10

ADAPTIVE IIR FILTERS

10.1 INTRODUCTION

Adaptive infinite impulse response (IIR) filters are those in which the zeros and poles of the filter can be adapted. For that benefit the adaptive IIR filters usually[1] have adaptive coefficients on the transfer function numerator and denominator. Adaptive IIR filters present some advantages as compared with the adaptive FIR filters, including reduced computational complexity. If both have the same number of coefficients, the frequency response of the IIR filter can approximate much better a desired characteristic. Therefore, an IIR filter in most cases requires fewer coefficients, mainly when the desired model has poles and zeros, or sharp resonances [2]-[1]. There are applications requiring hundreds and sometimes thousands of taps in an FIR filter where the use of an adaptive IIR filter is highly desirable. Among these applications are satellite-channel and mobile-radio equalizers, acoustic echo cancellation, etc.

The advantages of the adaptive IIR filters come with a number of difficulties, some of them not encountered in the adaptive FIR counterparts. The main drawbacks are: possible instability of the adaptive filter, slow convergence, and error surface with local minima or biased global minimum depending on the objective function [3].

In this chapter, several strategies to implement adaptive IIR filters will be discussed. First, adaptive IIR filters having as objective function the minimization of the mean-square output error are introduced. Several alternative structures are presented and some properties of the error surface are addressed. In addition, some algorithms based on the minimization of alternative objective functions are discussed. The algorithms are devised to avoid the multimodality inherent to the methods based on the output error.

[1] There are adaptive filtering algorithms with fixed poles.

P.S.R. Diniz, *Adaptive Filtering*, DOI: 10.1007/978-0-387-68606-6_10,

10.2 OUTPUT-ERROR IIR FILTERS

In the present section, we examine strategies to reduce a function of the output error given by

$$\xi(k) = F[e(k)] \tag{10.1}$$

using an adaptive filter with IIR structure. The output error is defined by

$$e(k) = d(k) - y(k) \tag{10.2}$$

as illustrated in Fig. 10.1.a. As usual, an adaptation algorithm determines how the coefficients of the adaptive IIR filter should change in order to get the objective function reduced.

Let us consider that the adaptive IIR filter is realized using the direct-form structure of Fig. 10.1.b. The signal information vector in this case is defined by

$$\boldsymbol{\phi}(k) = [y(k-1)\, y(k-2) \ldots y(k-N)\, x(k)\, x(k-1) \ldots x(k-M)]^T \tag{10.3}$$

where N and M are the adaptive filter denominator and numerator orders, respectively.

The direct-form adaptive filter can be characterized in time domain by the following difference equation

$$y(k) = \sum_{j=0}^{M} b_j(k)x(k-j) - \sum_{j=1}^{N} a_j(k)y(k-j) \tag{10.4}$$

In the system identification field [8], the above difference equation is in general described through polynomial operator as follows:

$$y(k) = \frac{B(k, q^{-1})}{A(k, q^{-1})}x(k) \tag{10.5}$$

where

$$\begin{aligned} B(k, q^{-1}) &= b_0(k) + b_1(k)q^{-1} + \cdots + b_M(k)q^{-M} \\ A(k, q^{-1}) &= 1 + a_1(k)q^{-1} + \cdots + a_N(k)q^{-N} \end{aligned}$$

and q^{-j} denotes a delay operation in a time domain signal of j samples, i.e., $q^{-j}x(k) = x(k-j)$. The difference equation (10.4) can also be rewritten in a vector form, which is more convenient for the algorithm description and implementation, as described below

$$y(k) = \boldsymbol{\theta}^T(k)\boldsymbol{\phi}(k) \tag{10.6}$$

where $\boldsymbol{\theta}(k)$ is the adaptive-filter coefficient vector given by

$$\boldsymbol{\theta}(k) = [-a_1(k)\ -a_2(k) \ldots -a_N(k)\, b_0(k)\, b_1(k) \ldots b_M(k)]^T \tag{10.7}$$

In a given iteration k, the adaptive-filter transfer function can be expressed as follows:

$$\begin{aligned} H_k(z) &= z^{N-M}\frac{b_0(k)z^M + b_1(k)z^{M-1} + \cdots + b_{M-1}(k)z + b_M(k)}{z^N + a_1(k)z^{N-1} + \cdots + a_{N-1}(k)z + a_N(k)} \\ &= z^{N-M}\frac{N_k(z)}{D_k(z)} \end{aligned} \tag{10.8}$$

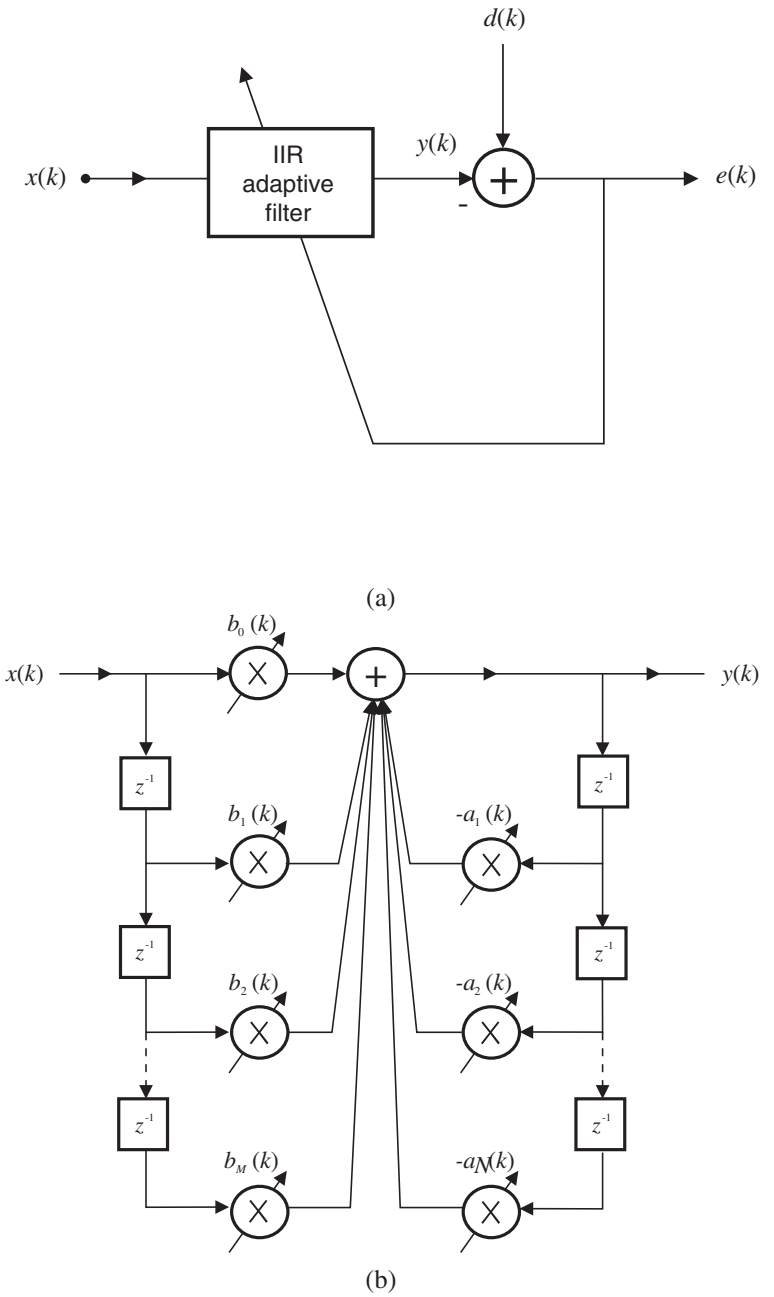


Figure 10.1 Adaptive IIR Filtering: (a) General configuration, (b) Adaptive IIR direct-form realization.

Given the objective function $F[e(k)]$, the gradient vector required to be employed in the adaptive algorithm is given by

$$\boldsymbol{g}(k) = \frac{\partial F[e(k)]}{\partial e(k)} \frac{\partial e(k)}{\partial \boldsymbol{\theta}(k)} \tag{10.9}$$

where $e(k)$ is the output error. The first derivative in the above gradient equation is a scalar dependent on the objective function, while the second derivative is a vector whose elements are obtained by

$$\frac{\partial e(k)}{\partial a_i(k)} = \frac{\partial [d(k) - y(k)]}{\partial a_i(k)} = -\frac{\partial y(k)}{\partial a_i(k)}$$

for $i = 1, 2, \ldots, N$, and

$$\frac{\partial e(k)}{\partial b_j(k)} = \frac{\partial [d(k) - y(k)]}{\partial b_j(k)} = -\frac{\partial y(k)}{\partial b_j(k)} \tag{10.10}$$

for $j = 0, 1, \ldots, M$, where we used the fact that the desired signal $d(k)$ is not dependent on the adaptive-filter coefficients.

The derivatives of $y(k)$ with respect to the filter coefficients can be calculated from the difference equation (10.4) as follows:

$$\frac{\partial y(k)}{\partial a_i(k)} = -y(k-i) - \sum_{j=1}^{N} a_j(k) \frac{\partial y(k-j)}{\partial a_i(k)}$$

for $i = 1, 2, \ldots, N$, and

$$\frac{\partial y(k)}{\partial b_j(k)} = x(k-j) - \sum_{i=1}^{N} a_i(k) \frac{\partial y(k-i)}{\partial b_j(k)} \tag{10.11}$$

for $j = 0, 1, \ldots, M$. The partial derivatives of $y(k-i)$ with respect to the coefficients, for $i = 1, 2, \ldots, N$, are different from zero because the adaptive filter is recursive. As a result, the present coefficients $a_i(k)$ and $b_j(k)$ are dependent on the past output samples $y(k-i)$. The precise evaluation of these partial derivatives is a very difficult task, and does not have a simple implementation. However, as first pointed out in [5] and [6], if small step sizes are used in the coefficient updating, the following approximations are valid

$$a_i(k) \quad \approx \quad a_i(k-j) \quad \text{for } i, j = 1, 2, \ldots, N$$

and

$$b_j(k) \quad \approx \quad b_j(k-i) \quad \text{for } j = 0, 1, \ldots, M \text{ and } i = 1, 2, \ldots, N \tag{10.12}$$

As a consequence, equations (10.11) can be rewritten as

$$-\frac{\partial y(k)}{\partial a_i(k)} \quad \approx \quad +y(k-i) - \sum_{j=1}^{N} a_j(k) \left[\frac{-\partial y(k-j)}{\partial a_i(k-j)} \right]$$

for $i = 1, 2, \ldots, N$, and

$$\frac{\partial y(k)}{\partial b_j(k)} \approx x(k-j) - \sum_{i=1}^{N} a_i(k) \frac{\partial y(k-i)}{\partial b_j(k-i)} \tag{10.13}$$

for $j = 0, 1, \ldots, M$. Note that these equations are standard difference equations.

The above equations can be implemented by all-pole filters having as input signals $-y(k-i)$ and $x(k-j)$ for the first and second set of equations, respectively. The implementation of the derivative signals of equations (10.13) is depicted in Fig. 10.2. The all-pole sections realization can be performed through IIR direct-form structure, with transfer function given by

$$S^{a_i}(z) = \mathcal{Z}\left[\frac{\partial y(k)}{\partial a_i(k)}\right] = \frac{-z^{N-i}}{D_k(z)} Y(z)$$

for $i = 1, 2, \ldots, N$, and

$$S^{b_j}(z) = \mathcal{Z}\left[\frac{\partial y(k)}{\partial b_i(k)}\right] = \frac{z^{N-j}}{D_k(z)} X(z) \tag{10.14}$$

for $j = 0, 1, \ldots, M$, respectively, where $\mathcal{Z}[\cdot]$ denotes the $\mathcal{Z}$-transform of $[\cdot]$.

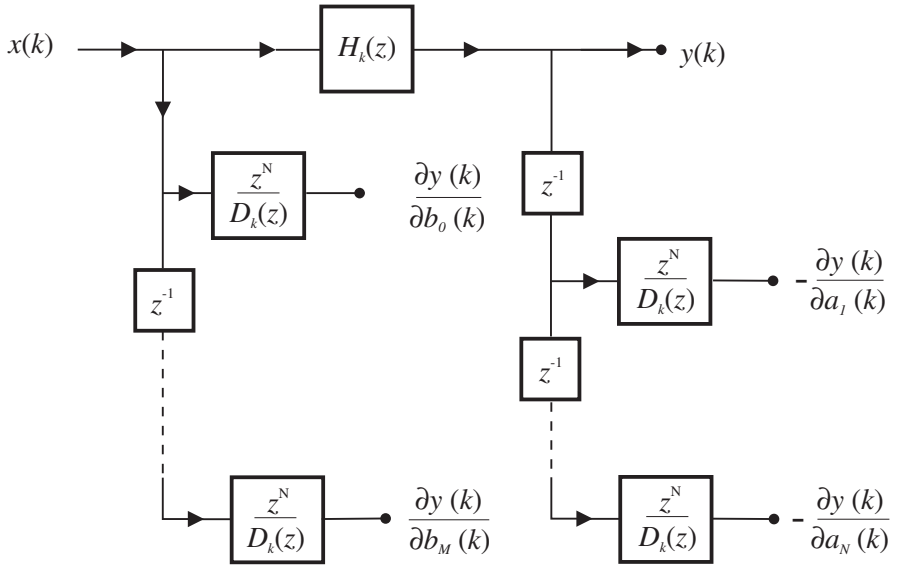


Figure 10.2 Derivative implementation.

The amount of computation spent to obtain the derivatives is relatively high, as compared with the adaptive-filter computation itself. A considerable reduction in the amount of computation can be achieved, if it is considered that the coefficients of the adaptive-filter denominator polynomial are slowly varying, such that

$$D_k(z) \approx D_{k-i}(z) \qquad \text{for } i = 1, 2, \ldots, \max(N, M) \tag{10.15}$$

where $\max(a, b)$ denotes maximum between a and b. The interpretation is that the denominator polynomial is kept almost constant for a number of iterations. With this approximation, it is possible to eliminate the duplicating all-pole filters of Fig. 10.2, and replace them by a single all-pole in front of the two sets of delays as depicted in Fig. 10.3.a. In addition, if the recursive part of the adaptive filter is implemented before the numerator part, one more all-pole section can be saved as illustrated in Fig. 10.3.b [7].

Note that in the time domain, the approximations of equation (10.15) imply the following relations

$$\frac{\partial y(k)}{\partial a_i(k)} \approx q^{-i+1} \frac{\partial y(k)}{\partial a_1(k)}$$

for $i = 1, 2, \ldots, N$, and

$$\frac{\partial y(k)}{\partial b_j(k)} \approx q^{-j} \frac{\partial y(k)}{\partial b_0(k)} \tag{10.16}$$

for $j = 0, 1, \ldots, M$, where $\frac{\partial y(k)}{\partial a_1(k)}$ represents the partial derivative of $y(k)$ with respect to the first non unit coefficient of the denominator polynomial, whereas $\frac{\partial y(k)}{\partial b_0(k)}$ is the partial derivative of $y(k)$ with respect to the first coefficient of the numerator polynomial.

10.3 GENERAL DERIVATIVE IMPLEMENTATION

The derivatives of the output signal as related to the adaptive-filter coefficients are always required to generate the gradient vector that is used in most adaptive algorithms. These derivatives can be obtained in a systematic form by employing a sensitivity property of digital filters with fixed coefficients [2]-[1], if the adaptive-filter coefficients are slowly varying as assumed in equation (10.12).

Refer to Fig. 10.4.a, where the multiplier with coefficient c is an internal multiplier of a digital filter with fixed coefficients. A good measure of how the digital filter characteristics change when the value of c changes is the sensitivity function, defined as the partial derivative of the digital filter transfer function $H(z)$ as related to the coefficient c. It is well known from classical digital filtering theory [2]-[1] that the partial derivative of the digital filter transfer function, with respect to a given multiplier coefficient c, is given by the product of the transfer function $H_{13}(z)$ from the filter input to the multiplier input and the transfer function $H_{42}(z)$ from the multiplier output to the filter output, that is

$$S^c(z) = H_{13}(z) \cdot H_{42}(z) \tag{10.17}$$

Fig. 10.4.b illustrates the derivative implementation. It can be noted that the implementation of the derivatives for the direct-form structure shown in Fig. 10.2 can be obtained by employing equation (10.17). In the time domain, the filtering operation performed in the implementation of Fig. 10.4.b is given by

$$\frac{\partial y(k)}{\partial c} = h_{13}(k) * h_{42}(k) * x(k) \tag{10.18}$$

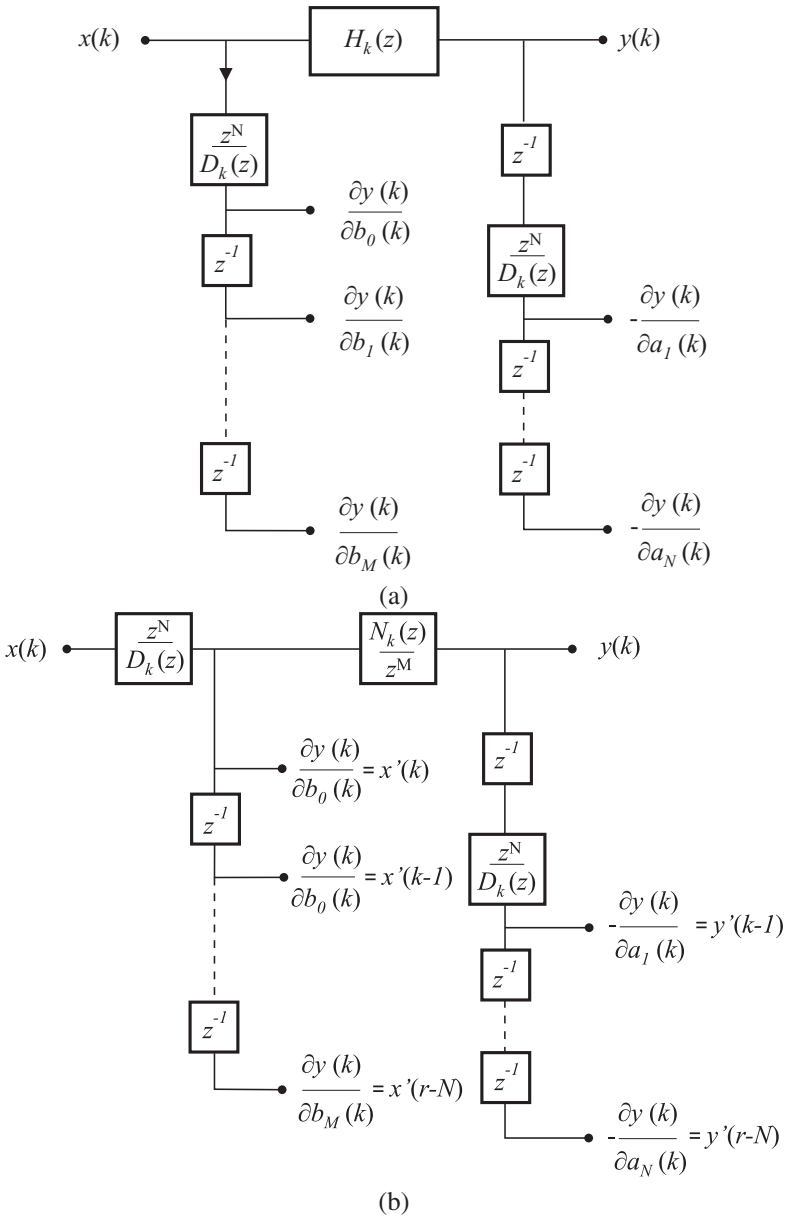


Figure 10.3 Simplified derivative implementation: (a) Simplification I, (b) Simplification II.

where $*$ denotes convolution and $h_{ij}(k)$ is the impulse response related to $H_{ij}(z)$. When the digital filter coefficients are slowly varying, the desired derivatives can be derived as in Fig. 10.4 for each adaptive coefficient. In this case, only an approximated derivative is obtained

$$\frac{\partial y(k)}{\partial c(k)} \approx h_{13k}(k) * h_{42k}(k) * x(k) \tag{10.19}$$

10.4 ADAPTIVE ALGORITHMS

In this section, the adaptation algorithms used in IIR adaptive filtering are described. In particular, we present the RLS, the Gauss-Newton, and the gradient-based algorithms.

10.4.1 Recursive Least-Squares Algorithm

A possible objective function for adaptive IIR filtering based on output error is the least-squares function[2]

$$\xi^d(k) = \sum_{i=0}^{k} \lambda^{k-i} e^2(i) = \sum_{i=0}^{k} \lambda^{k-i} [d(i) - \boldsymbol{\theta}^T(k)\boldsymbol{\phi}(i)]^2 \tag{10.20}$$

The forgetting factor λ is usually chosen in the range $0 \ll \lambda < 1$, with the objective of turning the distant past information increasingly negligible. By differentiating $\xi^d(k)$ with respect to $\boldsymbol{\theta}(k)$, it follows that

$$\begin{aligned} 2\mathbf{g}_D(k) &= \frac{\partial \xi^d(k)}{\partial \boldsymbol{\theta}(k)} \\ &= 2\sum_{i=0}^{k} \lambda^{k-i} \boldsymbol{\varphi}(i)[d(i) - \boldsymbol{\theta}^T(k)\boldsymbol{\phi}(i)] \\ &= 2\boldsymbol{\varphi}(k)e(k) + \lambda \frac{\partial \xi^d(k-1)}{\partial \boldsymbol{\theta}(k)} \end{aligned} \tag{10.21}$$

where the vector $\boldsymbol{\varphi}(k)$ is the derivative of $e(i)$ with respect to $\boldsymbol{\theta}(k)$, i.e.,

$$\boldsymbol{\varphi}(k) = \frac{\partial e(k)}{\partial \boldsymbol{\theta}(k)} = -\frac{\partial y(k)}{\partial \boldsymbol{\theta}(k)} \tag{10.22}$$

and without loss of generality we considered that $\xi^d(k-1)$ is a function of $\boldsymbol{\theta}(k)$ and not of $\boldsymbol{\theta}(k-1)$ as in the FIR case. The second-derivative matrix $2\mathbf{R}_D(k)$ of $\xi^d(k)$[3] with respect to $\boldsymbol{\theta}(k)$ is then given by

$$\frac{\partial^2 \xi^d(k)}{\partial \boldsymbol{\theta}^2(k)} = 2\mathbf{R}_D(k) = 2\lambda \mathbf{R}_D(k-1) + 2\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k) - 2\frac{\partial^2 y(k)}{\partial \boldsymbol{\theta}^2(k)} e(k) \tag{10.23}$$

[2]The reader should note that this definition of the deterministic weighted least squares utilizes the *a priori* error with respect to the latest data pair $d(k)$ and $x(k)$, unlike the FIR RLS case.

[3]By differentiating $2\mathbf{g}_D(k)$ in equation (10.21) with respect to $\boldsymbol{\theta}(k)$.

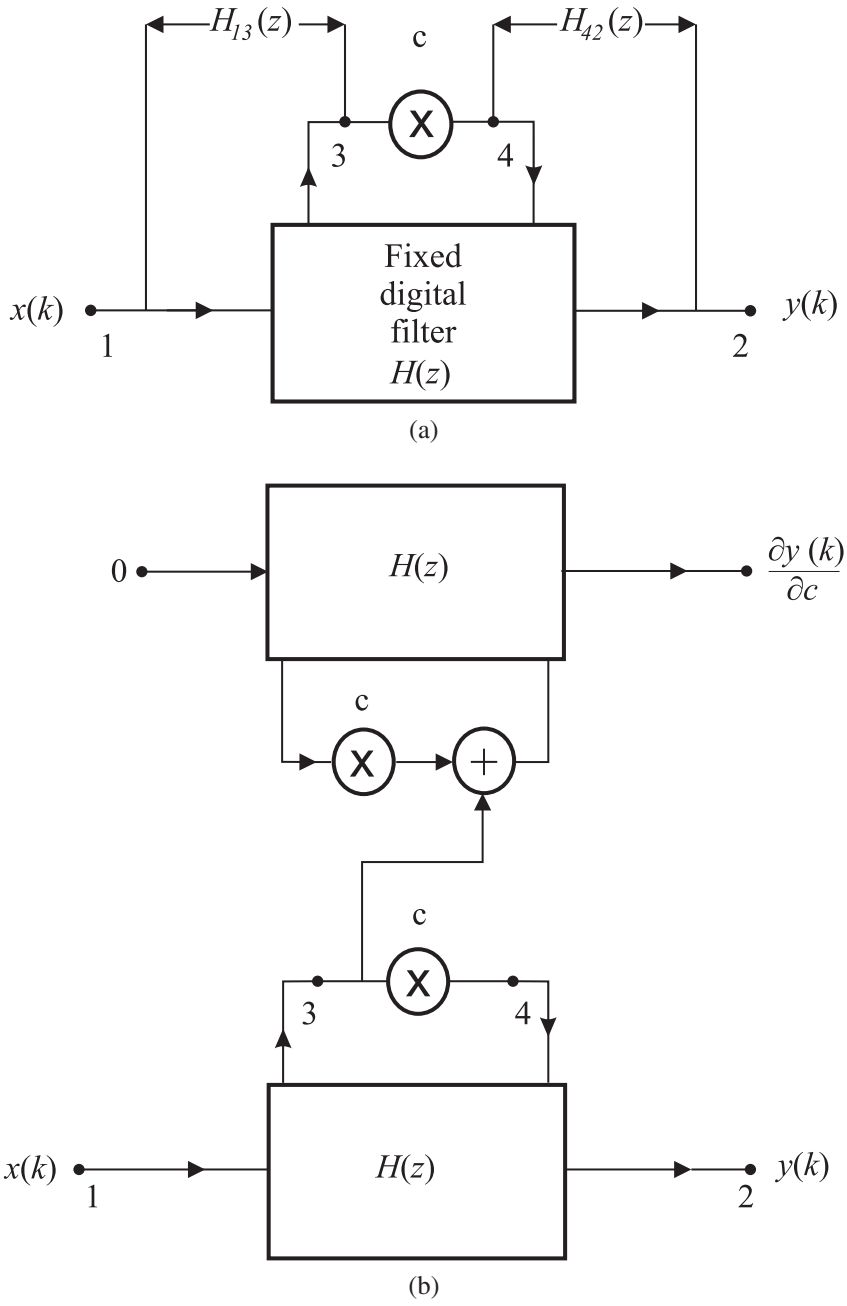


Figure 10.4 General derivative implementation: (a) General structure, (b) Derivative implementation.

Now, several assumptions are made to generate a recursive algorithm. The adaptive-filter parameters are considered to be updated by

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \mathbf{R}_D^{-1}(k)\mathbf{g}_D(k) \tag{10.24}$$

As can be noted from equations (10.21) and (10.23), the calculations of the last terms in both $\mathbf{R}_D(k)$ and $\mathbf{g}_D(k)$ require a knowledge of the signal information vector since the beginning of the algorithm operation, namely $\boldsymbol{\varphi}(i)$ for $i < k$. However, if the algorithm step sizes, i.e., the elements of $|\boldsymbol{\theta}(k+1) - \boldsymbol{\theta}(k)|$, are considered small, then

$$\frac{\partial \xi^d(k-1)}{\partial \boldsymbol{\theta}(k)} \approx 0 \tag{10.25}$$

assuming that the vector $\boldsymbol{\theta}(k)$ is the optimal estimate for the parameters at the instant $k-1$. This conclusion can be drawn by approximating $\xi^d(k-1)$ by a Taylor series around $\boldsymbol{\theta}(k-1)$ and considering only the first-order term [8]. Also, close to the minimum solution, the output error $e(k)$ can be considered approximately a white noise (if the measurement noise is also a white noise) and independent of $\frac{\partial^2 y(k)}{\partial \boldsymbol{\theta}^2(k)}$. This assumption allows us to consider the expected value of the last term in equation (10.23) negligible as compared to the remaining terms.

Applying the above approximations, an RLS algorithm for adaptive IIR filtering is derived in which the basic steps are:

$$e(k) = d(k) - \boldsymbol{\theta}^T(k)\boldsymbol{\phi}(k) \tag{10.26}$$

$$\boldsymbol{\varphi}(k) = -\frac{\partial y(k)}{\partial \boldsymbol{\theta}(k)} \tag{10.27}$$

$$\mathbf{S}_D(k) = \frac{1}{\lambda}\left[\mathbf{S}_D(k-1) - \frac{\mathbf{S}_D(k-1)\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k)\mathbf{S}_D(k-1)}{\lambda + \boldsymbol{\varphi}^T(k)\mathbf{S}_D(k-1)\boldsymbol{\varphi}(k)}\right] \tag{10.28}$$

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \mathbf{S}_D(k)\boldsymbol{\varphi}(k)e(k) \tag{10.29}$$

The description of the RLS adaptive IIR filter is given in Algorithm 10.1.

Note that the primary difference between the RLS algorithm for FIR and IIR adaptive filtering relies on the signal information vector, $\boldsymbol{\varphi}(k)$, that in the IIR case is obtained through a filtering operation while in the FIR case it corresponds to the input signal vector $\mathbf{x}(k)$.

10.4.2 The Gauss-Newton Algorithm

Consider as objective function the mean-square error (MSE) defined as

$$\xi = E[e^2(k)] \tag{10.30}$$

In the Gauss-Newton algorithm, the minimization of the objective function is obtained by performing searches in the Newton direction, using estimates of the inverse Hessian matrix and the gradient vector.

Algorithm 10.1

Output Error Algorithm, RLS Version

Initialization

$a_i(k) = b_i(k) = e(k) = 0$

$y(k) = x(k) = 0\,,\ k\ < 0$

$\mathbf{S}_D(0) = \delta^{-1}\mathbf{I}$

Definition

$\boldsymbol{\varphi}^T(k) = [-y'(k-1)\ldots - y'(k-N)\ -x'(k)\ -x'(k-1)\ldots - x'(k-M)]$

For each $x(k),\ d(k),\ k \geq 0,$ do

$y(k) = \boldsymbol{\phi}^T(k)\boldsymbol{\theta}(k)$

$y'(k) = -y(k) - \sum_{i=1}^{N} a_i(k)y'(k-i)$

$x'(k) = x(k) - \sum_{i=1}^{N} a_i(k)x'(k-i)$

$e(k) = d(k) - y(k)$

$\mathbf{S}_D(k) = \frac{1}{\lambda}\left[\mathbf{S}_D(k-1) - \frac{\mathbf{S}_D(k-1)\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k)\mathbf{S}_D(k-1)}{\lambda+\boldsymbol{\varphi}^T(k)\mathbf{S}_D(k-1)\boldsymbol{\varphi}(k)}\right]$

$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \mathbf{S}_D(k)\boldsymbol{\varphi}(k)e(k)$

Stability test

The gradient vector is calculated as follows:

$$\frac{\partial \xi}{\partial \boldsymbol{\theta}(k)} = E[2e(k)\boldsymbol{\varphi}(k)] \tag{10.31}$$

where $\boldsymbol{\varphi}(k) = \frac{\partial e(k)}{\partial \boldsymbol{\theta}(k)}$ as defined in equation (10.22).

The Hessian matrix is then given by

$$\frac{\partial^2 \xi}{\partial \boldsymbol{\theta}^2(k)} = 2E\left[\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k) + \frac{\partial^2 e(k)}{\partial \boldsymbol{\theta}^2(k)}e(k)\right] \tag{10.32}$$

where the expected value of the second term in the above equation is approximately zero, since close to a solution the output error $e(k)$ is "almost" a white noise independent of the following term

$$\frac{\partial^2 e(k)}{\partial \boldsymbol{\theta}^2(k)} = -\frac{\partial^2 y(k)}{\partial \boldsymbol{\theta}^2(k)}$$

The determination of the gradient vector and the Hessian matrix requires statistical expectation calculations. In order to derive a recursive algorithm, estimates of the gradient vector and Hessian matrix have to be used. For the gradient vector, the most commonly used estimation is the stochastic

gradient given by

$$\frac{\partial \hat{\xi}}{\partial \boldsymbol{\theta}(k)} = 2e(k)\boldsymbol{\varphi}(k) \tag{10.33}$$

where $\hat{\xi}$ is an estimate of ξ. Such approximation was also used in the derivation of the LMS algorithm. The name stochastic gradient originates from the fact that the estimates point to random directions around the true gradient direction.

The Hessian estimate can be generated by employing a weighted summation as follows:

$$\begin{aligned}\hat{\mathbf{R}}(k+1) &= \alpha\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k) + \alpha\sum_{i=0}^{k-1}(1-\alpha)^{k-i}\boldsymbol{\varphi}(i)\boldsymbol{\varphi}^T(i) \\ &= \alpha\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k) + (1-\alpha)\hat{\mathbf{R}}(k)\end{aligned} \tag{10.34}$$

where α is a small factor chosen in the range $0 < \alpha < 0.1$. By taking the expected value on both sides of the above equation and assuming that $k \to \infty$, it follows that

$$\begin{aligned}E[\hat{\mathbf{R}}(k+1)] &= \alpha\sum_{i=0}^{k}(1-\alpha)^{k-i}E[\boldsymbol{\varphi}(i)\boldsymbol{\varphi}^T(i)] \\ &\approx E[\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k)]\end{aligned} \tag{10.35}$$

Applying the approximation discussed and the matrix inversion lemma to calculate the inverse of $\hat{\mathbf{R}}(k+1)$, i.e., $\hat{\mathbf{S}}(k+1)$, the Gauss-Newton algorithm for IIR adaptive filtering is derived, consisting of the following basic steps

$$e(k) = d(k) - \boldsymbol{\theta}^T(k)\boldsymbol{\phi}(k) \tag{10.36}$$

$$\boldsymbol{\varphi}(k) = \frac{\partial e(k)}{\partial \boldsymbol{\theta}(k)} \tag{10.37}$$

$$\hat{\mathbf{S}}(k+1) = \frac{1}{1-\alpha}\left[\hat{\mathbf{S}}(k) - \frac{\hat{\mathbf{S}}(k)\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k)\hat{\mathbf{S}}(k)}{\frac{1-\alpha}{\alpha} + \boldsymbol{\varphi}^T(k)\hat{\mathbf{S}}(k)\boldsymbol{\varphi}(k)}\right] \tag{10.38}$$

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \mu\hat{\mathbf{S}}(k+1)\boldsymbol{\varphi}(k)e(k) \tag{10.39}$$

where μ is the convergence factor. In most cases, μ is chosen approximately equal to α.

In the updating of the $\hat{\mathbf{R}}(k)$ matrix, the factor $(1-\alpha)$ plays the role of a forgetting factor that determines the effective memory of the algorithm when computing the present estimate. The closer α is to zero the more important is the past information, in other words, the longer is the memory of the algorithm.

10.4.3 Gradient-Based Algorithm

If in the Gauss-Newton algorithm, the estimate of the Hessian matrix is replaced by the identity matrix, the resulting basic algorithm is given by

$$e(k) = d(k) - \boldsymbol{\theta}^T(k)\boldsymbol{\phi}(k) \tag{10.40}$$

$$\boldsymbol{\varphi}(k) = \frac{\partial e(k)}{\partial \boldsymbol{\theta}(k)} \tag{10.41}$$

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \mu\boldsymbol{\varphi}(k)e(k) \tag{10.42}$$

These are the steps of a gradient-based algorithm for IIR filtering. The computational complexity is much lower in gradient-based algorithm than in the Gauss-Newton algorithm. With the latter, however, faster convergence is in general achieved.

10.5 ALTERNATIVE ADAPTIVE FILTER STRUCTURES

The direct-form structure is historically the most widely used realization for the IIR adaptive filter. The main advantages of the direct form are the minimum number of multiplier coefficients required to realize a desired transfer function and the computationally efficient implementation for the gradient (which is possible under the assumption that the denominator coefficients are slowly varying, as illustrated in Fig. 10.3). On the other hand, the stability monitoring of the direct form is difficult because it requires either the factorization of a high-order denominator polynomial in each algorithm step or the use of a sophisticated stability test. In addition, the coefficient sensitivities and output quantization noise are known to be high in the direct form [2].

Alternate solutions are the cascade and parallel realizations using first- or second-order sections as building blocks [9]-[10]. Also, the lattice structures are popular in the implementation of adaptive filters [13]-[19]. All these structures allow easy stability monitoring while the parallel form appears to be most efficient in the gradient computation. The standard parallel realization, however, may converge slowly if two poles approach each other, as will be discussed later and, when a Newton-based algorithm is employed, the estimated Hessian matrix becomes ill-conditioned bringing convergence problems. This problem can be alleviated by applying a preprocessing to the input signal [10]-[11].

10.5.1 Cascade Form

Any Nth-order transfer function can be realized by connecting several first- or second-order sections in series, generating the so-called cascade form. Here we consider that all subfilters are second-order sections without loss of generality, and if an odd-order adaptive filter is required we add a single first-order section. Also, only filters with real multiplier coefficients are discussed. The cascade realization transfer function is given by

$$H_k(z) = \prod_{i=1}^{m} \frac{b_{0i}z^2 + b_{1i}(k)z + b_{2i}(k)}{z^2 + a_{1i}(k)z + a_{2i}(k)} = \prod_{i=1}^{m} H_{ki}(z) \tag{10.43}$$

where m denotes the number of sections.

The parameter vector in the cascade form is

$$\boldsymbol{\theta}(k) = [-a_{11}(k) \; -a_{21}(k) \; b_{01}(k) \; b_{11}(k) \; b_{21}(k) \ldots - a_{1m}(k) \; -a_{2m}(k) \; b_{0m}(k) \; b_{1m}(k) \; b_{2m}(k)]^T$$

The transfer function derivatives as related to the multiplier coefficients can be generated by employing the general result of Fig. 10.4. Fig. 10.5 depicts the cascade realization along with the generation of the derivative signals of interest, where the sections were realized through the direct form of Fig. 10.1.

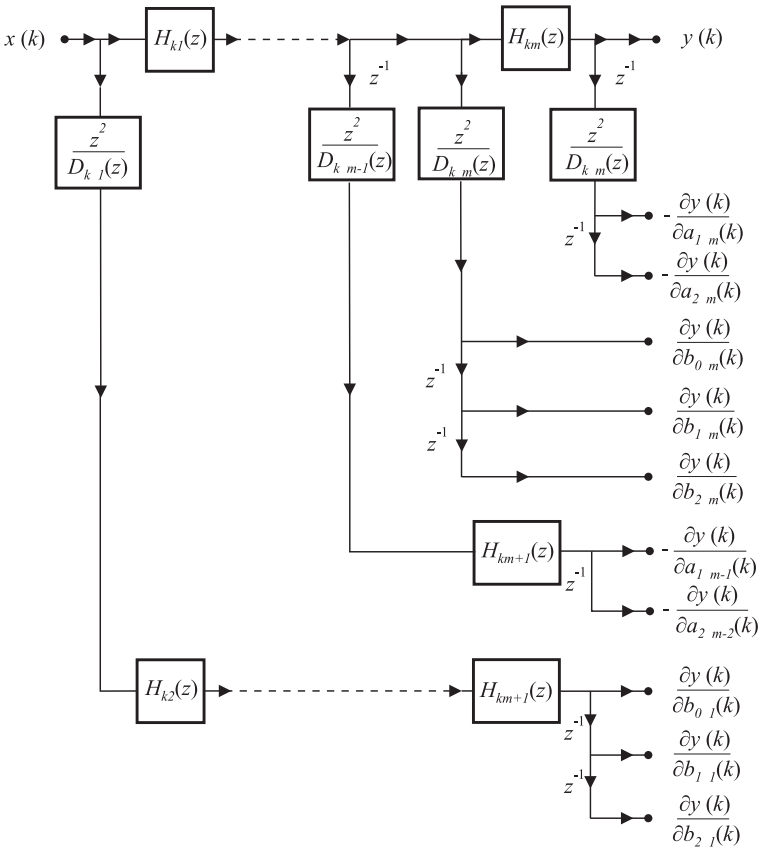


Figure 10.5 Cascade form.

Any alternative second-order section can be used in the cascade form and the appropriate choice depends on a trade-off between quantization effects, hardware resources, computation time, and other factors. The main drawbacks of the cascade form are the amount of extra computations required to generate the gradients, and the manifolds (see sections 10.6 and 10.7) generated on the error surface which may result in slow convergence of the gradient-based algorithms.

10.5.2 Lattice Structure

In this subsection we discuss the lattice algorithm starting from its realization. Although this might appear to be a recipe approach, the development presented here allows us to access the nice properties of the lattice realization. The book by Regalia [12] provides a detailed presentation of the various forms of lattice realization.

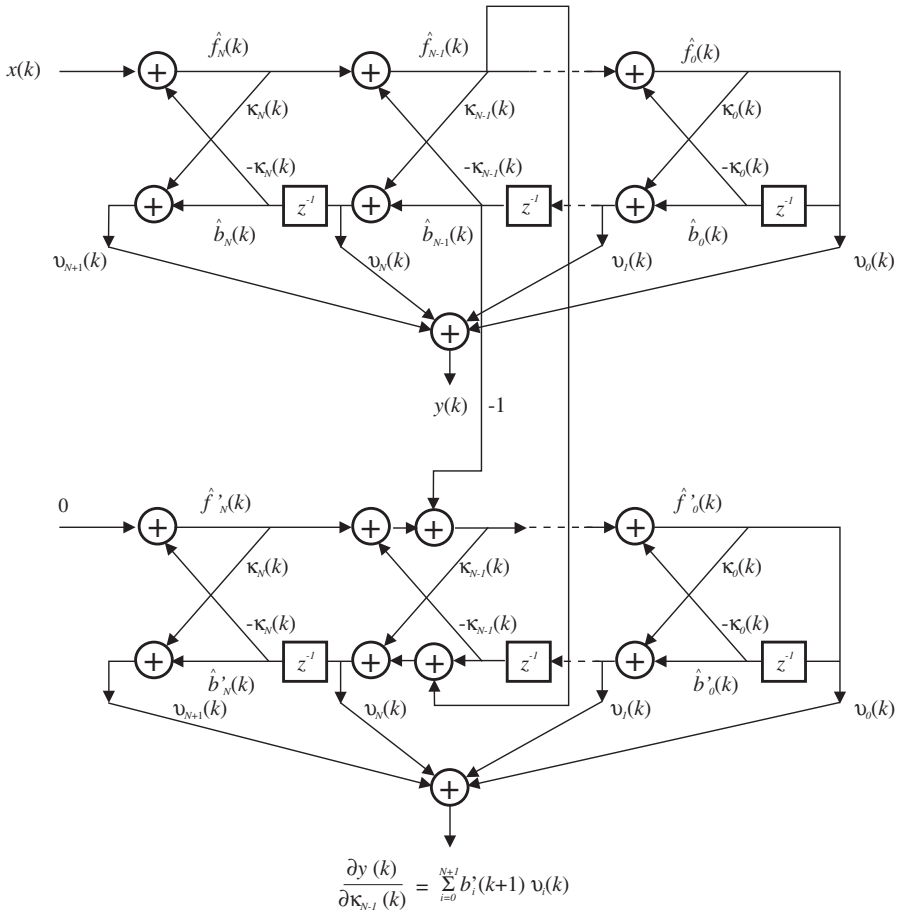


Figure 10.6 Lattice structure including a sample of gradient computation.

The two-multiplier lattice structure [13]-[18] for IIR filters is depicted in Fig. 10.6 with a sample of gradient computation. The coefficients $\kappa_i(k)$ in the recursive part of the structure are called reflection coefficients. The internal signals $\hat{f}_i(k)$ and $\hat{b}_i(k)$ are the forward and backward residuals,

respectively. These internal signals are calculated as follows:

$$\begin{aligned}\hat{f}_{N+1}(k) &= x(k)\\ \hat{f}_{N-i}(k) &= \hat{f}_{N-i+1}(k) - \kappa_{N-i}(k)\hat{b}_{N-i}(k)\\ \hat{b}_{N-i+1}(k+1) &= \kappa_{N-i}(k)\hat{f}_{N-i}(k) + \hat{b}_{N-i}(k)\end{aligned}$$

for $i = 0, 1, \ldots, N$, and

$$\hat{b}_0(k+1) = \hat{f}_0(k) \tag{10.44}$$

The zero placement is implemented by a weighted sum of the backward residuals $\hat{b}_i(k)$, generating the filter output according to

$$y(k) = \sum_{i=0}^{N+1} \hat{b}_i(k+1)v_i(k) \tag{10.45}$$

where $v_i(k)$, for $i = 0, 1, \ldots, N+1$, are the output coefficients.

The derivatives of the filter output $y(k)$ with respect to the output tap coefficients $v_i(k)$ are given by the backward residuals $\hat{b}_i(k+1)$. On the other hand, the derivatives of $y(k)$ as related to the reflection multiplier coefficients $\kappa_i(k)$ require one additional lattice structure for each $\kappa_i(k)$. In Fig. 10.6, the extra lattice required to calculate $\frac{\partial y(k)}{\partial \kappa_{N-1}(k)}$ is shown for illustration. The overall structure for the calculation of the referred partial derivative can be obtained by utilizing the general derivative implementation of Fig. 10.4.b. First note that the transfer functions from the filter input to the inputs of the multipliers $\pm\kappa_{N-1}(k)$ were realized by the original adaptive lattice filter. Next, the overall partial derivative is obtained by taking the input signals of $\pm\kappa_{N-1}(k)$ in the first lattice structure to their corresponding output nodes in a second lattice structure whose external input is zero. For each derivative $\frac{\partial y(k)}{\partial \kappa_j(k)}$, the following algorithm must be used

$$\begin{aligned}\hat{f}'_{N+1}(k) &= 0\\ \text{If } i &\neq N-j\\ \hat{f}'_{N-i}(k) &= \hat{f}'_{N-i+1}(k) - \kappa_{N-i}(k)\hat{b}'_{N-i}(k)\\ \hat{b}'_{N-i+1}(k+1) &= \kappa_{N-i}(k)\hat{f}'_{N-i}(k) + \hat{b}'_{N-i}(k)\\ \text{for } i &= 0, 1, \ldots, N-j-1, N-j+1, \ldots, N\\ \text{If } i &= N-j\\ \hat{f}'_j(k) &= \hat{f}'_{j+1}(k) - \kappa_j(k)\hat{b}'_j(k) - \hat{b}_j(k)\\ \hat{b}'_{j+1}(k+1) &= \kappa_j(k)\hat{f}'_j(k) + \hat{b}'_j(k) + \hat{f}_j(k)\\ \hat{b}'_o(k+1) &= \hat{f}_o(k)\\ &\text{Then}\\ \frac{\partial y(k)}{\partial \kappa_j(k)} &= \sum_{i=0}^{N+1} \hat{b}'_i(k+1)v_i(k)\end{aligned} \tag{10.46}$$

The main desirable feature brought about by the lattice IIR realization is the simple stability test. The stability requires only that reflection coefficients $\kappa_i(k)$ be maintained with modulus less than

one [17]. However, the gradient computations are extremely complex, and of order N^2 in terms of multiplication count. An approach for the gradient computations with order N multiplications and divisions was proposed [16], which is still more complex than for the direct-form realization. It should be noticed that in the direct form, all the signals at the multiplier's input are delayed versions of each other, and the transfer function from the multiplier's output to the filter output are the same. These properties make the gradient computational complexity in the direct form low. The lattice IIR realization does not have these features.

When the two-multiplier lattice structure is realizing a transfer function with poles close to the unit circle, the internal signals may present a large dynamic range, resulting in poor performance due to quantization effects. In this case, the normalized lattice [19] is a better choice despite its higher computational complexity. There are alternative lattice structures, such as the two-multiplier with distinct reflection coefficients and the one-multiplier structures [15], that can also be employed in adaptive filtering. For all these options the stability test is trivial, retaining the main feature of the two-multiplier lattice structure.

An application where adaptive IIR filtering is the natural choice is sinusoid detection using notch filters. A notch transfer function using direct-form structure is given by

$$H_{\mathrm{N}}(z) = \frac{1 - 2\cos\omega_0 z^{-1} + z^{-2}}{1 - 2r\cos\omega_0 z^{-1} + r^2 z^{-2}} \tag{10.47}$$

where ω_0 is the notch frequency and r is the pole radius [20]. The closer the pole radius is to the unit circle the narrower is the notch transfer function, leading to better estimate of the sinusoid frequency in a noisy environment. However, in the direct form the noise gain, caused by the notch transfer function, varies with the sinusoid frequency, causing a bias in the frequency estimate [12].

An alternative is to construct a notch filter by using a lattice structure. A second-order notch filter can be generated by

$$H_{\mathrm{N}}(z) = \frac{1}{2}\left[1 + H_{\mathrm{AP}}(z)\right] \tag{10.48}$$

where $H_{\mathrm{AP}}(z)$ is an all-pass transfer function which can be realized by a lattice structure by setting $v_2 = 1$ and $v_1 = v_0 = 0$ in Fig. 10.6. In this case,

$$H_{\mathrm{AP}}(z) = \frac{\kappa_1 + \kappa_0(1+\kappa_1)z^{-1} + z^{-2}}{1 + \kappa_0(1+\kappa_1)z^{-1} + \kappa_1 z^{-2}} \tag{10.49}$$

The notch frequency ω_0 and the relation between -3 dB attenuation bandwidth $\Delta\omega_{3\mathrm{dB}}$ and κ_1 are given by

$$\omega_0 = \cos^{-1}(-\kappa_0) \tag{10.50}$$

and

$$\kappa_1 = \frac{1 - \tan\frac{\Delta\omega_{3\mathrm{dB}}}{2}}{1 + \tan\frac{\Delta\omega_{3\mathrm{dB}}}{2}} \tag{10.51}$$

respectively. The main feature of the notch filter based on the lattice structure is the independent control of the notch frequency and the -3 dB attenuation bandwidth.

It is worth mentioning that an enhanced version of the sinusoid signal can be obtained by applying the noisy input signal to the bandpass filter whose transfer function is given by

$$H_{\mathrm{BP}}(z) = \frac{1}{2}\left[1 - H_{\mathrm{AP}}(z)\right] \tag{10.52}$$

For identification of multiple sinusoids the most widely used structure is the cascade of second-order sections, where each section identifies one of the sinusoids removing the corresponding sinusoid from the input to the following sections.

Sinusoid detection in noise utilizing adaptive notch filter has rather simple implementation as compared with other methods, and finds application in synchronization, tone detection and tracking for music signals among others.

Example 10.1

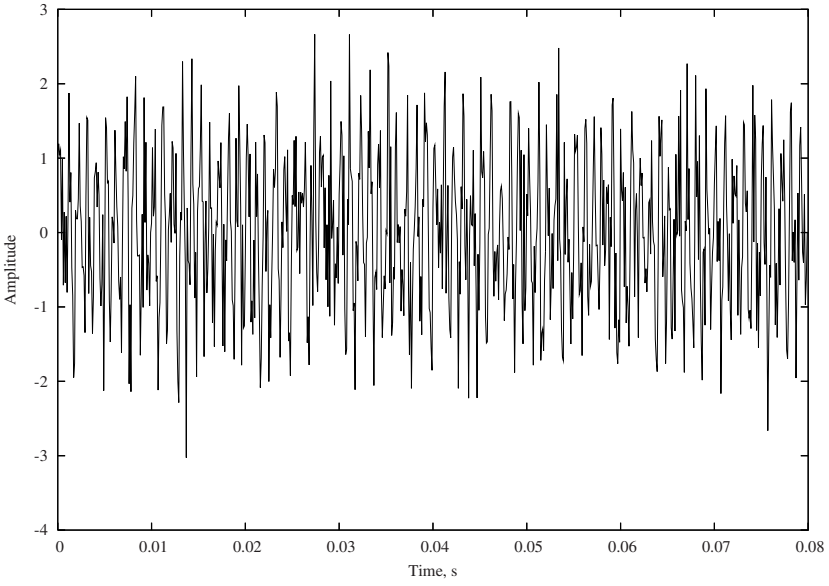
Apply an IIR notch adaptive filter using the second-order lattice structure to detect a sinusoid buried in noise.

The input signal noise is a Gaussian white noise with variance $\sigma_x^2 = 1$, whereas the sampling frequency is 10000Hz and the sinusoid to be detected is at 1000Hz. Use a gradient-based algorithm.
(a) Choose the appropriate value of μ.
(b) Run the algorithm using for signal to noise ratios of 0 and -5dB, respectively.
(c) Show the learning curves for the detected frequency, the input and the bandpass filtered output signal.

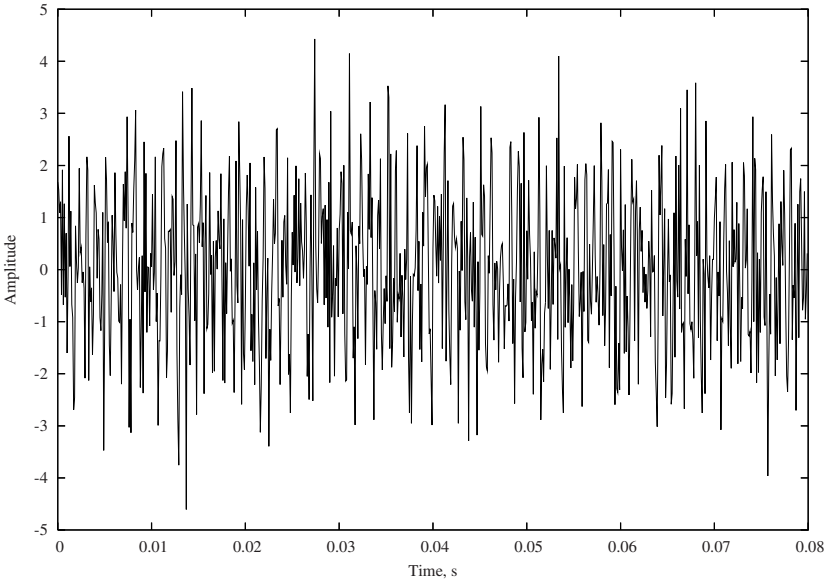
Solution:

A rather small convergence factor $\mu = 0.000001$ is used in this example. Higher values can be used for lower ratio between the sampling frequency and the sinusoid frequency. The starting search frequency is 1100Hz. A quality factor of 10 is used, where this factor measures ratio between the notch frequency and the frequencies with -3dB of attenuation with respect to the gain in the pass band of filter. The stopband width is then 100 Hz. Figs. 10.7.a and 10.7.b depict the input signals for the cases where the signal to noise ratios are 0 and -5 dB's, respectively. Figs. 10.8.a and 10.8.b show the learning curves for the sinusoid frequencies where in both cases the correct frequencies are detected in less than one second which is equivalent to 1000 iterations. As can be observed, the noisier input leads to noisier output. Figs. 10.9.a and 10.9.b depict the bandpass output signal where the sinusoidal components are clearly seen, and again the higher signal to noise ratio results in cleaner sinusoids. In these plots we froze the value of κ_0 at a given iteration after convergence in order to generate the band-passed signals.

□

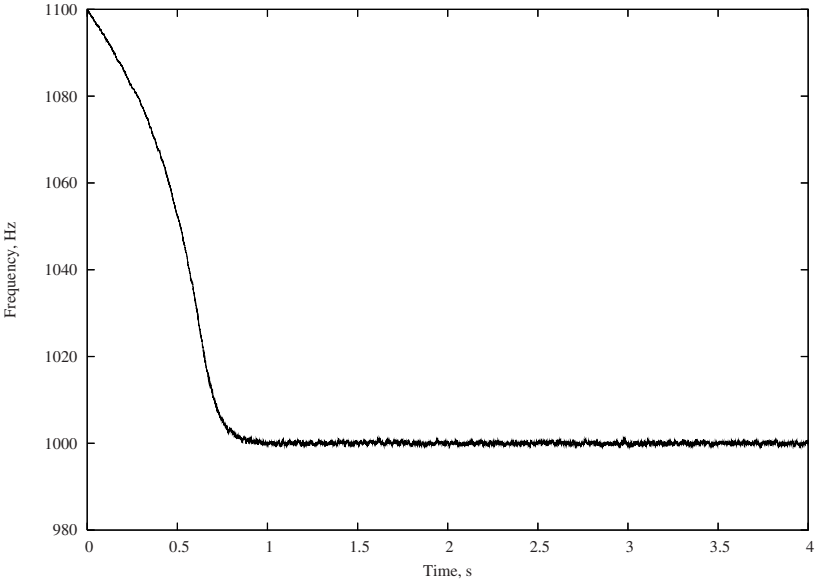


(a)

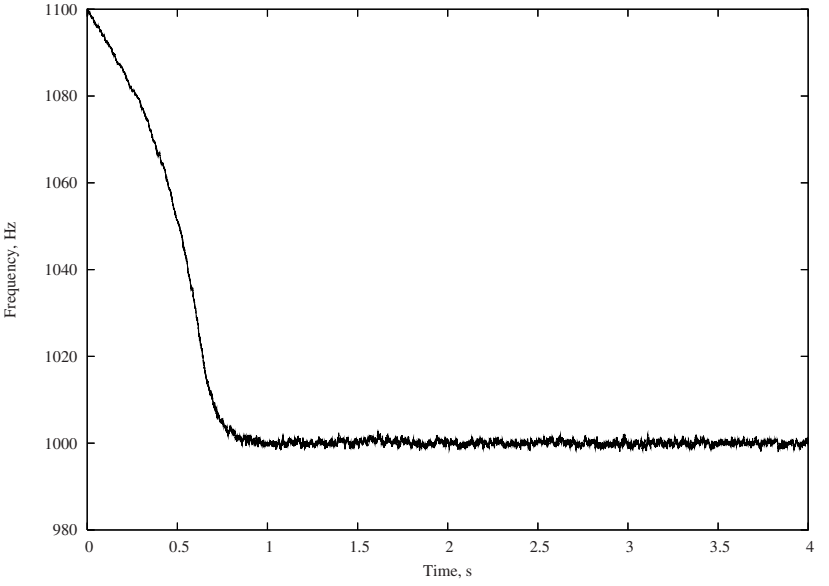


(b)

Figure 10.7 Sinusoid buried in noise for signal to noise ratio (a) 0dB, (b) −5dB.

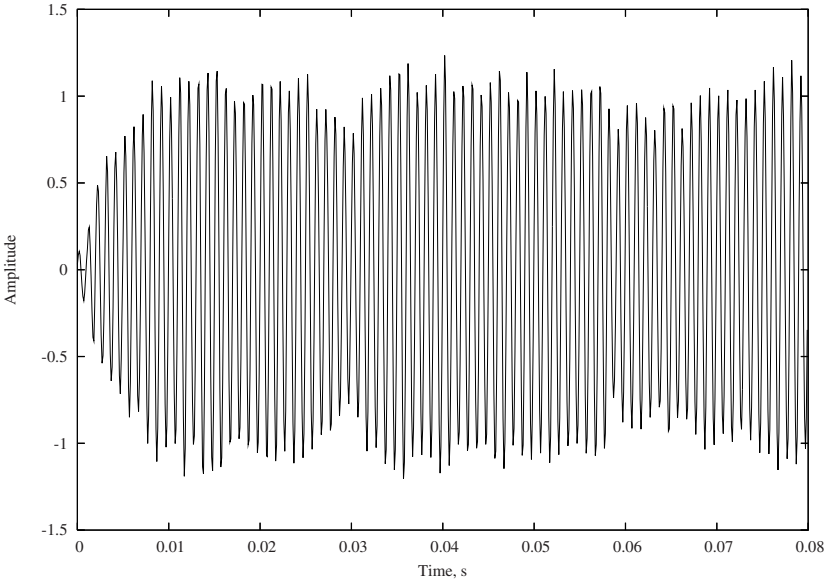


(a)

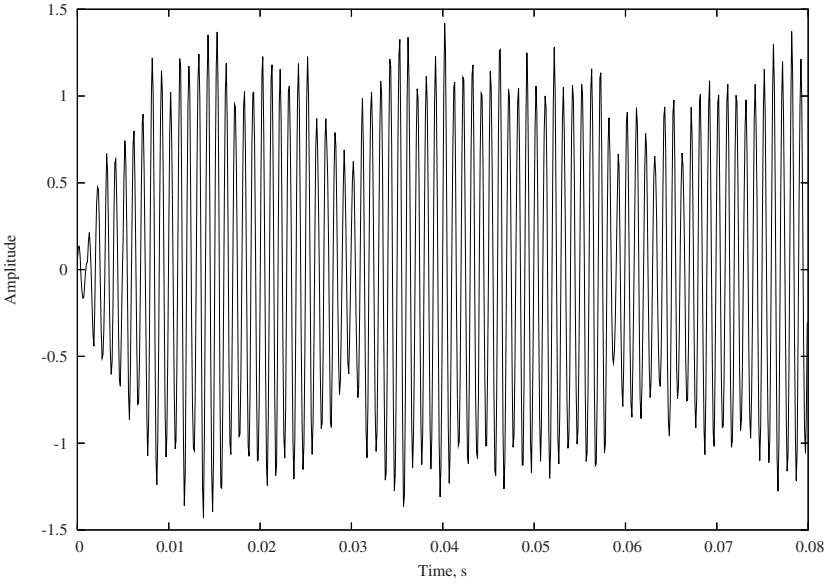


(b)

Figure 10.8 Learning curves of the sinusoid frequency (a) 0dB, (b) -5dB.



(a)



(b)

Figure 10.9 Band-passed output signals (a) 0dB, (b) −5dB.

10.5.3 Parallel Form

In the parallel realization, the transfer function is realized by a parallel connection of sections as shown in Fig. 10.10. The sections are in most of the cases of first- or second-order, making the stability test trivial. The transfer function when second-order sections are employed is given by

$$H_k(z) = \sum_{i=0}^{m-1} \frac{b_{0i}(k)z^2 + b_{1i}(k)z + b_{2i}(k)}{z^2 + a_{1i}(k)z + a_{2i}(k)} \tag{10.53}$$

The parameter vector for the parallel form is

$$\begin{aligned} \boldsymbol{\theta}(k) \;=\; & [-a_{10}(k) \; -a_{20}(k)\; b_{00}(k)\; b_{10}(k)\; b_{20}(k) \\ & \ldots - a_{1\,m-1}(k) \; -a_{2\,m-1}(k)\; b_{0\,m-1}(k)\; b_{1\,m-1}(k)\; b_{2\,m-1}(k)]^T \end{aligned} \tag{10.54}$$

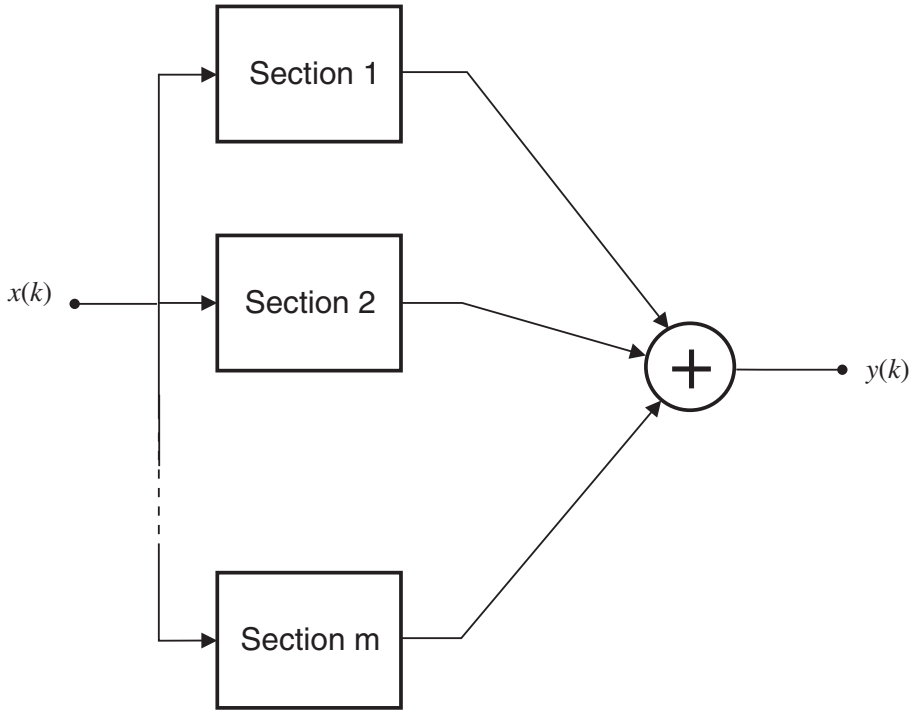


Figure 10.10 Parallel form.

The transfer function derivatives as related to the multiplier coefficients in the parallel form are simple to calculate, because they depend on the derivative of the individual section transfer function with respect to the multiplier coefficients belonging to that section. Basically, the technique of Fig. 10.4 can be applied to each section individually.

Since the interchange of sections in the parallel form does not alter the transfer function, there are $m!$ global minimum points each located in separate subregions of the MSE surface. These subregions are separated by boundaries that are reduced-order manifolds as will be discussed in section 10.7. These boundaries contain saddle points and if the filter parameters are initialized on a boundary, the convergence rate is most probably slow. Consider that the internal signals cross-correlation matrix is approximately estimated by

$$\hat{\mathbf{R}}(k+1) = \alpha \sum_{i=0}^{k} (1-\alpha)^{k-i} \boldsymbol{\varphi}(i) \boldsymbol{\varphi}^T(i) \tag{10.55}$$

when k is large. In this case, if the sections coefficients are identical the information vector consists of a set of identical subvectors $\boldsymbol{\varphi}(i)$, which in turn makes $\hat{\mathbf{R}}(k+1)$ ill-conditioned. The above discussion suggests that the sections in the parallel realization should be initialized differently, although there is no guarantee that this will avoid the ill-conditioning problems.

10.5.4 Frequency-Domain Parallel Structure

A possible alternative parallel realization first proposed in [10] incorporates a preprocessing of the input signal using a discrete-time Fourier transform, generating m signals that are individually applied as input to first-order complex-coefficients sections. With this strategy, the matrix $\hat{\mathbf{R}}(k)$ is more unlikely to become ill-conditioned. Also, it is more difficult for a gradient-based algorithm to get stuck on a reduced-order manifold, resulting in faster convergence. The parallel realization can also be implemented using a real-coefficient transform for the preprocessing and second-order sections.

The frequency-domain parallel structure is illustrated in Fig. 10.11, where $d(k)$ is the reference signal, $x(k)$ is the input signal, $n(k)$ is an additive noise source, and $y(k)$ is the output. The ith parallel section is represented by the transfer function

$$H_i(z) = \frac{b_{0i}(k)z^2 + b_{1i}(k)z + b_{2i}(k)}{z^2 + a_{1i}(k)z + a_{2i}(k)} \qquad k = 0, 1, \ldots, m-1 \tag{10.56}$$

where $a_{1i}(k), a_{2i}(k), b_{0i}(k), b_{1i}(k)$, and $b_{2i}(k)$ are adjustable real coefficients. The inputs of the filter sections are preprocessed as shown in Fig. 10.11.

The purpose of preprocessing in Fig. 10.11 is to generate a set of uncorrelated signals $x_1(k), x_2(k), \ldots, x_m(k)$ in order to reduce the probability that two or more sections converge to the same solution, to simplify the adaptation algorithm, and to improve the rate of convergence.

On employing the discrete-time cosine transform (DCT), the input signals to the subfilters in Fig. 10.11 are given by

$$x_0(k) = \frac{\sqrt{2}}{m} \sum_{l=0}^{m-1} x(k-l)$$

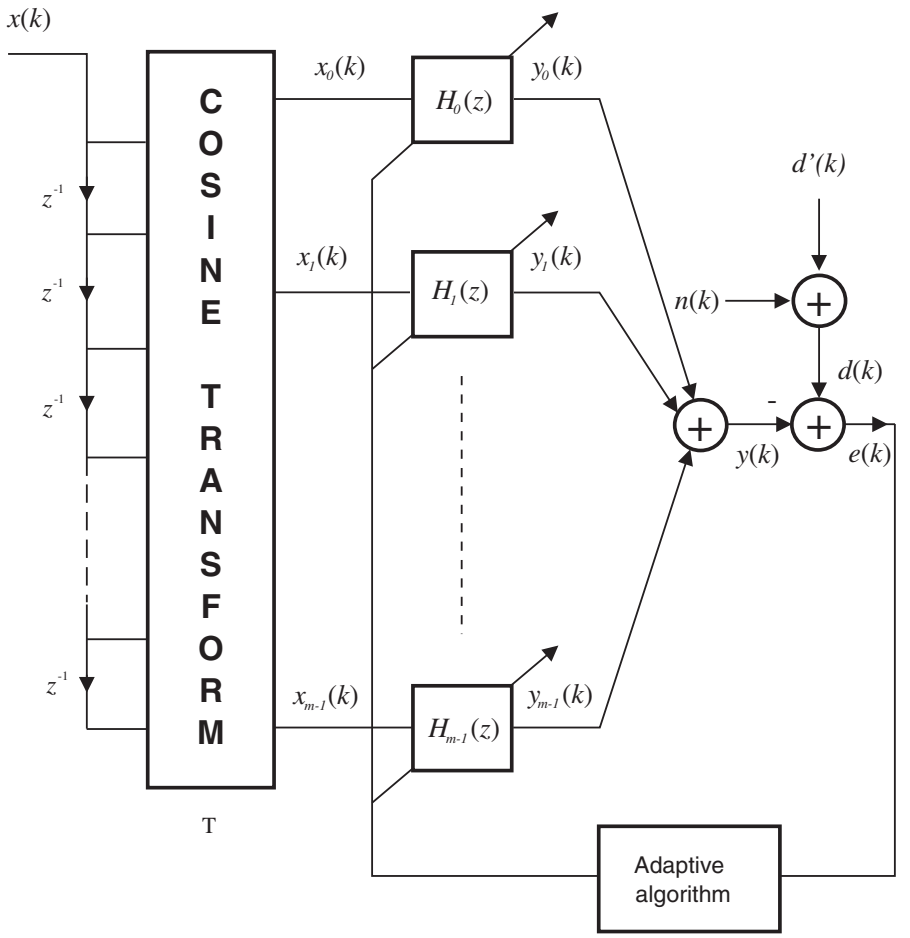


Figure 10.11 Real coefficient frequency-domain adaptive filter.

and

$$x_i(k) = \sqrt{\frac{2}{m}} \sum_{l=0}^{m-1} x(k-l) \cos[\pi i(2l+1)/(2m)] \tag{10.57}$$

The transfer function from the input to the outputs of the DCT preprocessing filter (or prefilter) can be described through the recursive frequency-domain description given by

$$T_i(z) = \frac{k_0}{m} \cos \tau_i \frac{[z^m - (-1)^i](z-1)}{z^{m-1}[z^2 - (2\cos 2\tau_i)z + 1]} \tag{10.58}$$

where

$$k_0 = \begin{cases} \sqrt{2} & \text{if } i = 0 \\ \sqrt{2m} & \text{if } i = 1, 2, \ldots, m-1 \end{cases}$$

and $\tau_i = \pi i/(2m)$. The DCT can be efficiently implemented through some fast algorithms, or by employing equation (10.58). In the latter case, special consideration must be given to the poles on the unit circle.

Alternatively, the transfer functions of the prefilter can be expressed as

$$T_i(z) = \frac{1}{m}\sum_{j=0}^{m-1} t_{ij} z^{-j} = \frac{1}{m}\prod_{r=0}^{m-2}\frac{(z-\tau_{ir})}{z} = \frac{1}{z^{m-1}}\frac{(z-1)[z^m-(-1)^i]}{[z^2-(2\cos\frac{\pi i}{m})z+1]} \tag{10.59}$$

where the t_{ij} are the coefficients of the transform matrix $\mathbf{T}$, and the τ_{ir} are the zeros of $T_i(z)$. The gain constants k_0 and $\cos\tau$ were dropped in equation (10.59) and will not be considered from now on, since they can be absorbed by the numerator coefficients $b_{0i}(k),\ b_{1i}(k)$, and $b_{2i}(k)$ of $H_i(z)$.

The overall transfer function of the frequency-domain adaptive filter of Fig. 10.11 is given by

$$\begin{aligned} H(z) &= \sum_{i=0}^{m-1} T_i(z)H_i(z) \\ &= \frac{1}{m}\left(\frac{1}{z^{m-1}}\right)\left[\sum_{i=0}^{m-1}\left(\frac{b_{0i}z^2+b_{1i}z+b_{2i}}{z^2+a_{1i}z+a_{2i}}\right)\prod_{r=0}^{m-2}(z-\tau_{ir})\right] \\ &= \frac{1}{m}\frac{1}{z^{3m+1}}\left[\sum_{i=0}^{m-1}(b_{0i}z^2+b_{1i}z+b_{2i})\frac{\prod_{j=0,\neq i}^{m-1}(z^2+a_{1j}z+a_{2j})\prod_{r=0}^{m-2}(z-\tau_{ir})}{\prod_{l=0}^{m-1}(z^2+a_{1l}z+a_{2l})}\right] \end{aligned} \tag{10.60}$$

Now assume that the realization discussed is used to identify a system of order $2N_p$ described by

$$H_D(z) = Kz^{2N_p-P}\frac{\prod_{r=0}^{P-1}(z-\gamma_r)}{\prod_{i=0}^{N_p-1}(z^2+\alpha_{1i}z+\alpha_{2i})} \tag{10.61}$$

where K is a gain constant, p_{0i} and p_{1i} are the poles of section i, and γ_r are the zeros of $H_D(z)$ such that

$$\gamma_r \neq p_{0i}, p_{1i} \text{ for } r = 0, \ldots, P-1 \text{ and for } i = 0, \ldots, N_p - 1$$

It can be shown that if the conditions outlined below are satisfied, the filter of Fig. 10.11 can identify exactly systems with $N_p \leq m$ and $P \leq 3m+1$. The sufficient conditions are:

i) The transformation matrix $\mathbf{T}$ of the prefilter is square and has linearly independent rows.

ii) $a_{1i} \neq a_{1j}$, and $a_{2i} \neq a_{2j}$ for $i \neq j$; a_{1i} and a_{2i} are not simultaneously zero for all i.

iii) The zeros of the prefilter do not coincide with the system's poles, i.e., $\tau_{ij} \neq p_{0l},\ \tau_{ij} \neq p_{1l}$, for all i, j, and l.

Adaptation Algorithm

The adaptation algorithm entails the manipulation of a number of vectors, namely, the coefficient vector

$$\boldsymbol{\theta}(k) = \left[\boldsymbol{\theta}_0^T(k) \ldots \boldsymbol{\theta}_{m-1}^T(k)\right]^T$$

where

$$\boldsymbol{\theta}_i(k) = [-a_{1i}(k)\ -a_{2i}(k)\ b_{0i}(k)\ b_{1i}(k)\ b_{2i}(k)]^T$$

the internal data vector

$$\boldsymbol{\phi}(k) = \left[\boldsymbol{\phi}_0^T(k) \ldots \boldsymbol{\phi}_{m-1}^T(k)\right]^T$$

where

$$\boldsymbol{\phi}_i(k) = [y_i(k-1)\ y_i(k-2)\ x_i(k)\ x_i(k-1)\ x_i(k-2)]^T$$

the gradient vector

$$\tilde{\boldsymbol{\varphi}}(k) = [\boldsymbol{\varphi}_0^T(k) \ldots \boldsymbol{\varphi}_{m-1}^T(k)]^T$$

where

$$\boldsymbol{\varphi}_i(k) = [-y_i'(k-1)\ -y_i'(k-2)\ -x_i'(k)\ -x_i'(k-1)\ -x_i'(k-2)]^T$$

and the matrix $\hat{\mathbf{S}}(k)$ which is an estimate of the inverse Hessian $\hat{\mathbf{R}}^{-1}(k)$.

The elements of the gradient vector can be calculated by using the relations

$$x_i'(k) = x_i(k) - a_{1i}(k)x_i'(k-1) - a_{2i}(k)x_i'(k-2)$$

and

$$y_i'(k) = -y_i(k) - a_{1i}(k)y_i'(k-1) - a_{2i}(k)y_i'(k-2)$$

An adaptation algorithm for updating the filter coefficients based on the Gauss-Newton algorithm is summarized in Algorithm 10.2. The algorithm includes the updating of matrix $\hat{\mathbf{S}}(k)$, which is obtained through the matrix inversion lemma.

The stability monitoring consists of verifying whether each set of coefficients $a_{1i}(k)$ and $a_{2i}(k)$ defines a point outside the stability triangle [2], i.e., by testing whether

$$1 - a_{1i}(k) + a_{2i}(k) < 0 \quad \text{or} \quad 1 + a_{1i}(k) + a_{2i}(k) < 0 \quad \text{or} \quad |a_{2i}(k)| \geq 1 \tag{10.62}$$

If instability is detected in a particular section, the poles must be projected back inside the unit circle. A possible strategy is to project each pole by keeping its angle and inverting its modulus. In this case, a_{2i} and a_{1i} should be replaced by $1/a_{2i}(k)$ and $-a_{1i}(k)/a_{2i}(k)$, respectively.

Algorithm 10.2

Frequency-Domain Parallel Algorithm, RLS Version

Initialization

$\hat{\mathbf{S}}(0) = \delta\mathbf{I}(\delta > 0)$

$\boldsymbol{\theta}_i(k), 0 \le i \le m-1$

For each $x(k)$ and $d(k)$ given for $k \ge 0$, compute:

$X_{\mathrm{DCT}}(k) = \mathrm{DCT}[x(k) \dots x(k-m+1)]$

Do for $i = 0, 1, \dots, m-1$:

$x_i'(k) = x_i(k) - a_{1i}(k)x_i'(k-1) - a_{2i}(k)x_i'(k-2)$

$y_i(k) = \boldsymbol{\theta}_i^T(k)\boldsymbol{\phi}_i(k)$

$y_i'(k) = -y_i(k) - a_{1i}(k)y_i'(k-1) - a_{2i}(k)y_i'(k-2)$

End

$e(k) = d(k) - \sum_{i=0}^{m-1} y_i(k)$

$\boldsymbol{h}(k) = \hat{\mathbf{S}}(k)\tilde{\boldsymbol{\varphi}}(k)$

$$\hat{\mathbf{S}}(k+1) = \left[\hat{\mathbf{S}}(k) - \frac{\boldsymbol{h}(k)\boldsymbol{h}^T(k)}{(\frac{1}{\alpha}-1)+\boldsymbol{h}^T(k)\tilde{\boldsymbol{\varphi}}(k)}\right](\frac{1}{1-\alpha})$$

$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \mu\hat{\mathbf{S}}(k+1)\tilde{\boldsymbol{\varphi}}(k)e(k)$

Carry out stability test.
End

If the outputs of the DCT prefilter $x_i(k)$ are sufficiently uncorrelated, the Hessian matrix is approximately block-diagonal consisting of 5×5 submatrices $\hat{\mathbf{R}}_i(k)$. In this case, instead of computing a $5m \times 5m$ inverse Hessian estimate $\hat{\mathbf{S}}(k)$, several 5×5 submatrices are computed and applied in the above algorithm as follows:

For $i = 0, 1, \ldots, m-1$

$$\boldsymbol{h}_i(k) = \hat{\mathbf{S}}_i(k)\boldsymbol{\varphi}_i(k)$$

$$\hat{\mathbf{S}}_i(k+1) = \left[\hat{\mathbf{S}}_i(k) - \frac{\boldsymbol{h}_i(k)\boldsymbol{h}_i^T(k)}{(\frac{1}{\alpha}-1) + \boldsymbol{h}_i^T(k)\boldsymbol{\varphi}_i(k)}\right](\frac{1}{1-\alpha})$$

$$\boldsymbol{\theta}_i(k+1) = \boldsymbol{\theta}_i(k) - \mu\hat{\mathbf{S}}_i(k+1)\boldsymbol{\varphi}_i(k)e(k)$$

□

The choice of the adaptive-filter realization has implications on the computational complexity as well as on the convergence speed. Some studies exploring this aspect related to the frequency-domain realization can be found in [21]. The exploration of realization related properties of the IIR adaptive MSE surface led to a fast parallel realization where no transform preprocessing is required [22]. In this approach, the reduced-order manifolds are avoided by properly configuring the parallel sections which are implemented with general purpose second-order sections [23]. An analysis of the asymptotic convergence speed of some adaptive IIR filtering algorithms from the realization point of view can be found in [24]. Another approach proposes a cascade/parallel orthogonal realization, with simplified gradient computation, by utilizing some of the ideas behind the derivation of improved parallel realizations [25].

Example 10.2

An IIR adaptive filter of sufficient order is used to identify a system with the transfer function given below.

$$H(z) = \frac{0.8(z^2 - 1.804z + 1)^2}{(z^2 - 1.512z + 0.827)(z^2 - 1.567z + 0.736)}$$

The input signal is a uniformly distributed white noise with variance $\sigma_x^2 = 1$, and the measurement noise is Gaussian white noise uncorrelated with the input with variance $\sigma_n^2 = 10^{-1.5}$. Use a gradient-based algorithm.
(a) Choose the appropriate values of μ.
(b) Run the algorithm using the direct-form structure, the lattice structure, the parallel realization with preprocessing, and the cascade realization with direct-form sections. Compare their convergence speed.
(c) Measure the MSE.
(d) Plot the obtained IIR filter frequency response at any iteration after convergence is achieved and compare with the unknown system. Consider for this item only the direct-form realization.

Solution:

A convergence factor $\mu = 0.004$ is used in all examples, except for the lattice realization where $\mu = 0.0002$ is employed for the internal coefficients and a larger $\mu = 0.002$ is employed for the updating of the feedforward coefficients, for stability reasons. Although the chosen value of μ is

not an optimal value in any sense, it led to the convergence of all algorithms. Fig. 10.12 depicts the magnitude response of the adaptive filter using the direct form at a given iteration after convergence. For comparison the magnitude response of the system being modeled is also plotted. As can be seen, the responses are close outside the frequency range where the unknown system has a notch. Fig. 10.13 shows the learning curves of the algorithms obtained by averaging the results of 200 independent runs. As can be seen the faster algorithms led to higher MSE. The cascade realization presented faster convergence, followed by the parallel and lattice realizations. The measured MSEs are given in Table 10.1.

There are very few results published in the literature addressing the finite-precision implementation of IIR adaptive filters. For this particular example, all algorithms are also implemented with fixed point arithmetic, with 12 and 16 bits. No sign of divergence is detected during the early 2000 iterations. However, the reader should not take this result as conclusive.

□

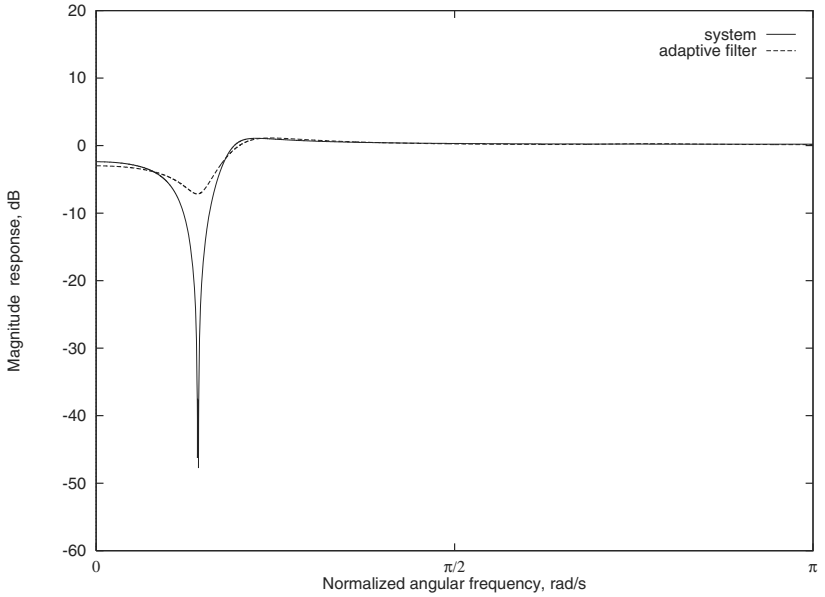
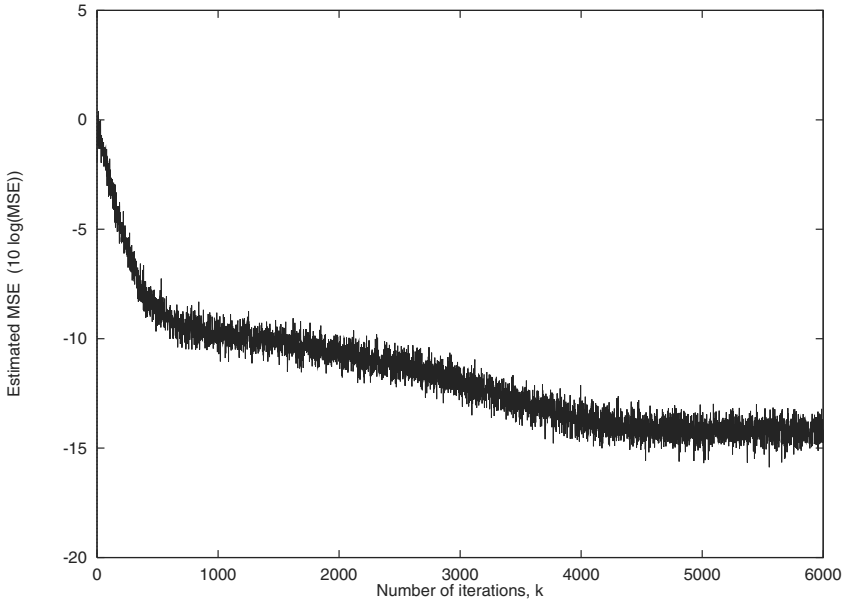
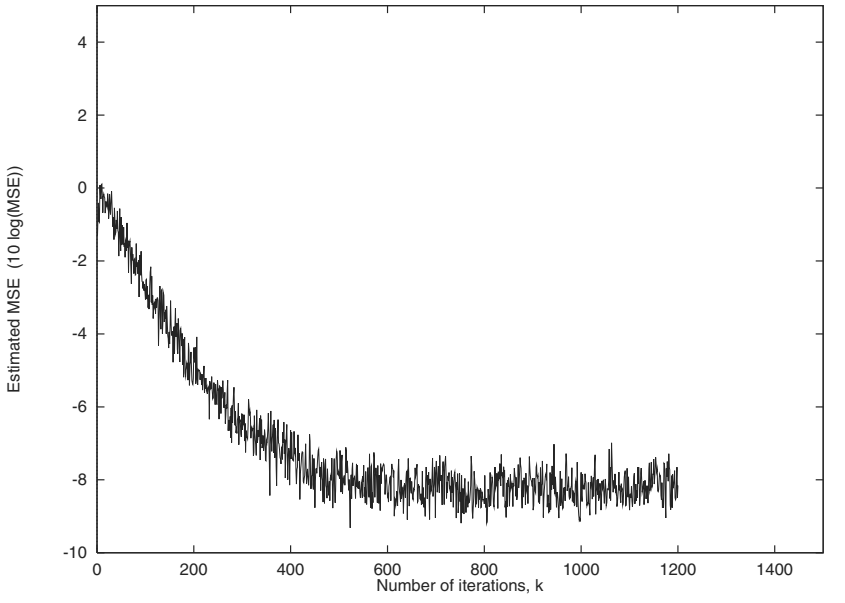


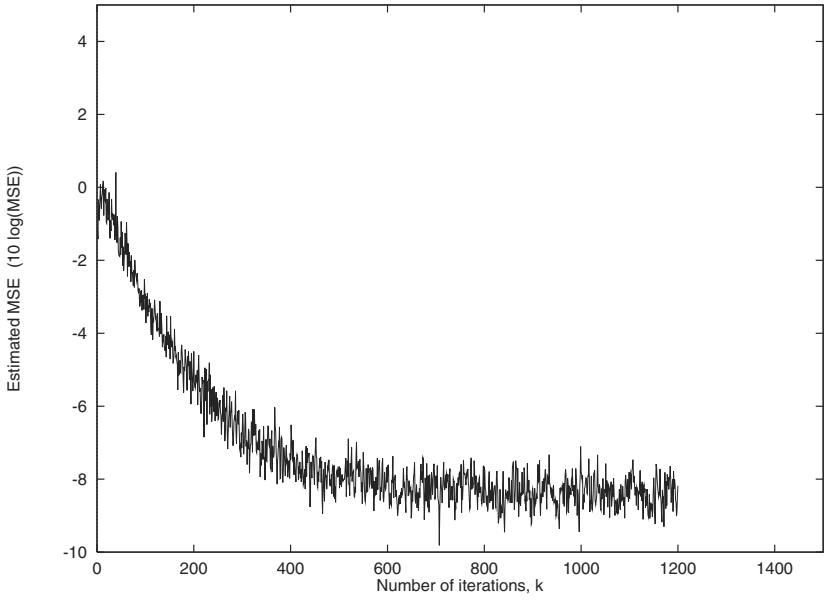
Figure 10.12 Magnitude response of the IIR adaptive filter with direct form at a given iteration after convergence.


Estimated MSE (10 log(MSE))
Number of iterations, k
5
0
-5
-10
-15
-20
0
1000
2000
3000
4000
5000
6000

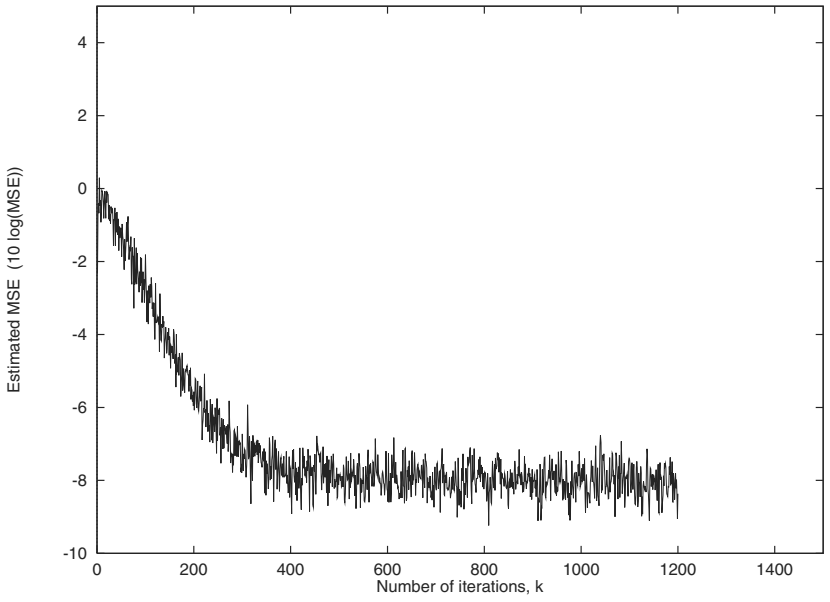
(a)


Estimated MSE (10 log(MSE))
Number of iterations, k
4
2
0
-2
-4
-6
-8
-10
0
200
400
600
800
1000
1200
1400

(b)



(c)



(d)

Figure 10.13 Learning curves for IIR adaptive filters with (a) Direct form, (b) Parallel form with preprocessing, (c) Lattice, and (d) Cascade realizations.

Table 10.1 Evaluation of the IIR Algorithms

Realization	MSE
Direct Form	0.0391
Lattice	0.1514
Transf. Dom. Parallel	0.1478
Cascade	0.1592

10.6 MEAN-SQUARE ERROR SURFACE

The error surface properties in the case of adaptive IIR filtering are key in understanding the difficulties in applying gradient-based algorithms to search for the optimal filter coefficient vector. In this section, the main emphasis is given to the system identification application where the unknown system is modeled by

$$d(k) = \frac{G(q^{-1})}{C(q^{-1})}x(k) + n(k) \tag{10.63}$$

where

$$G(q^{-1}) = g_0 + g_1 q^{-1} + \cdots + g_{M_d} q^{-M_d}$$
$$C(q^{-1}) = 1 + c_1 q^{-1} + \cdots + c_{N_d} q^{-N_d}$$

and $n(k)$ is the measurement noise that is considered uncorrelated with the input signal $x(k)$.

The unknown transfer function is

$$\begin{aligned} H_o(z) &= z^{N_d - M_d} \frac{g_0 z^{M_d} + g_1 z^{M_d - 1} + \cdots + g_{M_d - 1} z + g_{M_d}}{z^{N_d} + c_1 z^{N_d - 1} + \cdots + c_{N_d - 1} z + c_{N_d}} \\ &= z^{N_d - M_d} \frac{N_o(z)}{D_o(z)} \end{aligned} \tag{10.64}$$

The desired feature of the identification problem is that the adaptive-filter transfer function $H_k(z)$ approximates $H_o(z)$ as much as possible in each iteration. If the performance criterion is the mean-square error (MSE), the objective function is expressed in terms of the input signal and the desired signals as follows:

$$\begin{aligned} \xi &= E[e^2(k)] = E\{[d(k) - y(k)]^2\} \\ &= E[d^2(k) - 2d(k)y(k) + y^2(k)] \\ &= E\left\{ \left[\left(\frac{G(q^{-1})}{C(q^{-1})}x(k) + n(k) \right) - \frac{B(k, q^{-1})}{A(k, q^{-1})}x(k) \right]^2 \right\} \end{aligned} \tag{10.65}$$

Since $n(k)$ is not correlated to $x(k)$ and $E[n(k)] = 0$, equation (10.65) can be rewritten as

$$\xi = E\left\{\left[\left(\frac{G(q^{-1})}{C(q^{-1})} - \frac{B(k,q^{-1})}{A(k,q^{-1})}\right)x(k)\right]^2\right\} + E[n^2(k)] \tag{10.66}$$

The interest here is to study the relation between the objective function ξ and the model filter coefficients, independently if these coefficients are adaptive or not. The polynomials operators $B(k,q^{-1})$ and $A(k,q^{-1})$ will be considered fixed, denoted respectively by $B(q^{-1})$ and $A(q^{-1})$.

The power spectra of the signals involved in the identification process are given by

$$\begin{aligned}
R_{xx}(z) &= \mathcal{Z}[r_{xx}(l)] \\
R_{nn}(z) &= \mathcal{Z}[r_{nn}(l)] \\
R_{dd}(z) &= H_o(z)\, H_o(z^{-1})\, R_{xx}(z) + R_{nn}(z) \\
R_{yy}(z) &= H_k(z)\, H_k(z^{-1})\, R_{xx}(z) \\
R_{dy}(z) &= H_o(z)\, H_k(z^{-1})\, R_{xx}(z)
\end{aligned} \tag{10.67}$$

By noting that for any processes $x_1(k)$ and $x_2(k)$

$$E[x_1(k)x_2(k)] = \frac{1}{2\pi\jmath}\oint R_{x_1x_2}(z)\frac{dz}{z} \tag{10.68}$$

where the integration path is the counterclockwise unit circle, the objective function, as in equation (10.65), can be rewritten as

$$\begin{aligned}
\xi &= \frac{1}{2\pi\jmath}\oint \left[|H_o(z) - H_k(z)|^2 R_{xx}(z) + R_{nn}(z)\right]\frac{dz}{z} \\
&= \frac{1}{2\pi\jmath}\left[\oint H_o(z)H_o(z^{-1})R_{xx}(z)\frac{dz}{z} - 2\oint H_o(z)H_k(z^{-1})R_{xx}(z)\frac{dz}{z}\right. \\
&\quad \left. + \oint H_k(z)H_k(z^{-1})R_{xx}(z)\frac{dz}{z} + \oint R_{nn}(z)\frac{dz}{z}\right]
\end{aligned} \tag{10.69}$$

For the case the input and additional noise signals are white with variances respectively given by σ_x^2 and σ_n^2, the equation (10.69) can be simplified to

$$\xi = \frac{\sigma_x^2}{2\pi\jmath}\oint \left[H_o(z)H_o(z^{-1}) - 2H_o(z)H_k(z^{-1}) + H_k(z)H_k(z^{-1})\right]\frac{dz}{z} + \sigma_n^2 \tag{10.70}$$

This expression provides the relation between the MSE surface represented by ξ and the coefficients of the adaptive filter. The following example illustrates the use of the above equation.

Example 10.3

An all-pole adaptive filter of second-order is used to identify a system with transfer function

$$H_o(z) = \frac{1}{z^2 + 0.9z + 0.81}$$

The input signal and the measurement (additional) noise are white with $\sigma_x^2 = 1$ and $\sigma_n^2 = 0.1$, respectively. Compute the MSE as a function of the adaptive-filter multiplier coefficients.

Solution
The adaptive-filter transfer function is given by

$$H_k(z) = \frac{b_2}{z^2 + a_1 z + a_2}$$

Equation (10.70) can be solved by employing the residue theorem [1] which results in

$$\begin{aligned}\xi \;=\; & \frac{b_2^2(1+a_2)}{(1-a_2)(1+a_2-a_1)(1+a_2+a_1)} \\ & -\frac{2b_2(1-0.81a_2)}{1-0.9a_1-0.81a_2-0.729a_1a_2+0.81a_1^2+0.6561a_2^2} \\ & +3.86907339+0.1 \end{aligned} \tag{10.71}$$

If the adaptive-filter coefficients are set to their optimal values, i.e., $b_2 = 1,\ a_1 = 0.9$ and $a_2 = 0.81$, indicating a perfect identification of the unknown system, the resulting MSE is

$$\begin{aligned}\xi &= 3.86907339 - 7.73814678 + 3.86907339 + 0.1 \\ &= 0.1\end{aligned}$$

Note that the minimum MSE is equal to the measurement noise variance.

□

Equations (10.69) and (10.70), and more specifically equation (10.71), indicate clearly that the MSE surface is a nonquadratic function of the multiplier coefficients of the adaptive filter. This is particularly true for the multiplier coefficients pertaining to the denominator of the adaptive filter. As a consequence, the MSE surface may have several local minima, some of those corresponding to the desired global minimum. The multiplicity of minimum points depends upon the order of the adaptive IIR filter as compared to the unknown system that shapes the desired signal, and also upon the input signal properties when it is a colored noise.

Note that when the adaptive filter is FIR there is only a minimum point because the MSE surface is quadratic, independently of the unknown system and input signal characteristics. If the input or the desired signal are not stationary, the minimum point of the MSE surface moves in time but it is still unique.

The main problem brought about by the multimodality of the MSE surface is that gradient and Newton direction search algorithms will converge to a local minimum. Therefore, the adaptive filter may converge to a very bad point where the MSE assumes a large and unacceptable value. For example, in the system identification application, the generated transfer function may differ significantly from the unknown system transfer function.

Example 10.4
An unknown system with transfer function

$$H_o(z) = \frac{z - 0.85}{z + 0.99}$$

is supposed to be identified by a first-order adaptive filter described by

$$H_k(z) = \frac{bz}{z - a}$$

Plot the error surface, considering the input signal variance $\sigma_x^2 = 1$.

Solution
The expression for the MSE is given by

$$\xi = 171.13064 - \frac{(2 - 1.7a)b}{1 + 0.99a} + \frac{b^2}{1 - a^2}$$

The MSE surface is depicted in Fig. 10.14, where the MSE is clipped at 1 for a better view.

□

Several results regarding the uniqueness of the minimum point in the MSE surface are available in the literature [26]-[31]. Here, some of these results are summarized without proof, in order to give the designer some tools to support the appropriate choice of the adaptive IIR filter order.

First consider the case of inverse filtering or equalization, where the adaptive filter is placed in cascade with an unknown system and the desired signal is a delayed version of the overall cascade input signal. This case had been originally explored by Åström and Söderström [26], and they proved that if the adaptive filter is of sufficient order to find the inverse filter of the unknown system all the local minima will correspond to global minima if the input signal is a white noise. The sufficient order means that

$$N \quad \geq \quad M_d$$

and

$$M \quad \geq \quad N_d \tag{10.72}$$

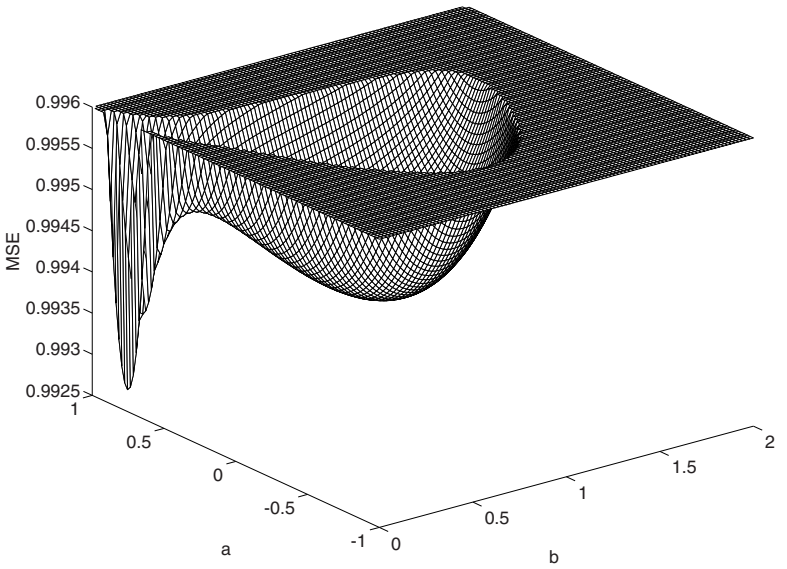
where N and M are the numerator and denominator orders of the adaptive filter as indicated in equation (10.5), N_d and M_d are the corresponding orders for the unknown system as indicated in equation (10.64).

When $N > M_d$ and $M > N_d$, there are infinitely many solutions given by

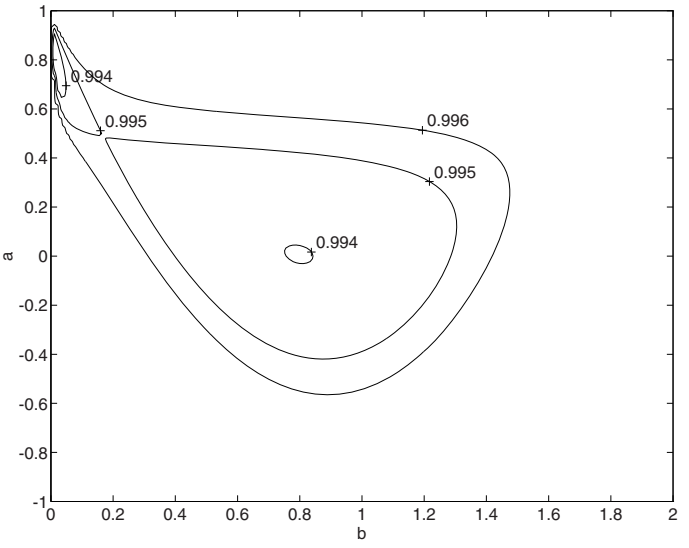
$$N(z) = L(z)D_o(z)$$

and

$$D(z) = L(z)N_o(z) \tag{10.73}$$



(a)



(b)

Figure 10.14 (a) MSE error surface, (b) MSE contours.

where $L(z) = z^{-N_l}(z^{N_l} + l_1 z^{N_l-1} + \cdots + l_{N_l})$, $N_l = \min(N - M_d, M - N_d)$, and l_i, for $i = 1, 2, \ldots, N_l$, are arbitrary.

The input signal can be colored noise generated for example by applying an IIR filter to a white noise. In this case, the adaptive filter must have order sufficient to generate the inverse of the unknown system and the input signal must be persistently exciting of order $\max(N + M_d, M + N_d)$, see for example [26]-[27], in order to guarantee that all local minima correspond to global minima.

For insufficient-order equalization, several local minima that do not correspond to a global minimum may occur. In this case, the MSE may not attain its minimum value after the algorithm convergence.

The situation is not the same in system identification application, as thought in the early investigations [28]. For this application, the sufficient order means

$$N \quad \geq \quad N_d$$

and

$$M \quad \geq \quad M_d \tag{10.74}$$

since the desired feature is to reproduce the unknown system frequency response, and not its inverse as in the equalization case. For $N > N_d$ and $M > M_d$, the local minima corresponding to global minima must satisfy the following conditions

$$N(z) = L(z)N_o(z)$$

and

$$D(z) = L(z)D_o(z) \tag{10.75}$$

where $L(z) = z^{-N_l}(z^{N_l} + l_i z^{N_l-1} + \cdots + l_{N_l})$, $N_l = \min(N - M_d, M - N_d)$, and l_i, for $i = 1, 2, \ldots, N_l$, are arbitrary.

The strongest result derived so far regarding the error surface property in system identification was derived by Söderström and Stoica [29]. The result states: For white noise input, all the stationary points correspond to global minima if

$$M \quad \geq \quad N_d - 1$$

and

$$\min(N - N_d, M - M_d) \quad \geq \quad 0 \tag{10.76}$$

Suppose that the input signal is an ARMA process generated by filtering a white noise with an IIR filter of orders M_n by N_n, and that there are no common zeros between the unknown system denominator and the input coloring IIR filter. In this case, all stationary points correspond to global minima if

$$M - N_d + 1 \quad \geq \quad N_n$$

and

$$\min(N - N_d, M - M_d) \quad \geq \quad M_n \tag{10.77}$$

The conditions summarized by equations (10.76) and (10.77) are sufficient but not necessary to guarantee that all stationary solutions correspond to the minimum MSE.

For $N = N_d = 1,\ M \geq M_d \geq 0$ and the input signal persistently exciting of order M_d there is a unique solution given by [29]

$$D(z) = D_o(z)$$

and

$$N(z) = N_o(z) \tag{10.78}$$

Also, when the adaptive filter and unknown system are all-pole second-order sections the unique solution is given by equation (10.78) [30].

Another particular result of some interest presented in [31], states that if

$$N - N_d = M - M_d = 0$$

and

$$M \geq N_d - 2 \tag{10.79}$$

the MSE surface has a unique stationary point corresponding to a global minimum.

For the case of insufficient-order identification [32], i.e., $\min(N - N_d, M - M_d) < 0$, or of sufficient order not satisfying the condition related to equations (10.77)-(10.79), the MSE surface may have local minima not attaining the minimum MSE, i.e., that are not global minima.

To satisfy any of the conditions of equations (10.77)-(10.79) a knowledge of the unknown system numerator and denominator orders is required. This information is not in general available or easy to obtain. This is one of the reasons adaptive IIR filters are not as popular as their FIR counterparts. However, there are situations where either a local minimum is acceptable or some information about the unknown system is available.

It should be noted that a vast literature is available for system identification [8],[33]-[34]. Here, the objective was to summarize some properties of the MSE surface, when the unknown system is modeled as an IIR filter with additive, white, and uncorrelated measurement noise. The assumptions regarding the measurement noise are quite reasonable for most applications of adaptive filtering.

10.7 INFLUENCE OF THE FILTER STRUCTURE ON THE MSE SURFACE

Some characteristics of the MSE surface differ when alternative structures are used in the realization of the adaptive filter. Each realization has a different relation between the filter transfer function and the multiplier coefficients, originating modifications in the MSE surface [35].

The MSE surfaces related to two alternative realizations for the adaptive filter can be described as functions of the filter multiplier coefficients by $F_1(\boldsymbol{\theta}_1)$ and $F_2(\boldsymbol{\theta}_2)$, respectively. Note that no index was used to indicate the varying characteristics of the adaptive-filter parameters, since this simplifies the notation while keeping the relevant MSE surface properties. It is assumed that the desired signal and the input signal are the same in the alternative experiments. Also, it is considered that for any set of parameters $\boldsymbol{\theta}_1$ leading to a stable filter, there is a continuous mapping given by $\boldsymbol{f}_3(\boldsymbol{\theta}_1) = \boldsymbol{\theta}_2$, where $\boldsymbol{\theta}_2$ also leads to a stable filter. Both $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are N' by 1 vectors.

The two alternative structures are equivalent if the objective functions are equal, i.e.,

$$F_1(\boldsymbol{\theta}_1) = F_2(\boldsymbol{\theta}_2) = F_2[\boldsymbol{f}_3(\boldsymbol{\theta}_1)] \tag{10.80}$$

First consider the case where $\boldsymbol{f}_3$ is differentiable, and then from the above equation it follows that

$$\frac{\partial F_1(\boldsymbol{\theta}_1)}{\partial \boldsymbol{\theta}_1} = \frac{\partial F_2[\boldsymbol{f}_3(\boldsymbol{\theta}_1)]}{\partial \boldsymbol{\theta}_1} = \frac{\partial F_2[\boldsymbol{f}_3(\boldsymbol{\theta}_1)]}{\partial \boldsymbol{f}_3(\boldsymbol{\theta}_1)} \frac{\partial \boldsymbol{f}_3(\boldsymbol{\theta}_1)}{\partial \boldsymbol{\theta}_1} \tag{10.81}$$

where the first partial derivative on the rightmost side of the above equation is an 1 by N' vector while the second partial derivative is a matrix with dimensions N' by N', where N' is the number of parameters in $\boldsymbol{\theta}_1$. Suppose that $\boldsymbol{\theta}_2'$ is a stationary point of $F_2(\boldsymbol{\theta}_2)$, it then follows that

$$\frac{\partial F_2(\boldsymbol{\theta}_2)}{\partial \boldsymbol{\theta}_2}\Big|_{\boldsymbol{\theta}_2=\boldsymbol{\theta}_2'} = \mathbf{0} = \frac{\partial F_1(\boldsymbol{\theta}_1)}{\partial \boldsymbol{\theta}_1}\Big|_{\boldsymbol{\theta}_1=\boldsymbol{\theta}_1'} \tag{10.82}$$

where $\boldsymbol{\theta}_2' = \boldsymbol{f}_3(\boldsymbol{\theta}_1')$. Note that the type of the stationary points of $F_1(\boldsymbol{\theta}_1)$ and $F_2(\boldsymbol{\theta}_2)$ are the same, since their second derivatives have the same properties at these stationary points (see problem 1).

Now consider the case where

$$\frac{\partial F_2[\boldsymbol{f}_3(\boldsymbol{\theta}_1)]}{\partial \boldsymbol{f}_3(\boldsymbol{\theta}_1)}\Big|_{\boldsymbol{\theta}_1=\boldsymbol{\theta}_1''} = \mathbf{0} \tag{10.83}$$

but

$$\frac{\partial F_1(\boldsymbol{\theta}_1)}{\partial \boldsymbol{\theta}_1}\Big|_{\boldsymbol{\theta}_1=\boldsymbol{\theta}_1''} \neq \mathbf{0} \tag{10.84}$$

that can happen only when $\boldsymbol{f}_3(\boldsymbol{\theta}_1)$ is not differentiable at $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_1''$. In this case, the chain rule of equation (10.81) does not apply. The new generated stationary points in $F_2(\boldsymbol{\theta}_2)$ can be shown to be saddle points (see problem 2).

Example 10.5
An unknown second-order system described by

$$H_o(z) = \frac{2z + c_1}{z^2 + c_1 z + c_2}$$

is to be identified by using two different structures for the adaptive filter, namely the direct form and the parallel form described respectively by

$$H_d(z) = \frac{2z + a_1}{z^2 + a_1 z + a_2}$$

and

$$H_p(z) = \frac{1}{z + p_1} + \frac{1}{z + p_2} = \frac{2z + p_1 + p_2}{z^2 + (p_1 + p_2)z + p_1 p_2}$$

verify the existence of new saddle points in the parallel realization.

Solution
The function relating the parameters of the two realizations can be given by

$$\boldsymbol{\theta}_2 = \begin{bmatrix} \frac{a_1 + \sqrt{a_1^2 - 4a_2}}{2} \\ \frac{a_1 - \sqrt{a_1^2 - 4a_2}}{2} \end{bmatrix} = \boldsymbol{f}_3(\boldsymbol{\theta}_1)$$

where function $\boldsymbol{f}_3(\boldsymbol{\theta}_1)$ is not differentiable when $a_2 = \frac{a_1^2}{4}$.

The inverse of the matrix $\frac{\partial \boldsymbol{f}_3(\boldsymbol{\theta}_1)}{\partial \boldsymbol{\theta}_1}$ is given by

$$\left[\frac{\partial \boldsymbol{f}_3(\boldsymbol{\theta}_1)}{\partial \boldsymbol{\theta}_1}\right]^{-1} = \begin{bmatrix} 1 & 1 \\ p_2 & p_1 \end{bmatrix}$$

and, if $p_1 = p_2$, the above matrix is singular, which makes it possible that $\frac{\partial F_1(\boldsymbol{\theta}_1)}{\partial \boldsymbol{\theta}_1} \neq 0$ when $\frac{\partial F_2(\boldsymbol{\theta}_2)}{\partial \boldsymbol{\theta}_2} = 0$, as previously mentioned in equations (10.81) and (10.82).

Note that, as expected, $p_1 = p_2$ only when $a_2 = \frac{a_1^2}{4}$. On this parabola, the objective function $F_1(\boldsymbol{\theta}_1)$ has a minimum that corresponds to a saddle point of the function $F_2(\boldsymbol{\theta}_2)$. Also, this is the situation where the parallel realization is of reduced order, i.e., first order.

□

Basically, the manifold generated by the parallel realization is due to the fact that a given section can identify any pole of the unknown system, leaving the other poles to the remaining sections in parallel. This means that in a sufficient-order identification problem, if for the direct-form realization there is a unique global minimum point, in the case of parallel realization with first-order sections there

will be $N!$ global minima, where N is the number of poles in the unknown system. When using a parallel realization it is assumed that no multiple poles exist in the unknown system.

In the initialization of the algorithm, the adaptive-filter parameters should not be in a reduced-order manifold, because by employing a gradient-based algorithm the parameters may be kept in the manifold and eventually reach a saddle point. The measurement noise, that is in general present in the adaptive-filtering process, will help the parameters to skip the manifolds, but despite that the convergence will be slowed. A similar phenomenon occurs with the cascade realization of the adaptive filter.

10.8 ALTERNATIVE ERROR FORMULATIONS

The error signal (in some cases the regressor) can be chosen in alternative ways in order to avoid some of the drawbacks related to the output error formulation, as for example the multiple local minima. Several formulations have been investigated in the literature [36]-[37], [39], [40]-[42], [45]-[46], [51]-[52], where each of them has its own advantages and disadvantages. The choice of the best error formulation depends on the application and on the information available about the adaptive-filtering environment. In this section, we present two alternative error formulations, namely the equation error and Steiglitz-McBride methods, and discuss some of their known properties. Throughout the section other error formulations are briefly mentioned.

10.8.1 Equation Error Formulation

In the equation error (EE) formulation, the information vector instead of having past samples of the adaptive-filter output, uses delayed samples of the desired signal as follows:

$$\boldsymbol{\phi}_e(k) = [d(k-1)\, d(k-2) \ldots d(k-N)\, x(k)\, x(k-1) \ldots x(k-M)]^T \tag{10.85}$$

The equation error is defined by

$$e_e(k) = d(k) - \boldsymbol{\theta}^T(k)\boldsymbol{\phi}_e(k) \tag{10.86}$$

as illustrated in Fig. 10.15. The parameter vector $\boldsymbol{\theta}(k)$ is given by

$$\boldsymbol{\theta}(k) = [-a_1(k)\ -a_2(k) \ldots -a_N(k)\, b_0(k) \ldots b_M(k)]^T \tag{10.87}$$

The equation error can be described in a polynomial form as follows:

$$e_e(k) = A(k, q^{-1})d(k) - B(k, q^{-1})x(k) \tag{10.88}$$

where, once again

$$B(k, q^{-1}) = b_0(k) + b_1(k)q^{-1} + \cdots + b_M(k)q^{-M}$$
$$A(k, q^{-1}) = 1 + a_1(k)q^{-1} + \cdots + a_N(k)q^{-N}$$

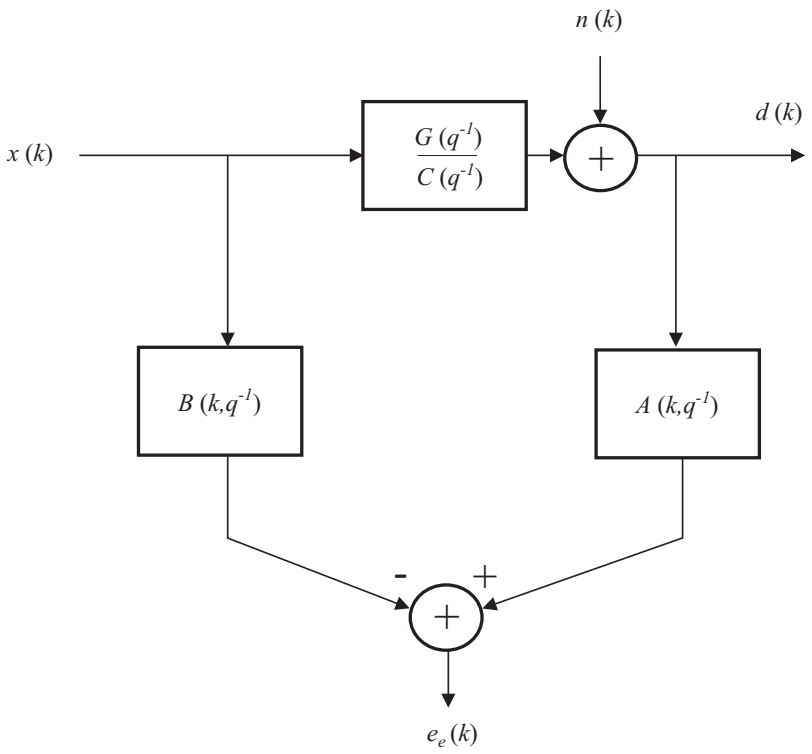


Figure 10.15 Equation error configuration.

The output signal related to the EE formulation is obtained through the following linear difference equation

$$\begin{aligned} y_e(k) &= \sum_{j=0}^{M} b_j(k)x(k-j) - \sum_{j=1}^{N} a_j(k)d(k-j) \\ &= \boldsymbol{\theta}^T(k)\boldsymbol{\phi}_e(k) \end{aligned} \tag{10.89}$$

As can be noted, the adaptive filter does not have feedback and $y_e(k)$ is a linear function of the parameters.

In the EE formulation, the adaptation algorithm determines how the coefficients of the adaptive IIR filter should change in order to minimize an objective function which involves $e_e(k)$ defined as

$$\xi_e = F[e_e(k)] \tag{10.90}$$

Usually, the objective function to be minimized is the mean-squared value of the EE (MSEE), i.e.,

$$\xi_e(k) = E[e_e^2(k)] \tag{10.91}$$

Since the input and desired signals are not functions of the adaptive-filter parameters, it can be expected that the sole approximation in the gradient computation is due to the estimate of the expected

value required in practical implementations. The key point is to note that since the MSEE is a quadratic function of the parameters, only a global minimum exists provided the signals involved are persistently exciting. When the estimate of the MSEE is the instantaneous squared equation error, the gradient vector is proportional to minus the information vector. In this case, the resulting algorithm is called LMSEE algorithm whose coefficient updating equation is given by

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + 2\mu\boldsymbol{\phi}_e(k)e_e(k) \tag{10.92}$$

A number of approaches with different points of view are available to analyze the convergence properties of this method. A particularly interesting result is that if the convergence factor is chosen in the range

$$0 < \mu < \frac{1}{\lambda_{\max}} \tag{10.93}$$

the convergence in the mean of the LMSEE algorithm can be guaranteed [37], where $\lambda_{\max}$ is the maximum eigenvalue of $E[\boldsymbol{\phi}_e(k)\boldsymbol{\phi}_e^T(k)]$. This result can be easily proved by exploring the similarity between the LMSEE algorithm and the standard FIR LMS algorithm. Some stability results of the LMSEE algorithm can be found in [38].

An alternative objective function for adaptive IIR filtering based on equation error is the least-squares function

$$\xi_e(k) = \sum_{i=0}^{k} \lambda^{k-i} e_e^2(i) = \sum_{i=0}^{k} \lambda^{k-i}[d(i) - \boldsymbol{\theta}^T(k)\boldsymbol{\phi}_e(i)]^2 \tag{10.94}$$

The forgetting factor λ, as usual is chosen in the range $0 << \lambda < 1$, allowing the distant past information to be increasingly negligible. In this case, the corresponding RLS algorithm consists of the following basic steps

$$e(k) = d(k) - \boldsymbol{\theta}^T(k)\boldsymbol{\phi}_e(k) \tag{10.95}$$

$$\mathbf{S}_{De}(k+1) = \frac{1}{\lambda}\left[\mathbf{S}_{De}(k) - \frac{\mathbf{S}_{De}(k)\boldsymbol{\phi}_e(k)\boldsymbol{\phi}_e^T(k)\mathbf{S}_{De}(k)}{\lambda + \boldsymbol{\phi}_e^T(k)\mathbf{S}_{De}(k)\boldsymbol{\phi}_e(k)}\right] \tag{10.96}$$

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + \mathbf{S}_{De}(k+1)\boldsymbol{\phi}_e(k)e_e(k) \tag{10.97}$$

In a given iteration k, the adaptive IIR filter transfer function related to the EE formulation can be expressed as follows:

$$H_k(z) = z^{N-M}\frac{b_0(k)z^M + b_1(k)z^{M-1} + \cdots + b_{M-1}(k)z + b_M(k)}{z^N + a_1(k)z^{N-1} + \cdots + a_{N-1}(k)z + a_N(k)} \tag{10.98}$$

In Fig. 10.16 an alternative structure for the EE approach where the IIR adaptive filter appears explicitly is depicted. Note that the structure shows clearly that the polynomial $A(k, q^{-1})$ is meant to model the denominator polynomial of the unknown system, in system identification applications. During the adaptation process, it is necessary to monitor the stability of the poles, as described for the output error method. The full description of the RLS equation error algorithm is given in Algorithm 10.3.

The basic problem related to this method is the parameter bias induced by the measurement noise [37]-[38], even for sufficient-order case. The bias is caused by the fact that the additional noise $n(k)$

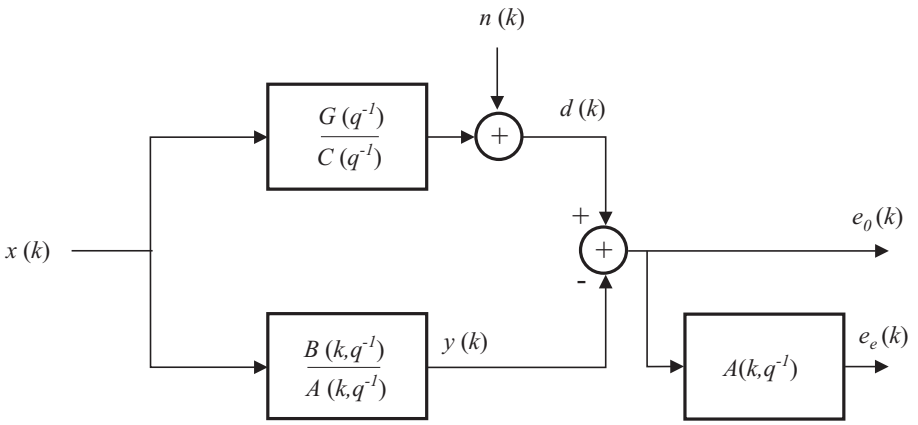


Figure 10.16 Basic configuration for system identification using equation error.

Algorithm 10.3

EE Algorithm, RLS Version

Initialization

$a_i(k) = b_i(k) = e(k) = 0$

$y(k) = x(k) = 0\,,\ k\ < 0$

$\mathbf{S}_{De}(0) = \delta^{-1}\mathbf{I}$

For each $x(k),\ d(k),\ k \geq 0,$ do

$e_e(k) = d(k) - \boldsymbol{\phi}_e^T(k)\boldsymbol{\theta}(k)$

$\mathbf{S}_{De}(k+1) = \frac{1}{\lambda}\left[\mathbf{S}_{De}(k) - \frac{\mathbf{S}_{De}(k)\boldsymbol{\phi}_e(k)\boldsymbol{\phi}_e^T(k)\mathbf{S}_{De}(k)}{\lambda + \boldsymbol{\phi}_e^T(k)\mathbf{S}_{De}(k)\boldsymbol{\phi}_e(k)}\right]$

$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + \mathbf{S}_{De}(k+1)\boldsymbol{\phi}_e(k)e_e(k)$

Stability test

is filtered by the FIR filter represented by the polynomial $A(k, q^{-1})$. Since the coefficients of this polynomial are updated with the objective of minimizing the EE signal, they also attempt to minimize the contribution of $n(k)$ to the EE power. The bias is induced by the fact that the additional noise does not belong to the unknown system model. An increase in the power of $n(k)$ leads to higher bias in the parameter estimate.

The Instrumental Variable methods [39] were proposed to solve the bias problem. In these methods the stability cannot be guaranteed under the same general conditions as for the LMSEE method.

Another approach was proposed in [40], and extended in [41] and [42], where a family of asymptotically stable algorithms was introduced. The resulting algorithms are based on a modification of the basic LMSEE updating equations, that within sufficiently general conditions lead to consistent parameter estimates. These algorithms employ a type of output error feedback to the information vector. There are also algorithms that combine different algorithms to define the objective function [43]-[44].

10.8.2 The Steiglitz-McBride Method

The Steiglitz-McBride (SM) error formulation [45], by employing some extra all-pole filtering, leads to algorithms whose behavior resembles the EE approach in the initial iterations and the output error approach after convergence. The main motivation of the SM method is the global convergence behavior for some cases of insufficient-order system identification. Such interest sparked investigations which resulted in a number of on-line algorithms based on the SM method that are suitable for adaptive IIR filtering [46]. The main problem associated with the SM method is the inconsistent behavior when the measurement noise is colored [47]. Since the on-line method converges asymptotically to the off-line solution, the bias error also affects the on-line algorithms proposed in [46].

In order to introduce the SM method, consider the identification of a system whose model is described by

$$d(k) = \frac{G(q^{-1})}{C(q^{-1})}x(k) + n(k) = y_d(k) + n(k) \tag{10.99}$$

where $d(k)$ is the reference signal, $x(k)$ is the input signal, $n(k)$ is the measurement noise, and $y_d(k)$ is the output signal of the plant, with $C(q^{-1}) = 1 - \sum_{i=1}^{N_d} c_i q^{-i}$ and $G(q^{-1}) = \sum_{i=0}^{M_d} g_i q^{-i}$ coprime. The polynomial $C(q^{-1})$ has zeros inside the unit circle, and the input signal $x(k)$ and the measurement noise $n(k)$ are assumed independent. The estimation of the parameters associated with the polynomials $C(q^{-1})$ and $G(q^{-1})$ through the SM method is based on the minimization of the following criterion [45]

$$\xi_s(\boldsymbol{\theta}(k+1)) = E\left\{\left[A(k+1,q^{-1})\frac{d(k)}{A(k,q^{-1})} - B(k+1,q^{-1})\frac{x(k)}{A(k,q^{-1})}\right]^2\right\} \tag{10.100}$$

where $A(k,q^{-1}) = 1 + \sum_{i=1}^{N} a_i(k)q^{-i}$ and $B(k,q^{-1}) = \sum_{i=0}^{M} b_i(k)q^{-i}$ are the denominator and numerator estimator polynomials, respectively, and

$$\boldsymbol{\theta}(k) = [-a_1(k)\ -a_2(k)\ \ldots - a_N(k)\ b_0(k)\ \ldots\ b_M(k)]^T \tag{10.101}$$

is the adaptive-filter parameter vector.

The estimate $\boldsymbol{\theta}(k+1)$ is obtained by minimizing equation (10.100) assuming $\boldsymbol{\theta}(k)$ known. The solution of this MSE minimization problem at iteration $(k+1)$ is

$$\begin{aligned}\boldsymbol{\theta}(k+1) &= \left[E\left\{\boldsymbol{\phi}_s(k)\boldsymbol{\phi}_s^T(k)\right\}\right]^{-1} E\left[\boldsymbol{\phi}_s(k)\frac{d(k)}{A(k,q^{-1})}\right] \\ &= \left[E\left\{\boldsymbol{\phi}_s(k)\boldsymbol{\phi}_s^T(k)\right\}\right]^{-1} E\left[\boldsymbol{\phi}_s(k)d_f(k)\right]\end{aligned} \tag{10.102}$$

where

$$\begin{aligned}\boldsymbol{\phi}_s(k) &= \left[\frac{d(k-1)}{A(k,q^{-1})} \ldots \frac{d(k-N)}{A(k,q^{-1})} \; \frac{x(k)}{A(k,q^{-1})} \ldots \frac{x(k-M)}{A(k,q^{-1})}\right]^T \\ &= \left[d_f(k-1) \ldots d_f(k-N)\, x_f(k) \ldots x_f(k-M)\right]^T \end{aligned} \tag{10.103}$$

is the regressor related to the SM method.

If the input signal is persistently exciting of sufficient order and the adaptive filter has strictly sufficient order, some properties of the estimate resulting from equation (10.102) are known [47]: a) The estimate that minimizes equation (10.100) is unique; b) If the measurement noise is not white, the estimate resulting from equation (10.102) is biased.

In real-time signal processing applications, it is important to consider an on-line version of the SM method. In this case, some approximations are necessary. First note that the error criterion whose variance is to be minimized in equation (10.102) is

$$e_s(k) = \frac{d(k)}{A(k,q^{-1})} - \boldsymbol{\theta}^T(k+1)\boldsymbol{\phi}_s(k) \tag{10.104}$$

The SM error is computed as illustrated in Fig. 10.17. Assuming a sufficiently slow parameter variation, we can consider that $\boldsymbol{\theta}(k+1) \approx \boldsymbol{\theta}(k)$. Therefore, equation (10.104) can be rewritten as follows:

$$e_s(k) \approx \frac{d(k)}{A(k,q^{-1})} - \boldsymbol{\theta}^T(k)\boldsymbol{\phi}_s(k) \tag{10.105}$$

The exact implementation of the regressor $\boldsymbol{\phi}_s(k)$ requires an independent filtering of each component by an all-pole filter with denominator polynomial $A(k,q^{-1})$. A useful approximation that reduces considerably the computational complexity is possible by assuming slow parameter variation [46] in such a way that

$$\boldsymbol{\theta}(k-1) \approx \boldsymbol{\theta}(k-2) \ldots \approx \boldsymbol{\theta}(k-N) \tag{10.106}$$

With these simplifications only one all-pole filtering is required. Note that a hypothesis similar to equation (10.106) was utilized in the output error method in order to simplify the implementation. However, in the case of the output error method, the measurement noise does not affect the regressor, since the regressor vector is composed of delayed samples of the adaptive-filter input and output. For the SM method, except for white measurement noise, the simplification in equation (10.106) is not easily justified.

The updating equation of the on-line SM algorithm for system identification employing a stochastic gradient search is given by

$$\begin{aligned}\boldsymbol{\theta}(k+1) &= \boldsymbol{\theta}(k) + 2\mu\boldsymbol{\phi}_s(k)\left[\frac{d(k)}{A(k,q^{-1})} - \boldsymbol{\phi}_s^T(k)\boldsymbol{\theta}(k)\right] \\ &= \boldsymbol{\theta}(k) + 2\mu\boldsymbol{\phi}_s(k)e_s(k) \end{aligned} \tag{10.107}$$

The description of a gradient SM algorithm is given in the Algorithm 10.4.

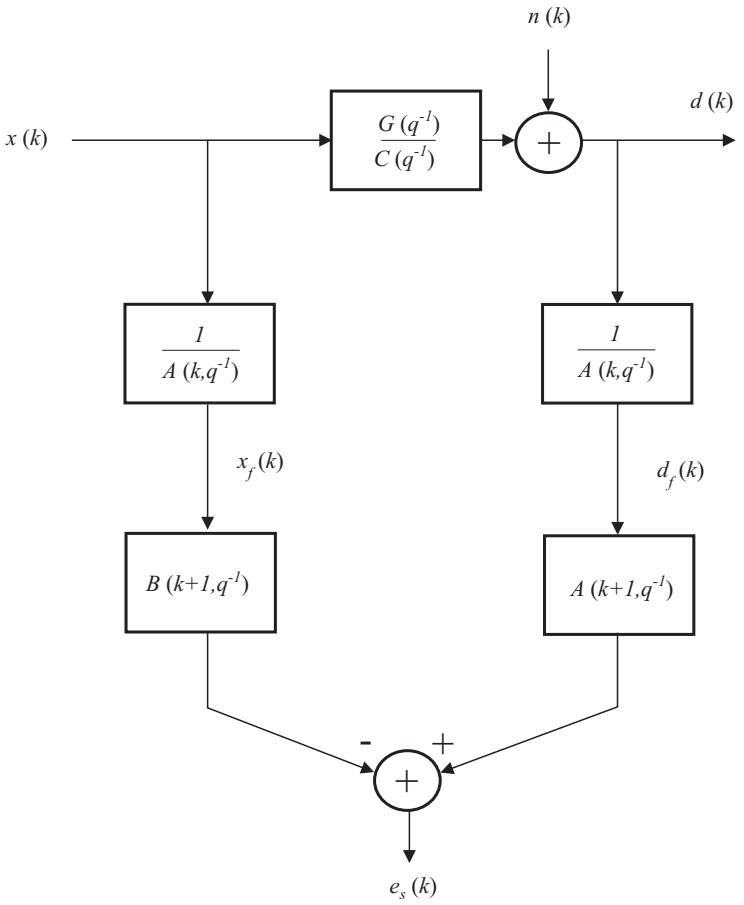


Figure 10.17 Steiglitz-McBride configuration.

The SM method can be implemented using different realizations such as cascade [48], lattice [49], and the series-parallel realization [50]. These realizations allow easy stability monitoring, and their choice affects the convergence speed [50].

It should be mentioned that a family of algorithms based on the SM method that solves the problem of inconsistency of the parameter estimates was proposed in [51]-[52]. These algorithms are very attractive for adaptive IIR filtering due to their behavior in terms of consistency (i.e., definition of stationary points) and convergence properties. In [55], simulation results as well as an alternative implementation for the consistent SM method was presented.

The interested reader can also find some interesting results about the convergence behavior of the SM-based algorithms in [53]-[54] and in the references therein. Also, applications of the SM algorithm to equalization can be found in [56].

Algorithm 10.4

SM Based Algorithm, Gradient Version

Initialization

$a_i(k) = b_i(k) = 0$

$d_f(k) = x_f(k) = 0\,,\ k\ < 0$

For each $x(k),\ d(k),\ k \geq 0$ do

$x_f(k) = x(k) - \sum_{i=1}^{N} a_i(k)\, x_f(k-i)$

$d_f(k) = d(k) - \sum_{i=1}^{N} a_i(k)\, d_f(k-i)$

$e_s(k) = d_f(k) - \boldsymbol{\phi}_s^T(k)\boldsymbol{\theta}(k)$

$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + 2\mu\boldsymbol{\phi}_s(k)e_s(k)$

Stability test

10.9 CONCLUSION

It is recognized that the adaptive IIR filter can be potentially used in a number of applications due to its superior system modeling owing to poles. These advantages come with drawbacks such as possible local minima in the performance surface and the possible instability during the adaptation process. Also, the nonlinear relation between the adaptive-filter parameters and the internal signals in some formulations makes the gradient computation and convergence analysis much more complicated as compared to the FIR case. In this chapter, the theory of adaptive IIR filters was presented exposing several solutions to the above mentioned drawbacks, such that the designer can decide which is the best configuration for a given application.

In this chapter, an example of application of adaptive IIR filters in system identification was presented. In this example, some of the realizations presented here were tested and compared. Another example exploited the use of notch filters for sinusoid detection in noise.

10.10 REFERENCES

1. P. S. R. Diniz, E. A. B. da Silva, and S. L. Netto, *Digital Signal Processing: System Analysis and Design*, Cambridge University Press, Cambridge, UK, 2002.

2. A. Antoniou, *Digital Signal Processing: Signals, Systems, and Filters*, McGraw Hill, New York, NY, 2005.

3. C. R. Johnson, Jr., "Adaptive IIR filtering: Current results and open issues," *IEEE Trans. on Information Theory*, vol. IT-30, pp. 237-250, Mar. 1984.

4. J. J. Shynk, "Adaptive IIR filtering," *IEEE ASSP Magazine*, vol. 6, pp. 4-21, April 1989.

5. S. A. White, "An adaptive recursive digital filter," *Proc. 9th Asilomar Conf. on Circuits, Systems, and Computers*, Pacific Grove, CA, pp. 21-25, Nov. 1975.

6. S. Hovarth, Jr., "A new adaptive recursive LMS filter," *Proc. Digital Signal Processing Conference*, Florence, Italy, pp. 21-26, 1980.

7. T. C. Hsia, "A simplified adaptive recursive filter design," *Proceedings of the IEEE*, vol. 69, pp. 1153-1155, Sept. 1981.

8. L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, MA, 1983.

9. R. A. David, "A modified cascade structure for IIR adaptive algorithms," *Proc. 15th Asilomar Conf. on Circuits, Systems, and Computers*, Pacific Grove, CA, pp. 175-179, Nov. 1981.

10. J. J. Shynk, "Adaptive IIR filtering using parallel-form realization," *IEEE Trans. on Acoust., Speech, and Signal Processing*, vol. 37, pp. 519-533, April 1989.

11. P. S. R. Diniz, J. E. Cousseau, and A. Antoniou, "Improved parallel realization of IIR adaptive filters," *Proceedings of the IEE Part G: Circuits, Devices, and Systems*, vol. 140, pp. 322-328, Oct. 1993.

12. P. A. Regalia, *Adaptive IIR Filtering for Signal Processing and Control*, Marcel Dekker, New York, NY, 1995.

13. D. Parikh, N. Ahmed, and S. D. Stearns, "An adaptive lattice algorithm for recursive filters," *IEEE Trans. on Acoust., Speech, and Signal Processing*, vol. ASSP-28, pp. 110-112, Feb. 1980.

14. I. L. Ayala, "On a new adaptive lattice algorithm for recursive filters," *IEEE Trans. on Acoust., Speech, and Signal Processing*, vol. ASSP-30, pp. 316-319, April 1982.

15. J. J. Shynk, "On lattice-form algorithms for adaptive IIR filtering," *Proc. IEEE Intern. Conf. on Acoust., Speech, Signal Processing*, New York, NY, pp. 1554-1557, April 1988.

16. J. A. Rodríguez-Fonollosa and E. Masgrau, "Simplified gradient calculation in adaptive IIR lattice filters," *IEEE Trans. on Signal Processing*, vol. 39, pp. 1702-1705, July 1991.

17. A. H. Gray, Jr., and J. D. Markel, "Digital lattice and ladder filter synthesis," *IEEE Trans. on Audio Electroacoust.*, vol. AU-21, pp. 492-500, Dec. 1973.

18. F. Itakura and S. Saito, "Digital filtering techniques for speech analysis and synthesis," *Proc. 7th Intern. Congr. Acoustics*, Paper 25C-1, Budapest, Hungary, pp. 261-264, 1971.

19. M. Tummala, "New adaptive normalised lattice algorithm for recursive filters," *Electronics Letters*, vol. 24, pp. 659-661, May 1988.

20. J. M. Romano and M. G. Bellanger, "Fast least-squares adaptive notch filtering," *IEEE Trans. on Acoust. Speech and Signal Processing*, vol. 36, pp. 1536-1540, Sept. 1988.

21. H. Fan and Y. Yang, "Analysis of a frequency-domain adaptive IIR filter," *IEEE Trans. on Acoust. Speech, and Signal Processing*, vol. 38, pp. 864-870, May 1990.

22. P. S. R. Diniz, J. E. Cousseau, and A. Antoniou, "Fast parallel realization for IIR adaptive filters," *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 41, pp. 561-567, Aug. 1994.

23. P. S. R. Diniz and A. Antoniou, "Digital-filter structures based on the concept of the voltage-conversion generalized immittance converter," *Canadian J. of Electrical and Computer Engineering*, vol. 13, pp. 90-98, April 1988.

24. H. Fan, "A structural view of asymptotic convergence speed of adaptive IIR filtering algorithms: Part I-Infinite precision implementation," *IEEE Trans. on Signal Processing*, vol. 41, pp. 1493-1517, April 1993.

25. J. E. Cousseau, P. S. R. Diniz, G. Sentoni, and O. Agamennoni, "On orthogonal realizations for adaptive IIR filters," *International Journal Circuit Theory and Applications*, John Wiley & Sons, NY, vol. 28, pp. 481-500, Sept. 2000.

26. K. J. Åström and T. Söderström, "Uniqueness of the maximum likelihood estimates of the parameters of an ARMA model," *IEEE Trans. on Automatic Control*, vol. AC-19, pp. 769-773, Dec. 1974.

27. T. Söderström, "On the uniqueness of maximum likelihood identification," *Automatica*, vol. 11, pp. 193-197, Mar. 1975.

28. S. D. Stearns, "Error surfaces of recursive adaptive filters," *IEEE Trans. on Acoust., Speech, and Signal Processing*, vol. ASSP-29, pp. 763-766, June 1981.

29. T. Söderström and P. Stoica, "Some properties of the output error model," *Automatica*, vol. 18, pp. 93-99, Jan. 1982.

30. H. Fan and M. Nayeri, "On error surfaces of sufficient order adaptive IIR filters: Proofs and counterexamples to a unimodality conjecture," *IEEE Trans. on Acoust., Speech, and Signal Processing*, vol. 37, pp. 1436-1442, Sept. 1989.

31. M. Nayeri, "Uniqueness of MSOE estimates in IIR adaptive filtering; a search for necessary conditions," *Proc. IEEE Intern. Conf. on Acoust., Speech, Signal Processing*, Glasgow, Scotland, pp. 1047-1050, May 1989.

32. M. Nayeri, H. Fan, and W. K. Jenkins, "Some characteristics of error surfaces for insufficient order adaptive IIR filters," *IEEE Trans. on Acoust., Speech, and Signal Processing*, vol. 38, pp. 1222-1227, July 1990.

33. L. Ljung, *System Identification: Theory for the User*, Prentice Hall Inc., Englewood Cliffs, NJ, 1987.

34. T. Söderström and P. Stoica, *System Identification*, Prentice Hall International, Hempstead, Hertfordshire, 1989.

35. M. Nayeri and W. K. Jenkins, "Alternate realizations to adaptive IIR filters and properties of their performance surfaces," *IEEE Trans. on Circuits and Systems*, vol. 36, pp. 485-496, April 1989.

36. S. L. Netto, P. S. R. Diniz, and P. Agathoklis, "Adaptive IIR filtering algorithms for system identification: A general framework," *IEEE Trans. on Education*, vol. 26, pp. 54-66, Feb. 1995.

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

38. T. Söderström and P. Stoica, "On the stability of dynamic models obtained by least squares identification," *IEEE Trans. on Automatic Control*, vol. AC-26, pp. 575-577, April 1981.

39. T. Söderström and P. Stoica, *Instrumental Variable Methods for System Identification*, Springer Verlag, New York, NY, 1983.

40. J. -N. Lin and R. Unbehauen, "Bias-remedy least mean square equation error algorithm for IIR parameter recursive estimation," *IEEE Trans. on Signal Processing*, vol. 40, pp. 62-69, Jan. 1992.

41. P. S. R. Diniz and J. E. Cousseau, "A family of equation-error based IIR adaptive algorithms," *IEEE Proc. Midwest Symposium of Circuits and Systems*, Lafayette, LA, pp. 1083-1086, 1994.

42. J. E. Cousseau and P. S. R. Diniz, "A general consistent equation-error algorithm for adaptive IIR filtering," *Signal Processing*, vol. 56, pp. 121-134, 1997.

43. J. B. Kenney and C. E. Rohrs, "The composite regressor algorithm for IIR adaptive systems," *IEEE Trans. on Signal Processing*, vol. 41, pp. 617-628, Feb. 1993.

44. S. L. Netto and P. S. R. Diniz, "Composite algorithms for adaptive IIR filtering," *IEE Electronics Letters*, vol. 28, pp. 886-888, April 1992.

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

46. H. Fan and W. K. Jenkins, "A new adaptive IIR filter," *IEEE Trans. on Circuits and Systems*, vol. CAS-33, pp. 939-947, Oct. 1986.

47. P. Stoica and T. Söderström, "The Steiglitz-McBride identification algorithm revisited–convergence analysis and accuracy aspects," *IEEE Trans. on Automatic Control*, vol. AC-26, pp. 712-717, June 1981.

48. B. E. Usevitch and W. K. Jenkins, "A cascade implementation of a new IIR adaptive digital filter with global convergence and improved convergence rates," *Proc. IEEE Intern. Symposium of Circuits and Systems*, Portland, OR, pp. 2140-2142, 1989.

49. P. A. Regalia, "Stable and efficient lattice algorithms for adaptive IIR filtering," *IEEE Trans. on Signal Processing*, vol. 40, pp. 375-388, Feb. 1992.

50. J. E. Cousseau and P. S. R. Diniz, "A new realization of IIR echo cancellers using the Steiglitz-McBride method," *Proc. IEEE Intern. Telecommunication Symposium*, Rio de Janeiro, Brazil, pp. 11-14, 1994.

51. J. E. Cousseau and P. S. R. Diniz, "A consistent Steiglitz-McBride algorithm," *Proc. IEEE Intern. Symposium of Circuits and Systems*, Chicago, IL, pp. 52-55, 1993.

52. J. E. Cousseau and P. S. R. Diniz, "New adaptive IIR filtering algorithms based on Steiglitz-McBride method," *IEEE Trans. on Signal Processing*, vol. 45, pp. 1367-1371, May 1997.

53. H. Fan and M. Doroslovački, "On 'global convergence' of Steiglitz-McBride adaptive algorithm," *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 40, pp. 73-87, Feb. 1993.

54. M. H. Cheng and V. L. Stonick, "Convergence, convergence point and convergence rate for Steiglitz-McBride method: A unified approach," *Proc. IEEE Intern. Conf. on Acoust., Speech and Signal Processing*, Adelaide, Australia, 1994.

55. V. L. Stonick and M. H. Cheng, "Adaptive IIR filtering: composite regressor method," *Proc. IEEE Intern. Conf. on Acoust., Speech and Signal Processing*, Adelaide, Australia, 1994.

56. P. M. Crespo and M. L. Honig, "Pole-zero decision feedback equalization with a rapidly converging adaptive IIR algorithm," *IEEE J. on Selected Areas in Communications*, vol. 9, pp. 817-828, Aug. 1991.

10.11 PROBLEMS

1. Show that the stationary points related to two equivalent adaptive realizations of the type in equation (10.82) have the same nature, i.e., are minimum, maximum or saddle point.

2. Show that the new stationary points generated by the discontinuity in $\boldsymbol{f}_3(\boldsymbol{\theta}_1)$ as discussed after equation (10.84) are saddle points.

3. Describe how the manifolds are formed in the MSE surface when a cascade realization is used for the adaptive-filter implementation. Give a generic example.

4. Derive a general expression for the transfer function of the two-multiplier lattice structure.

5. Derive an adaptive-filtering algorithm which employs the canonic direct-form structure shown in Fig. 10.18. Consider that the adaptive-filter parameters are slowly varying in order to derive an efficient implementation for the gradient vector.

6. A second-order all-pole adaptive filter is used to find the inverse model of the signal $x(k) = 1.7n(k-1)+0.81n(k-2)+n(k)$, where $n(k)$ is Gaussian white noise with variance 0.1. Using the gradient algorithm, calculate the error and the filter coefficients for the first 10 iterations. Start with $a_1(0) = 0, a_2(0) = 0$.

7. Repeat the problem 6 using the Gauss-Newton algorithm.

8. Use an IIR adaptive filter of sufficient order to identify a system with the transfer function given below. The input signal is a uniformly distributed white noise with variance $\sigma_x^2 = 1$, and the measurement noise is Gaussian white noise uncorrelated with the input with variance $\sigma_n^2 = 10^{-2}$. Use a Gauss-Newton based algorithm and the direct-form structure.

$$H(z) = \frac{0.000058(z^2 - 2z + 1)^3}{(z^2 + 1.645z + 0.701)(z^2 + 1.575z + 0.781)(z^2 + 1.547z + 0.917)}$$

   (a) Run the algorithm for three values of μ. Comment on the convergence behavior in each case.
   (b) Measure the MSE in each example.
   (c) Plot the obtained IIR filter frequency response at any iteration after convergence is achieved and compare with the unknown system.

9. Repeat the previous problem using a second-order adaptive filter and interpret the results.

10. A sinusoid of normalized frequency equal to $\frac{\pi}{4}$ with unit amplitude is buried in noise. The signal to noise ratio is 0 dB. Detect the sinusoid with notch filters using the lattice and the direct-form structures.

    (a) After convergence compute an estimate of the frequency by averaging the result of ten samples for each structure and comment on the result.

    (b) Depict the input signal and the output signal for the bandpass filter based on the lattice structure.

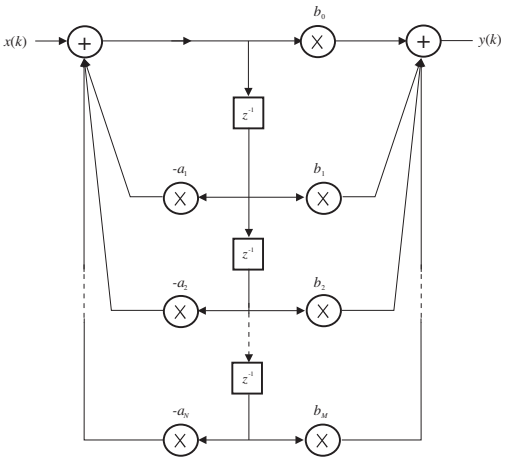


Figure 10.18 Direct form of Problem 5.

11. Replace the direct-form structure in problem 8 by the parallel realization with preprocessing.

12. Replace the direct-form structure in problem 8 by the cascade realization.

13. Repeat problem 8 in case the input signal is a uniformly distributed white noise with variance $\sigma_{n_x}^2 = 0.1$, filtered by all-pole filter given by

$$H(z) = \frac{z}{z - 0.95}$$

14. In problem 8 consider that the additional noise has the following variances (a) $\sigma_n^2 = 0$, (b) $\sigma_n^2 = 1$. Comment on the results obtained in each case.

15. Perform the equalization of a channel with the following transfer function

$$H(z) = \frac{z^2 - 1.359z + 0.81}{z^2 - 1.919z + 0.923}$$

using a known training signal that consists of a binary (-1,1) random signal. An additional Gaussian white noise with variance 10^{-2} is present at the channel output.
(a) Apply a Newton-based algorithm with direct-form structure.
(b) Plot the magnitude response of the cascade of the channel and the adaptive-filter transfer functions. Comment on the result.

16. In a system identification problem the input signal is generated by an autoregressive process given by

$$x(k) = -1.2x(k-1) - 0.81x(k-2) + n_x(k)$$

where $n_x(k)$ is zero-mean Gaussian white noise with variance such that $\sigma_x^2 = 1$. The unknown system is described by

$$H(z) = \frac{80z^3(z^2 + 0.81)(z - 0.9)}{(z^2 - 0.71z + 0.25)(z^2 + 0.75z + 0.56)(z^2 - 0.2z + 0.81)}$$

The adaptive filter is also a sixth-order IIR filter.
Choose an appropriate λ, run an ensemble of 20 experiments, and plot the average learning curve. Use the RLS algorithm for IIR filters.

17. A second-order IIR adaptive-filtering algorithm is applied to identify a 3rd-order time-varying unknown system whose coefficients are first-order Markov processes with $\lambda_{\mathbf{W}} = 0.999$ and $\sigma_{\mathbf{W}}^2 = 0.001$. The initial time-varying system multiplier coefficients are

$$\mathbf{w}_o^T = [0.03490 \ -0.011 \ -0.06864 \ 0.22391]$$

The input signal is Gaussian white noise with variance $\sigma_x^2 = 0.7$, and the measurement noise is also Gaussian white noise independent of the input signal and of the elements of $\mathbf{n}_{\mathbf{W}}(k)$, with variance $\sigma_n^2 = 0.01$.
Simulate the experiment described and plot the learning curve, by using the direct-form structure with a gradient-type algorithm.

18. Suppose a second-order IIR digital filter, with multiplier coefficients given below, is identified by an adaptive IIR filter of the same order using the gradient algorithm. Considering that fixed-point arithmetic is used, measure the values of $E[||\Delta\boldsymbol{\theta}(k)_Q||^2]$ and $\xi(k)_Q$ for the case described below. Plot the learning curves for the finite- and infinite-precision implementations. Also plot an estimate of the expected value of $||\Delta\boldsymbol{\theta}(k)||^2$ versus k in both cases.
    Additional noise: white noise with variance $\sigma_n^2 = 0.0015$
    Coefficient wordlength: $b_c = 16$ bits
    Signal wordlength: $b_d = 16$ bits
    Input signal: Gaussian white noise with variance $\sigma_x^2 = 0.7$

$$H(z) = \frac{z^2 - 1.804z + 1}{z^2 - 1.793z + 0.896}$$

19. Repeat the above problem for the following cases
    (a) $\sigma_n^2 = 0.01$, $b_c = 9$ bits, $b_d = 9$ bits, $\sigma_x^2 = 0.7$.
    (b) $\sigma_n^2 = 0.1$, $b_c = 10$ bits, $b_d = 10$ bits, $\sigma_x^2 = 0.8$.
    (c) $\sigma_n^2 = 0.05$, $b_c = 8$ bits, $b_d = 16$ bits, $\sigma_x^2 = 0.8$.

20. Replace the direct-form structure in problem 18 by the lattice structure, and comment on the results.

21. Repeat problem 8 using the LMSEE algorithm.

22. Show the inequality in equation (10.93).

23. Repeat problem 15 using the LMSEE algorithm.

24. Repeat problem 8 using a gradient-type algorithm based on the SM method.

25. Repeat problem 15 using a gradient-type algorithm based on the SM method.

26. Derive the RLS-type algorithm based on the SM method.