Adaptive IIR Filtering

John J. Shynk

This paper presents an overview of several methods, filter structures, and recursive algorithms used in adaptive IIR filtering. Both the equation-error and output-error formulations are described, although the paper focuses on the adaptive algorithms and properties of the output-error configuration. These parameter-update algorithms have the same generic form and they are based on a prediction-error performance criterion. A direct-form implementation of the adaptive filters is emphasized, but alternative realizations such as the parallel and lattice forms are briefly discussed. Several important issues associated with adaptive IIR filtering, including stability monitoring, the SPR condition, and convergence, are also addressed.

INTRODUCTION

OVER THE LAST SEVERAL YEARS, adaptive infinite-impulse-response (IIR) filtering has been an active area of research [Jo84, Tr85], and it has been considered for a variety of problems in signal processing and communications. Examples of some important applications include linear prediction [Hv76], adaptive notch filtering [Fr84a, Ne85], adaptive differential pulse code modulation (ADPCM) [Ja84], channel equalization [Pr83], echo cancellation [Lo87], and adaptive array processing [Gc86]. In addition, several techniques used in adaptive IIR filtering have been derived from the fields of system identification [As71, La74, Fr82, Lj83] and adaptive control [La79, Gd84] where it is often assumed that the underlying models have a pole-zero structure. Many of the known convergence results for adaptive IIR algorithms require that the filter be operating in a system identification configuration such that the unknown system can be represented by a stable rational transfer function.

Fig. 1 illustrates the general structure and components of an adaptive IIR filter with input $x(n)$ and output $y(n)$. Observe that it is comprised of a *time-varying filter,* characterized by the adjustable coefficients $\theta(n)$, and a *recursive algorithm* that adjusts $\theta(n)$ so that $y(n)$ approximates some desired response $d(n)$, which is determined by the particular application [Wi85]. For example, Fig. 2 shows the adaptive filter in a system identification configuration where θ_* are the unknown system parameters, and $d(n)$ is simply the measured output of the system, which usually includes an additive noise process $v(n)$. The objective of the algorithm is to minimize a performance criterion that is based on the prediction error $e(n)$ (sometimes called the estimation error), defined by $e(n) = d(n) - y(n)$. One commonly used criterion is the mean-square error (MSE), $\xi = E[e^2(n)]$, where E is statistical expectation; the corresponding coefficient recursions are called stochastic gradient (SG) algorithms or recursive Gauss-Newton (GN) algorithms. Another criterion is based on the method of least squares, and the resulting algorithms are known as recursive least squares (RLS).

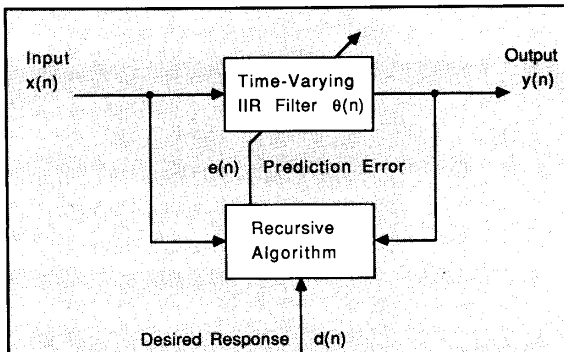

Figure 1. Adaptive IIR Filter. An adaptive IIR filter is comprised of two basic components: a time-varying IIR filter with input $x(n)$ and output $y(n)$, and a recursive algorithm that adjusts the filter coefficients $\theta(n)$ to optimize a performance criterion that is based on the prediction error $e(n)$. The goal of the adaptive filter is to generate a suitable estimate $y(n)$ of the desired response $d(n)$.

Fundamentally, there have been two approaches to adaptive IIR filtering that correspond to different formulations of the prediction error; these are known as *equation error* and *output error* methods. In the equation-error formulation [Mn73, Gc83], the feedback coefficients of an IIR filter are updated in an all-zero, nonrecursive[1] form which are then copied to a second filter implemented in an all-pole form. This formulation is essentially a type of

[1]Pole-zero (IIR) and all-zero (FIR) filters are often denoted recursive and nonrecursive, respectively. This description follows from the structure of their difference equations, and it should not be confused with the recursive nature of adaptive algorithms.

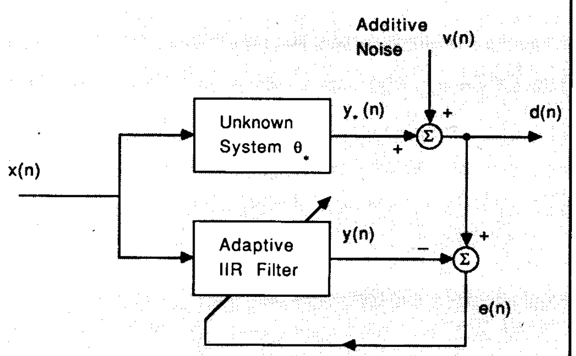

Figure 2. System Identification Configuration. The adaptive IIR filter is shown in a system identification configuration where it is estimating the parameters θ_* of some unknown system $H_*(z)$, often called the "plant." The system dynamics are usually more complicated than that described by a difference equation, so the adaptive filter is an approximate model of the system characteristics. The additive signal $v(n)$ is often measurement noise, and it is usually uncorrelated with the input $x(n)$. This configuration is an example of the output-error formulation (Fig. 4), and in the control literature it is generally referred to as a model reference adaptive system (MRAS) [La79].

adaptive FIR (finite-impulse-response) filtering, and the corresponding algorithms have properties that are well understood and predictable. Unfortunately, the equation-error approach can lead to biased estimates of the coefficients θ_*. The output-error formulation [La79, Jo84] updates the feedback coefficients directly in a pole-zero, recursive form and it does not generate biased estimates. However, the adaptive algorithms can converge to a local minimum of ξ leading to an incorrect estimate of θ_*, and their convergence properties are not easily predicted. As a result, there is a trade-off between converging to a biased estimate of the coefficients and converging to a local minimum of ξ.

The primary advantage of an adaptive IIR filter is that it can provide significantly better performance than an adaptive FIR filter having the same number of coefficients. This is a consequence of the output feedback which generates an infinite impulse response with only a finite number of parameters. A desired response or, equivalently, its frequency response can be approximated more effectively by the output of a filter that has both poles and zeros compared to one that has only zeros. For example, an adaptive IIR filter with sufficient order can exactly model an unknown pole-zero system (represented by θ_*), whereas an adaptive FIR filter can only approximate such a system. This improved performance is also obtained in other applications. Alternatively, to achieve a specified level of performance, an IIR filter generally requires considerably fewer coefficients than the corresponding FIR filter. Because of the potential savings in computational complexity, it is anticipated that the adaptive IIR filter will replace the widely-used adaptive FIR filter in many applications.

The goal of this paper is to provide a basic understanding of the algorithms used to update the coefficients of adaptive IIR filters. Both the equation-error and output-error formulations are initially examined; the paper then focuses on output-error algorithms and their properties. One such algorithm is the *recursive prediction-error (RPE)* algorithm which is a gradient-descent approach that is based on an instantaneous estimate of ξ. An approximation to the gradient leads to a simpler algorithm known as the *pseudolinear regression (PLR)* algorithm. A stochastic convergence analysis of these algorithms, which involves the study of an associated ODE (ordinary differential equation), is then outlined. Important issues related to convergence, such as stability monitoring of the poles and the SPR (strictly positive real) condition, are also discussed. As we shall see, the properties of an adaptive IIR filter are considerably more complex than those of the conventional adaptive FIR filter, and it is more difficult to predict the behavior of an adaptive IIR algorithm in a general way.

Most adaptive IIR algorithms have been derived for a direct-form implementation of the filter coefficients. However, some disadvantages of the direct form such as finite-precision effects and the complexity of stability monitoring have led to the development of algorithms for alternative structures. The computational complexity and convergence properties of adaptive algorithms can vary widely depending on the filter realization used. This paper concentrates on direct-form implementations, but it also briefly describes the adaptive algorithms designed for parallel- and lattice-form realizations.

EQUATION-ERROR AND OUTPUT-ERROR FORMULATIONS

Equation-error formulation

Consider the equation-error adaptive IIR filter shown in Fig. 3 which is characterized by the *nonrecursive* difference equation:

$$y_e(n) = \sum_{m=1}^{N-1} a_m(n)d(n-m) + \sum_{m=0}^{M-1} b_m(n)x(n-m), \qquad (1)$$

where $\{a_m(n), b_m(n)\}$ are the adjustable coefficients and the subscript e is used to distinguish this output from that of the output-error formulation. Observe that (1) is a two-input, single-output filter that depends on delayed samples of the input $x(n-m)$, $m = 0, \ldots, M-1$, and of the desired response $d(n-m)$, $m = 1, \ldots, N-1$. It does not depend on delayed samples of the output and, therefore, the filter does not have feedback; the output is clearly a *linear* function of the coefficients. This property greatly simplifies the derivation of gradient-based algorithms. Since $d(n)$ and $x(n)$ are not functions of the coefficients, the derivative of $y_e(n)$ with respect to the coefficients is nonrecursive and is easy to compute.

This expression can be rewritten in a more convenient form using delay-operator notation as follows:

$$y_e(n) = A(n,q)d(n) + B(n,q)x(n), \qquad (2)$$

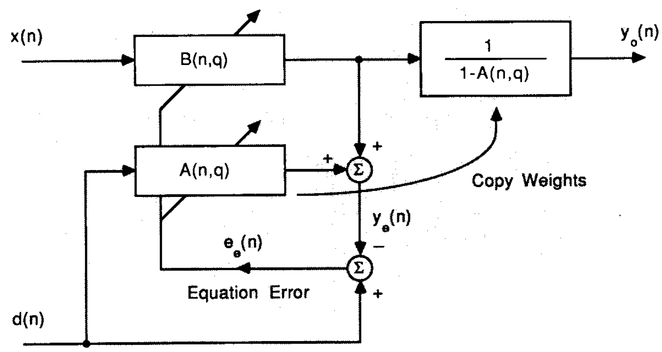

Figure 3. Equation-Error Formulation. The equation-error formulation filters both the input signal $x(n)$ and the *delayed* desired response $d(n-1)$ to generate an estimate of $d(n)$. It essentially operates as a two-input, single-output adaptive FIR filter such that the polynomial associated with the poles, $1 - A(n, q)$, is adapted in an all-zero form. After each update of the weights, the inverse of $1 - A(n, q)$ is copied to an all-pole filter which is in cascade with $B(n, q)$. The impulse response from $x(n)$ to $y_o(n)$ is infinite, and the cascaded filters have the same form as that of the output-error formulation (Fig. 4). However, the converged coefficients obtained with this approach are generally different from those generated by the output-error formulation.

where the polynomials in q represent time-varying filters and are defined by

$$A(n,q) = \sum_{m=1}^{N-1} a_m(n)q^{-m} \quad \text{and} \quad B(n,q) = \sum_{m=0}^{M-1} b_m(n)q^{-m}. \tag{3}$$

Note that the lower limit of the sum for $A(n,q)$ begins with $m = 1$; consequently, $A(n,q)d(n)$ depends only on *delayed* samples of d. The argument n emphasizes the time dependence of the coefficients and q^{-1} is the delay operator such that $q^{-m}x(n) = x(n-m)$. These functions of q operate on time signals only from the left as in (2).[2] By replacing q with the complex variable z, the expressions in (3) become z-transforms (transfer functions), assuming that the coefficients are fixed (independent of time), i.e., $a_m(n) \rightarrow a_m$ and $b_m(n) \rightarrow b_m$ so that $A(n,q) \rightarrow A(z)$ and $B(n,q) \rightarrow B(z)$. This form can be used to find the zeros of the adaptive filter at any instant of time. For example, after each update of the coefficients and before the coefficients $\{a_m(n)\}$ are copied to the inverse filter (Fig. 3), it will be necessary to monitor the zeros of $1 - A(z)$ to determine if its inverse is a stable system. If it is not stable, then some method of projecting the roots inside the unit circle will be necessary before the inverse filter is formed.

The equation error is given by $e_e(n) = d(n) - y_e(n)$. It is called this because it is generated by subtracting two difference equations: $[1 - A(n,q)]d(n)$ and $B(n,q)x(n)$. Note that $e_e(n)$ is also a linear function of the coefficients; as a result, the mean-square-equation error (MSEE) is a quadratic function with a single global minimum (provided the data correlation matrix is nonsingular) and no local minima [Wi85]. In many ways, the performance of an equation-error adaptive IIR filter is like that of an adaptive FIR filter (where $A(n,q) = 0$). They have similar adaptive algorithms with similar convergence properties; the convergence rate and stability of the coefficient updates are usually determined by the eigenvalues of the Hessian matrix [Wi76]. The main difference is that the equation-error adaptive IIR filter can operate as a *pole-zero model* by copying and inverting the polynomial $1 - A(n,q)$. The adaptive FIR filter is strictly an all-zero model since $A(n,q) = 0$.

Equation (1) can also be compactly written as the inner product

$$y_e(n) = \theta^T(n)\phi_e(n), \tag{4}$$

where the coefficient vector θ and the signal vector ϕ_e each have length $M + N - 1$ and are defined as

$$\theta(n) = [a_1(n), \ldots, a_{N-1}(n), b_0(n), \ldots, b_{M-1}(n)]^T \tag{5a}$$

$$\phi_e(n) = [d(n-1), \ldots, d(n-N+1), x(n), \ldots, x(n-M+1)]^T. \tag{5b}$$

Observe that (4) has the form of a *linear regression*, which is commonly used in statistics [Mo77], where θ corresponds to the estimated parameters and ϕ_e is the regression vector (containing the data). The regressor is clearly independent of the coefficients since the data $d(n)$ and $x(n)$ are not functions of $A(n,q)$ or $B(n,q)$. Many of the techniques and algorithms used in parametric statistical inference can be used here to find the optimal set of parameters. Some examples of these estimation methods are maximum likelihood [Mn87], maximum a posteriori [Mn87], least squares [Ha86], and mean-square error [Wi85]. The RLS (recursive-least-squares) algorithm [Ha86] is one approach that recursively minimizes a least-squares criterion; the LMS (least-mean-square) algorithm [Wi76] is a recursive gradient-descent method that searches for the minimum of the MSEE.

[2] In the literature, functions of the delay operator are often written as $A(n, q^{-1})$. To simplify the notation and to make it consistent with the usual definition of the z-transform, we will use the representations defined in (3).

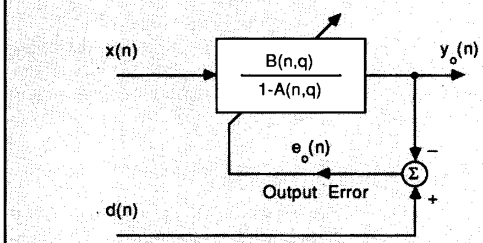

Figure 4. Output-Error Formulation. The output-error formulation filters only the input $x(n)$ to generate an estimate of $d(n)$. Unlike the equation-error formulation, the pole-polynomial $1 - A(n, q)$ is adapted directly in an IIR filter form. In a sense, this formulation is a "natural" generalization of the adaptive FIR filter where $A(n, q) = 0$.

Output-error formulation

The output-error adaptive IIR filter shown in Fig. 4 is implemented in direct form and is characterized by the following *recursive* difference equation (the subscript *o* denotes the output-error approach):

$$y_o(n) = \sum_{m=1}^{N-1} a_m(n)y_o(n-m) + \sum_{m=0}^{M-1} b_m(n)x(n-m), \tag{6}$$

which depends on past output samples $y_o(n-m)$, $m = 1, \ldots, N-1$. This output feedback significantly influences the form of the adaptive algorithm, adding greater complexity compared to that of the equation-error approach. Analogous to (2) and (4), this expression can be rewritten as the filter

$$y_o(n) = \left(\frac{B(n,q)}{1 - A(n,q)}\right)x(n), \tag{7}$$

and as the inner product

$$y_o(n) = \theta^T(n)\phi_o(n), \tag{8}$$

where the coefficient vector θ is given in (5a) and the signal vector in this case is

$$\phi_o(n) = [y_o(n-1), \ldots, y_o(n-N+1), x(n), \ldots, x(n-M+1)]^T. \tag{9}$$

The output $y_o(n)$ is clearly a *nonlinear* function of θ because the delayed output signals $y_o(n-k)$ of ϕ_o depend on previous coefficient values [i.e., they depend on $A(n-k,q)$ and $B(n-k,q)$].[3] Equation (8) is not a linear regression, but it has the same form as (4) and is often referred to as a *pseudolinear regression* [Lj83]. Similar statistical techniques and algorithms can be applied here to solve for the optimal coefficients, but it can be shown that the solution may be suboptimal unless a certain transfer function is strictly positive real (SPR) [La79]. To overcome this SPR condition, additional processing (filtering) of the regression vector or of the output error is generally necessary, as will be shown later.

The output error is given by $e_o(n) = d(n) - y_o(n)$, and it is called this simply because it is generated by subtracting the output in (7) from $d(n)$. Clearly, $e_o(n)$ is also a nonlinear function of θ; the mean-square-output error (MSOE) is, therefore, not a quadratic function and it can have multiple local minima [St81]. Adaptive algorithms that are based on gradient-search methods could converge to one of these local solutions, resulting in suboptimal performance and inaccurate estimates of θ_*.

Coefficient bias and local minima

As mentioned above, algorithms based on the equation-error formulation may converge to a result that is biased away from the optimal (Wiener) solution. In a system identification application, this corresponds to incorrect estimates of θ_* such that $E[\theta(n)] = \theta_* + bias$ in the limit as $n \to \infty$. It can be shown that this bias will be zero if either the additive noise signal $v(n) = 0$ or $A(n,q) = 0$ (corresponding to an FIR filter). Note that the equation error can be expressed as a filtered version of the output error (compare Figs. 3 and 4): $e_e(n) = [1 - A(n,q)]e_o(n)$. The equation-error and output-error formulations are obviously identical if $A(n,q) = 0$, and the amount of bias is directly influenced by the power of $v(n)$ because $v(n)$ is also filtered by $1 - A(n,q)$. In effect, the adaptive filter is attempting to minimize the noise power that reaches $e_e(n)$, in addition to identifying the system poles. These conflicting goals are the cause of the bias. Although equation-error adaptive IIR filters have rapid convergence properties, their performance may be completely unsatisfactory if the bias becomes significant. An example of this is shown in Fig. 5.

Adaptive algorithms based on the output-error formulation are generally more complicated than those based on the equation error, but they do not lead to biased solutions. However, they may converge to a local minimum of the MSOE surface [St81]. An example of a local minimum for a first-order adaptive filter is shown in Fig. 6. Sufficient conditions for which there are no local minima have been investigated for the system identification configuration [So75, So82, Na88b]. It can be shown that none exist if: (1) the adaptive filter transfer function has sufficient order (poles and zeros) to exactly model the unknown system (the order of the adaptive filter can be greater than that of the unknown system), (2) the input $x(n)$ is a white-noise sequence, and (3) the order of the adaptive filter numerator exceeds that of the unknown system denominator. For configurations other than system identification, there may be a similar set of conditions but there are relatively few analytical results. It should be mentioned that noise in the adaptive filter, such as that introduced by the gradient estimate, may induce the adaptive algorithm to escape from a local minimum and then possibly converge to the global minimum. It is not clear how often this actually occurs in practice.

[3] This dependence on the coefficients is often emphasized by writing the signal vector as $\phi_o(n, \theta)$. To keep the notation simple in this paper, we do not include θ in the argument of ϕ_o; the subscript *o* essentially serves the same purpose.

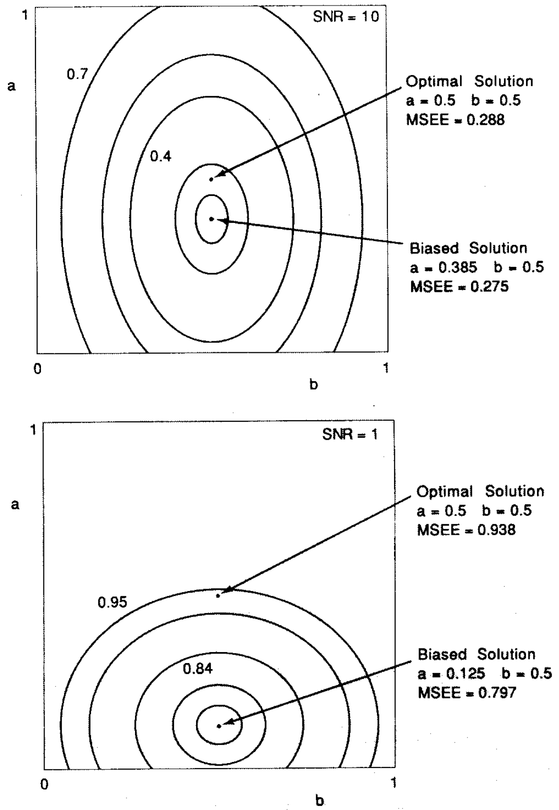

Figure 5. Coefficient Bias. These contours of constant MSEE (normalized to have a maximum value of 1) illustrate the coefficient bias that can occur when using the equation-error formulation. A system identification configuration was examined with the following first-order system:

$$H_*(z) = \frac{\beta}{1 - \alpha z^{-1}},$$

where $\alpha = \beta = 0.5$. The adaptive filter was also first-order such that $y_e(n) = aq^{-1}d(n) + bx(n)$. The input $x(n)$ and additive noise $v(n)$ were uncorrelated, white sequences with a signal-to-noise ratio (SNR $= \sigma_x^2/\sigma_v^2$) of (a) SNR = 10 and (b) SNR = 1. It is straightforward to show that the equation-error estimates of the system coefficients are $a = s \cdot \alpha$ and $b = \beta$, where the scale factor

$$s = \frac{\beta^2 \mathrm{SNR}}{1 - \alpha^2 + \beta^2 \mathrm{SNR}}.$$

In this case, only a is biased; the amount of bias, defined by $a = \alpha + bias$, is given by

$$bias = \frac{-(1 - \alpha^2)\alpha}{1 - \alpha^2 + \beta^2 \mathrm{SNR}}.$$

As the SNR decreases the bias increases ($s \rightarrow 0$), shifting the contours away from the optimal solution of $a = b = 0.5$. The noise power contained in $y_e(n)$ is given by $(1 + a^2)\sigma_v^2$. Clearly, this is minimized when $a = 0$, a result that is obtained in the limit as SNR $\rightarrow 0$. This illustrates the trade-off between minimizing the noise power that is contained in $e_e(n)$, and correctly estimating the unknown coefficient α. Observe that the contours have the form of an ellipse; the MSEE surface is a paraboloid with a single (biased) global minimum and no local minima.

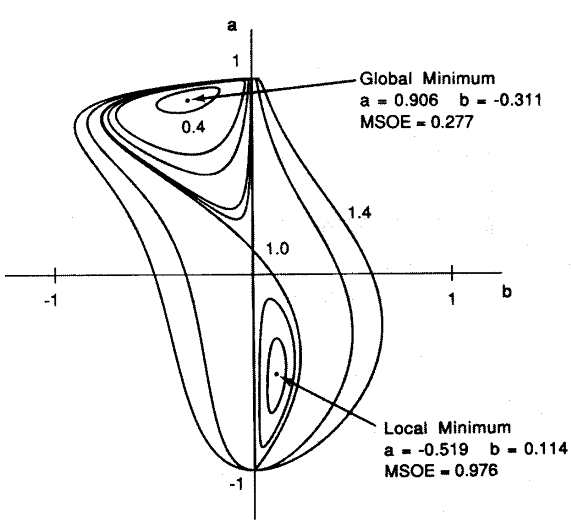

Figure 6. Local Minima. The above contours of constant MSOE were generated from a system identification configuration where the unknown system was second-order:

$$H_*(z) = \frac{\beta_0 + \beta_1 z^{-1}}{1 - \alpha_1 z^{-1} - \alpha_2 z^{-2}},$$

with $\alpha_1 = 1.1314$, $\alpha_2 = -0.25$, $\beta_0 = 0.05$, $\beta_1 = -0.4$, and the adaptive filter was first-order such that

$$y_o(n) = \left(\frac{b}{1 - aq^{-1}}\right) x(n).$$

The input signal $x(n)$ was a white sequence with variance $\sigma_x^2 = 1$, and the additive noise $v(n) = 0$. It can be shown that [Jo77]

$$\mathrm{MSOE} = \sigma_d^2 - 2bH_*(a^{-1}) + \frac{b^2}{1 - a^2},$$

where σ_d^2 is the variance of the desired response $d(n)$, and $H_*(z)$ has been evaluated at $z = a^{-1}$. Observe that this expression is a highly nonlinear function of the feedback coefficient a. As a result, the MSOE surface generally is not a paraboloid and it can have local minima (as shown in the figure). If the coefficients are initialized near a local minimum, then the adaptive IIR algorithm might converge to this suboptimal solution (see [Jo77]).

The initial conditions for $\theta(n)$ can also influence to which minimum the algorithm will converge; it is obviously desirable that $\theta(0)$ be near or lie on a trajectory to the global minimum.

Clearly, there is a trade-off between the two error formulations. On the one hand, the equation error is a linear function of the coefficients so that the MSEE surface has only a global minimum and no local minima. The adaptive algorithms generally have fast convergence, but they converge to a biased solution since there is always some additive noise. On the other hand, the output error is a nonlinear function of the coefficients and the MSOE surface can have multiple local minima. The corresponding adaptive algorithms usually converge more slowly and they may converge to a local minimum. One might argue, however, that the output-error formulation is the "correct" approach because the adaptive filter is operating only on $x(n)$ such that the output $y(n)$ estimates the desired response $d(n)$. In contrast, the equation-error approach uses past values of the desired response as well as $x(n)$ to estimate $d(n)$.

Alternative error formulations

There has also been a significant amount of research on other error formulations that are based on more general system models such as the ARMAX and Box-Jenkins models [Lj87]. These are often used in system identification applications, and they provide a more complete description of the additive noise process $v(n)$. This paper is primarily concerned with models that make sense in the framework of adaptive filtering where there may not be an underlying system, such as in the adaptive notch filter. The equation-error and output-error formulations are two special cases of these more general models that are useful for adaptive filtering purposes. Further discussion of other model formulations can be found in [Lj83] and [Lj87].

GENERAL FORM OF THE ADAPTIVE ALGORITHMS

All of the adaptive IIR algorithms discussed in this paper have the following generic form which is often called the recursive Gauss-Newton algorithm [Lj83]:

$$\theta(n+1) = \theta(n) + \alpha R^{-1}(n+1)\phi_f(n)e_f(n), \qquad (10)$$

where $\phi_f(n)$ and $e_f(n)$ are *filtered* versions of $\phi(n)$ and $e(n)$, respectively, according to $\phi_f(n) = F(n,q)\phi(n)$ and $e_f(n) = G(n,q)e(n)$. The filtered regression (signal) vector is often called the *information vector,* and it can be shown that $E[\phi_f(n)\phi_f^T(n)]$ is the Fisher information matrix used in statistical inference [Lh83]. The information matrix determines the Cramer-Rao lower bound[4] [Lj83, Fr84b] for the coefficients in (10). Note that ϕ and e are derived from either the equation-error (ϕ_e, e_e) or output-error (ϕ_o, e_o) formulations. The time-varying filters $F(n,q)$ and $G(n,q)$ are defined in a manner similar to $B(n,q)$ in (3). The scalar (positive) step size α controls the algorithm convergence rate and $R(n)$ is an estimate of the Hessian matrix updated according to

$$R(n+1) = \lambda R(n) + \alpha\phi_f(n)\phi_f^T(n), \qquad (11)$$

where $\lambda = 1 - \alpha$ is the so-called forgetting factor. Typically, λ would have a value between 0.9 and 0.99, corresponding to an effective memory[5] between 10 samples and 100 samples, respectively. Since computing the inverse of R is computationally expensive, R^{-1} is generally updated directly using the matrix inversion lemma [Ka80]. In this case we have

$$R^{-1}(n+1) = \frac{1}{\lambda}\left(R^{-1}(n) - \frac{R^{-1}(n)\phi_f(n)\phi_f^T(n)R^{-1}(n)}{\lambda/\alpha + \phi_f^T(n)R^{-1}(n)\phi_f(n)}\right). \qquad (12)$$

The (estimated) Hessian matrix is often incorporated to improve the algorithm convergence rate but at the expense of an increase in the computational complexity. Otherwise, the update in (11) or (12) is not performed and $R(n+1)$ in (10) is set equal to I, the identity matrix. This form, often referred to as a stochastic gradient algorithm [Ha86], generally has a slower convergence. However, its complexity is much less than that of (10)—on the order of $M + N$ operations compared to order $(M + N)^2$.

The algorithm is usually initialized with $\theta(0) = \underline{0}$ and $R(0) = \delta I$ where $\underline{0}$ is a vector of zeros and δ is a small positive scalar. If some a priori information is known in a particular application, it may be desirable to initialize the parameters to some other appropriate values. In order for (10) to converge, it is important that R always be positive definite (so that it is invertible), and that the poles of $1 - A(z)$ always lie inside the unit circle (so that the filter is stable). The above initial conditions satisfy these two requirements, but it may be necessary to monitor θ and R after each update of the algorithm. This depends on the filters chosen for F and G.

For the equation-error formulation, it can easily be shown that $\phi_f(n) = \phi_e(n)$ and $e_f(n) = e_e(n)$ so that $F(n,q) = G(n,q) = 1$; i.e., there is no filtering of the regressor or of the equation error. The corresponding algorithm is known as the normalized-gain RLS algorithm [Lj87], and if $R = I$ it is simply the LMS algorithm. The properties of these algorithms have been studied extensively in the context of adaptive FIR filtering (see, for example, [Wi76, Hr81, Ha86]), and they extend in a straightforward manner to the equation-error formulation. A convergence analysis of the equation-error algorithm can easily be done by assuming that ϕ_e and θ are independent (which is often done for the LMS algorithm [Wi85][6]). This assumption cannot be used, however, for the output-error algorithms described later in this paper (where F and G may not be equal to 1). In that case ϕ_o

[4]The Cramer-Rao lower bound is the minimum achievable covariance of $\theta(n)$ at convergence; i.e., it determines the lower limit of $E[\theta(n) - \theta_*][\theta(n) - \theta_*]^T$ assuming that $\theta(n)$ is unbiased: $E[\theta(n)] = \theta_*$.

[5]The effective memory is defined as $\tau = 1/(1 - \lambda)$.

[6]Although the independence assumption is not strictly valid, it is "approximately" correct for slow adaptation (small α), and it yields results that accurately predict the convergence properties of the LMS algorithm.

and θ are not independent because the regressor contains the filter output y_o, which is a function of the coefficients. Note that the correlation matrix $E[\phi_e(n)\phi_e^T(n)]$ of the equation-error formulation is independent of time when $x(n)$ and $d(n)$ are wide-sense stationary random processes. It is this property that greatly simplifies the convergence analysis of equation-error adaptive IIR filters. On the other hand, the correlation matrix $E[\phi_o(n)\phi_o^T(n)]$ of the output-error formulation is *not* independent of time—even for a stationary input signal because ϕ_o depends on the time-varying filter coefficients. This property complicates the convergence analysis of output-error adaptive IIR filters.

Because the equation-error formulation is closely related to adaptive FIR filtering, it will not be discussed further in this paper. A more thorough discussion of equation-error algorithms and their properties can be found in [Mn73, Gc83, Lj83]. A recently developed algorithm that exhibits properties of both the equation-error and output-error formulations is described in [Fa86]. The rest of the paper presents some adaptive algorithms for the output-error formulation.

GRADIENT-BASED METHODS

Recursive prediction error algorithm

The recursive prediction error (RPE) adaptive algorithm [Lj83] adjusts the filter coefficients to minimize the MSOE cost function $\xi = E[e_o^2(n)]$ where $e_o(n)$ is the output error. Because ξ is generally unknown or the signals are nonstationary, the algorithm is designed to minimize at each instant of time an *instantaneous* estimate of ξ given by $\zeta(n) = e_o^2(n)$. In effect, we are using the current sample of the output error as an estimate of the ensemble. Clearly, this approximation will lead to noisy estimates of the filter parameters, but it can be shown that they are *unbiased* so that the average convergence properties are comparable to those that would be observed if ξ was available.

The RPE algorithm updates the coefficient vector $\theta(n)$ along the (negative) gradient of $\zeta(n)$, defined as[7]

$$\nabla_\theta \zeta(n) \equiv \frac{\partial \zeta(n)}{\partial \theta(n)} = e_o(n)\nabla_\theta e_o(n) = -e_o(n)\nabla_\theta y_o(n)\,, \tag{13}$$

where the gradient $\nabla_\theta y_o(n)$ is the following column vector:

$$\nabla_\theta y_o(n) = \left[\frac{\partial y_o(n)}{\partial a_1(n)}, \ldots, \frac{\partial y_o(n)}{\partial a_{N-1}(n)}, \frac{\partial y_o(n)}{\partial b_0(n)}, \ldots, \frac{\partial y_o(n)}{\partial b_{M-1}(n)}\right]^T. \tag{14}$$

The last expression in (13) follows from the definition of the output error and from the fact that $d(n)$ is independent of the coefficients. Observe from (8) that $\nabla_\theta y_o(n) = \nabla_\theta[\theta^T(n)\phi_o(n)]$. If ϕ_o was independent of θ, then the gradient would simply be $\nabla_\theta y_o(n) = \phi_o(n)$ [which is analogous to the gradient in the equation-error formulation, i.e., $\nabla_\theta y_e(n) = \phi_e(n)$]. However, ϕ_o contains delayed samples of y_o which depend on θ according to (6). As a result, the gradient expression is considerably more complicated, as we now show; the simplified form, $\nabla_\theta y_o(n) = \phi_o(n)$, is only an approximation and the corresponding algorithm is known as a pseudolinear regression algorithm (which is discussed in the next section). Taking the derivative of both sides of (6) with respect to $a_k(n)$ and noting that $b_m(n)$ and $x(n-m)$ are independent of $a_k(n)$, we obtain

$$\frac{\partial y_o(n)}{\partial a_k(n)} = y_o(n-k) + \sum_{m=1}^{N-1} a_m(n)\frac{\partial y_o(n-m)}{\partial a_k(n)}. \tag{15a}$$

Similarly, the derivative with respect to $b_k(n)$ is

$$\frac{\partial y_o(n)}{\partial b_k(n)} = x(n-k) + \sum_{m=1}^{N-1} a_m(n)\frac{\partial y_o(n-m)}{\partial b_k(n)}. \tag{15b}$$

The partial derivatives on the right-hand side of (15) arise because the filter in (6) has feedback, whereby previous output samples depend on previous coefficient values which, in turn, are related to the current coefficient values via successive updates of the algorithm in (10). Observe that these derivatives are with respect to the *present* values of a_k and b_k so that the expressions are not recursive and cannot be expressed in the form of a filter using the delay-operator notation. However, if the step size α is chosen sufficiently small so that the coefficients adapt slowly, then the following approximation can be made: $\theta(n) \approx \theta(n-1) \approx, \ldots, \approx \theta(n-N+1)$ [Jo84]. This is a reasonable assumption in many applications, particularly when N is small; in cases where this assumption is not valid only a small degradation in the performance of the RPE algorithm is observed in practice. Consequently, we can replace (15) by

$$\frac{\partial y_o(n)}{\partial a_k(n)} \approx y_o(n-k) + \sum_{m=1}^{N-1} a_m(n)\frac{\partial y_o(n-m)}{\partial a_k(n-m)} = \left(\frac{1}{1-A(n,q)}\right)y_o(n-k) \tag{16a}$$

and

$$\frac{\partial y_o(n)}{\partial b_k(n)} \approx x(n-k) + \sum_{m=1}^{N-1} a_m(n)\frac{\partial y_o(n-m)}{\partial b_k(n-m)} = \left(\frac{1}{1-A(n,q)}\right)x(n-k)\,. \tag{16b}$$

These derivatives are now recursive in the partial derivatives since terms in the middle expression correspond to delayed versions of the left-hand side. This observation leads to the delay-operator form in the right-hand expression where each component of the signal vector ϕ_o is filtered by the inverse pole-polynomial of the adaptive filter—a result that is characteristic of adaptive IIR filtering. This additional processing does not arise in adaptive FIR filtering or the equation-error formulation because

[7] In order to obtain the convenient form in (13), a factor of one-half has been included in the definition of $\zeta(n)$.

there is no feedback.

We now see that $F(n,q) = 1/[1 - A(n,q)]$ and $G(n,q) = 1$; the complete algorithm is thus given by

$$\theta(n+1) = \theta(n) + \alpha R^{-1}(n+1)\left(\frac{1}{1 - A(n,q)}\right)\phi_o(n)e_o(n), \tag{17}$$

where the all-pole filter F operates on each component of the signal-vector ϕ_o, as shown in Fig. 7. Observe that there are $M + N - 1$ identical filters operating in parallel to compute the gradient components. They are generated by copying the feedback coefficients from the adaptive filter after each update of θ. These filters differ only in that they are driven by different input signals, corresponding to delayed samples of the adaptive filter input and output. This algorithm, with $R = I$, was first derived in the context of adaptive filters by White [Wh75]. Similar algorithms were derived earlier for other system identification models, such as the recursive maximum likelihood (RML) algorithm developed by Åström and Söderström [As74].

Simplified RPE algorithm

It is clear that computing the components of ϕ_f contributes a significant amount of complexity to the RPE algorithm, and that a large amount of storage is needed for past values of them. Fortunately, the coefficient approximation described previously allows a considerable simplification in the calculation of ϕ_f as shown in Fig. 8. To see this, first define

$$y_f(n-1) \equiv \frac{\partial y_o(n)}{\partial a_1(n)} \quad \text{and} \quad x_f(n) \equiv \frac{\partial y_o(n)}{\partial b_0(n)}, \tag{18}$$

which are the initial gradient terms in (16a) and (16b) for $k = 1$ and $k = 0$, respectively. Then, for the other components of ϕ_f, the approximation permits us to substitute

$$\frac{\partial y_o(n)}{\partial a_k(n)} \approx y_f(n-k) \quad \text{and} \quad \frac{\partial y_o(n)}{\partial b_k(n)} \approx x_f(n-k) \tag{19}$$

for $k = 2, \ldots, N-1$ and $k = 1, \ldots, M-1$, respectively. That is, each component of the gradient is simply a delayed version of one of the two initial components defined in (18). As a result, $y_f(n-k)$ and $x_f(n-k)$ depend

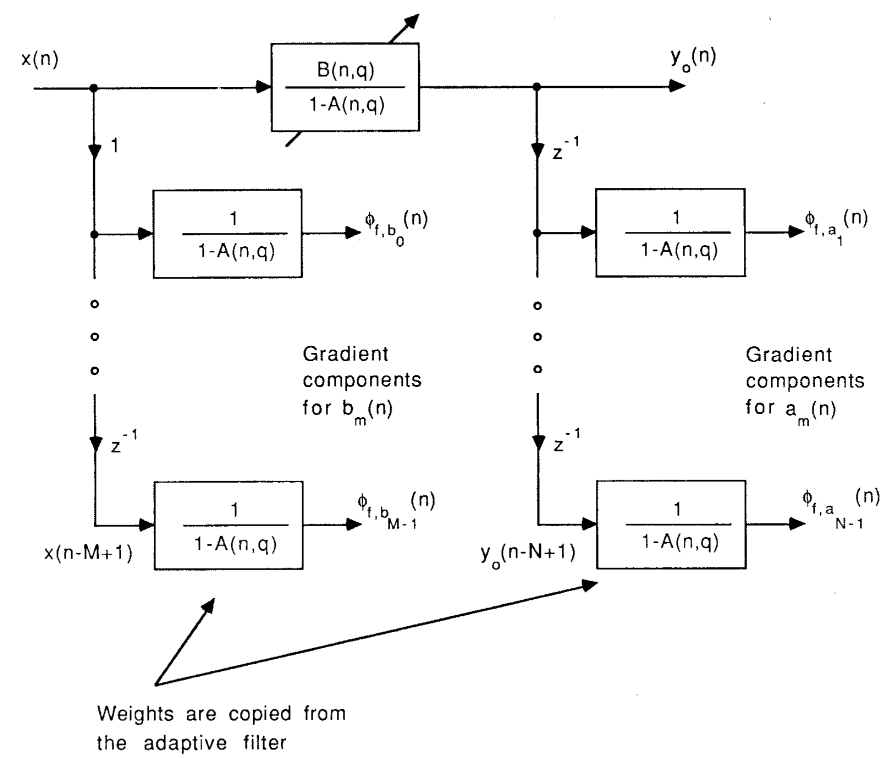

Figure 7. Full Gradient. Each component of the gradient vector $\phi_f(n)$ is obtained by filtering either $x(n-k)$ or $y(n-k)$ by the inverse of the pole polynomial $1 - A(n,q)$. For example, the first gradient element, denoted by $\phi_{f,b_0}(n)$, corresponds to the derivative of $y_o(n)$ with respect to the feedforward coefficient $b_0(n)$; it is generated by filtering $x(n)$ with the inverse polynomial. Similarly, $\phi_{f,a_1}(n)$ is computed by filtering $y_o(n-1)$. Observe that the output $y_o(n)$ is first delayed before generating the gradient components for the feedback coefficients.

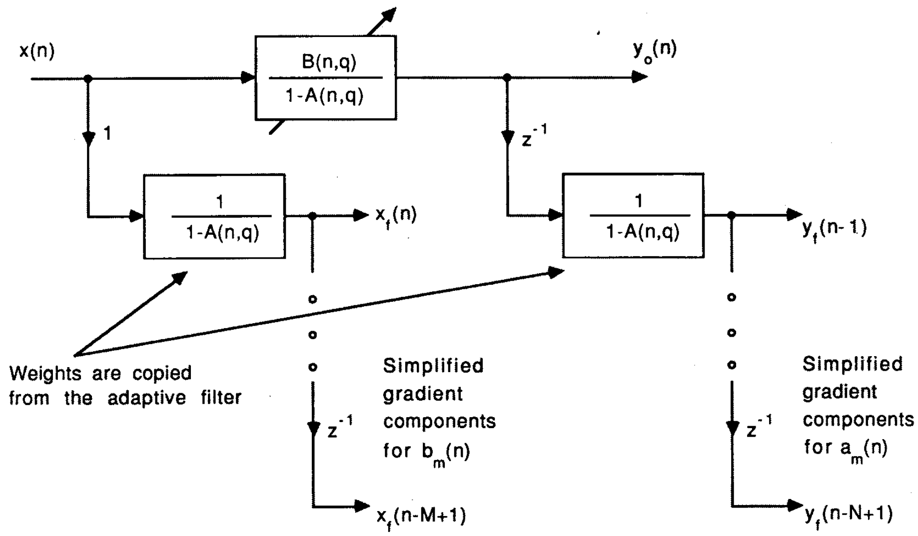

Figure 8. Simplified Gradient. The simplified gradient is obtained by observing that the gradient components in Fig. 7 differ only in that the input signals of the inverse filters are delayed versions of either $x(n)$ or $y_o(n)$. This suggests that the gradient components might approximately be delayed versions of each other. If the coefficients are slowly varying, we can substitute $A(n-k,q)$ for $A(n,q)$, where the argument $n-k$ matches that of the inverse filter input signal [either $x(n-k)$ or $y_o(n-k)$]. As a result,

$$\phi_{f,b_k}(n) \approx \frac{x(n-k)}{1-A(n-k,q)} = \phi_{f,b_0}(n-k) \equiv x_f(n-k)$$
$$k = 0,\ldots,M-1$$

and

$$\phi_{f,a_k}(n) \approx \frac{y_o(n-k)}{1-A(n-k,q)} = \phi_{f,a_1}(n-k) \equiv y_f(n-k)$$
$$k = 1,\ldots,N-1 .$$

This is equivalent to the simplified structure shown above where only two inverse filters are needed to compute all of the gradient components.

on the *delayed* coefficients $A(n-k,q)$ instead of the most recent ones $A(n,q)$ (compare Figs. 7 and 8). We can therefore replace the previous information vector with

$$\phi_f(n) = [y_f(n-1),\ldots,y_f(n-N+1), \quad x_f(n),\ldots,x_f(n-M+1)]^T, \tag{20}$$

which requires only two filters [Hv80]. In effect, the delay lines used to generate the components of ϕ_f in Fig. 7 have been moved "through" the pole-polynomial filters so that only one such filter is needed for each delay line. This simplification introduces essentially no degradation in performance and is generally used in practice. The resulting simplified RPE algorithm is thus (10) coupled with (18) and (20), which clearly requires less storage and computation than using (17). For convenience, the complete algorithm with initial conditions is summarized in Table 1. A generalization of this algorithm for complex coefficients is derived in [Sh86].

Stability monitoring

One of the major drawbacks associated with the RPE algorithm is that the pole polynomial of the filter, which is also used to compute the gradient components, may become unstable during adaptation. This can occur if one or more poles of $1-A(z)$ accidentally updates outside the unit circle and remains there for a significant length of time. As a result, the output can grow without bound and it is difficult for the algorithm to overcome this problem unless there is some form of stability monitoring. Because of the noisy gradient estimate (i.e., $\zeta(n)$ is used instead of ξ), it is not unlikely that the poles will update outside the unit circle, particularly if the application requires that they be near the unit circle. Fig. 9 shows the stable region, called the *stability triangle,* for the feedback coefficients of a second-order adaptive filter.

One of the simplest tests of stability is to check after each update of the algorithm that the sum of $|a_m(n)|$ is less than 1 [Hv76]. All unstable updates will be detected by this approach, but it can be shown that the coefficient space is severely restricted (see Fig. 9), especially for large N; i.e., the test will indicate that a polynomial is unstable when in fact it is not. Jury's test [Ju64] (also called the modified Schur-Cohn test [Tt76]) is a somewhat more complex method of determining whether or not a polynomial has minimum phase (all roots lie inside the unit circle), and it does not restrict the coefficient space. This test does not reveal *which* poles are unstable; to obtain

TABLE 1
RPE Algorithm with Simplified Gradient

INITIALIZATION:

$$a_m(0) = b_m(0) = 0$$

$$x(n-m) = y_o(n-m) = x_f(n-m) = y_f(n-m) = 0$$

$$R(0) = \delta I$$

VECTOR DEFINITIONS:

$$\theta(n) = [a_1(n), \ldots, a_{N-1}(n), b_0(n), \ldots, b_{M-1}(n)]^T$$

$$\phi_o(n) = [y_o(n-1), \ldots, y_o(n-N+1), x(n), \ldots, x(n-M+1)]^T$$

$$\phi_f(n) = [y_f(n-1), \ldots, y_f(n-N+1), x_f(n), \ldots, x_f(n-M+1)]^T$$

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

$$x_f(n) = x(n) + \sum_{m=1}^{N-1} a_m(n) x_f(n-m)$$

$$y_o(n) = \theta^T(n)\phi_o(n)$$

$$y_f(n) = y_o(n) + \sum_{m=1}^{N-1} a_m(n) y_f(n-m)$$

$$e_o(n) = d(n) - y_o(n)$$

$$R^{-1}(n+1) = \frac{1}{\lambda}\left(R^{-1}(n) - \frac{R^{-1}(n)\phi_f(n)\phi_f^T(n)R^{-1}(n)}{\lambda/\alpha + \phi_f^T(n)R^{-1}(n)\phi_f(n)}\right)$$

$$\theta(n+1) = \theta(n) + \alpha R^{-1}(n+1)\phi_f(n)e_o(n)$$

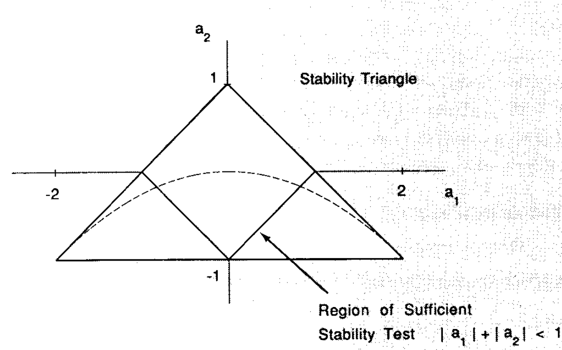

Figure 9. Stability Triangle. The stable region for the second-order IIR filter

$$H(z) = \frac{1}{1 - a_1 z^{-1} - a_2 z^{-2}}$$

is shown as a function of the feedback coefficients a_1 and a_2. This region is called the *stability triangle*, and it can easily be shown that the poles lie inside the unit circle if $1 + a_1 - a_2 > 0$, $1 - a_1 - a_2 > 0$, and $1 + a_2 > 0$. This is a simple stability test that can be applied to second-order adaptive IIR filters, such as the second-order sections of the parallel form shown in Fig. 12. It is also easy to determine the pole locations exactly, and to project any unstable roots back inside the unit circle to some appropriate location. The region corresponding to the sufficient stability test, $|a_1| + |a_2| < 1$, is also shown. Observe that it is significantly smaller than the entire stable region and, in particular, that it excludes the lower right corner of the triangle which corresponds to the region near $z = 1$ on the z-plane. Note that in the region below the dashed curve the poles form a complex conjugate pair.

this information the polynomial must be factored—a computationally expensive operation for $N > 2$. If this were done, any unstable poles could easily be projected back inside the stable region to some appropriate location. In most cases, however, factorization is not practical, so instead the previous coefficient update is usually ignored, i.e., $\theta(n+1) = \theta(n)$. Of course, this approach requires much less complexity than factoring the polynomial, but it is not a robust method because the algorithm can lock up in this state for an indefinite period of time [Jo84], thereby degrading the convergence rate and the overall performance.

Other approaches to stability monitoring have been suggested, but they are either computationally expensive or nonrobust. The problem is still an ongoing area of research. A recent method is based on Kharitonov's theorem [Br88] which requires that four related polynomials be tested to ascertain the stability of the pole polynomial. The complexity of this method is relatively high, and it has been shown that it also restricts the size of the coefficient space. Other recent work has shown that alternative realizations such as the parallel and lattice forms can easily overcome potential instabilities. The parallel form is comprised of second-order sections which are trivial to factor, and the lattice form requires only that each reflection coefficient have a magnitude less than 1.

Finally, it should be noted that when the poles lie inside the unit circle, the filter is guaranteed to be stable only when it is linear and time invariant. For time-varying systems such as the adaptive IIR filter, it is not sufficient to monitor the poles at discrete time intervals; even if the poles always lie inside the unit circle it is possible for the system to become unstable for certain "pathological" input signals [Vi78]. This potential problem is often ignored in practice and is usually not observed in computer simulations.

APPROXIMATE GRADIENT METHODS

Pseudolinear regression algorithm

The RPE algorithm can be simplified further by using an *approximate gradient* such that $F(n,q) = G(n,q) = 1$ and (17) becomes

$$\theta(n+1) = \theta(n) + \alpha R^{-1}(n+1)\phi_o(n)e_o(n). \qquad (21)$$

There is no filtering of the signal vector ϕ_o and the gradient is approximated by $\nabla_\theta y_o(n) \approx \phi_o(n)$. This approach is known as a pseudolinear regression (PLR) algorithm because the adaptive filter output signal is a nonlinear function of the coefficients [see (7)], yet the algorithm (gradient) ignores the fact that the signal (regression) vector depends on the coefficients (i.e., the product rule of differentiation was not applied when calculating the gradient, as in the previous section). Observe that the PLR algorithm is similar in form to the RLS algorithm used in the equation-error formulation except that (ϕ_o, e_o) are substituted for (ϕ_e, e_e). The computational complexity and storage requirements of the PLR algorithm are comparable to that of the RLS algorithm, and they are clearly less than that of the RPE algorithm.

The adaptive algorithm in (21) (with $R = I$) was first derived for adaptive filters by Feintuch [Fe76] and for system identification applications by Landau [La76]. Although satisfactory convergence results have been achieved for specific cases [Fe76], it was shown later that (21) may not converge to a minimum (local or global) of the MSOE surface unless the denominator polynomial associated with θ_*, denoted by $1 - A_*(q)$, satisfies a *strictly positive real* (SPR) condition. If this condition (defined below) is not satisfied, the algorithm may converge to an arbitrary point on the MSOE surface and the overall performance may be unacceptable. This has been demonstrated in a system identification configuration where the order of the adaptive filter was less than that of the unknown system [Jo77].

Unlike the RPE algorithm, stability monitoring is not needed here [Lj83]. In effect, the PLR algorithm has a self-stabilizing feature whereby unstable poles have a tendency to migrate back into the stable region. It has been suggested that when RPE updates become unstable, PLR updates could be done instead [i.e., set $F = 1$ in (17)] until the filter is again stable. This technique may result in improved performance compared to that obtained when unstable RPE updates are simply skipped.

Filtered-error algorithm

To overcome the convergence problem associated with the SPR condition, the output error in (21) can be filtered according to $[1 + C(n,q)]e_o(n)$, resulting in the following filtered-error (FE) algorithm (see Fig. 10):

$$\theta(n+1) = \theta(n) + \alpha R^{-1}(n+1)\phi_o(n)[1 + C(n,q)]e_o(n). \tag{22}$$

If C is chosen such that the SPR condition is satisfied, (22) will have *global* convergence, as described later. This algorithm has the same general form as that in (10) with $\phi_f(n) = \phi_o(n)$ and $e_f(n) = [1 + C(n,q)]e_o(n)$; i.e., $F(n,q) = 1$ and $G(n,q) = 1 + C(n,q)$. In order to have satisfactory convergence it is desirable that $1 + C \approx 1 - A_*$, a condition that is not always possible to achieve in practice since A_* is usually unknown. Table 2 summarizes the FE algorithm with the standard initial conditions.

Although we have expressed C as a time-varying filter, the coefficients are often *fixed* so that stability monitoring will not be necessary. Simultaneous adaptation of C could help to satisfy the SPR condition in a real-time manner, but the algorithm will no longer be self stabilizing [Jo84]. The same stability tests used for the RPE algorithm would be needed here. Clearly, a time-varying C offsets a desirable feature of the approximate gradient methods.

A somewhat more complicated form of the FE algo-

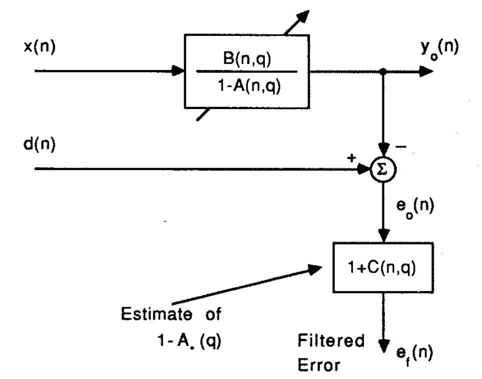

Figure 10. Filtering of the Output Error. The filtered-error (FE) algorithm is obtained from the PLR algorithm simply by filtering the output error $e_o(n)$ with the polynomial $1 + C(n, q)$, which serves as an estimate of the underlying system poles $1 - A_*(q)$. If the ratio of $1 + C(n, q)$ and $1 - A_*(q)$ is close to 1, then the SPR region associated with the unknown system nearly fills the entire stability region.

TABLE 2
PLR Algorithm with Error Filtering

INITIALIZATION:

$$a_m(0) = b_m(0) = 0$$

$$x(n-m) = y_o(n-m) = e_o(n-m) = 0$$

$$R(0) = \delta I$$

VECTOR DEFINITIONS:

$$\theta(n) = [a_1(n), \ldots, a_{N-1}(n), b_0(n), \ldots, b_{M-1}(n)]^T$$

$$\phi_o(n) = [y_o(n-1), \ldots, y_o(n-N+1), x(n), \ldots, x(n-M+1)]^T$$

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

$$y_o(n) = \theta^T(n)\phi_o(n)$$

$$e_o(n) = d(n) - y_o(n)$$

$$e_f(n) = \sum_{m=0}^{P-1} c_m(n)e_o(n-m)$$

$$R^{-1}(n+1) = \frac{1}{\lambda}\left(R^{-1}(n) - \frac{R^{-1}(n)\phi_o(n)\phi_o^T(n)R^{-1}(n)}{\lambda/\alpha + \phi_o^T(n)R^{-1}(n)\phi_o(n)}\right)$$

$$\theta(n+1) = \theta(n) + \alpha R^{-1}(n+1)\phi_o(n)e_f(n)$$

rithm is known as HARF (hyperstable adaptive recursive filter) [Jo79]. We will not discuss the HARF algorithm here except to mention that the FE algorithm can be derived from HARF if we assume that the coefficients adapt slowly, as we did to obtain the simplified RPE algorithm. In the literature, the FE algorithm is also called SHARF (simplified HARF) [Tr78, Le80]. Further discussion of HARF and SHARF can be found in [Tr85] and [Tr87].

SPR condition

In order to guarantee convergence of the PLR algorithm in (22) it is necessary that the following SPR condition be satisfied:

$$\mathrm{Re}\left(\frac{1 + C(z)}{1 - A_*(z)}\right) - \gamma > 0, \quad \text{for all } |z| = 1, \tag{23}$$

where Re (u) denotes the real part of u and the scalar $\gamma = 1/2$.[8] Note that (23) applies only to points on the unit circle. The SPR condition for the algorithm in (21) is also given by (23) with $C = 0$. That is, the pole polynomial of the unknown system must be such that (23) is valid. In most cases the SPR condition is not satisfied, as demonstrated in Fig. 11 for a second-order system. By filtering the error with $1 + C$ we can expand the SPR region to include more coefficient values, although this requires some knowledge of $1 - A_*$ as mentioned before. Recently, there has been some progress on minimizing the effects of the SPR condition by increasing the step size α [Ta87], but there is no general method yet to eliminate the condition entirely. This is the major drawback with PLR algorithms.

When $C = 0$, it can be shown that (23) with $\gamma = 1/2$ is equivalent to: $|A_*(z)| < 1$, for all $|z| = 1$. From this, we may interpret the SPR condition as a measure of how close A_* is to zero. It is an indication of when ϕ_o in (21) is a reasonable approximation to ϕ_f in (17).

The SPR condition is related to the concept of *hyperstability*, which describes the output stability of feedback systems that may have both nonlinear and time-varying components [Po73]. Hyperstability refers to the asymptotic convergence to zero of a state vector that characterizes the system [La79]. In the framework of adaptive IIR filtering, this state vector corresponds to the coefficient error vector $\tilde{\theta}(n) = \theta_* - \theta(n)$. In addition to the SPR condition, hyperstability requires certain restrictions on the data and on the adaptive filter configuration.

CONVERGENCE ANALYSIS

Stochastic ODE approach

The ODE (ordinary differential equation) approach to the convergence analysis of adaptive IIR filters is a powerful technique that requires relatively weak assumptions. Subject to certain smoothness conditions on the algorithm, a boundedness restriction on the *random* data, and other technical conditions [Lj83], it is possible to represent (10) and (11) by the following coupled pair of ODEs:

$$\frac{d}{d\tau}\theta_D(\tau) = R_D^{-1}(\tau)\, f[\theta_D(\tau)] \tag{24a}$$

$$\frac{d}{d\tau}R_D(\tau) = g[\theta_D(\tau)] - R_D(\tau), \tag{24b}$$

where the subscript D distinguishes these nonrandom recursions from the stochastic algorithm, and the update directions are defined by $f[\theta_D(\tau)] = E[\phi_f(\tau)e_f(\tau)]$ and $g[\theta_D(\tau)] = E[\phi_f(\tau)\phi_f^T(\tau)]$. The variable τ is a compressed time scale that essentially allows us to observe the asymptotic properties of (24), which on average represent the asymptotic properties of (10) and (11) (because of the expectations used to define f and g).

There have been relatively few analyses of the convergence properties of adaptive IIR filters. Most of the results are derived from work in system identification where it is often assumed that the step size $\alpha = \alpha(n)$ decreases to zero with time; that is, the adaptive algorithm eventually shuts off. These results are particularly useful for the system identification application where the unknown system is time-invariant and the signals are stationary. In this case, the algorithm in (10) will converge to a stable point of the ODE *with probability one* provided the data is asymptotically mean stationary and exponentially stable [Lj83]. Most of the results are based on work by Ljung [Lj77] and Söderström [So78].

In many adaptive filter applications, however, it is generally necessary that the algorithm be capable of tracking time variations in the system or signal statistics, such as in the equalization of communication channels. It is therefore desirable that α be a constant. There are some convergence results for this case, but they are weaker in that the algorithm converges only *in probability*. A stable range of values for α is not determined here, but it must be "sufficiently small." These results are based on work by Benveniste [Be80], Kushner [Ku84], and others [Ma83, Fa88].

Consider the case where $\alpha(n)$ decreases to zero (typically $\alpha(n) = 1/n$) and assume that R is always positive definite. This is usually satisfied if the data is sufficiently rich in frequency content.[9] Roughly speaking, there must be at least as many distinct frequency components in the data as the number of coefficients in the adaptive filter [Lj83]. If we assume that there is some method of stability monitoring to ensure that the poles remain inside the unit circle, then it is possible to prove the following convergence property of the RPE algorithm: with probability one as $n \to \infty$, $\theta(n)$ will converge to θ_s such that $\xi(\theta_s) = 0$, corresponding to a stable local minimum, or it will converge to a cluster point at the boundary of the stability region [Lj83]. Basically, the algorithm performs as expected. The average asymptotic properties of $\theta(n)$ are determined by the solutions of the ODE, and it is possible to examine these properties by studying the trajectories of the ODE

[8] For convenience in computing the SPR regions in Fig. 11 we have used $\gamma = 0$; similar results are obtained for other values of γ.

[9] In the control literature, this condition is called persistent excitation [Bi84].

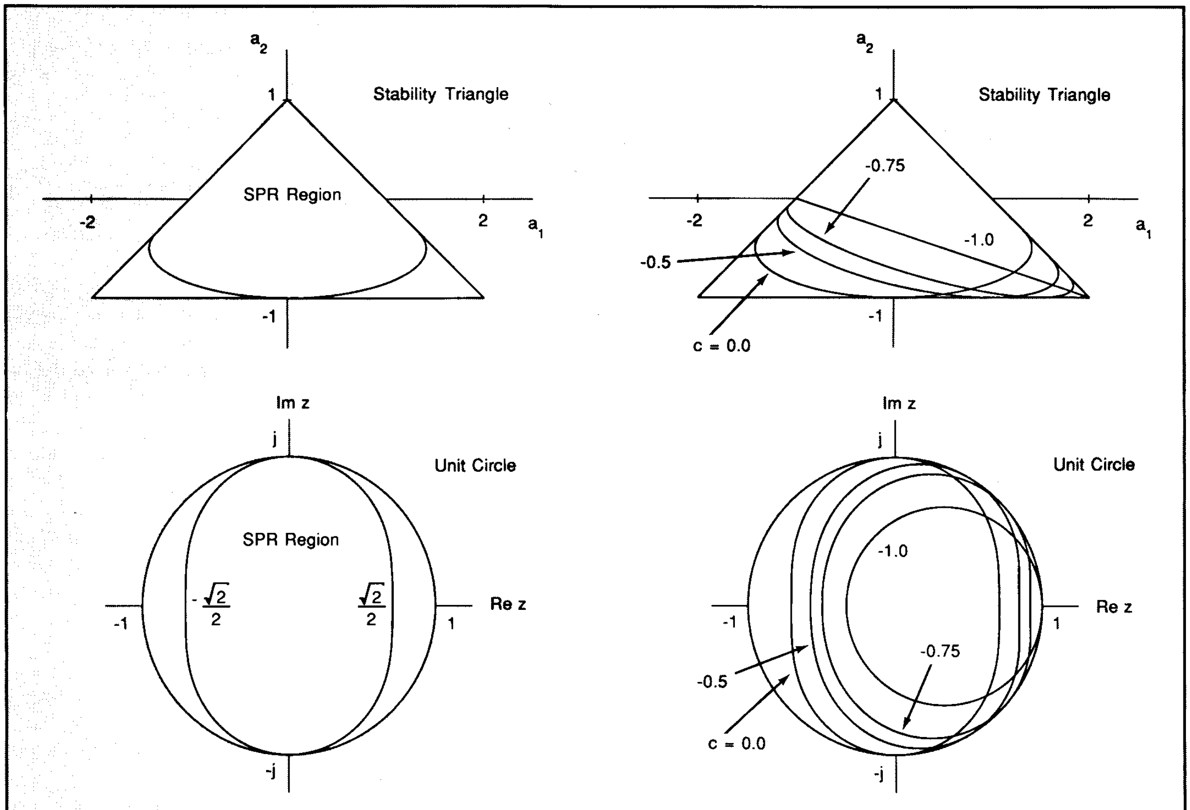

Figure 11. SPR Region. The SPR (strictly positive real) region for the second-order IIR system $H(z)$ of Fig. 9 is shown above in two ways with $\gamma = 0$ [La79, Tr87]: (a) within the stability triangle in the coefficient space, and (b) within the unit circle in the z-plane. Observe that the SPR region is somewhat smaller than the entire stable region. This result can be generalized to higher-order systems, and the conclusion is that only a subset of all systems satisfy the SPR condition. This is particularly important for oversampled systems because in that case the poles tend to cluster near $z = 1$, which is not in the SPR region. Note that the regions near $z = 1$ and $z = -1$ correspond to the lower right and lower left corners, respectively, of the stability triangle. It is possible to modify the SPR region by adding zeros. For example, consider

$$H(z) = \frac{1 + cz^{-1}}{1 - a_1z^{-1} - a_2z^{-2}}.$$

The effect of this zero is illustrated in (c) and (d) for three nonzero values of c. The SPR region lies *above* the curves in the coefficient space, and *inside* the contours in the z-plane. Note that $c = -1$ results in a straight line in the coefficient space and a circle in the z-plane which includes the region near $z = 1$.

solutions via computer simulations.

For the PLR algorithm in (21) and the FE algorithm in (22), we must assume that the adaptive filter is operating in a system identification configuration and that it has sufficient order to model the unknown system coefficients θ_*. An equivalent requirement is that there exist $H_1(q)$ and $H_2(q)$ such that we can express the filtered output error as

$$e_f(n) = H_1(q)\phi_o^T(n)[\theta_* - \theta(n)] + H_2(q)v(n), \quad (25)$$

where $H_2(q)v(n)$ is independent of $\phi_o(n)$ [i.e., $v(n)$ is independent of $x(n)$]. If we also assume that $H_1(z) - 1/2$ is SPR, then it is possible to prove the following convergence property of the PLR and FE algorithms: with probability one as $n \rightarrow \infty$, $\theta(n)$ will converge to a stable point such that $E[y_o(n) - y_*(n)]^2 = 0$, where $y_*(n) = d(n) - v(n)$ is the output of the unknown system (before the measurement noise is added). In effect, the algorithm has global convergence to θ_*. It can easily be shown that $H_1 = 1/(1 - A_*)$ and $H_2 = 1$ for the PLR algorithm, and that $H_1 = (1 + C)/(1 - A_*)$ and $H_2 = 1 + C$ for the FE algorithm. This convergence result does not require stability monitoring (for fixed C) [Lj83], but it is more restrictive than that of the RPE algorithm since it assumes the system identification configuration implied by (25).

The ODE method does not prove convergence of the algorithm to the global minimum of ξ, except when there are no local minima. Furthermore, it does not provide any information concerning the *rate* of convergence, only that asymptotically the algorithm will converge. Thus, there are some limitations to the ODE method as compared to the standard techniques used for adaptive FIR filtering. The eigenvalues of the Hessian matrix probably influence the rate of convergence, but unlike the Hessian matrix in adaptive FIR filtering, it is time-varying even when the data is stationary. Computer simulations are still the only method of studying the rate of convergence, and these can be misleading because it is known that the rate is greatly influenced by the initial conditions on θ and R. In summary, the convergence analysis of adaptive IIR filters is somewhat limited and the problem is still largely unsolved.

Hyperstability and averaging approaches

There have been other approaches to the convergence analysis of adaptive IIR filters which we briefly mention here. For deterministic signals, hyperstability theory can be used to prove convergence of the PLR and FE algorithms under assumptions similar to those described above. This was done in [Jo79] for the HARF algorithm, and in [Jo81] for the SHARF algorithm (which is essentially equivalent to the FE algorithm). Averaging analysis [An86] is another important approach that is analogous to the stochastic ODE method except that it replaces the ensemble averages in (24) with suitable *time averages*. Basically, similar convergence results as those stated above with probability one are obtained here, but α need not be a decreasing function because a *strict* boundedness condition on the data is assumed.

ALTERNATIVE REALIZATIONS

To resolve some of the problems associated with direct-form adaptive IIR filters, algorithms for alternative realizations such as the parallel, cascade, and lattice forms have been developed. These structures offer simple stability monitoring and are less sensitive to finite-precision effects (coefficient round-off).

Parallel form

The parallel form [Hv76, Je86] is derived from a partial fraction expansion of the pole-zero filter in (7), resulting in the sum of $L = (N - 1)/2$ second-order sections as shown in Fig. 12. Stability monitoring is trivial in that we require only that $|a_{m,2}| < 1$ and $|a_{m,1}| < 1 - a_{m,2}$ for each section, $m = 0, \ldots, L - 1$ (see the stability triangle in Fig. 9). The gradient components are also easy to compute because the sections are essentially independent; the gradient of one section does not depend on the coefficients of any other section. The exact gradient requires only order $4L$ computations, and it has essentially the same complexity as the simplified gradient of the direct form. Note that a simplified gradient is computed in Fig. 12 for each section, resulting in a reduction of complexity by a factor of two since there are two feedback and two feedforward coefficients in each section.

A disadvantage of the parallel form is that there are now many different *global* minima which can be obtained by reordering the poles among the different sections. It has been shown that this property leads to saddle points on the manifold of equivalent sections that separate the global minima [Na88a]. If the algorithm is initialized on such a manifold (e.g., $\theta(0) = \underline{0}$), then the convergence rate may be slower. It is therefore desirable to initialize the poles to different locations; for example, they might be equally spaced around a circle with a small radius on the z-plane. Another approach that has improved convergence properties uses a DFT to first preprocess the input signal to generate N signals in parallel [Sh89]. These (complex) signals are then filtered by a bank of first-order, pole-zero filters in a manner similar to the parallel form above. In this way the section input signals are now statistically different (because of the spectral shaping of the DFT) which can lead to faster convergence.

It should be noted that the cascade form is very similar to the parallel form in that it is generated by factoring the filter in (7) into the product of $L = (N - 1)/2$ second-order sections [Da81]. Stability monitoring is also trivial here, but the complexity of the gradient is significantly greater. This can be understood by noting that the output signal of each section depends on the coefficients of that section as well as all *previous* sections. Furthermore, it is not possible to generate an adequate simplified form of the gradient with a complexity as low as that of the simplified direct- and parallel-form gradients. Simulations demonstrate that the cascade form can have a slower convergence rate than that of the other realizations [Sh87].

Lattice form

The lattice form adaptive IIR filter [Hv76, Pa80] is shown in Fig. 13, along with an example of how the gradient components are computed. The complexity of the gradient is comparable to that of the cascade form, and no simplified form has been found yet that has satisfactory convergence properties.[10] The primary advantage of the lattice structure is that stability monitoring is even simpler than that of the parallel form, requiring only that each reflection coefficient satisfy $|a_m(n)| < 1$ [Mk76]. Another advantage over the parallel form is that it does not have any saddle points; there is a unique lattice representation for any set of direct-form coefficients. Computer simulations indicate that the adaptive lattice filter has convergence properties similar to those of the direct form [Sh87], and this appears to be related to its unique representation.

Observe that the feedforward coefficients weight the lattice node signals (backward residuals [Ha86]) in the same way an adaptive FIR filter weights the delayed input signal of a tapped delay line. It can be shown that when the input $x(n)$ of the lattice is a white random process, these signals will be mutually uncorrelated. Consequently, fast convergence of the feedforward coefficients can be achieved using only a stochastic gradient algorithm. This property is not shared by the parallel and cascade forms because each subsection is implemented in a direct form.

[10] A simplified gradient has been derived in [Ay82], but it was shown that the resulting algorithm has convergence problems.

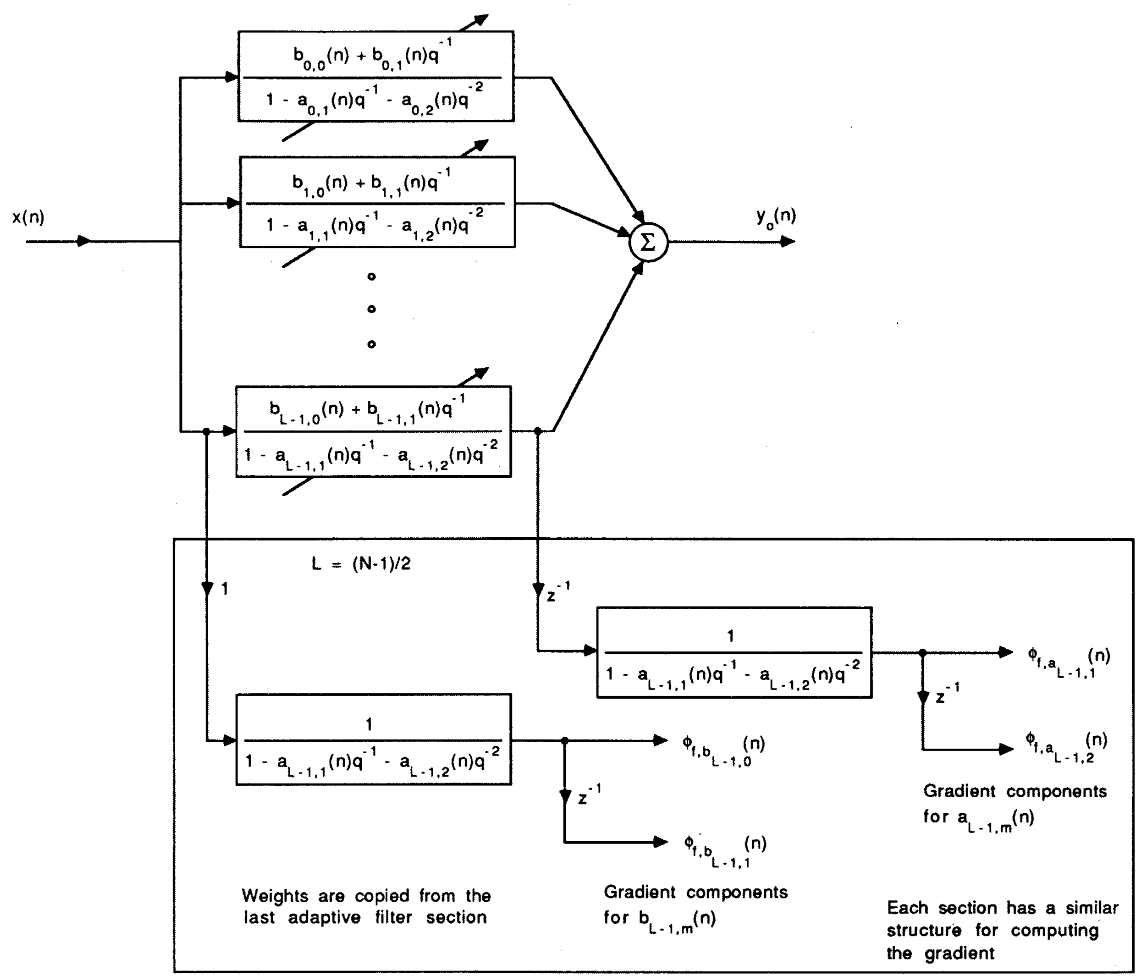

Figure 12. Parallel-Form Adaptive IIR Filter. A parallel form of second-order sections is obtained from a partial fraction expansion of the direct form [$L = (N - 1)/2$]. The gradient of the output with respect to the filter coefficients is uncoupled in the sense that the coefficients in one section do not affect the output of any other section. As a result, the $(N - 1)$-order inverse polynomials used to generate the gradient components of the direct form are reduced to only *second-order* polynomials—a different one for each section. Further simplification is obtained by assuming that the coefficients change slowly. In this case, only two filters are needed for each section: one for the feedback coefficients and another for the feedforward coefficients. This simplified form is illustrated above where the gradient components are shown only for the last section.

CONCLUSION AND SUMMARY

This paper has presented an overview of the important structures and algorithms used in adaptive IIR filtering. All of the output-error realizations have some form of feedback and, consequently, the adaptive algorithms are more complicated than those used in adaptive FIR filtering. There are several important issues associated with convergence, such as the need for stability monitoring, satisfying the SPR condition, and the possible existence of local minima and saddle points. The theory of adaptive IIR filters is incomplete because the analysis involves highly nonlinear systems, but there has been significant progress in recent years. Computer simulation studies of the algorithms are usually necessary in order to determine convergence rates and to predict the overall performance. It is anticipated that future research will focus on alternative realizations such as the lattice form and on improved convergence analysis techniques. Because they offer a significant reduction in computational complexity, adaptive IIR filters are an important alternative to conventional adaptive FIR filtering.

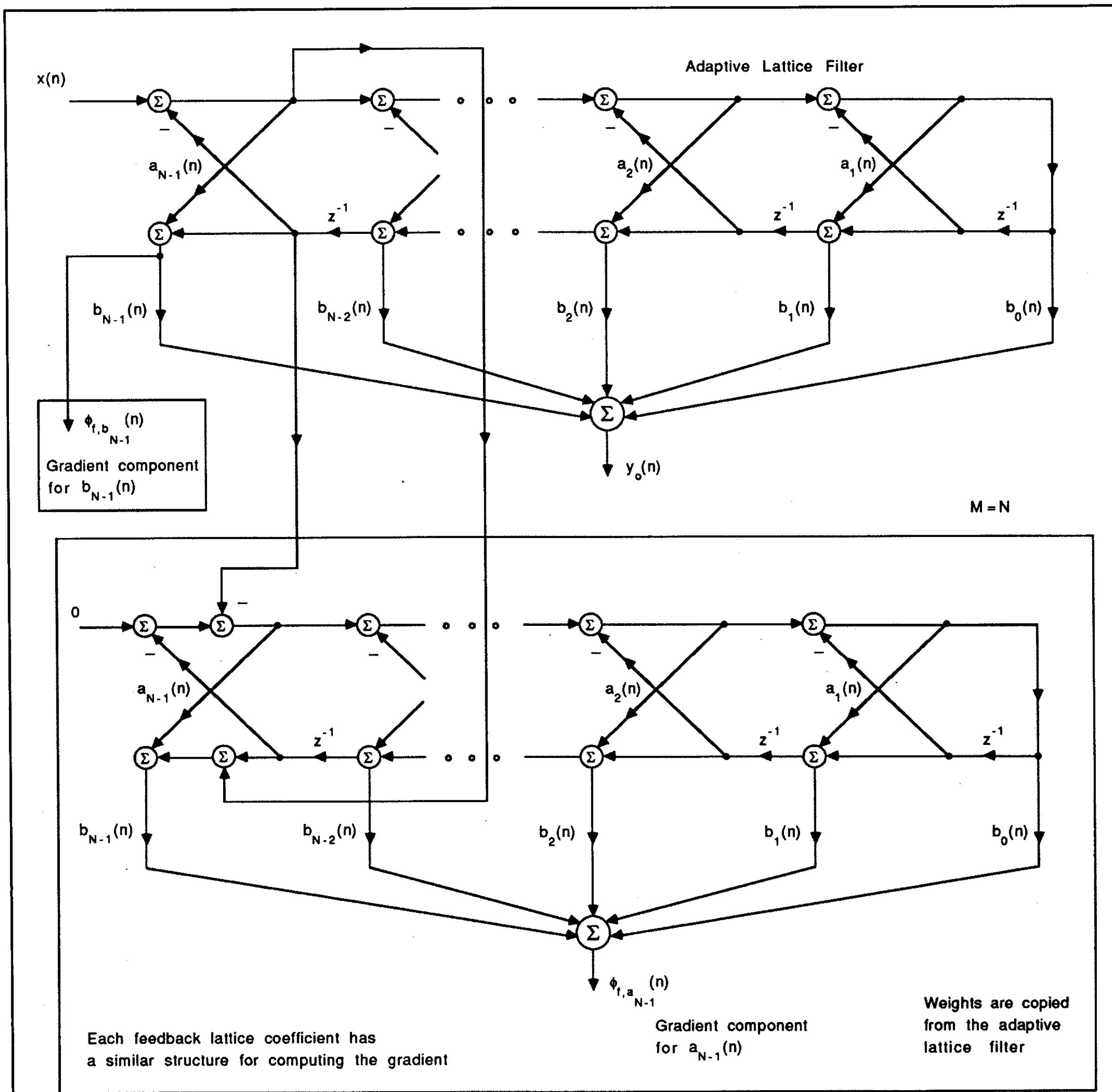

Figure 13. Lattice-Form Adaptive IIR Filter. The adaptive IIR lattice filter consists of a feedback lattice structure characterized by the reflection coefficients $\{a_m(n)\}$ and a feedforward structure characterized by the coefficients $\{b_m(n)\}$. Observe that the gradient component for a feedback coefficient [shown only for $a_{N-1}(n)$] requires a separate lattice filter using intermediate signals of the adaptive lattice filter as input signals. This is considerably more complicated than that of the parallel form, and there does not appear to be a satisfactory way of simplifying this result. The gradient components for the feedforward section are simply the feedback node signals (backward residuals), as shown above for $b_{N-1}(n)$.

REFERENCES

[An86] B. D. O. Anderson, R. R. Bitmead, C. R. Johnson, Jr., P. V. Kokotovic, R. L. Kosut, I. M. Y. Mareels, L. Praly, and B. D. Riedle, *Stability of Adaptive Systems: Passivity and Averaging Analysis*. MIT Press, Cambridge, 1986.

[As71] K. J. Åström and P. Eykhoff, "System identification—A survey," *Automatica*, vol. 7, no. 2, pp. 123–162, Mar. 1971.

[As74] K. J. Åström and T. Söderström, "Uniqueness of the maximum likelihood estimates of the parameters of an ARMA model," *IEEE Trans. Automatic Control*, vol. AC-19, no. 6, pp. 769–773, Dec. 1974.

[Ay82] I. L. Ayala, "On a new adaptive lattice algorithm for recursive filters," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-30, no. 2, pp. 316–319, Apr. 1982.

[Be80] A. Benveniste, M. Goursat, and G. Ruget, "Analysis of stochastic approximation schemes with discontinuous and dependent forcing terms with applications to data communications algorithms," *IEEE Trans. Automatic Control*, vol. AC-25, no. 12, pp. 1042–1058, Dec. 1980.

[Bi84] R. R. Bitmead, "Persistence of excitation conditions and the convergence of adaptive schemes," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 2, pp. 183–191, Mar. 1984.

[Br88] T. Brennan, "Bounding adaptive-filter poles using Kharitonov's theorem," in *Proc. 22nd Asilomar Conf. Signals, Systems, Computers*, Pacific Grove, California, Nov. 1988.

[Da81] R. A. David, "IIR adaptive algorithms based on gradient search techniques," Ph.D. diss. Stanford University, 1981.

[Fa86] H. Fan and W. K. Jenkins, "A new adaptive IIR filter," *IEEE Trans. Circuits Systems*, vol. CAS-33, no. 10, pp. 939–947, Oct. 1986.

[Fa88] H. Fan, "Application of Benveniste's convergence results in a study of adaptive IIR filtering algorithms," *IEEE Trans. Inform. Theory*, vol. 34, no. 4, pp. 692–709, July 1988.

[Fe76] P. L. Feintuch, "An adaptive recursive LMS filter," *Proc. IEEE*, vol. 64, no. 11, pp. 1622–1624, Nov. 1976.

[Fr82] B. Friedlander, "System identification techniques for adaptive signal processing," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-30, no. 2, pp. 240–246, Apr. 1982.

[Fr84a] B. Friedlander and J. O. Smith, "Analysis and performance evaluation of an adaptive notch filter," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 2, pp. 283–295, Mar. 1984.

[Fr84b] B. Friedlander, "On the computation of the Cramer-Rao bound for ARMA parameter estimation," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-32, no. 4, pp. 721–727, Aug. 1984.

[Gc83] R. P. Gooch, "Adaptive pole-zero filtering: the equation-error approach," Ph.D. diss. Stanford University, 1983.

[Gc86] R. P. Gooch and J. J. Shynk, "Wide-band adaptive array processing using pole-zero digital filters," *IEEE Trans. Antennas and Propagation*, vol. AP-34, no. 3, pp. 355–367, Mar. 1986.

[Gd84] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*. Prentice-Hall, Englewood Cliffs, N.J., 1984.

[Ha86] S. S. Haykin, *Adaptive Filter Theory*. Prentice-Hall, Englewood Cliffs, N. J., 1986.

[Hr81] L. L. Horowitz and K. D. Senne, "Performance advantage of complex LMS for controlling narrow-band adaptive arrays," *IEEE Trans. Circuits Systems*, vol. CAS-28, no. 6, pp. 562–576, June 1981.

[Hv76] S. Horvath, Jr., "Adaptive IIR digital filters for on-line time-domain equalization and linear prediction," presented at IEEE Arden House Workshop on Dig. Sig. Proc., Harriman, N.Y. Feb. 1976.

[Hv80] S. Horvath, Jr., "A new adaptive recursive LMS filter," in *Digital Signal Processing*, V. Cappellini and A. G. Constantinides, eds. Academic Press, New York, 1980, pp. 21–26.

[Ja84] N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Prentice-Hall, Englewood Cliffs, N.J., 1984.

[Je86] W. K. Jenkins and M. Nayeri, "Adaptive filters realized with second order sections," in *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Proc.*, Tokyo, Japan, Apr. 1986, pp. 2103–2106.

[Jo77] C. R. Johnson, Jr., and M. G. Larimore, "Comments on and additions to 'An adaptive recursive LMS filter,'" *Proc. IEEE*, vol. 65, no. 9, pp. 1399–1402, Sep. 1977.

[Jo79] C. R. Johnson, Jr., "A convergence proof for a hyperstable adaptive recursive filter," *IEEE Trans. Inform. Theory*, vol. IT-25, no. 6, pp. 745–749, Nov. 1979.

[Jo81] C. R. Johnson, Jr., M. G. Larimore, J. R. Treichler, and B. D. O. Anderson, "SHARF convergence properties," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-29, no. 3, pp. 659–670, June 1981.

[Jo84] C. R. Johnson, Jr., "Adaptive IIR filtering: Current results and open issues," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 2, pp. 237–250, Mar. 1984.

[Ju64] E. I. Jury, *Theory and Applications of the Z-Transform Method*. Wiley, New York, 1964.

[Ka80] T. Kailath, *Linear Systems*. Prentice-Hall, Englewood Cliffs, N.J., 1980.

[Ku84] H. J. Kushner and A. Shwartz, "Weak convergence and asymptotic properties of adaptive filters with constant gains," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 2, pp. 177–182, Mar. 1984.

[La74] I. D. Landau, "A survey of model reference adaptive techniques—Theory and applications," *Automatica*, vol. 10, no. 4, pp. 353–379, July 1974.

[La79] I. D. Landau, *Adaptive Control: The Model Reference Approach*. Marcel Dekker, New York, 1979.

[Le80] M. G. Larimore, J. R. Treichler, and C. R. Johnson, Jr., "SHARF: An algorithm for adapting IIR digital filters," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-28, no. 4, pp. 428–440, Aug. 1980.

[Lh83] E. L. Lehmann, *Theory of Point Estimation*. Wiley, New York, 1983.

[Lj77] L. Ljung, "Analysis of recursive stochastic algorithms," *IEEE Trans. Automatic Control*, vol. AC-22, no. 4, pp. 551–575, Aug. 1977.

[Lj83] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*. MIT Press, Cambridge, 1983.

[Lj87] L. Ljung, *System Identification: Theory for the User*. Prentice-Hall, Englewood Cliffs, N.J., 1987.

[Lo87] G. Long, D. Shwed, and D. D. Falconer, "Study of a pole-zero adaptive echo canceller," *IEEE Trans. Circuits Systems*, vol. CAS-34, no. 7, pp. 765–769, July 1987.

[Ma83] O. Macchi and E. Eweda, "Second-order convergence analysis of stochastic adaptive linear filtering," *IEEE Trans. Automatic Control*, vol. AC-28, no. 1, pp. 76–85, Jan. 1983.

[Mk76] J. D. Markel and A. H. Gray, Jr., *Linear Prediction* *of Speech*. Springer-Verlag, New York, 1976.

[Mn73] J. M. Mendel, *Discrete Techniques of Parameter Estimation: The Equation Error Formulation*. Marcel Dekker, New York, 1973.

[Mn87] J. M. Mendel, *Lessons in Digital Estimation Theory*. Prentice-Hall, Englewood Cliffs, N.J., 1987.

[Mo77] F. Mosteller and J. W. Tukey, *Data Analysis and Regression*. Addison-Wesley, Reading, Mass., 1977.

[Na88a] M. Nayeri and W. K. Jenkins, "Analysis of alternate realizations of adaptive IIR filters," in *Proc. IEEE Int. Symp. Circuits Systems*, Espoo, Finland, June 1988, pp. 2157–2160.

[Na88b] M. Nayeri, "A weaker sufficient condition for the unimodality of error surfaces associated with exactly matching adaptive IIR filters," in *Proc. 22nd Asilomar Conf. Signals, Systems, Computers*, Pacific Grove, California, Nov. 1988.

[Ne85] A. Nehorai, "A minimal parameter adaptive notch filter with constrained poles and zeros," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-33, no. 4, pp. 983–996, Aug. 1985.

[Pa80] D. Parikh, N. Ahmed, and S. D. Stearns, "An adaptive lattice algorithm for recursive filters," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-28, no. 1, pp. 110–111, Feb. 1980.

[Po73] V. M. Popov, *Hyperstability of Control Systems*. Springer-Verlag, New York, 1973.

[Pr83] J. G. Proakis, *Digital Communications*. McGraw-Hill, New York, 1983.

[Sh86] J. J. Shynk, "A complex adaptive algorithm for IIR filtering," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-34, no. 5, pp. 1342–1344, Oct. 1986.

[Sh87] J. J. Shynk, "Performance of alternative adaptive IIR filter realizations," in *Proc. 21st Asilomar Conf. Signals, Systems, Computers*, Pacific Grove, California, Nov. 1987, pp. 144–150.

[Sh89] J. J. Shynk, "Adaptive IIR filtering using parallel-form realizations," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-37, no. 4, pp. 519–533, Apr. 1989.

[So75] T. Söderström, "On the uniqueness of maximum likelihood identification," *Automatica*, vol. 11, no. 2, pp. 193–197, Mar. 1975.

[So78] T. Söderström, L. Ljung, and I. Gustavsson, "A theoretical analysis of recursive identification methods," *Automatica*, vol. 14, no. 3, pp. 231–244, May 1978.

[So82] T. Söderström and P. Stoica, "Some properties of the output error method," *Automatica*, vol. 18, no. 1, pp. 93–99, Jan. 1982.

[St81] S. D. Stearns, "Error surfaces of recursive adaptive filters," *IEEE Trans. Circuits Systems*, vol. CAS-28, no. 6, pp. 603–606, June 1981.

[Ta87] K. Tang and C. E. Rohrs, "The use of large adaptation gains to remove the SPR condition from recursive adaptive algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Proc.*, Dallas, Texas, Apr. 1987, pp. 129–132.

[Tr78] J. R. Treichler, M. G. Larimore, and C. R. Johnson, Jr., "Simple adaptive IIR filtering," in *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Proc.*, Tulsa, Oklahoma, Apr. 1978, pp. 118–122.

[Tr85] J. R. Treichler, "Adaptive algorithms for infinite impulse response filters," in *Adaptive Filters*, C. F. N. Cowen and P. M. Grant, eds. Prentice-Hall, Englewood Cliffs, N.J., 1985, ch. 4, pp. 60–90.

[Tr87] J. R. Treichler, C. R. Johnson, Jr., and M. G. Larimore, *Theory and Design of Adaptive Filters*, Wiley, New York, 1987.

[Tt76] S. A. Tretter, *Introduction to Discrete-Time Signal Processing*. Wiley, New York, 1976.

[Vi78] M. Vidyasagar, *Nonlinear Systems Analysis*. Prentice-Hall, Englewood Cliffs, N.J., 1978.

[Wh75] S. A. White, "An adaptive recursive digital filter," in *Proc. 9th Asilomar Conf. Circuits, Systems, Computers*, Pacific Grove, California, Nov. 1975, pp. 21–25.

[Wi76] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, Jr., "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, no. 8, pp. 1151–1162, Aug. 1976.

[Wi85] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Prentice-Hall, Englewood Cliffs, N.J., 1985.

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, 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 IIR 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.