed from (29) and (31) (exact modeling) where $N = 2$ ($\beta = 1$ and $k_o = 0$). Because the magnitudes of the system poles are identical, the CRLB is the same for the FDAF poles a_0 and a_1.

Observe from the table that the bound increases significantly as the poles move closer to the origin. Intuitively, this occurs because the time constant of the system impulse response (approximately $\tau = 1/(1 - |p|)$ for a first-order system with a pole at $z = |p|$) becomes smaller as the poles move toward the origin; for example, $\tau = 1.01$ for $|p| = 0.01$ compared to $\tau = 100$ for $|p| = 0.99$. As a result, the system is characteristically similar to an FIR filter and can be modeled reasonably well by an adaptive FIR filter of similar complexity. The large per-sample variance indicates that it is difficult to identify poles close to the origin—a result that has been observed in computer simulations [20].

For poles close to the unit circle, the effective impulse response can be so long that an adaptive FIR filter cannot model the system adequately except with a very large number of coefficients. Here, the adaptive poles must be close to the true values for a significant reduction in the

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 13,2010 at 21:39:13 UTC from IEEE Xplore. Restrictions apply.

MSE. The small per-sample variance indicates that poles close to the unit circle are more easily identified.

D. Simulation Results

The configuration described by (25) was used in all simulations, and $x(n)$ was a white random process with a uniform distribution and unit variance; in the simulations with additive noise, $v(n)$ was also a white (uniform) random process uncorrelated with $x(n)$ and with variance σ_v^2. The frequency-domain adaptive algorithm in Table I with a full Hessian matrix was used except where noted; the shift parameter $k_o = 0.5$. The FDAF step size α, initial variance δ, and order N are shown in the figures of the convergence plots. All coefficients $\{a_k(n), b_k(n), c_k(n)\}$ were initialized to zero. An estimate of the MSE at each instant of time was obtained by averaging $|e(n)|^2$ over 25 independent computer runs. The coefficients were similarly averaged to have smooth trajectories.

1) Exact Modeling: In this case, $G(z)$ was a fourth-order system with distinct poles located well inside the unit circle. Its transfer function was

$$G(z) = K\frac{(1 - 0.9z^{-1})(1 + 0.81z^{-2})}{(1 - 0.71z^{-1} + 0.25z^{-2})(1 + 0.75z^{-1} + 0.56z^{-2})} \tag{38}$$

which has poles at $p_{1,2} = 0.5/\pm 45°$, $p_{3,4} = 0.75/\pm 120°$, and zeros at $z_{1,2} = 0.9/\pm 90°$, $z_3 = 0.9$, $z_4 = 0$. The gain K was chosen such that $G(q^{-1})\,x(n)$ had unit power, and was similarly chosen for the other simulations. Fig. 7 shows the MSE learning curve and the pole trajectories of the FDAF. Observe that only two poles are shown because the other two are the complex conjugates of them.

Because $N = 4$, the FDAF has sufficient order to exactly model $G(z)$. We see that the FDAF poles converge to the true poles of $G(z)$ (indicated by the dots) and that the MSE converges essentially to zero. Two points on each pole trajectory and the corresponding points on the MSE learning curve have been labeled. Observe that the poles have very different rates of convergence; a_2 converges rapidly, approximately by iteration 2000, and a_1 requires twice as many iterations, converging by iteration 4000. These results (and that of other simulations) suggest that poles closest to the unit circle have faster rates of convergence. Notice that a_1 first "wanders" over to frequency bin $k = 2$, corresponding to the reduced MSE convergence rate between iterations 1000 and 2000, and then converges in bin $k = 1$. It may converge faster if constrained adaptation is applied to control the pole trajectories; however, this would require *a priori* knowledge of the pole locations of $G(z)$.

It is possible to calculate from (31) the values of the numerator coefficients $\{b_k, c_k\}$ corresponding to the converged poles in Fig. 7(b); the results are listed in Table IV. From the trajectories in Fig. 8, we see that the FDAF

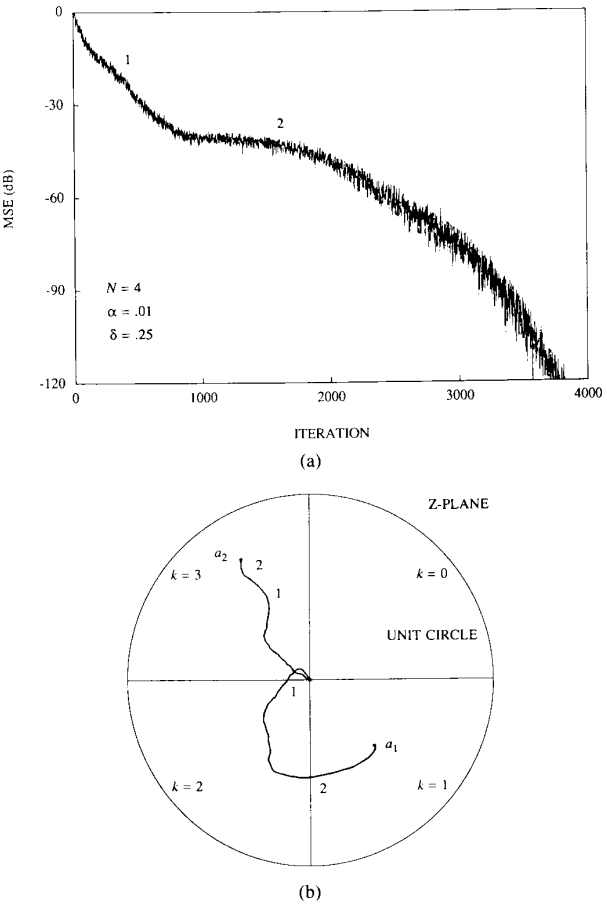


Fig. 7. Exact-modeling simulation. (a) MSE learning curve. (b) Pole trajectories.

TABLE IV
PARTIAL FRACTION EXPANSION FOR THE EXACT-MODELING SIMULATION

Index k	Filter Coefficients		
	a_k	b_k	c_k
0	0.354 + 0.354j	0.375 - 0.362j	-0.134 - 0.076j
1	0.354 - 0.354j	0.375 + 0.362j	-0.134 + 0.076j
2	-0.375 + 0.650j	0.840 + 0.640j	0.496 - 0.341j
3	-0.375 - 0.650j	0.840 - 0.640j	0.496 + 0.341j

numerator coefficients converge to the values given in the table. These results confirm the exact-modeling theory and demonstrate that the FDAF can model a system of the same denominator order with distinct poles.

Different forms of the Hessian matrix can be employed in the adaptive algorithm, as discussed in Section III. Fig. 9 compares the MSE learning curve in Fig. 7(a), which corresponds to a full Hessian, with those for block-diagonal and diagonal Hessians. It can be seen that the slowest

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 13,2010 at 21:39:13 UTC from IEEE Xplore. Restrictions apply.

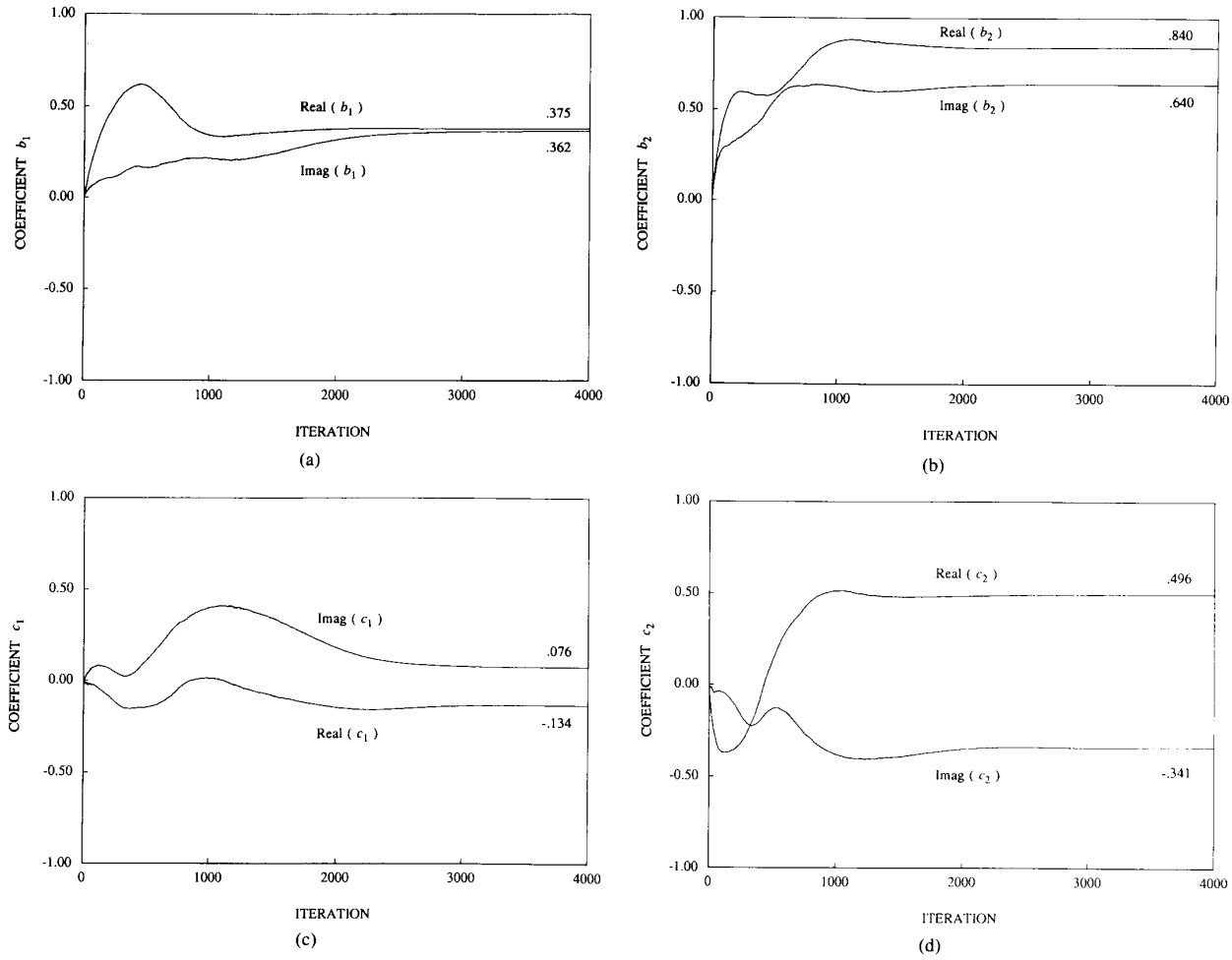


Fig. 8. Exact-modeling simulation. (a) Trajectory of b_1. (b) Trajectory of b_2. (c) Trajectory of c_1. (d) Trajectory of c_2.

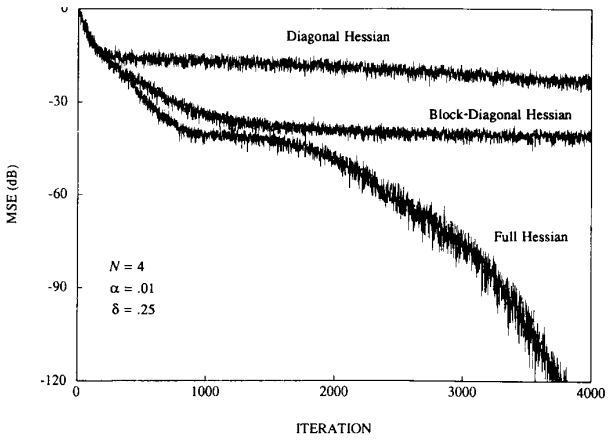


Fig. 9. Exact-modeling simulation. MSE learning curves for the full, block-diagonal, and diagonal Hessian matrices.

rate of convergence is obtained with the diagonal Hessian, as expected. Using the block-diagonal Hessian requires only slightly more complexity than the diagonal Hessian and improves the performance by more than 10 dB. Notice that the MSE is less than -30 dB by iteration 1000. In applications when there is additive noise, the block-diagonal Hessian may provide satisfactory performance.

2) Overmodeling: In this simulation, $G(z)$ was the second-order system

$$G(z) = K\frac{1 - 0.9z^{-1}}{1 - 0.71z^{-1} + 0.25z^{-2}} \qquad (39)$$

with distinct poles at $p_{1,2} = 0.5/\pm 45°$ and zeros at $z_1 = 0.9$, $z_2 = 0$. Fig. 10 shows the MSE learning curve and pole trajectories for the FDAF with $N = 4$; again, the trajectories of only two poles are shown.

Observe that the poles have converged to the *same* value in frequency bin $k = 0$, as predicted by the analysis in the Appendix. Because the MSE has converged to zero, the FDAF transfer function is equivalent to $G(z)$; however, the convergence rate is somewhat slower than that of the simulation shown in Fig. 7. If $N = 2$ had been chosen instead, the MSE convergence would be faster and the

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 13,2010 at 21:39:13 UTC from IEEE Xplore. Restrictions apply.

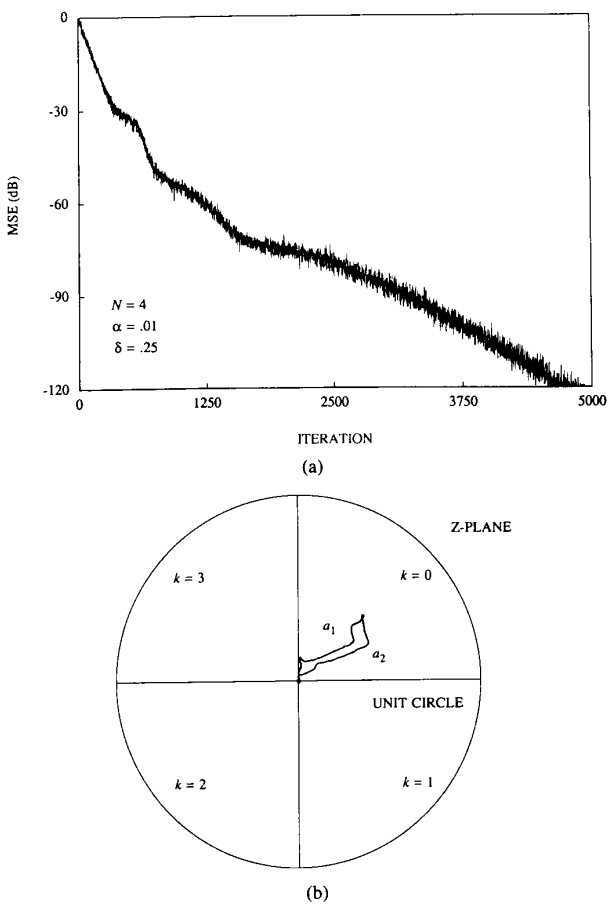


Fig. 10. Overmodeling simulation. (a) MSE learning curve. (b) Pole trajectories.

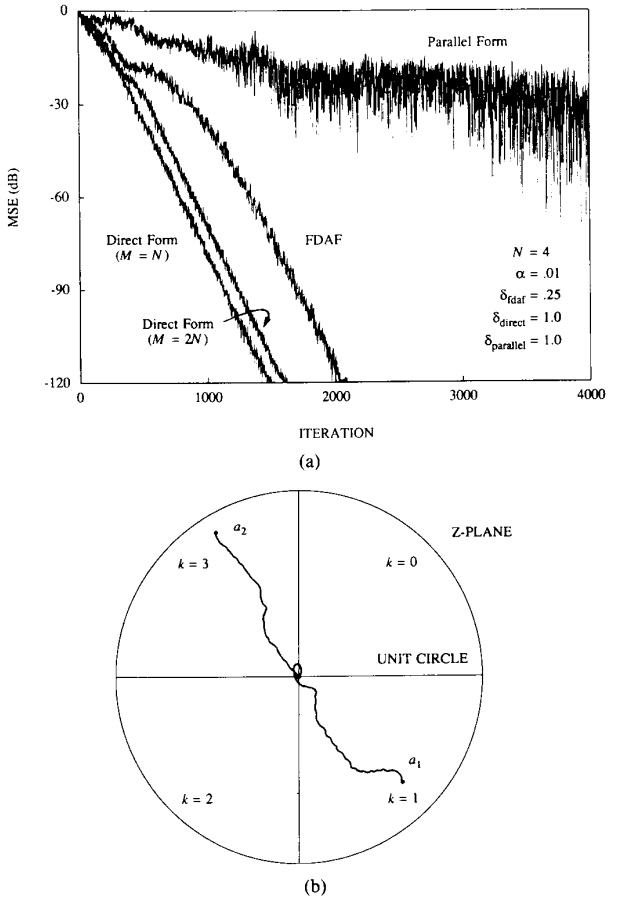


Fig. 11. Simulation with poles close to the unit circle. (a) MSE learning curves. (b) Pole trajectories.

FDAF would have converged to the exact-modeling solution in (31).

3) Poles Close to the Unit Circle: The system in (38) was also used in this simulation, except the poles were at $p_{1,2} = 0.8/\underline{\pm 45°}$ and $p_{3,4} = 0.9/\underline{\pm 120°}$. Observe from Fig. 11 that the convergence rate of the MSE is fast—converging by iteration 2000. Furthermore, the poles have converged exactly to the system poles as expected since $N = 4$.

The MSE learning curves of the direct-form adaptive IIR filter are also plotted in Fig. 11 for two values of M (the numerator order). When $M = 2N$, the direct form has the same number of coefficients ($3N$) as the FDAF and similar algorithm complexity. Observe that the direct form converges slightly faster than the FDAF, which is not unexpected for two reasons. First, the FDAF is clearly a more complicated model of the system. In parameter estimation theory, a model with sufficient order generally achieves the best performance, which is known as the parsimony principle [10]; models with too many parameters often have degraded performance. Because the FDAF must have $3N$ filter coefficients to model an Nth-order system, it cannot be expected to always perform as well as the direct form.

The second reason is related to the nonuniqueness of the FDAF coefficients in (31) where there are $N!$ sets of $\{b_k, c_k\}$ that produce the same transfer function. Consequently, it is possible for some of the FDAF poles to initially move in one direction and then, as other poles converge, change direction to satisfy (31). An example of this effect was illustrated in Fig. 7(b). Notice, however, that the convergence rates of the FDAF and direct form are very similar up to iteration 500 where the MSE is approximately -20 dB. For applications with additive noise or nonzero minimum MSE, the difference in convergence rates of the two filters becomes less significant.

The MSE learning curve for the parallel form without preprocessing of the input signal (Fig. 1, using second-order sections) is also shown in Fig. 11. Because the subfilters were initialized to zero, observe that convergence is considerably slower. This result is consistent with the singularity condition described in Section II. The simulations in [18] show that when the subfilters are initialized with different values, convergence is more rapid, approaching that of the FDAF. Proper initialization of the parallel form can lead to satisfactory performance, although it is clear that the convergence rate of the GN algorithm can be severely degraded whenever the coeffi-

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 13,2010 at 21:39:13 UTC from IEEE Xplore. Restrictions apply.

cients lie on (or are near) the surface manifold. The convergence rate of the GN algorithm for the FDAF is also slower near this manifold, but the effect is clearly less significant.

Additive noise with a variance of $\sigma_v^2 = 0.001$ was included in the next simulation where the system in (38) was again used, except with poles at $p_{1,2} = 0.95/\pm 45°$ and $p_{3,4} = 0.98/\pm 120°$. The simplified IIR FDAF algorithm in Table II, which incorporates a block-diagonal Hessian, updated the coefficients. This example with noise and system poles very close to the unit circle was designed to stress the FDAF. It can be seen in Fig. 12 that, even in this difficult case, the convergence rate of the algorithm is fast. The average number of times the poles attempted to update outside the unit circle was approximately 75 over the 4000 iterations. As a result of stability monitoring, the FDAF remained stable during convergence. (Unstable updates were simply skipped as described in Section II.)

The MSE learning curve of the FIR FDAF in Fig. 12 is based on the IIR FDAF with the coefficients $\{a_k, c_k\}$ set to zero, and corresponds to the adaptive filter introduced by Narayan [23]. It is apparent that the performance of the IIR FDAF is much better even though it has fewer coefficients; it would be necessary to increase significantly the order of the FIR FDAF to achieve similar MSE performance [19]. The FIR FDAF can only approximately model this IIR system.

4) Frequency-Sampling Realization: In this simulation, the FS implementation of the DFT in (19) was employed instead of the FFT. Because this implementation is less complex than the FFT, it is preferred if there is no degradation in performance. The system was as in (38), but with poles at $p_{1,2} = 0.5/\pm 30°$ and $p_{3,4} = 0.75/\pm 120°$. Fig. 13 displays the results for $\beta = 0.9$; for comparison, the MSE learning curve of the FDAF using the FFT is also shown. Observe that the performance of the FS-FDAF is similar to that of the FFT-FDAF, and the FS-FDAF poles have converged to those of $G(z)$. These results and those of other simulations indicate that the FS implementation results in essentially the same performance as the FFT. The imperfect pole-zero cancellation discussed in Section III does not appear to adversely affect the performance of the FDAF.

V. Conclusion

Parallel-form realizations for adaptive IIR filtering have been examined and, in particular, a frequency-domain adaptive IIR filter (FDAF) has been presented. The FDAF has a parallel structure of first-order sections that facilitates stability monitoring. It is comprised of a DFT that operates as a bank of bandpass filters to preprocess the input signal and transform a wide-band adaptive filter into several narrow-band adaptive filters (Fig. 3). A Gauss–Newton adaptive algorithm to adjust the coefficients was detailed; two simpler versions of the algorithm that exploit orthogonality properties of the DFT were also discussed. The DFT can be implemented by a frequency-sampling structure that has similar performance as the FFT but is less complex. It was also shown that the FDAF can be extended to a general parallel-form adaptive IIR filter using bandpass filters to preprocess the input signal (Fig. 5).

Several performance characteristics of the FDAF in a system identification application were presented. It was

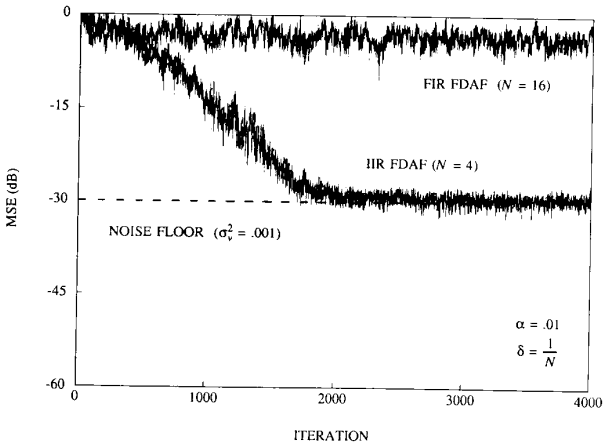
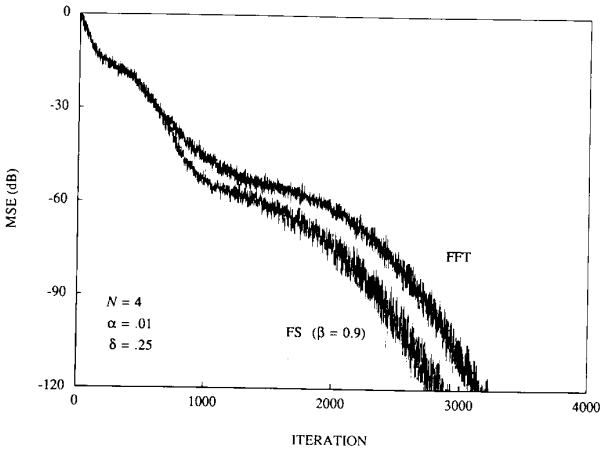
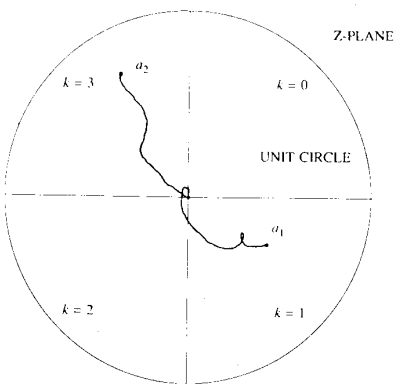


Fig. 12. Simulation with poles close to the unit circle and additive noise. MSE learning curves.



(a)



(b)

Fig. 13. Frequency-sampling simulation using the full Hessian matrix. (a) MSE learning curves. (b) FS-FDAF pole trajectories.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 13,2010 at 21:39:13 UTC from IEEE Xplore. Restrictions apply.

shown that the FDAF can model exactly any proper rational system with distinct poles if the order of the FDAF (DFT) exceeds the denominator order of the system. This modeling result is actually more general and has significance in many other applications such as linear prediction and channel equalization. The MSE performance surface of a second-order FDAF was shown to be finite at certain boundaries of the stability region. Analysis of the CRLB illustrated the sensitivity of the FDAF to various pole locations of the unknown system. Finally, computer simulations of the FDAF for several systems were presented that demonstrate convergence of the adaptive algorithm and confirm the system-modeling analysis (Fig. 7).

Appendix

This appendix describes the modeling properties of the FDAF in system identification when the order of the FDAF is different than that of the system being modeled.

1) Overmodeling: Assume that $N > P$ so that the adaptive filter has more poles than $G(z)$. To exactly model $G(z)$, one solution requires that $N - P$ feedback coefficients $\{a_k\}$ be zero. From (29),

$$\begin{aligned} G(z) &= \frac{(1 - z^{-N})}{N}\left[\sum_{k=0}^{P-1}\left(\frac{A_k}{1 - a_k z^{-1}} + \frac{B_k}{1 - W_N^k z^{-1}}\right) + \sum_{k=P}^{N-1}\frac{B_k}{1 - W_N^k z^{-1}}\right] \\ &= \frac{(1 - z^{-N})}{N}\left(\sum_{k=0}^{P-1}\frac{b_k + c_k z^{-1}}{(1 - a_k z^{-1})(1 - W_N^k z^{-1})} + \sum_{k=P}^{N-1}\frac{b_k}{(1 - W_N^k z^{-1})}\right) \\ &= \frac{(1 - z^{-N})}{N}\left(\sum_{k=0}^{P-1}\frac{H_k(z)}{(1 - W_N^k z^{-1})} + \sum_{k=P}^{N-1}\frac{b_k}{(1 - W_N^k z^{-1})}\right), \end{aligned} \tag{A.1}$$

which is similar to (31) except that $N - P$ subfilters (second summation) each contains only one coefficient b_k. Although this is an exact model of $G(z)$ that will achieve zero MSE, simulations indicate that the FDAF generally does not converge to this solution. Another solution is obtained when all subfilters have nonzero poles, such that *identical* poles occur in some of the DFT frequency bins. Equation (29) can then be rewritten as

$$G(z) = \frac{(1 - z^{-N})}{N}\left[\sum_{k=0}^{P-1}\left(\frac{C_k}{1 - a_k z^{-1}} + \frac{D_k}{1 - a_k z^{-1}}\right) + \sum_{k=0}^{N-1}\frac{B_k}{1 - W_N^k z^{-1}}\right] \tag{A.2}$$

where $C_k + D_k = A_k$. If $N = 2P$ is assumed without loss of generality, each W_N^k can be paired with a pole of $G(z)$ so that

$$\begin{aligned} G(z) &= \frac{(1 - z^{-N})}{N}\left(\sum_{k=0}^{P-1}\frac{b_k + c_k z^{-1}}{(1 - a_k z^{-1})(1 - W_N^k z^{-1})} + \sum_{k=P}^{N-1}\frac{b_k + c_k z^{-1}}{(1 - a_{k-P} z^{-1})(1 - W_N^k z^{-1})}\right) \\ &= \frac{(1 - z^{-N})}{N}\sum_{k=0}^{N-1}\frac{H_k(z)}{1 - W_N^k z^{-1}}. \end{aligned} \tag{A.3}$$

Note that the second summation begins at $k = P$ so that the corresponding $\{b_k, c_k\}$ generally differ from those of the first; however, for each pole in the first summation, there is an identical pole in the second (indicated by the subscript of a_{k-P}). Because of the parallel configuration, the overall filter will have only P distinct poles. Equation (A.3) is a valid solution, therefore, and will achieve a zero MSE (see the simulation in Fig. 10). Notice that there is an infinite number of minima because of the infinitely many values for $\{C_k, D_k\}$. Each solution is optimal, however, with a transfer function equivalent to $G(z)$.

2) Undermodeling: If $G(z)$ has more poles than the FDAF ($N < P$), the FDAF can only approximate $G(z)$ and it is not possible to have a zero MSE for any set of coefficients. Analogous to (A.1),

$$\begin{aligned} G(z) &= \frac{(1 - z^{-N})}{N}\left[\sum_{k=0}^{N-1}\left(\frac{A_k}{1 - a_k z^{-1}} + \frac{B_k}{1 - W_N^k z^{-1}}\right) + \sum_{k=N}^{P-1}\frac{A_k}{1 - a_k z^{-1}}\right] \\ &= \frac{(1 - z^{-N})}{N}\left(\sum_{k=0}^{N-1}\frac{b_k + c_k z^{-1}}{(1 - a_k z^{-1})(1 - W_N^k z^{-1})} + \sum_{k=N}^{P-1}\frac{b_k}{(1 - a_k z^{-1})}\right) \\ &= \frac{(1 - z^{-N})}{N}\left(\sum_{k=0}^{N-1}\frac{H_k(z)}{(1 - W_N^k z^{-1})} + \sum_{k=N}^{P-1}\frac{b_k}{(1 - a_k z^{-1})}\right). \end{aligned} \tag{A.4}$$

Because $P - N$ poles cannot be matched with any W_N^k of the FS structure, it is clear that the FDAF can only approximate $G(z)$. This is true for any adaptive IIR filter with insufficient order.

3) Multiple Poles: Because of the parallel structure, it is not possible for the FDAF to model exactly a system that has multiple poles of the same value. For example, consider a system that has two poles at $z = a$. If two of the FDAF subfilters have $a_k = a$, the overall transfer function will have a single pole at $z = a$. Because the two subfilters are in parallel, the numerator coefficients can be combined over the common denominator. (Clearly, the parallel realization in Fig. 1 also has this property.)

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 13,2010 at 21:39:13 UTC from IEEE Xplore. Restrictions apply.

The FDAF can be modified so that double-pole systems can be modeled if one or more $H_k(n, z)$ in (11) are replaced by second-order subfilters as follows:

$$y_k(n) = \left(\frac{b_k^*(n) + c_k^*(n)q^{-1} + d_k^*(n)q^{-2}}{1 - \alpha_k^*(n)q^{-1} - \beta_k^*(n)q^{-2}} \right) x_k(n). \tag{A.5}$$

With this change, a subfilter can have two poles of the same value, and it is still easy to confirm stability.

If any of the subfilters are required to have an order >2, stability verification is no longer trivial and the FDAF will encounter the same robustness problems as the direct form. (A possible solution, however, may be to implement the high-order subfilters in a cascade or lattice form [5].) As a result, the FDAF of Fig. 3 and the parallel form of Fig. 1 are restricted to applications where the modeling of multiple poles is not important.

REFERENCES

[1] C. R. Johnson, Jr., "Adaptive IIR filtering: Current results and open issues," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 237–250, Mar. 1984.

[2] J. R. Treichler, "Adaptive algorithms for infinite impulse response filters," in *Adaptive Filters*, C. F. N. Cowen and P. M. Grant, Eds. Englewood Cliffs, NJ: Prentice-Hall, 1985, ch. 4, pp. 60–90.

[3] S. A. White, "An adaptive recursive digital filter," in *Proc. 9th Asilomar Conf. Circuits, Syst., Comput.*, Pacific Grove, CA, Nov. 1975, pp. 21–25.

[4] P. L. Feintuch, "An adaptive recursive LMS filter," *Proc. IEEE*, vol. 64, pp. 1622–1624, Nov. 1976.

[5] S. Horvath, Jr., "Adaptive IIR digital filters for on-line time-domain equalization and linear prediction," presented at IEEE Arden House Workshop Dig. Sig. Proc., Harriman, NY, Feb. 1976.

[6] S. D. Stearns, G. R. Elliot, and N. Ahmed, "On adaptive recursive filtering," in *Proc. 10th Asilomar Conf. Circuits, Syst., Comput.*, Pacific Grove, CA, Nov. 1976, pp. 5–11.

[7] M. G. Larimore, J. R. Treichler, and C. R. Johnson, Jr., "SHARF: An algorithm for adapting IIR digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 428–440, Aug. 1980.

[8] B. Friedlander, "System identification techniques for adaptive signal processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 240–246, Apr. 1982.

[9] H. Fan and W. K. Jenkins, "A new adaptive IIR filter," *IEEE Trans. Circuits Syst.*, vol. CAS-33, pp. 939–947, Oct. 1986.

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

[11] D. Parikh, N. Ahmed, and S. D. Stearns, "An adaptive lattice algorithm for recursive filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 110–111, Feb. 1980.

[12] R. A. David, "IIR adaptive algorithms based on gradient search techniques," Ph.D. dissertation, Dep. Elec. Eng., Stanford Univ., Stanford, CA, Aug. 1981.

[13] W. K. Jenkins and M. Nayeri, "Adaptive filters realized with second order sections," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tokyo, Japan, Apr. 1986, pp. 2103–2106.

[14] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

[15] J. J. Shynk, "A complex adaptive algorithm for IIR filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1342–1344, Oct. 1986.

[16] M. Nayeri, D. F. Marshall, and W. K. Jenkins, "Relationships between error surfaces and network architectures in adaptive filters," in *Proc. IEEE Int. Symp. Circuits Syst.*, Philadelphia, PA, May 1987, pp. 777–780.

[17] M. Nayeri and W. K. Jenkins, "Analysis of alternate realizations of adaptive IIR filters," in *Proc. IEEE Int. Symp Circuits Syst.*, Espoo, Finland, June 1988, pp. 2157–2160.

[18] J. J. Shynk, "Performance of alternative adaptive IIR filter realizations," in *Proc. 21st Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 1987, pp. 144–150.

[19] J. J. Shynk and R. P. Gooch, "Frequency-domain adaptive pole-zero filtering," *Proc. IEEE*, vol. 73, pp. 1526–1528, Oct. 1985.

[20] J. J. Shynk, "Frequency-domain adaptive pole-zero filtering," Ph.D. dissertation, Dep. Elec. Eng., Stanford Univ., Stanford, CA, Dec. 1986.

[21] M. Dentino, J. McCool, and B. Widrow, "Adaptive filtering in the frequency domain," *Proc. IEEE*, vol. 66, pp. 1658–1659, Dec. 1978.

[22] E. R. Ferrara, Jr., "Fast implementation of LMS adaptive filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 474–475, Aug. 1980.

[23] S. S. Narayan and A. M. Peterson, "Frequency domain least-mean-square algorithm," *Proc. IEEE*, vol. 69, pp. 124–126, Jan. 1981.

[24] R. R. Bitmead and B. D. O. Anderson, "Adaptive frequency sampling filters," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 524–534, June 1981.

[25] D. Mansour and A. H. Gray, Jr., "Unconstrained frequency-domain adaptive filter," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 726–734, Oct. 1982.

[26] G. A. Clark, S. R. Parker, and S. K. Mitra, "A unified approach to time- and frequency-domain realization of FIR adaptive digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1073–1083, Oct. 1983.

[27] N. J. Bershad and P. L. Feintuch, "The recursive adaptive LMS filter—A line enhancer application and analytical model for the mean weight behavior," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 652–660, Dec. 1980.

[28] D. F. Elliott and K. R. Rao, *Fast Transforms: Algorithms, Analyses, Applications*. Orlando, FL: Academic, 1982.

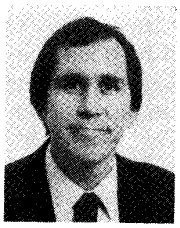
[29] H. Fan, Y. Yang, and M. Nayeri, "On the performance surfaces of a frequency domain adaptive IIR filter," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, New York, NY, Apr. 1988, pp. 1564–1567.

[30] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[31] J. J. Shynk and B. Widrow, "Bandpass adaptive pole-zero filtering," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tokyo, Japan, Apr. 1986, pp. 2107–2110.

[32] K. J. Astrom and P. Eykhoff, "System identification—A survey," *Automatica*, vol. 7, pp. 123–162, Mar. 1971.

[33] B. Friedlander, "On the computation of the Cramer–Rao bound for ARMA parameter estimation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 721–727, Aug. 1984.



John J. Shynk (S'78–M'86) was born in Lynn, MA, on June 20, 1956. He received the B.S. degree in systems engineering from Boston University, Boston, MA, in 1979, and the M.S. degree in electrical engineering and in statistics, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1980, 1985, and 1987, respectively.

From 1979 to 1982 he was a member of Technical Staff in the Data Communications Performance Group at Bell Laboratories, Holmdel, NJ, where he formulated performance models for voiceband data communications. He was a Research Assistant from 1982 to 1986 in the Electrical Engineering Department at Stanford University where he worked on frequency-domain implementations of adaptive filter algorithms. From 1985 to 1986 he was also an Instructor at Stanford teaching courses on digital signal processing and adaptive systems. Since 1987 he has been an Assistant Professor in the Department of Electrical and Computer Engineering at the University of California, Santa Barbara. His current research interests include developing and analyzing efficient adaptive signal processing algorithms for applications in system identification, communications, and adaptive array processing.

Dr. Shynk is a member of Tau Beta Pi and Sigma Xi.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 13,2010 at 21:39:13 UTC from IEEE Xplore. Restrictions apply.