ADAPTIVE IIR FILTER ALGORITHMS FOR REAL-TIME APPLICATIONS

Thesis submitted in accordance with the requirements of the University of Liverpool for the Degree of Doctor of Philosophy

by

MARTIN CHRISTOPHER HALL

Department of Electrical Engineering and Electronics,
University of Liverpool

January 1987

Adaptive IIR Filter Algorithms for Real-Time Applications

MARTIN C. HALL

The characteristics of a digital filter are determined by a set of multiplier values (the filter coefficients). By changing the coefficient values under program control, the filter characteristics can be optimized with respect to a chosen performance criterion. Adaptive filters are used in applications where fixed filter design is impractical or impossible, such as when signal properties are unknown or vary with time. Adaptive IIR filters contain both poles and zeros in their transfer function, and hence are more powerful than adaptive non-recursive (FIR) filters. However, the algorithms required for IIR filter adaptation are more complex than those for FIR filter adaptation. In addition, IIR filter algorithms have several disadvantages associated with them which have restricted their use in the past. With the increasing need for low-cost, dedicated microprocessor-based systems, the development of simple, robust algorithms for IIR filter adaptation is of great importance.

In this thesis, previously published adaptation algorithms for IIR filters are examined with a view to real-time implementation. It is shown that the algorithms developed to date are not sufficiently robust, and that many are too complex for microprocessor implementation. The algorithm assessment suggests that simple gradient-based algorithms for adaptation of filter structures with guaranteed stability, are the most applicable for hardware implementation.

Three new gradient adaptive IIR filter algorithms are proposed for filter structures consisting of parallel and cascade arrangements of second-order filter sections. An assessment of the algorithms is made using a series of system modelling tests which show advantages and disadvantages of the differing filter structures; in particular a local minimum problem arising from constraints on the filter coefficients is highlighted. It is shown that a modified cascade structure offers the best overall performance of the structures considered. More complex gradient algorithms applicable to this filter structure are studied, and two new algorithms are proposed, the Valley-Search and R-Cosθ Algorithms. Results are presented to demonstrate the improved rates of convergence obtained by these algorithms.

The modified cascade structure is applied to a practical echo cancelling problem which demonstrates advantages of using IIR filters in preference to FIR filters. In addition, a method for the avoidance of local minima in the IIR filter output error performance surface is presented.

An examination of the equation error performance criterion is presented and a new algorithm for minimization of the equation error is proposed. By combining advantages of output error and equation error minimization, a novel gradient adaptive IIR filter structure is proposed. The Master-Slave Algorithm uses adaptive FIR filters for the adaptation of both the transfer function numerator and denominator coefficients, and offers faster convergence than conventional output error minimizing adaptive filters. Implementation of the algorithm requires very little additional computation, and stability of the filter structure is maintained easily. In addition, the Master-Slave Algorithm offers possible adaptation to the global minimum of a multi-modal output error performance surface.

ACKNOWLEDGEMENTS

I wish to express my gratitude to Dr. P.M. Hughes for his guidance, support and encouragement throughout the course of this research, and to Dr. B.M.G. Cheetham for his supervision in the latter stages of the project. I should also like to thank Dr. J.M. Rollett of British Telecom Research Laboratories for sponsorship of the project and for the industrial experience gained at B.T.R.L..

Thanks also to Mr. P. Adams and Dr. B. Carpenter at B.T.R.L. and the Speech Group at Liverpool University.

Martin C. Hall.

This work was a C.A.S.E. project supported by the U.K. Science and Engineering Research Council and British Telecom Research Laboratories.

CONTENTS

Figures are located at the end of the relevant Chapters.

CHAPTER 1

1.0 INTRODUCTION

An Adaptive Filter is a filtering device which can modify its characteristics in response to an external stimulus. A digital filter is ideally suited to such a task, as the filter characteristics are determined by a set of multiplier (coefficient) values which may be changed easily under program control. The coefficients are adjusted to optimize the performance of the filter with respect to an error criterion, and hence achieve the required filter performance.

Adaptive digital filters are used in applications where *a-priori* filter design is either impractical or impossible, i.e. when signal properties are unknown or are variable with time. In recent years, an increasing number of signal processing and control engineering applications have been found for such filters. These include channel equalization[1], echo cancellation[53,58,59], noise cancellation[2,3], signal enhancement[20,66], adaptive differential pulse code modulation[4,21], system identification[13-17], and signal prediction[5,21,56]. Currently, problems associated with the adaptation of IIR (Infinite Impulse Response) filters have resulted in adaptive FIR (Finite Impulse Response) filters being used in the vast majority of cases. However, in certain applications, the use of an IIR filter can offer substantial computational savings and other benefits.

Adaptive IIR Filters have the advantage of being able to synthesize both poles and zeros in their responses, whereas FIR Filters may only synthesize zeros. Consequently, in applications where an adaptive filter having a highly selective frequency response is required (or alternatively a long impulse response), the order of the FIR filter needed may be prohibitively high for implementation. In addition, pole-zero models are preferable for adaptive control engineering applications since the filter parameters may be used to enhance the stability of an unknown system as well as controlling it[6].

Two major problems are, however, associated with the adaptation of IIR filters and have restricted their use. The migration of poles outside the z-plane unit circle will result in filter instability, and must therefore be avoided. Hence, maintaining filter stability during the adaptation process is of major importance. The second difficulty arises from characteristics of the performance criterion relating to the filter adaptation. In general, adaptive filters used in signal processing applications adapt their coefficients to minimize a mean-square-output error criterion[7,9]. A plot of this function with respect to the adaptive filter coefficients produces a multidimensional 'performance surface'. For an FIR ladder filter, the performance surface has been shown to be quadratic in shape[9]. Such a function has a single minimum which can be sought using simple and robust techniques[10] of which the Widrow LMS Algorithm[8] is perhaps the one most widely used. However, for an IIR filter, the corresponding surface is non-uniform and may contain local minima[11,12,22]. Both of these factors are extremely undesirable.

Despite the drawbacks of adapting IIR filters when compared with the adaptation of FIR filters, the increasing need for low cost, microprocessor-based real time systems encourages the development of a simple and robust IIR filter adaptation algorithm.

1.1 ADAPTIVE SYSTEMS

Adaptive systems can be grouped into three categories depending upon the objective of the adaptation. The classes are:

a) System Identification Systems.

b) Signal Estimation Systems.

c) Signal Correction Systems.

Adaptive techniques for the solution of Class (a) problems have been utilized in control engineering for many years[13-17]. Several of the techniques were first developed for off-line applications such as fitting a model to a block of data. To allow on-line operation, the algorithms were first modified by incorporating a recursive update formula, and then later stochastic approximations were developed to reduce the computation further. Methods used include Maximum Likelihood, Least Squares, Instrumental Variables, and Kalman Filtering. In general, the techniques involve the minimization of a performance criterion based on the prediction or equation error[18]. For the majority of signal processing applications, these algorithms and performance criteria are unsatisfactory.

Unfortunately, the complexity of the algorithms makes them unsuitable for a dedicated hardware environment. In addition, many of the algorithms have coefficient updates which decay to zero during the adaptation process. While being satisfactory for identification of stationary systems, many signal processing applications involve filter adaptation in a non-stationary environment, for which the coefficient updates cannot decay towards zero. Furthermore, whilst system identification is generally the target for adaptive systems in control engineering, the vast majority of signal processing applications fall into the signal estimation and correction categories[7]. The prediction/equation error criteria are not suited to these problems. Instead, minimization of a performance criterion based on the mean-square-output error is generally the target for the adaptation.

1.2 ADAPTIVE DIGITAL FILTERING

An adaptive digital filter is a signal processing device that sequentially adjusts its coefficients to optimize its

performance with respect to an error criterion. A generalized model of an adaptive digital filter is shown in figure 1.1. For the z-transform of an input X(z), the filter transfer function $\hat{H}(z)$ is required to adapt to minimize a function of the error signal E(z), the difference signal formed between the filter output Y(z) and a desired response D(z). E(z) is obtained from:

$$E(z) = D(z) - Y(z) \tag{1.1}$$

with error e(k) at time $t = kT$.

Ideally, each individual signal processing application should have its own optimization criterion depending upon its practical objective. For example, with Linear Predictive Coding (L.P.C.) analysis of speech, the quality of the reproduced speech to the human ear is the ultimate criterion for the filter adaptation. While other performance criteria have been suggested for filter adaptation[19], in practice, minimization of the mean-squared-output error ($\overline{e^2(k)}$) is chosen in the vast majority of cases. There are several reasons for this[7], viz:

(1) The ease with which an algorithm can be derived for optimization.

(2) If the error is small or reaches zero, the actual desired criterion will usually be met.

(3) The mean-square-error criterion is equivalent to the maximum likelihood criterion for Gaussian stochastic processes.

(4) Small deviations from the optimum usually do not give a large degradation in performance.

To minimize this error criterion, for the given input signal the adaptive filter changes its coefficients to give an output which forms a best least squares fit to the desired response. While the objective for the filter adaptation differs depending upon the application, for each of the three adaptation categories listed in Section 1.1, the necessary filter configurations can be reduced to the

general model shown in figure 1.1. The differences between each of the three classes are now examined in greater detail.

a) System Identification

The System Identification Configuration is shown in figure 1.2. An unknown "plant" H(z) and the adaptive filter $\hat{H}(z)$ are both excited by the same input signal X(z). This input signal is generally either White Noise or a pseudo-random sequence. The objective of the filter adaptation is to determine the parameters of the unknown system. This is achieved by adapting the adaptive filter to minimize $\overline{e^2(k)}$. To completely identify the unknown plant, the adaptive filter must have at least as many poles and zeros in its transfer function as that of the system. In this case, the mean-squared-error (m.s.e.) signal will tend to zero.

Adaptive control systems are almost invariably of this type. The adaptive filter estimates the unknown system parameters (or its state). These parameters can be used to control the system by changing its inputs and/or its parameters[7]. Signal processing tasks which fall into this category are, characterization of the layers of the ground in seismology through the adaptive filter parameters, and indications of significant events in Electroencephalograph (EEG) analysis and other event detection applications through variations in the parameters.

b) Signal Estimation

The basic configuration for this class of system is shown in Figure 1.3. At first appearances, the problem seems identical to that of system identification. The filter configuration is the same as that for the previous category, and in both cases the filter is adapted to minimize $\overline{e^2(k)}$. However, there are several important differences between the two classes. Parameters determined by this class generally do not correspond to those of the unknown system. For example, typical

applications from this category such as echo cancellation and noise cancellation will generally have an insufficient adaptive filter (ie. one having a smaller number of poles or zeros than the unknown system). Any set of filter coefficients which can give a sufficiently accurate copy of the unknown system output to reduce $\overline{e^2(k)}$ below the necessary threshold will be acceptable.

In addition, this only has to be achieved for the type of input signal used. For instance, if the input signal X(z) will always be band limited, then the response of the adaptive filter need model that of the unknown system only for signals with this band limitation.

c) Signal Correction

Figure 1.4 shows the basic structure for a signal correction system. Signal processing tasks which fall into this category are channel equalization, line enhancement, and linear prediction of speech. For these applications, the wanted signal X(z) is distorted or hidden by an unknown system, and the task for the adaptive filter is to retrieve it.

As with adaptive filters from the signal estimation class, adaptive filters from this class do not have to form an accurate model of the unknown system. For example, for channel equalization, the adaptive filter only has to remove the distortion components which are harmful to the receiver, and does not necessarily have to form a complete inverse of H(z). Similarly, for LPC analysis of speech, any set of filter coefficients which can reproduce the speech to the quality necessary and can be coded efficiently will be acceptable.

Minimization of the mean-square-output error criterion enables the adaptive filter to accomplish the goals of all three classes of adaptive systems listed above. However, in the case of an IIR filter, the nature of the m.s.e. performance surface creates problems for the

filter adaptation. In the following section, characteristics of the IIR filter performance surface are studied, and the inherent difficulties for filter adaptation are highlighted.

1.3 THE IIR FILTER PERFORMANCE SURFACE

The IIR Digital Filter

The output of an IIR digital filter at time t = kT can be represented by a difference equation of previous input and output values and the present input:

$$y(k) = \sum_{i=0}^{n} x(k-i).a_i - \sum_{i=1}^{m} y(k-i).b_i \qquad (1.2)$$

where $\{a_i\}$ are the feedforward coefficients, $\{b_i\}$ are the feedback coefficients, and x(k-i),y(k-i) are past input and output values respectively. The order of the filter is given by the larger value of the m and n indices. The IIR Filter gives rise to an Autoregressive Moving Average (ARMA) process. The first summation term in equation (1.2) is a moving average process, and the second summation term is an autoregressive process. If the $\{b_i\}$ coefficients are all zero, the resulting filter is an FIR ladder filter.

Taking z-transforms of equation (1.2) gives the general transfer function H(z) of an IIR filter:

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_n z^{-n}}{1 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_m z^{-m}} \qquad (1.3)$$

Where:

$$H(z) = \frac{Y(z)}{X(z)} \qquad (1.4)$$

The transfer function can be realized in a number of different structures, such as parallel and cascade arrangements of lower order filters, or alternatively as a lattice structure. However, in terms of hardware implementation, the simplest filter structures are the canonic forms, so called because they have the minimum number of component elements[65]. Figure 1.5 shows a Direct Form Canonic Implementation (Direct Form II Structure in[64]) of the transfer function given in equation (1.3). The Direct Form Filter Structure is utilized for the majority of adaptive IIR filtering algorithms. The m-s-e performance surface for the Direct Form IIR Filter Structure Coefficients was studied in detail by Stearns in[12]. A brief review of his analysis is presented below.

The Direct Form IIR Filter Performance Surface

For the general adaptive filter structure shown in figure 1.1, the mean-square-error surface is generated from:

$$E[e^2(k)] = E[d^2(k)] + E[y^2(k)] - E[2d(k)y(k)] \qquad (1.5)$$

$$\forall \ (a_i, b_j) \quad \begin{cases} i=0,1,..,n \\ j=1,2,..,m \end{cases}$$

Where $E[u]$ is the expected value of u.

Equation (1.5) can be written in terms of the adaptive filter transfer function ($\hat{H}(z)$) by noting that the expected value of two stochastic signals u and v is given by:

$$E[uv] = \frac{1}{2\pi j} \oint \Phi_{uv}(z) \frac{dz}{z} \qquad (1.6)$$

where $\Phi_{uv}(z)$ is the cross power spectrum and the path of integration is the unit circle in the z-plane. Strictly, u and v should be stationary, random signals with zero mean and finite variance.

Utilizing the transfer function relationships:

$$\Phi_{yy}(z) = \Phi_{xx}(z) \left| \hat{H}(z) \right|^2 \tag{1.7}$$

$$\Phi_{dy}(z) = \Phi_{dx}(z)\hat{H}(z) \tag{1.8}$$

, equation (1.5) can be expressed as:

$$2\pi j E[e^2(k)] = \oint \Phi_{dd}(z)\frac{dz}{z} + \oint \Phi_{xx}(z)\hat{H}(z)\hat{H}^*(z)\frac{dz}{z} - 2\oint \Phi_{dx}(z)\hat{H}(z)\frac{dz}{z} \tag{1.9}$$

Where $\hat{H}^*(z) = \hat{H}(z^{-1})$, since the contour for the integration is the unit circle, $\hat{H}(z)\hat{H}^*(z) = \left|\hat{H}(z)\right|^2$. It is seen from equations (1.9) and (1.3) that $E[e^2(k)]$ is quadratic with respect to the $\{a_i\}$ filter coefficients.

For the Direct Form IIR Filter Structure, $\hat{H}(z) = \hat{A}(z)/\hat{B}(z)$

where:

$$\hat{A}(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_n z^{-n} \tag{1.10}$$

$$\hat{B}(z) = 1 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_m z^{-m} \tag{1.11}$$

Setting the derivative of equation (1.9) with respect to each $\{a_i\}$ coefficient equal to zero, and noting that $\Phi_{xd}(z) = \Phi_{dx}^*(z)$ for any power spectrum yields:

$$\oint \Phi_{xx}(z) \frac{\hat{A}(z)z^k}{\hat{B}(z)\hat{B}^*(z)}\frac{dz}{z} = \oint \Phi_{xd}(z) \frac{z^k}{\hat{B}^*(z)}\frac{dz}{z} \qquad k=0,1,\ldots n \tag{1.12}$$

The set of equations generated from equation (1.12) are linear with respect to the $\{a_i\}$ coefficients and hence define a single global

minimum of $E[e^2(k)]$. If equation (1.12) has a singular coefficient matrix, then the minimum will be distributed. Setting the derivatives of $E[e^2(k)]$ with respect to the $\{b_i\}$ coefficients in equation (1.9) equal to zero yields:

$$\oint \Phi_{xx}(z) \frac{\hat{A}(z)\hat{A}^*(z)z^k}{\hat{B}(z)\hat{B}^*(z)\hat{B}^*(z)} \frac{dz}{z} = \oint \Phi_{xd}(z) \frac{\hat{A}^*(z)z^k}{\hat{B}^*(z)\hat{B}^*(z)} \frac{dz}{z} \qquad (1.13)$$

$$k=1,2,\ldots,m$$

The set of equations generated from equation (1.13) are nonlinear with respect to the $\{b_i\}$ coefficients. Hence the error surface with respect to the $\{b_i\}$ coefficients is non-quadratic and may contain local minima. To examine characteristics of the error surface with respect to the $\{b_i\}$ filter coefficients, Stearns utilized the system identification configuration shown in figure 1.2.

The adaptive filter $\hat{H}(z)$ and a known fixed plant $H(z)$ are excited by a white Gaussian noise source x of unit power density (hence $\Phi_{xx} = 1$). Applying transfer function relationships to the system identification configuration yields:

$$\Phi_{dd}(z) = \Phi_{xx}(z)\left|H(z)\right|^2 \qquad (1.14)$$

$$\Phi_{dx}(z) = \Phi_{xx}(z)H(z) \qquad (1.15)$$

Substituting equations (1.14) and (1.15) in equation (1.9) and setting $\Phi_{xx} = 1$ yields:

$$E[e^2(k)] = \frac{1}{2\pi j} \oint \left[H(z)H^*(z) + \hat{H}(z)\hat{H}^*(z) - 2\hat{H}(z)H^*(z)\right]\frac{dz}{z} \qquad (1.16)$$

As the $E[d^2(k)]$ term in equation (1.5) will be constant for a fixed plant, a normalized version of the error surface $\xi(a_i,b_i)$ can be obtained from equation (1.16), viz:

$$\xi(a_i,b_i) = 1 + \frac{1}{2\pi\sigma^2_j} \oint \hat{H}(z)\left[\hat{H}^*(z) - 2H^*(z)\right]\frac{dz}{z} \qquad (1.17)$$

where:

$$\sigma^2 = \frac{1}{2\pi j}\oint |H(z)|^2 \frac{dz}{z} = E[d^2(k)] \qquad (1.18)$$

Stearns demonstrated that for a sufficient sized adaptive filter, ie. the adaptive filter has at least as many poles and zeros as the plant, then ξ has a single minimum at which $\xi = 0$. This minimum will be distributed if the adaptive filter is over-sufficient.

As previously shown, the error surface is quadratic with respect to the (a_i) coefficients. To illustrate the non-uniform nature of the performance surface with respect to the (b_i) coefficients, plots were obtained of ξ for several second order all-pole plants and a second order all-pole adaptive filter. The adaptive filter had the transfer function $\hat{H}(z)$ given by:

$$\hat{H}(z) = \frac{a_0}{1 + b_1 z^{-1} + b_2 z^{-2}} \qquad (1.19)$$

For a second order adaptive filter the stable region for the poles of the filter transfer function, that is, the interior of the unit circle in the z-plane, maps onto a triangular region in the (b_1,b_2) coefficient plane. This triangular region is shown in figure 1.6. The parabola $b_2 = b_1^2/4$ separates the real poles from complex conjugate

pairs. The radius (r) and angle (θ) of the complex poles are related to the b_1 and b_2 coefficients by:

$$b_1 = -2.r.\cos\theta \tag{1.20}$$

$$b_2 = r^2 \tag{1.21}$$

Plots of the normalized error ξ for four different plants are shown in figure 1.7. The contours of constant m-s-e illustrate the highly irregular nature of the surfaces. The plots are characterised by steep convex-sided valleys and in general the gradient vector from a point on the surface will not point towards the minimum of the surface.

If the adaptive filter is insufficient, ie. has fewer poles or zeros than the plant, Stearns showed that local minima may exist in the error surface. However, conditions for the existence of local minima have been investigated in only very limited situations[12,29]. In an insufficient case, the error surface remains quadratic with respect to the $\{a_i\}$ coefficients, irrespective of the order of the adaptive filter. Hence, problems in the adaptation of the filter coefficients lie principally with the $\{b_i\}$ coefficients, as the $\{a_i\}$ coefficients can adapt to optimum values provided that the $\{b_i\}$ coefficients adapt to their optimum values.

An example of an insufficient adaptive filter is given in figure (1.8). The plant had a transfer function consisting of a single zero, while the adaptive filter transfer function was that given in equation (1.19). For this example, it is seen that the error surface is bimodal with respect to the $\{b_i\}$ coefficients.

The irregular nature of the IIR mean-square-error surface suggests that adaptation times using gradient techniques will be generally slower than those for FIR filters. However, a long FIR ladder filter, for example, will need a small adaptation parameter[8] and hence will have a slow convergence rate. A low order IIR filter may adapt in

fewer iterations and require less computation. In addition, as suggested by Etter and Stearns[22], the tracking capabilities required with non-stationary signals may be superior in the IIR case because of the steep error surface near the minimum.

1.4 THESIS CONTENTS

The work presented in this thesis is aimed towards the development of a robust algorithm for the adaptation of IIR Filters, with a view to real-time hardware implementation. Due to the necessary limitations in speed and complexity, the required computation should be kept to a minimum. The filter must remain stable and ideally should converge quickly to the minimum m.s.e. solution. As has been demonstrated in the previous sections of this chapter, the nature of the IIR filter m.s.e. performance surface makes this a very difficult problem.

In Chapter (2) a literature survey is presented of previously published adaptation algorithms for IIR Filters. It is shown that algorithms can be divided into two major classes, Gradient Algorithms and Hyperstable Algorithms, and that neither class offers a robust algorithm at present. However, for filter simplicity, stability and speed of adaptation, the Cascade and Lattice structure Gradient-Based algorithms are the most applicable for hardware implementation.

In Chapter (3), three new Gradient-Based Adaptive IIR Filter Algorithms are proposed using filter structures consisting of parallel and cascade arrangements of second order filter sections. It is shown that the new algorithms offer simple, stable adaptation with the "Modified Cascade Structure" offering the best performance of the algorithms considered.

In Chapter (4), methods for increasing the speed of convergence of the modified cascade structure are considered. Two new algorithms are

proposed; the "Valley-Search" algorithm and the "R-Cosθ" IIR Filter algorithm. Results demonstrate that the algorithms offer improvements in convergence times when compared with the algorithms of Chapter 3.

The Modified Cascade Structure is applied to a practical problem in Chapter (5) - that of Echo Cancellation for a 144 kbit Duplex Data Transmission System. The results obtained demonstrate the advantages of using adaptive IIR Filters in preference to FIR filters. In addition, a method for the avoidance of error surface local minima is presented.

In Chapter (6) a new approach to adaptive IIR filtering is presented which results in the Master-Slave Adaptation Algorithm. An examination of the equation error performance criterion is presented and a new algorithm for adaptation of a equation error minimizing adaptive filter is proposed. By combining advantages of output error and equation error minimization, a novel adaptive IIR Filter-structure is proposed. Using an arrangement of three FIR filters, the filter structure is stable, very simple to implement and has a fast speed of convergence. It is demonstrated that the algorithm gives possible adaptation to the global minimum of a multimodal performance surface.

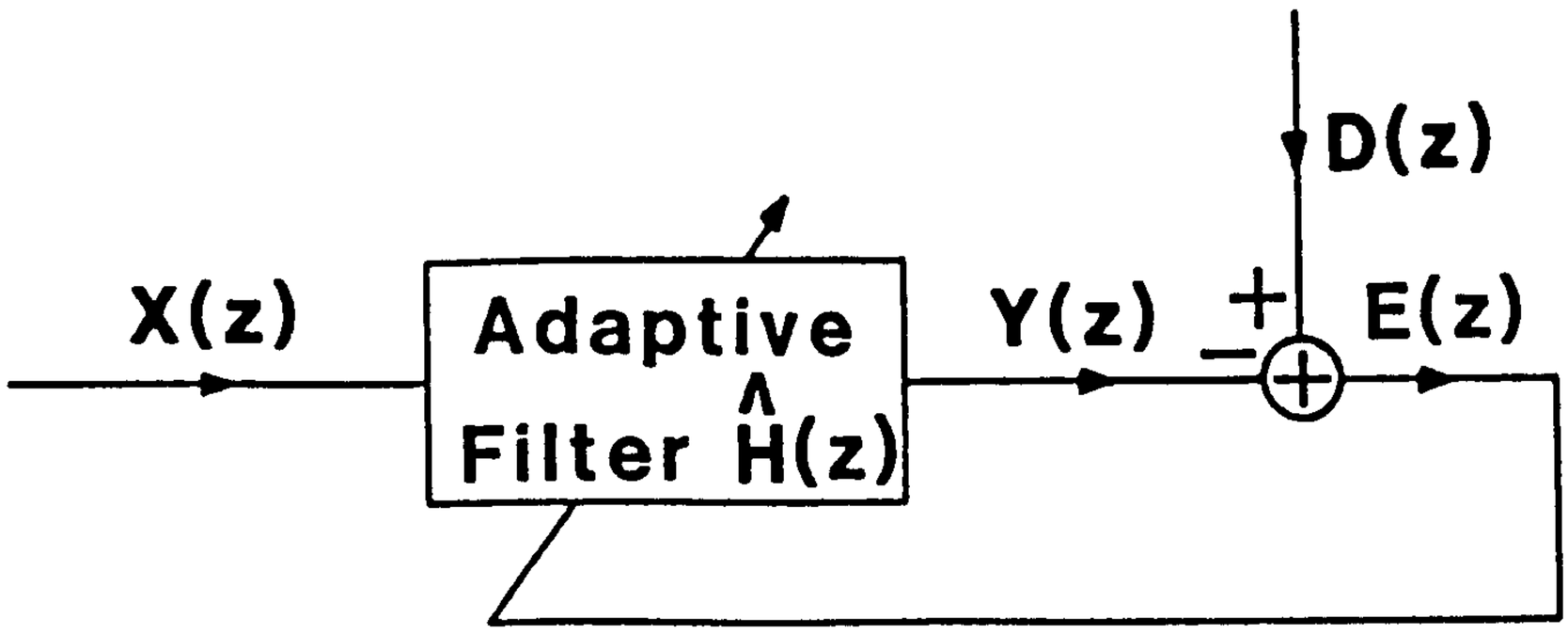


Figure 1.1 General Adaptive Digital Filter.

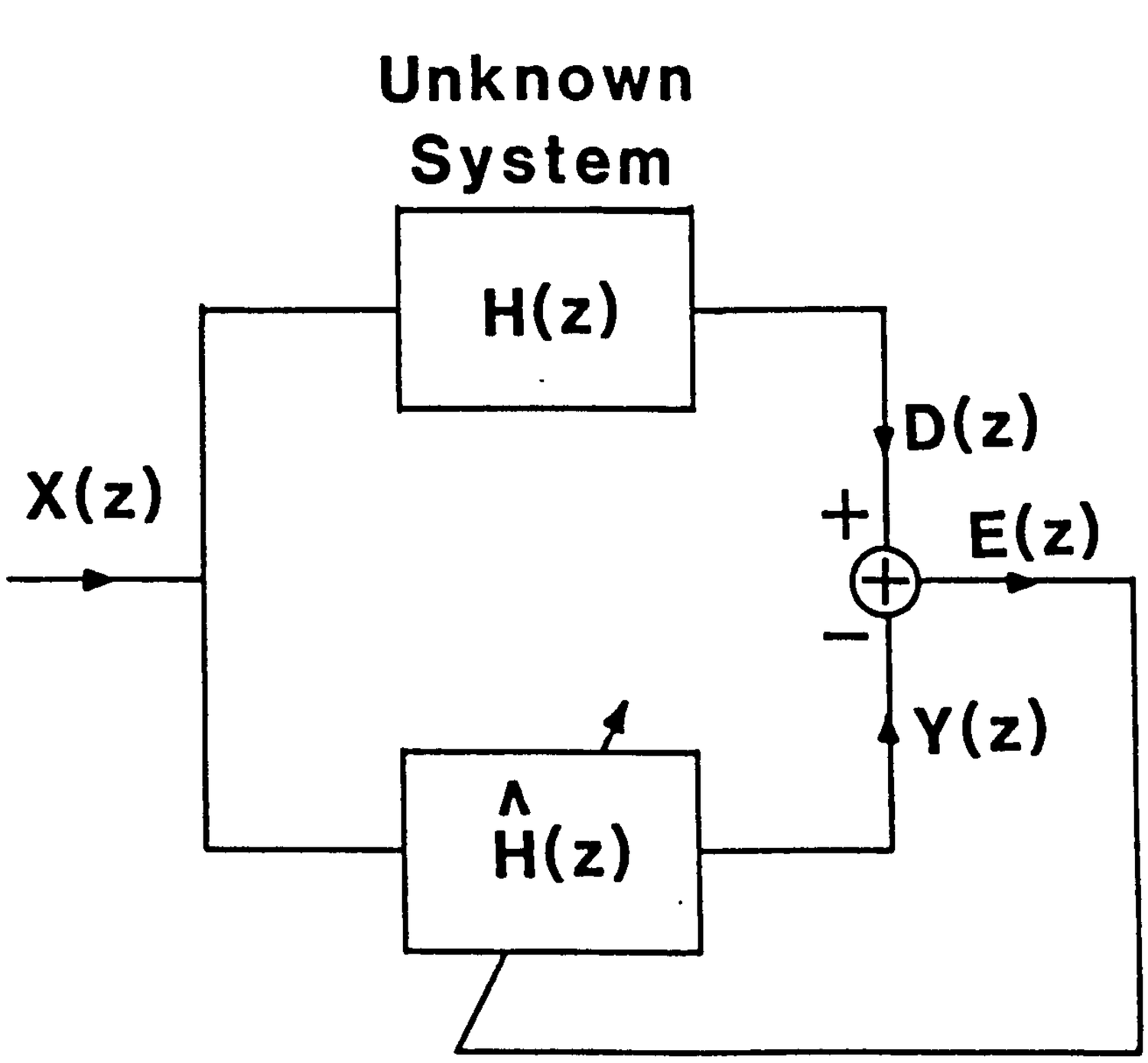


Figure 1.2 System Identification Structure.

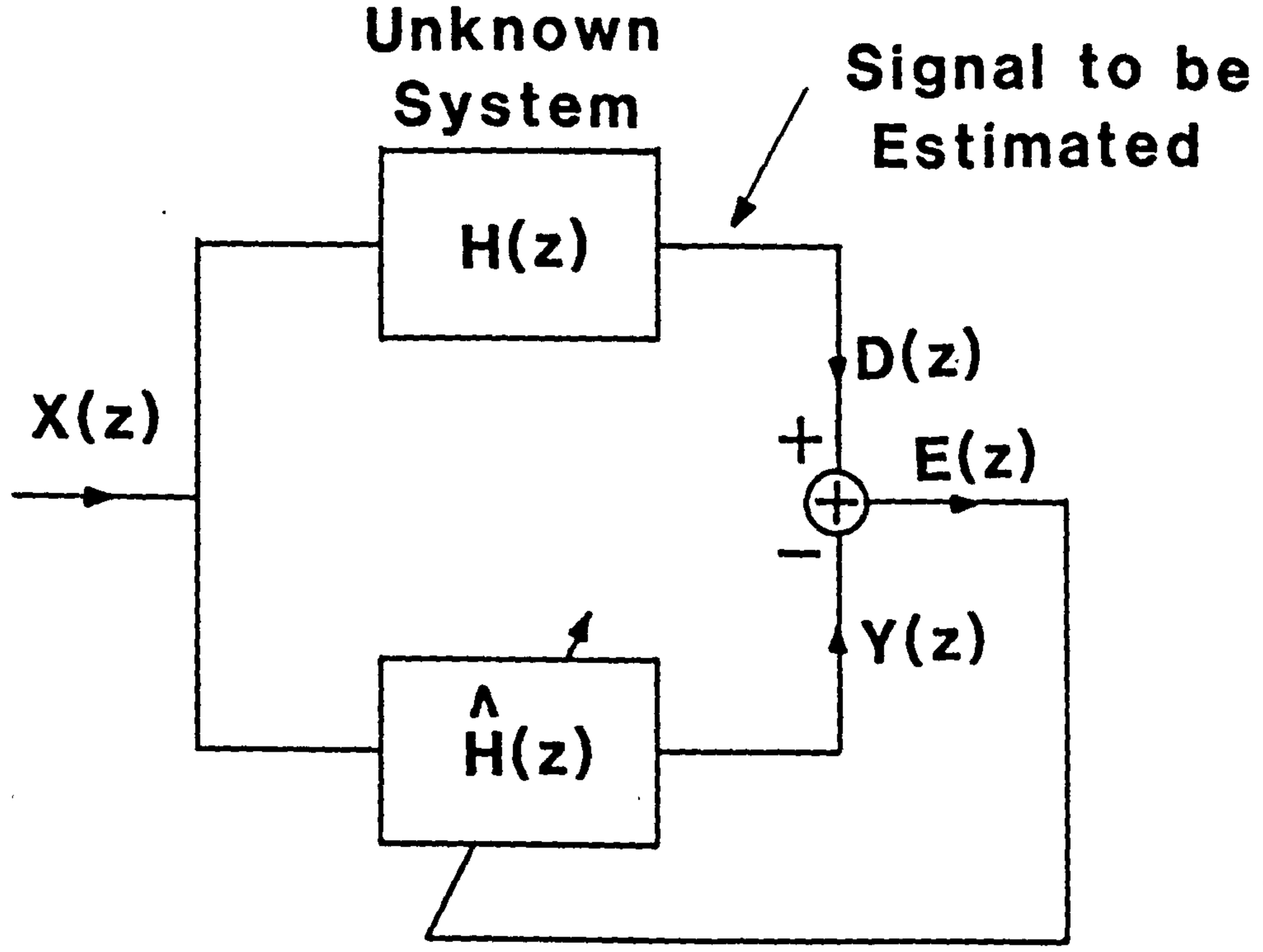


Figure 1.3 Signal Estimation Structure.

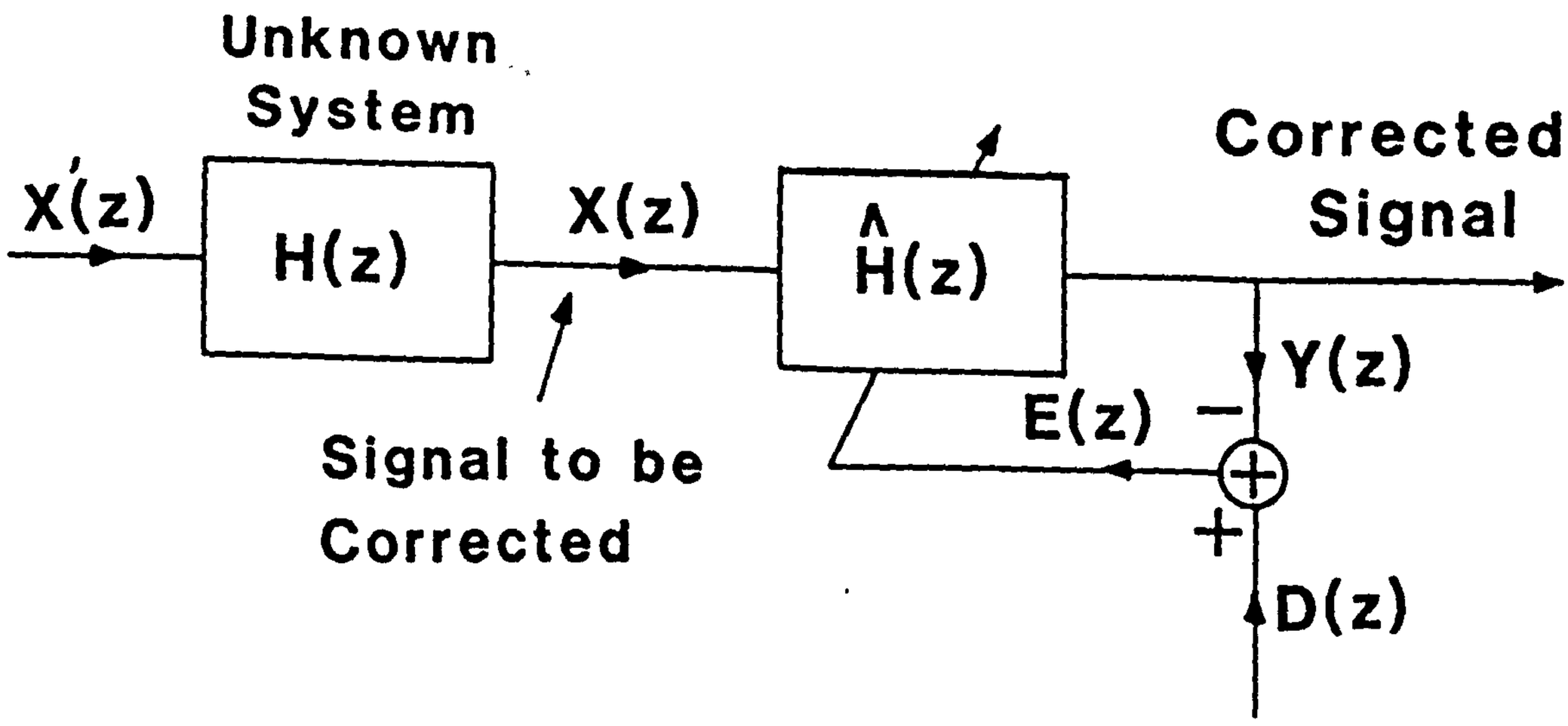


Figure 1.4 Signal Correction Structure.

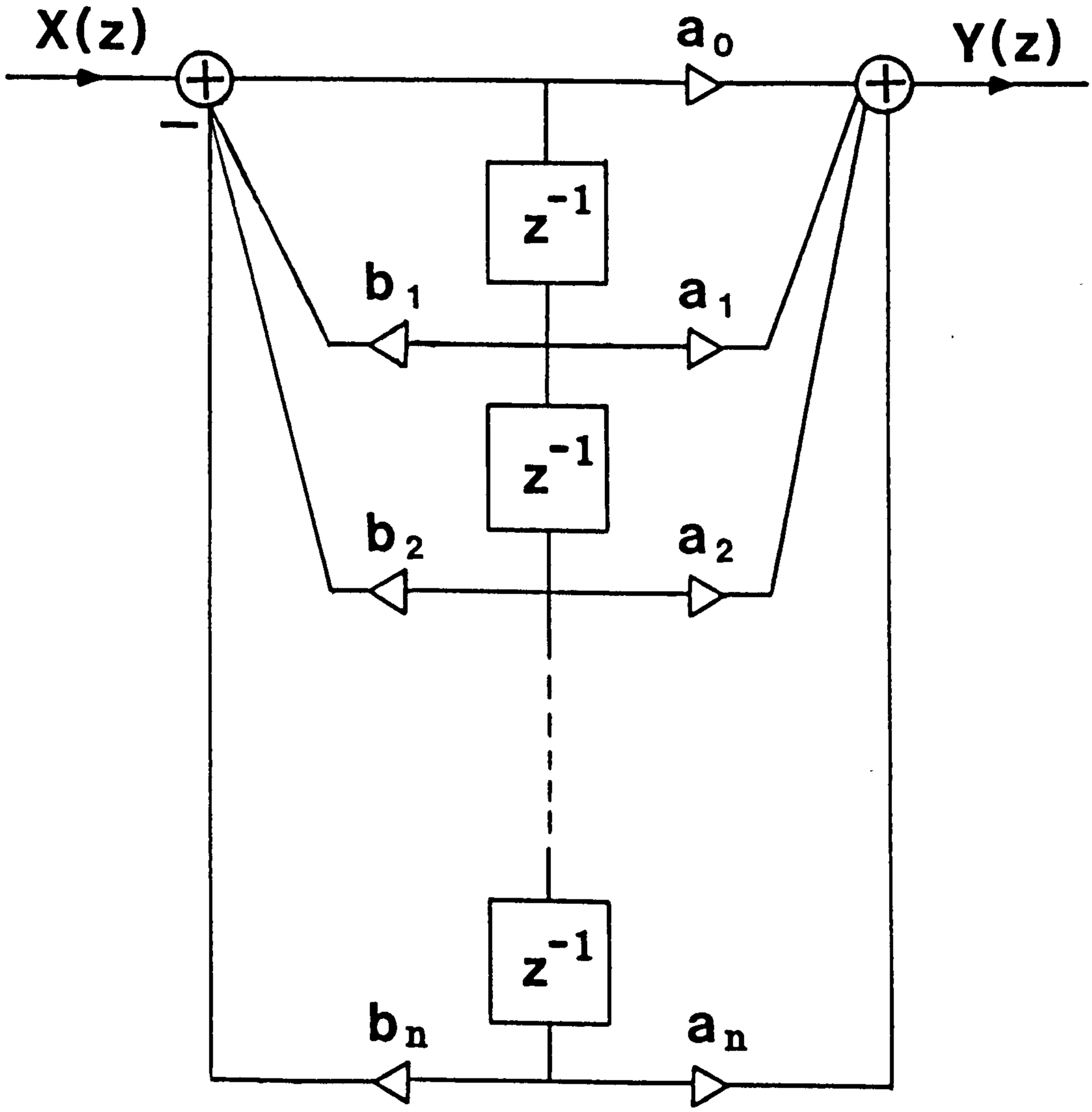


Figure 1.5 Direct Form IIR Filter Structure.

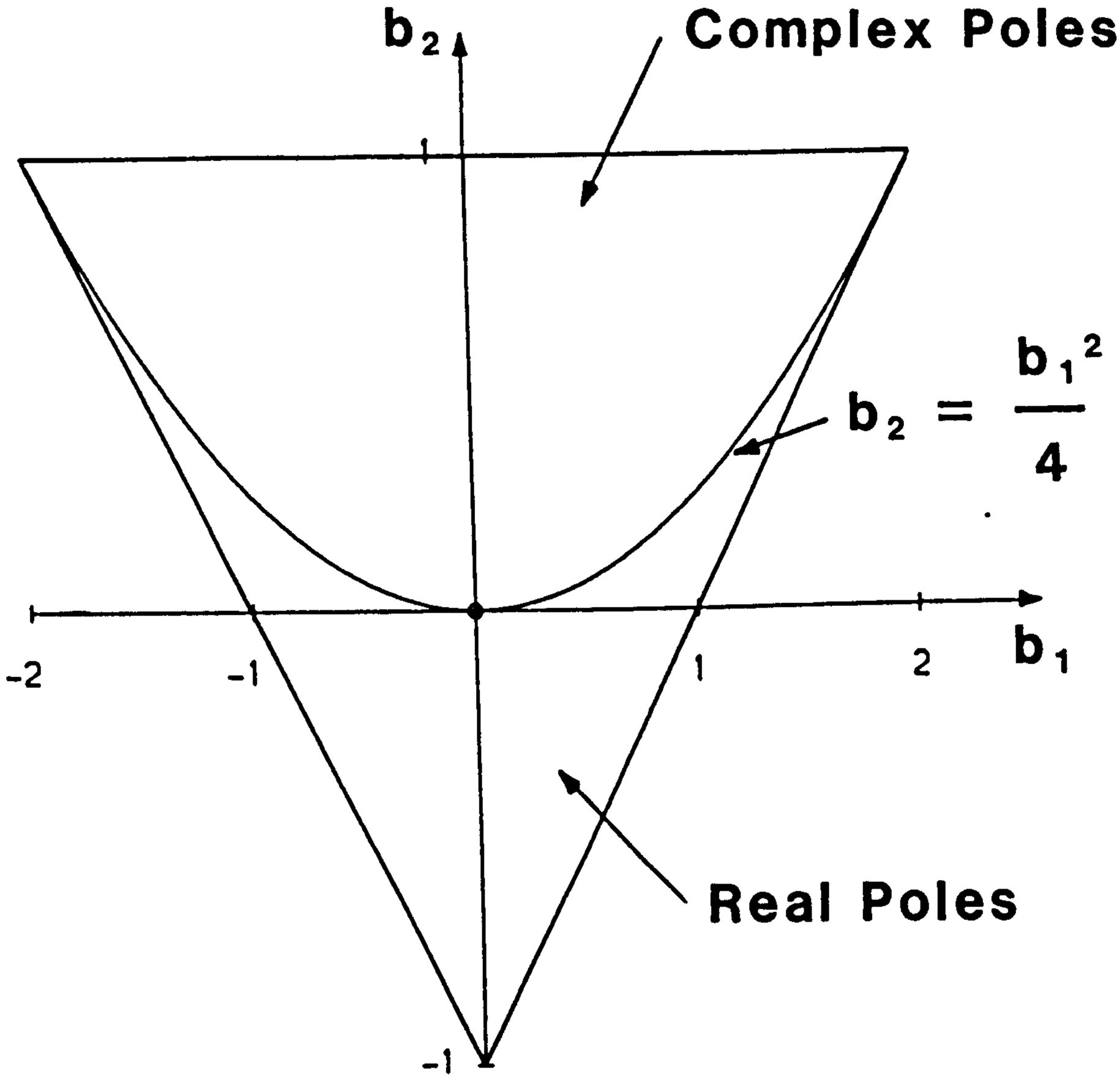
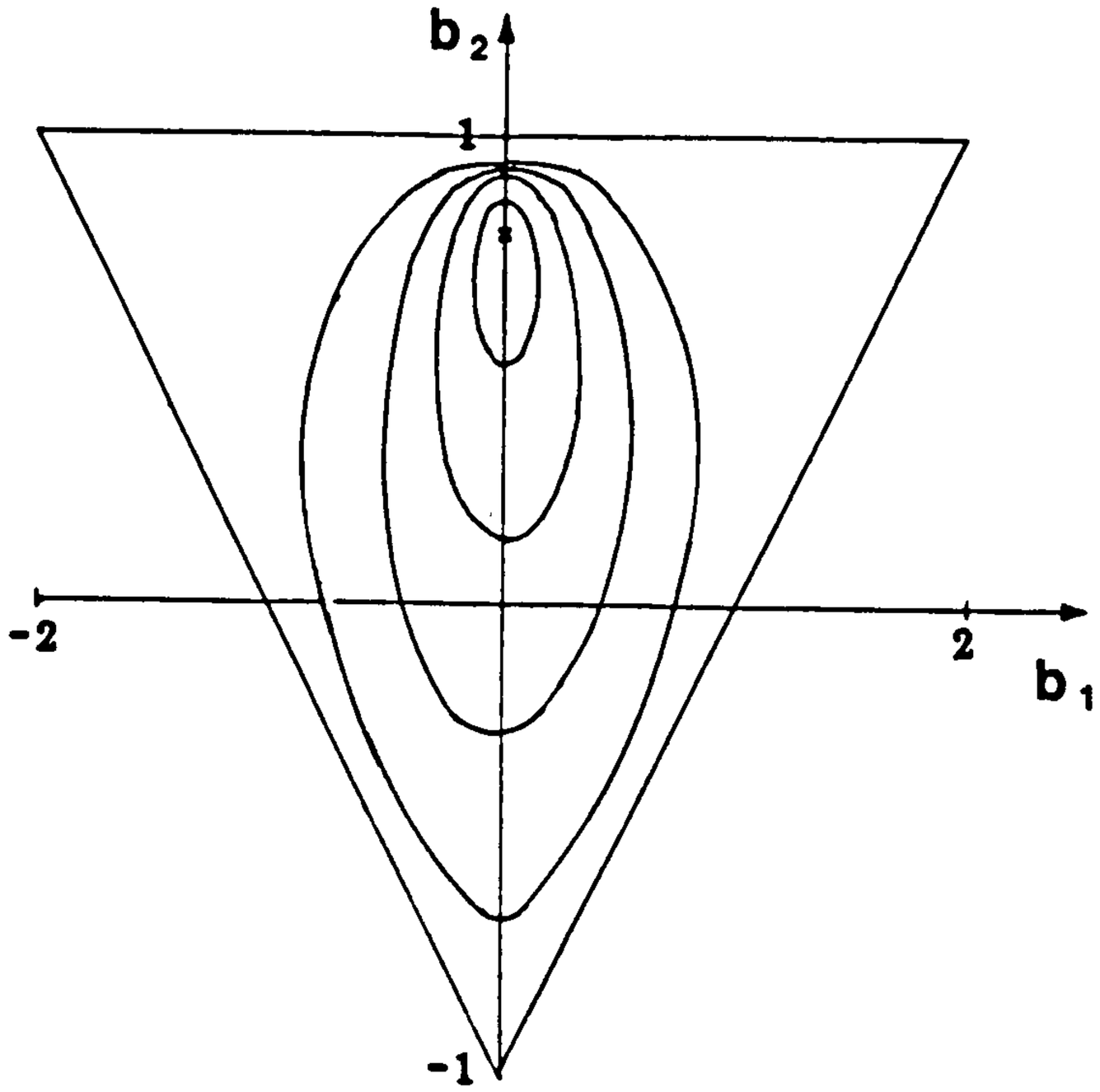
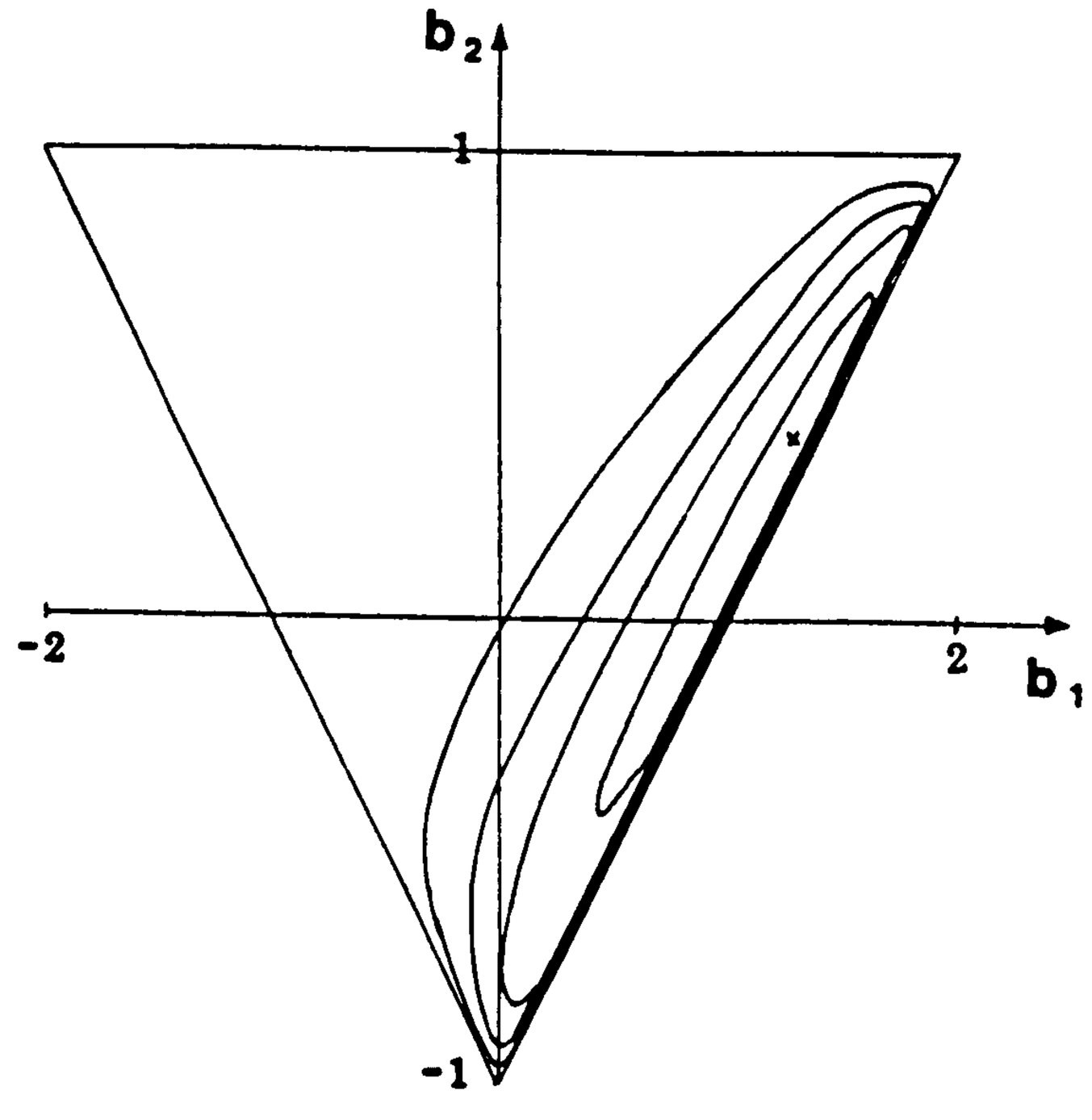


Figure 1.6 Stability Triangle.



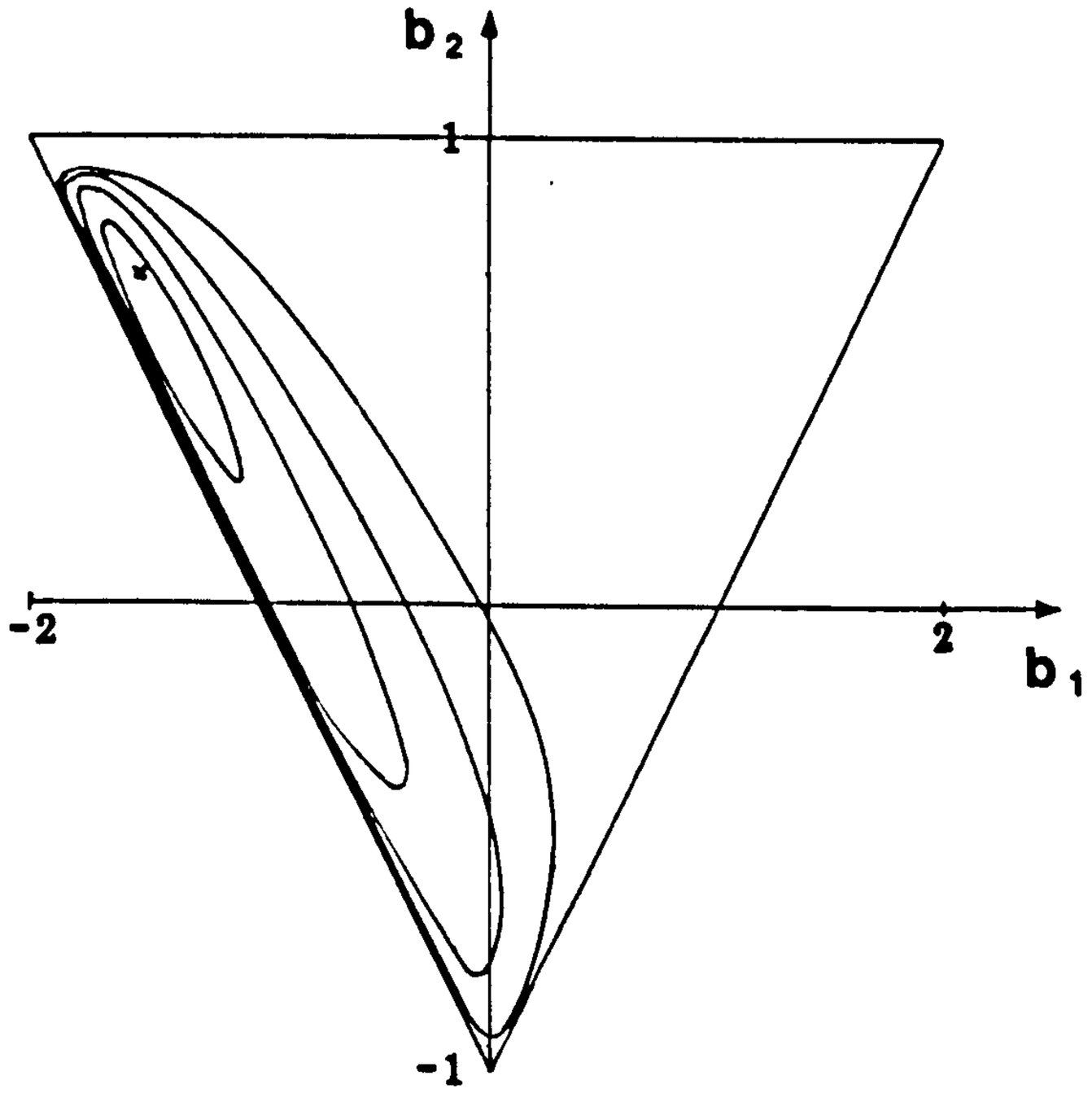
a) Target Plant: $(b_1,b_2) = (0.0,0.8)$



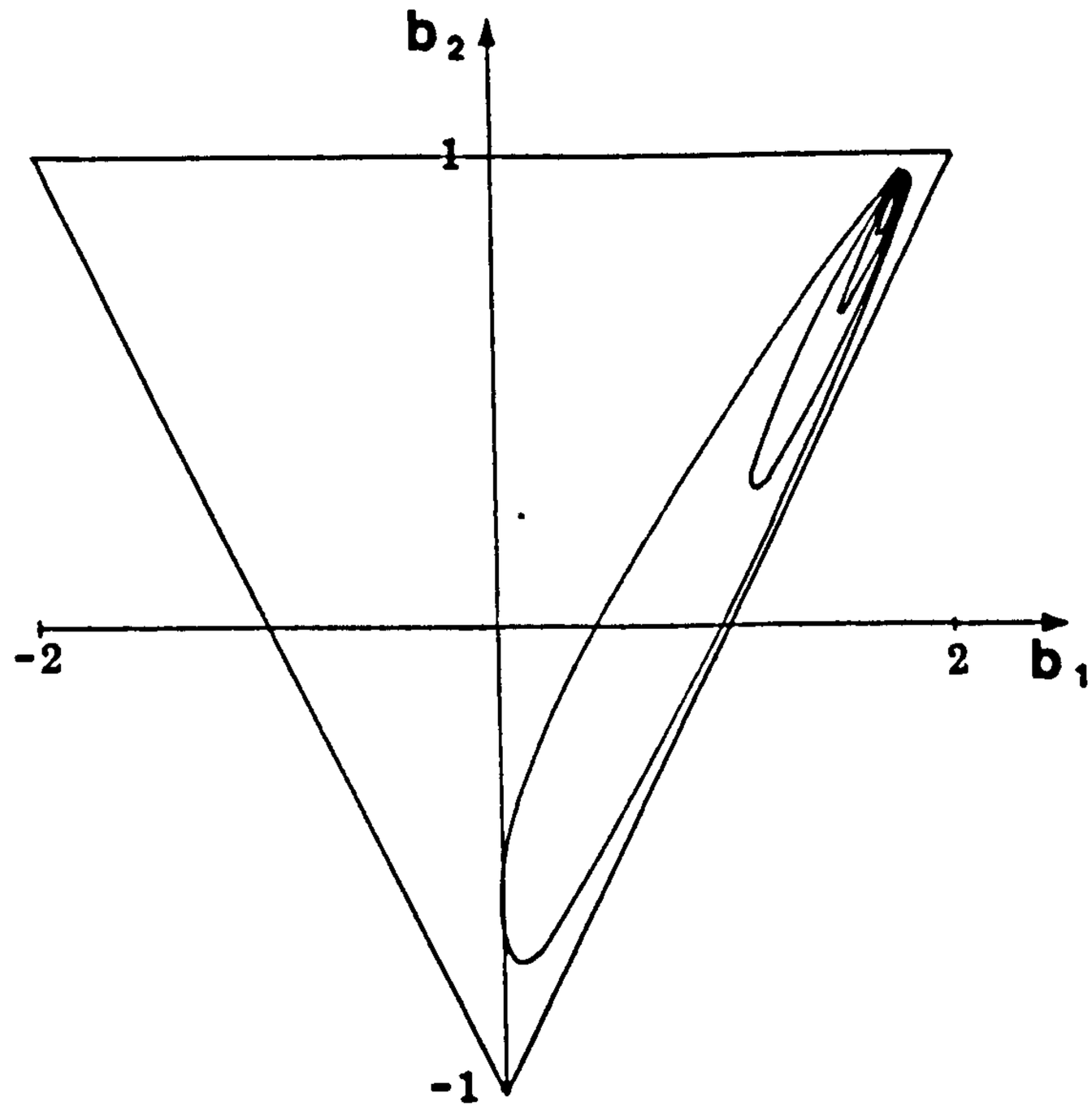
b) Target Plant: $(b_1,b_2) = (1.3,0.4)$

Figure 1.7 Normalized Error Surface Plots.

Contours at 0.05, 0.27, 0.5, 0.73.



c) Target Plant: $(b_1,b_2) = (-1.5,0.7)$



d) Target Plant: $(b_1,b_2) = (1.8,0.95)$

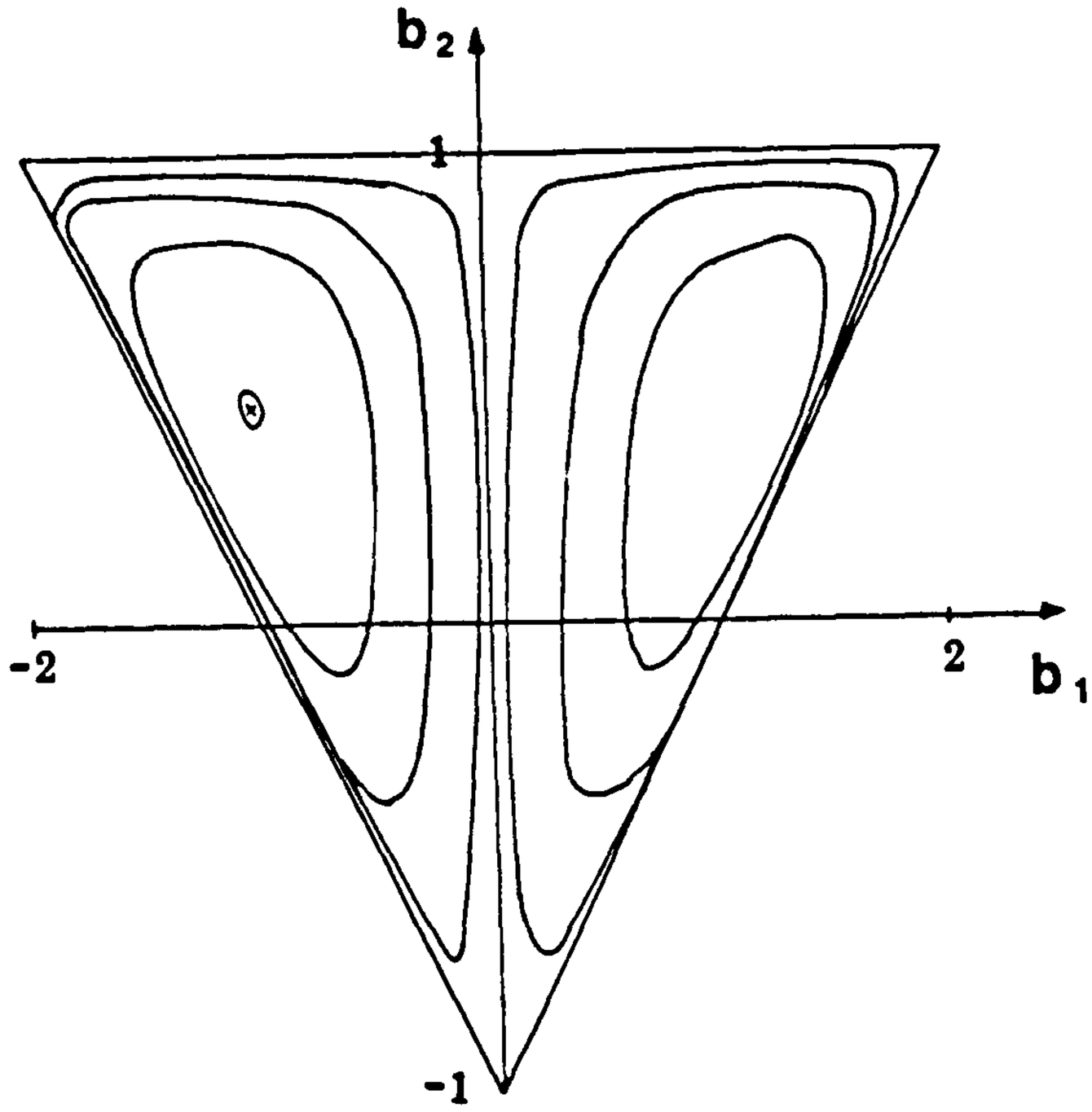


Figure 1.8 Insufficient Order Error Surface Example.

Plant Transfer Function $H(z) = 1 + 10z^{-1}$

CHAPTER 2

2.0 ADAPTATION ALGORITHMS FOR IIR FILTERS

At present, IIR filter adaptation algorithms can be separated into three classes; 1) Gradient Algorithms, 2) Hyperstable Algorithms and 3) Random Search Algorithms. The majority of the algorithms utilize the Direct Form IIR Filter Structure, although algorithms have been proposed for other structures, such as cascaded filter structures and the recursive lattice filter. In this chapter, brief derivations of the adaptation algorithms are presented. An assessment is made of each algorithm based on results published in the literature, and tests and comparisons made by the author. In Section 2.1 the Direct Form Structure algorithms are considered, whilst in Section 2.2 algorithms which employ other filter structures are examined. Conclusions on the relative merits of the algorithms are presented in Section 2.3 as a foundation for the work presented in Chapters 3 → 6 of the thesis.

2.1 DIRECT FORM FILTER STRUCTURE ALGORITHMS

2.11 Gradient Algorithms

Algorithms belonging to this class have the filter adaptation process driven by derivatives of the m.s.e. surface with respect to the filter coefficients. The first IIR filter Gradient Algorithm was proposed in 1975 by White[23]. By extending the Widrow LMS algorithm for the adaptation of FIR Ladder Filters[8] to a Direct Form IIR Filter Structure, White proposed a steepest descent approach now termed Recursive Least Mean Squares (RLMS). The algorithm was further developed by Stearns et al[24,25].

The steepest descent approach is a first order gradient method, and as such is relatively simple to implement. To adapt the filter cofficients, the slope of the error surface is calculated for the present set of coefficients, and the coefficient values are adjusted in

the opposite direction to the gradient vector. Given a particular coefficient set, the updated coefficients are calculated using:

$$\underline{W}_{k+1} = \underline{W}_k + \underline{\mu}(- \nabla_{\underline{W}_k} \overline{e^2(k)}) \qquad (2.1)$$

where $\underline{W}_k$ is the vector of coefficients $\{a_i\}$ and $\{b_i\}$ at time $t = kT$, $\nabla_{\underline{W}_k} \overline{e^2(k)}$ is the vector of partial derivatives of the m.s.e. and $\underline{\mu}$ is a diagonal matrix of convergence factors controlling the rate of adaptation. Provided that the values in $\underline{\mu}$ are correctly chosen, continual adaptation of each coefficient value in the opposite direction to the gradient vector drives the filter into a local minimum of the error surface.

In addition to the White algorithm, other direct form structure algorithms employing the steepest descent approach are the Hsia/Horvath and Feintuch algorithms. The differences between the algorithms lie in simplifications made in the calculation of the gradient vector.

2.111 White/Stearns Algorithm

From the general adaptive filter structure shown in figure 1.1, the error at time $t = kT$ is given by :

$$e(k) = d(k) - y(k) \qquad (2.2)$$

Hence, the m.s.e. value is obtained from:

$$\overline{e^2(k)} = \overline{d^2(k)} + \overline{y^2(k)} - \overline{2d(k)y(k)} \qquad (2.3)$$

As $\overline{e^2(k)}$ is not readily available, the instantaneous squared error $e^2(k)$ is used as a local estimate. It was shown[6] that this

estimate is asymptotically unbiased. From equation (2.3), taking partial derivatives of the instantaneous squared error $e^2(k)$ with respect to the $\{a_i\}$ and $\{b_i\}$ filter coefficients yields:

$$- \frac{\partial e^2(k)}{\partial a_i} = 2.e(k).\alpha_i(k) \qquad i = 0,1,..,n \tag{2.4}$$

$$- \frac{\partial e^2(k)}{\partial b_i} = 2.e(k).\beta_i(k) \qquad i = 1,2,..,m \tag{2.5}$$

Where:

$$\alpha_i(k) \triangleq \frac{\partial y(k)}{\partial a_i} \tag{2.6}$$

$$\beta_i(k) \triangleq \frac{\partial y(k)}{\partial b_i} \tag{2.7}$$

Due to the recursive nature of the filter, the output y(k) at time t = kT contains transients from past coefficient changes. This makes calculation of the $\alpha_i(k)$ and $\beta_i(k)$ terms a very difficult process. The problem was overcome by White by allowing only small coefficient updates at each iteration. The filter then remains approximately constant over a small number of iterations, and y(k) can be assumed to be the output from a fixed filter. An alternative approach would be to adapt the weights intermittently allowing time for the transients to decay sufficiently after each coefficient update. This would, however, slow the adaptation time considerably[26].

From equations (1.3),(1.4), the z-transform of the adaptive filter output, Y(z), is given by:

$$Y(z) = X(z) \frac{\hat{A}(z)}{\hat{B}(z)} \tag{2.8}$$

where:

$$\hat{A}(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_n z^{-n} \tag{2.9}$$

$$\hat{B}(z) = 1 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_m z^{-m} \tag{2.10}$$

By taking the partial derivatives of Y(z) with respect to the filter coefficients, equations (2.6) → (2.10) yield:

$$\alpha_i(z) = \frac{X(z).z^{-i}}{\hat{B}_k(z)} \qquad i = 0,1,..,n \tag{2.11}$$

$$\beta_i(z) = \frac{-Y(z).z^{-i}}{\hat{B}_k(z)} \qquad i = 1,2,..,m \tag{2.12}$$

Where $\hat{B}_k(z) \triangleq \hat{B}(z)$ at time t = kT. Inspection of equations (2.11) and (2.12) reveals that the $\alpha_i(z)$ and $\beta_i(z)$ terms can be generated by a series of time varying recursive filters with delayed versions of X(z) and Y(z) respectively as inputs. The filters are copies of the current $\hat{B}(z)$ adaptive filter section.

Hence, using equations (2.4) and (2.5), the general steepest descent equation (2.1) yields the following coefficient update equations:

$$a_i(k+1) = a_i(k) + \mu_a.e(k).\alpha_i(k) \qquad i = 0,1,..,n \tag{2.13}$$

$$b_i(k+1) = b_i(k) + \mu_b.e(k).\beta_i(k) \qquad i = 1,2,..,m \tag{2.14}$$

with the $\alpha_i(k)$ and $\beta_i(k)$ terms being generated by an additional set of time varying filters. The complete adaptive filter structure is shown in figure 2.1. It is seen that a considerable amount of computation is required for even relatively low order filters.

2.112 Hsia / Horvath Algorithm

Hsia[26] and Horvath[27] separately suggested the same simplification for the White/Stearns algorithm. The simplification drastically reduces the computation required to generate the $\alpha_i(k)$ and $\beta_i(k)$ gradient terms. By assuming that only small coefficient updates will be made at each iteration, $\hat{B}(z)$ at time $t = kT$ will approximate $\hat{B}(z)$ at times up to $t = (k-n)T$, where n is the order of the adaptive filter. The $\alpha_i(k)$ and $\beta_i(k)$ gradient terms can now be generated using:

$$\alpha_i(z) = \frac{X(z).z^{-i}}{\hat{B}_{k-i}(z)} \qquad i = 0,1,..,n \tag{2.15}$$

$$\beta_i(z) = \frac{-Y(z).z^{-i}}{\hat{B}_{k-i}(z)} \qquad i = 1,2,..,m \tag{2.16}$$

Where $\hat{B}_{k-i}(z)$ is the transfer function of the recursive section of the adaptive filter at time $t = (k-i)T$. The White filter structure is simplified to that shown in figure 2.2. The $\alpha_i(k)$ terms are already generated in the delay line of the adaptive filter, and hence do not

require any additional computation. Generation of the $\beta_i(k)$ terms requires a single copy of the present $\hat{B}(z)$ filter section. By comparing equations (2.11) and (2.12) with equations (2.15) and (2.16), it is seen that the Hsia/Horvath filter will have different transient behaviour to the White filter in the early stages of adaptation as the coefficients are rapidly varying, but in the later stages of adaptation $\hat{B}_{k-i}(z)$ will be a good approximation of $\hat{B}_k(z)$ as the coefficients approach their steady states. It is also observed that the transient differences will be greater for higher order filters. However, comparisons of the Hsia/Horvath and White/Stearns algorithms have shown that the difference in performance is negligible for low order filters.

2.113 Feintuch Algorithm

Feintuch proposed a further steepest descent algorithm[28] by approximating the adaptive filter input and output statistics. From equation (1.2), the output of the direct form adaptive filter structure at time t = kT is given by:

$$y(k) = \hat{A}^T X(k) - \hat{B}^T Y(k) \tag{2.17}$$

where:

$$\hat{A}^T = (a_0, a_1, a_2, \ldots, a_n) \tag{2.18}$$

$$\hat{B}^T = (b_1, b_2, b_3, \ldots, b_m) \tag{2.19}$$

$$X^T(k) = (x(k), x(k-1), \ldots, x(k-n)) \tag{2.20}$$

$$Y^T(k) = (y(k-1), y(k-2), \ldots, y(k-m)) \tag{2.21}$$

Hence, the error signal at time t = kT is given by:

$$e(k) = d(k) - (\hat{A}^T X(k) - \hat{B}^T Y(k)) \tag{2.22}$$

From equation (2.22), $E[e^2(k)]$ is obtained from:

$$E[e^2(k)] = E[d^2(k)] + \hat{A}^T R_{XX}\hat{A} + \hat{B}^T R_{YY}\hat{B} - 2\hat{A}^T R_{dX} + 2\hat{B}^T R_{dY} - 2\hat{A}^T R_{XY}\hat{B} \tag{2.23}$$

where:

$$\left.\begin{aligned} R_{XX} &= E[X(k)X^T(k)] \\ R_{YY} &= E[Y(k)Y^T(k)] \\ R_{dX} &= E[d(k)X(k)] \\ R_{dY} &= E[d(k)Y(k)] \\ R_{XY} &= E[X(k)Y^T(k)] \end{aligned}\right\} \tag{2.24}$$

Making the assumption that the Wiener solution for the adaptive filter is available [8], Feintuch observed that at the solution, the R_{XY}, R_{dY} and R_{YY} terms from equation (2.24) are constants when differentiated with respect to the vectors $\hat{A}$ and $\hat{B}$.

Hence, differentiation of equation (2.23) with respect to $\hat{A}$ and $\hat{B}$ yields:

$$\nabla_{\hat{A}}(E[e^2(k)]) = 2R_{XX}\hat{A} - 2R_{dX} - 2R_{XY}\hat{B} \tag{2.25}$$

$$\nabla_{\hat{B}}(E[e^2(k)]) = 2R_{YY}\hat{B} + 2R_{dY} - 2R^T_{XY}\hat{A} \tag{2.26}$$

If the statistics are known *a priori*, the optimum values of $\hat{A}$ and $\hat{B}$ can be calculated directly by equating equations (2.25) and (2.26) with zero (as the gradient vectors will be zero at the minimum). However, as in general the statistics are not available, Feintuch replaced the unknown matrices by local estimates and used a steepest descent procedure to update the coefficients. The gradient vectors can be approximated by:

$$\nabla_{\hat{A}}(E[e^2(k)]) = 2[X(k)X^T(k)\hat{A}(k) - d(k)X(k) - X(k)Y^T(k)\hat{B}(k)] \tag{2.27}$$

$$\nabla_{\hat{B}}(E[e^2(k)]) = 2[Y(k)Y^T(k)\hat{B}(k) + d(k)Y(k) - Y(k)X^T(k)\hat{A}(k)] \tag{2.28}$$

Using equation (2.22), equations (2.27) and (2.28) simplify to:

$$\nabla_A^{\wedge}(E[e^2(k)]) = -2X(k)e(k) \tag{2.29}$$

$$\nabla_B^{\wedge}(E[e^2(k)]) = 2Y(k)e(k) \tag{2.30}$$

A comparison of equations (2.29) and (2.30) with the gradient term equations (2.4) and (2.5) for the White/ Stearns algorithm reveals that the $\alpha_i(k)$ and $\beta_i(k)$ terms for the Feintuch algorithm are given by:

$$\alpha_i(k) = x(k-i) \qquad i = 0,1,..,n \tag{2.31}$$

$$\beta_i(k) = -y(k-i) \qquad i = 1,2,..,m \tag{2.32}$$

The filter structure of the Feintuch algorithm is shown in Figure 2.3. As for the previous two algorithms, the filter coefficients are updated every sample using equations (2.13) and (2.14).

Following the publication of the Feintuch algorithm in[28], two publications questioning the validity of the algorithm appeared in the literature[29,30]. Feintuch assumed in the derivation of the algorithm that the covariance terms R_{XY}, R_{YY} and R_{dY} are constants when differentiating with respect to the coefficients. However, this is not the case if the coefficients are adaptive. Hence, the Feintuch algorithm is effectively a gross approximation of the White algorithm. The White/Stearns $\alpha_i(k)$ and $\beta_i(k)$ terms are truncated to the Feintuch terms by ignoring the recursive calculators.

In[29] and[30], it was stated that, in general, the Feintuch algorithm will not converge to minimize the mean squared error. In[29] an example was given in which a two pole, two zero plant was modelled using a single pole, single zero adaptive filter. The resulting

error surface was shown to be bimodal. The Feintuch algorithm converged to neither the global minimum, nor the local minimum in the m.s.e. surface. This result confirms that the algorithm is not strictly a steepest descent algorithm.

However, simulations by Feintuch suggest that if the order of the adaptive filter is of the same or higher order than the plant, then the algorithm will converge to the optimum solution.

2.114 Sequential Regression Algorithm

An Adaptive IIR Filtering algorithm employing a sequential regression technique was proposed by Parikh and Ahmed in[31] and further developed in[32 → 35]. The coefficient update algorithm uses an approximation to Newtons method which has the advantage of straight line convergence toward the minimum of a quadratic performance surface, rather than the more circuitous steepest descent approach. Although the IIR error surface is non-quadratic as a whole, a similar technique can be used by making small coefficient adjustments and assuming the performance surface to be locally quadratic. The algorithm involves the generation of the inverse of the Hessian matrix of second order derivatives. While this is achieved using a matrix inversion lemma, the computation required is still considerable.

The Sequential Regression (SER) algorithm is derived to minimize the cost function $J(\underline{W})$ where:

$$J(\underline{W}) = q \sum_{k} e^2(k) + \underline{W}^T\underline{W} \qquad (2.33)$$

and $\underline{W}$ is the vector of filter coefficients. The parameter q is a scalar which determines the relative influence of the squared error, and the weights on the cost function. The $\underline{W}^T\underline{W}$ term in the cost function is required in cases when the error surface has a distributed minimum. That

is, the adaptive filter is over-sufficient. In such cases, the algorithm can run into computational difficulties if the $\underline{W}^T\underline{W}$ term is excluded due to the recursive estimate of a matrix inverse[35].

The error signal, e(k), is obtained from:

$$e(k) = d(k) - \underline{W}_k^T \underline{V}_k \tag{2.34}$$

where:

$$\underline{V}_k^T = (x(k),x(k-1),..x(k-n),-y(k-1),-y(k-2),..,-y(k-m)) \tag{2.35}$$

Letting $\nabla_{\underline{W}}J(\underline{W})$ denote the gradient of $J(\underline{W})$ with respect to $\underline{W}$, then conditions for a minimum in $J(\underline{W})$ are obtained by equating $\nabla_{\underline{W}}J(\underline{W})$ with zero which yields:

$$(I + \sum_k q\ \underline{\lambda}_k \underline{V}_k^T)\underline{W} = \sum_k d(k)\ \underline{\lambda}_k \tag{2.36}$$

where I is the identity matrix and $\underline{\lambda}_k = \nabla_{\underline{W}}y(k)$. Hence:

$$\underline{\lambda}_k^T = (\alpha_0(k),\alpha_1(k),\ldots,\alpha_n(k),\beta_1(k),\beta_2(k),\ldots,\beta_m(k)) \tag{2.37}$$

Where the $\alpha_i(k)$ and $\beta_i(k)$ terms are generated using equations (2.11) and (2.12) derived for the White algorithm. By estimating $\underline{W}$ using r iterations, equation (2.36) yields:

$$\underline{W}_r = P_r^{-1} \sum_{k=1}^{r} d(k)\ \underline{\lambda}_k \tag{2.38}$$

where:

$$P_r = I + \sum_{k=1}^{r} q\ \underline{\lambda}_k \underline{V}_k^T \tag{2.39}$$

$\underline{W}_r$ is computed recursively using:

$$\underline{W}_k = \underline{W}_{k-1} + q\, P_k^{-1}\, \underline{\lambda}_k\, e(k) \qquad k=1,2,\ldots,r \tag{2.40}$$

where $\underline{W}_0 = 0$ and P_k^{-1} is calculated via a matrix inversion lemma using:

$$P_k^{-1} = P_{k-1}^{-1} - \frac{1}{\gamma}\, P_{k-1}^{-1}\, \underline{\lambda}_k\, \underline{V}_k^T\, P_{k-1}^{-1} \tag{2.41}$$

where:

$$\gamma = \frac{1}{q} + \underline{V}_k^T\, P_{k-1}^{-1}\, \underline{\lambda}_k \tag{2.42}$$

and $P_0^{-1} = I$.

Hence, the adaptive filter coefficients are updated every iteration using equation (2.40) with P_k^{-1} calculated using equation (2.41) and $\underline{\lambda}_k$ generated using the White/Stearns filter structure. The matrix P_k is an approximation to the Hessian matrix at the kth iteration.

Simulations by Parikh and Ahmed[31-35] have demonstrated several applications for the SER algorithm. One interesting property of the algorithm was shown in[33]. The insufficient adaptive filter example from[29] was tested using the SER algorithm and it was shown that the adaptive filter converged to the global minimum of the error surface, and not the local minimum, irrespective of which minimum the filter was started closer to. However, in general, convergence to the global minimum cannot be guaranteed[36].

2.12 Hyperstable Algorithms

The concept of hyperstability was developed by V.M. Popov in[69]. Hyperstability extends the linear systems theory of stability and positivity, to non-linear and time varying systems. A discrete-time, recursive output error technique based on hyperstability was proposed by Landau[16] for identifying the parameters of an ARMA plant. Known as the Hyperstable Error Identifier, present and past values of the output error signal were used to satisfy conditions for hyperstability, and the error asymptotically decayed to zero. However, the method involved complicated stochastic approximations and employed a decaying step-size which was satisfactory for stationary input sequences only.

The technique was adapted by Larimore, Treichler and Johnson in[37 → 41] for use in adaptive filtering applications. The first algorithm, called HARF (Hyperstable Adaptive Recursive Filter), employed a fixed step-size and recursions from the Landau algorithm were sequenced to give a strictly causal filter. HARF was the first adaptive IIR filter algorithm to have provable convergence properties. Due to its complexity however, it was too computationally demanding for many signal processing tasks. Hence, the SHARF (Simplified HARF) algorithm was proposed[37]. The SHARF algorithm requires the same order of computation as the gradient-based algorithms. A further modification to the HARF algorithm resulted in MHARF (Modified HARF)[42]. Johnson[44] showed that all hyperstable algorithms can be derived from a central form called CHARF (Complete HARF).The algorithms are discussed in greater detail in the following sections.

2.121 HARF Algorithm

The difference equations required for the implementation of the HARF algorithm are given in equations (2.43) and (2.44). In addition to the main adaptive filter, an extra ARMA process is required for the

formation of the filter output, y(k). The auxiliary process generates:

$$f(k-1) = \sum_{i=0}^{n} a_i(k)x(k-i-1) - \sum_{i=1}^{m} b_i(k)f(k-i-1) \qquad (2.43)$$

With the filter output given by:

$$y(k) = \sum_{i=0}^{n} a_i(k)x(k-i) - \sum_{i=1}^{m} b_i(k)f(k-i) \qquad (2.44)$$

If the adaptive process converges on a solution, $a_i(k+1) = a_i(k)$, $b_i(k+1) = b_i(k)$, and y(k) asymptotically converges to f(k). The coefficients are updated every sample using:

$$a_i(k+1) = a_i(k) + \frac{\mu_a}{q(k)} .x(k-i).v(k) \qquad (2.45)$$

$$i=0,1,\ldots,n$$

$$b_i(k+1) = b_i(k) - \frac{\mu_b}{q(k)} .f(k-i).v(k) \qquad (2.46)$$

$$i=1,2,\ldots,m$$

where v(k) is the augmented error given by:

$$v(k) = (d(k)-f(k)) + \sum_{i=1}^{p} c_i(d(k-i)-f(k-i)) \qquad (2.47)$$

and q(k) is a normalizing factor obtained from:

$$q(k) = 1 + \sum_{i=1}^{m} \mu_b f^2(k-i) + \sum_{i=0}^{n} \mu_a x^2(k-i) \qquad (2.48)$$

For the adaptive filter to converge to its optimum coefficients, two conditions must be met.

a) The order of the adaptive filter transfer function numerator and denominator must be at least as great as that of the plant. ie. $\hat{m} > m$, $\hat{n} > n$.

b) The $\{c_i\}$ error smoothing coefficients must be chosen so that the transfer function G(z) given by:

$$G(z) = \frac{1 + \sum_{i=1}^{p} c_i z^{-i}}{1 + \sum_{i=1}^{m} b_i z^{-i}} \qquad (2.49)$$

is Strictly Positive Real (S.P.R.), ie. $\mathrm{Re}\{G(z)\} > 0$ for $|z| = 1$. Under these conditions, the modified error v(k) given in equation (2.47) converges to zero and therefore $y(k) \rightarrow f(k) \rightarrow d(k)$ which is the desired performance for the filter.

A complete explanation of the algorithm with a heuristic description of positive reality and hyperstability is provided in[37,39]. A proof of the algorithm's convergence properties is given in[38,41]. However, a considerable amount of computation is required for implementation of HARF, which is undesirable for real time applications. To make it more amenable, certain simplifications were made leading to SHARF.

2.122 SHARF Algorithm

Simplifications for the HARF algorithm were suggested by Larimore, Treichler and Johnson in[37] resulting in SHARF. By specifying the adaptation constants μ_a and μ_b to be sufficiently small, the coefficients in equations (2.43) and (2.44) change very little from iteration to iteration.

A comparison of equations (2.43) and (2.44) indicates that $f(k) \simeq y(k)$, and hence the filter output in equation (2.44) can be approximated by:

$$y(k) = \sum_{i=0}^{n} a_i(k)x(k-i) - \sum_{i=1}^{m} b_i(k)y(k-i) \qquad (2.50)$$

The augmented error v(k) in equation (2.48) becomes:

$$v(k) = (d(k)-y(k)) + \sum_{i=1}^{p} c_i(d(k-i)-y(k-i))$$

$$= e(k) + \sum_{i=1}^{p} c_i.e(k-i) \qquad (2.51)$$

v(k) is now a moving average of the output error e(k).

The normalizing factor q(k) was utilized in HARF to reduce the effective step size for large values of filter input and output. Assuming that μ_a and μ_b are small values, equation (2.48) can be simplified to:

$$q(k) = 1 \qquad (2.52)$$

Hence, from equations (2.45) and (2.46), the coefficient update equations for the SHARF algorithm are given by:

$$a_i(k+1) = a_i(k) + \mu_a.x(k-i).v(k) \qquad i=0,1,..,n \qquad (2.53)$$

$$b_i(k+1) = b_i(k) - \mu_b.y(k-i).v(k) \qquad i=1,2,..,m \qquad (2.54)$$

The complete filter structure for the SHARF algorithm is shown in figure 2.4. A comparison with the Feintuch algorithm of Section 2.113 reveals that if $(c_i) = 0$, $v(k) = e(k)$ and the two algorithms are

identical. However, in general, the hyperstable algorithms do not behave as gradient algorithms (In[38] it was shown that the Hessian matrix for HARF is irregular and may have complex eigenvalues). For small values of μ_a and μ_b, the SHARF (and hence the Feintuch algorithm) adaptation process will converge provided that G(z) in equation (2.49) is S.P.R.. For $\{c_i\} = 0$, this criterion means that the poles of the adaptive filter can only lie within a restricted region of the unit circle[40]. This region is shown in figure 2.5, together with S.P.R. regions for different values of c_1. It is seen that with suitable choice of $\{c_i\}$ coefficients, the S.P.R. region can be tailored to cover different regions of the unit circle. However, to guarantee convergence for the adaptive filter, the choice of $\{c_i\}$ coefficients requires *a priori* knowledge of the plant pole positions. In addition, it was shown[37,40] that rates of convergence are highly dependent on the choice of $\{c_i\}$ coefficients.

2.123 Modified HARF

To eliminate the need for an *a-priori* choice of $\{c_i\}$ coefficients, HARF was modified by Parikh, Sinha and Ahmed in[42] and[43] to provide an adaptive S.P.R. region. This was achieved by adaptation of the $\{c_i\}$ coefficients in a similar way to the $\{a_i\}$ and $\{b_i\}$ coefficients. The $\{a_i\}$, $\{b_i\}$ and $\{c_i\}$ coeffficients are updated using:

$$a_i(k+1) = a_i(k) + \frac{\mu_a}{q'(k)} v(k)x(k-i) \qquad i=0,n \tag{2.55}$$

$$b_i(k+1) = b_i(k) - \frac{\mu_b}{q'(k)} v(k)y(k-i) \qquad i=1,m \tag{2.56}$$

$$c_i(k+1) = c_i(k) + \frac{\mu_c}{q'(k)} v(k)e(k-i) \qquad i=1,p \tag{2.57}$$

where:

$$q'(k) = q(k) + \sum_{i=1}^{p} \mu_c e^2(k-i) \qquad (2.58)$$

q(k) is calculated using equation (2.48) and v(k) is obtained from equation (2.47). The MHARF algorithm has the advantage over the HARF algorithm of not requiring a pre-determined S.P.R. region. However, convergence of the MHARF algorithm cannot be guaranteed due to the adaptive (c_i) coefficients masking possible instability of the adaptive filter[36].

2.124 CHARF (Complete HARF)

This is the central algorithmic form of the hyperstable algorithms. Johnson[44] demonstrated that the HARF,SHARF and MHARF algorithms can be generated from CHARF.

2.13 Random Search Algorithms

In general, gradient techniques cannot guarantee convergence to the global minimum due to the non-unimodal nature of the IIR filter mean-square-error surface. An adaptive Genetic Algorithm was proposed by Etter in[45] and modified in[46] to circumvent the problem through the use of a random search approach with a "survival of the fittest" selection process that retains information from past trials.

2.131 Genetic Algorithm

Each filter in the search space (ie., all possible stable filters allowed under the accuracy constraints) have their coefficients coded into binary numbers using the Gray code, and combined into a binary string. Initially, a set of filters are chosen at random from the

search space. This is termed the first generation. Each filter has a mean-square-error value associated with it.

The binary strings are acted upon by one of two operations to form the next generation. The operations are mutation and crossover. Mutation acts on a filter string by complementing bits in the string with a predetermined probability depending on the filter's associated m.s.e.. Crossover takes two strings and breaks them at random point. The first part of one string is then combined with the second part of the other to produce two new strings. The probability of selection of a particular bit string to modify and/or replicate is inversely proportional to its estimated m.s.e. in relation to the m.s.e. of the other bit strings.

The algorithm was designed to slowly change from a random search to a converging process. Hence, in the initial stages of adaptation there is a higher probability that the random operator mutation will take place rather than the converging crossover. The probabilities are controlled by a decaying exponential governed by a constant, viz:

$$p(\text{mutation}) = 0.8333e^{-\alpha t} \tag{2.59}$$

$$p(\text{crossover}) = 1 - 0.8333e^{-\alpha t} \tag{2.60}$$

The search is terminated when either:

1) The outcome of all crossover operations are identical.

2) The minimum and maximum m.s.e. of a generation are the same.

3) A filter with zero error is found.

The evolution process enables the characteristics of above average filters to reproduce their patterns and hence information is retained through generations. The target during the random phase is to

generate optimal substrings. These subpatterns are termed Optimal Independent Schema and are bits within a string that are optimal, irrespective of the settings of other bits. During the convergence stage these Independent Schemas reproduce their bit patterns and eventually dominate the generation.

Simulations in[45,46] show that convergence to a global minimum in a bimodal error surface example was achieved with a high probability of success. The actual rates of convergence and minima obtained were dependent on the choice of α. If the algorithm changed quickly from the random search to the convergence mode, then it quickly converged but not always to the global minimum. Alternatively, with a slower changeover, there is an increased probability of convergence to the global minimum, but also an increase in convergence times.

However, the simulations by Etter used only second order All-Pole filters. The search space for the adaptive filter is therefore the triangular region shown in figure 1.6. For higher order filters the search space size would rapidly increase in size, and with 10 filters for each generation (as used in[46]), the computation required is large.

2.2 ALGORITHMS FOR OTHER FILTER STRUCTURES

The Direct Form filter structure featured in the previous algorithms is optimal in terms of hardware implementation, but suffers from two major drawbacks when the filter coefficients are adapted. The first is the absence of a simple stability check, which applies to even low order filters. For the Gradient Algorithms, noise introduced by the coefficient adaptation, or short term trends in the error signal, can cause poles to migrate outside the unit circle resulting in an unstable filter. Hence, to maintain stability of the Direct Form Structure during adaptation using gradient methods, it is necessary to either factorize

the denominator of the filter transfer function and check the pole positions, or alternatively to use some other equally complicated stability criterion.

Both of these approaches are computationally intensive and hence undesirable, as the test must be repeated following each coefficient update. Thus, stability of the Direct Form gradient algorithms is not generally guaranteed for simple real time operation, although the problem of instability is reduced to some extent by making very small coefficient changes at each update.

The second drawback is caused by the high sensitivity of the Direct Form filter coefficients to pole and zero positions near either $z = 1$ or $z = -1$ in the z-plane[64]. As a change in one denominator coefficient will change all the pole positions (and similarly a change in a numerator coefficient will alter all the zero positions), the filter characteristics can be very sensitive to small coefficient changes.

To reduce the problem of maintaining stability of a Direct Form filter structure, adaptation algorithms have been proposed for other filter structures which may be kept stable easily. Structures which have been utilized are the recursive lattice structure[47,48,49], a cascade of second order biquadratic filter sections[6], and a modified cascade structure[50].

2.21 Recursive Lattice

Several adaptive IIR lattice structures have appeared in the literature. The first was proposed by Parikh, Ahmed and Stearns[47]. This algorithm was simplified by Ayala in[48]. Recently, a SHARF algorithm using a lattice structure was published[49].

Figure 2.6 shows the IIR Lattice filter structure. The filter consists of two sections, a recursive lattice structure with

coefficients (k_i) which implement the poles, and a linear combiner with coefficients (v_i) which implement the zeros. There is a direct correspondence between the lattice filter coefficients and the direct form filter coefficients given in equation (1.3).

The filter remains stable provided the (k_i) recursive coefficients satisfy:

$$|k_i| < 1 \qquad 1 < i < N \tag{2.61}$$

The input and output of the filter at time t = nT is given by:

$$y(n) = \sum_{i=1}^{N} v_i(n).g_i(n) \tag{2.62}$$

where:

$$g_i(n) = g_{i-1}(n-1) + k_i(n)f_{i-1}(n) \qquad i=1,2,..,N \tag{2.63}$$

$$f_{i-1}(n) = f_i(n) - k_i(n)g_{i-1}(n-1) \qquad i=1,2,..,N \tag{2.64}$$

$$f_N(n) = x(n) \tag{2.65}$$

$$g_0(n) = f_0(n) \tag{2.66}$$

In the following two sections, the Parikh, Ahmed, Stearns and Ayala gradient algorithms for adapting the recursive lattice structure are studied.

2.211 Parikh,Ahmed and Stearns Algorithm

The algorithm proposed by Parikh,Ahmed and Stearns uses the method of steepest decent to minimize the m.s.e.. The filter coefficients are adapted using:

$$v_i(n+1) = v_i(n) + \frac{\mu}{\sigma_i^2(n)} e(n)\phi(n) \qquad i = 1,N \tag{2.67}$$

$$k_i(n+1) = k_i(n) + \frac{\mu}{\nu_i^2(n)} e(n)\psi(n) \qquad i = 1,N \tag{2.68}$$

Where:

$$\phi_i(n) = \frac{\partial y(n)}{\partial v_i(n)} \tag{2.69}$$

$$\psi_i(n) = \frac{\partial y(n)}{\partial k_i(n)} \tag{2.70}$$

μ is an adaptation constant and $\sigma_i^2(n)$ and $\nu_i^2(n)$ are the estimates of the power at the ith stage of the lattice. $\sigma_i^2(n)$ and $\nu_i^2(n)$ are calculated using:

$$\sigma_i^2(n) = \rho\sigma_i^2(n-1) + (1-\rho)g_i^2(n) \tag{2.71}$$

$$\nu_i^2(n) = \rho\nu_i^2(n-1) + (1-\rho)[g_{i-1}^2(n-1) + f_i^2(n)] \tag{2.72}$$

$0<\rho<1$ determines the bandwidth of the low pass filter operation.

The $\phi_i(n)$ terms are calculated using:

$$\phi_i(n) = g_i(n) \tag{2.73}$$

ie. they are generated within the filter itself.

The calculation of the $\psi_i(n)$ terms involves a complicated series of recursive lattice filters, (similar to the White Direct Form Structure algorithm which requires copies of the recursive section of the filter to generate the $\alpha_i(k)$ and $\beta_i(k)$ terms).

To calculate the $\psi_i(n)$ terms:

$$\psi_i(n) = \sum_{j=0}^{N} v_j(n)\beta_{i,j}(n) \qquad i=0,1,..,N-1 \tag{2.74}$$

where the $\beta_{i,j}(n)$ terms are generated using equations (2.75) and (2.76):

$$\alpha_{i,j}(n) = \frac{\partial f_j(n)}{\partial k_i(n)} \tag{2.75}$$

$$= \begin{cases} \beta_{i,j}(n) & j=0 \\ \alpha_{i,j+1}(n) - k_{j+1}(n)\beta_{i,j}(n-1) & j \neq i+1 \\ \alpha_{i,j+1}(n) - k_{j+1}(n)\beta_{i,j}(n-1) - g_j(n-1) & j=i+1 \end{cases}$$

And:

$$\beta_{i,j}(n) = \frac{\partial g_j(n)}{\partial k_i(n)} \tag{2.76}$$

$$= \begin{cases} \beta_{i,j-1}(n-1) + k_j(n)\alpha_{i,j-1}(n) & i \neq j \\ \beta_{i,j-1}(n-1) + k_j(n)\alpha_{i,j-1}(n) + f_{j-1}(n) & i = j \end{cases}$$

The calculation of the $\psi_i(n)$ terms requires a great deal of computation and hence the algorithm is unsuitable for hardware implementation.

2.212 Ayala Algorithm

The Ayala algorithm differs from the Parikh,Ahmed,Stearns algorithm in the calculation of the $\psi_i(n)$ terms. Ayala simplified the calculation by expressing equation (2.62) in terms of the reflective coefficients. The vector $\underline{g}(n)$ of forward residuals $g_i(n)$ can be expressed as:

$$\underline{g}(n) = A(n)\underline{g}(n-1) + f_N(n)\underline{k}(n) \tag{2.77}$$

Where $\underline{k}(n)$ is the vector of $k_i(n)$ coefficients and:

$$A(n) = \begin{bmatrix} -k_0k_1 & -k_0k_2 \cdots\cdots & -k_0k_N & 0 \\ 1-k_1^2 & -k_1k_2 \cdots\cdots & -k_1k_N & 0 \\ 0 & 1-k_2^2 \; -k_2k_3 \cdots & -k_2k_N & 0 \\ \vdots & 0 & \cdot & \cdot \\ 0 & \cdots\vdots\cdots & 1-k_N^2 & 0 \end{bmatrix} \tag{2.78}$$

By neglecting Second Order effects, equations (2.65),(2.69),(2.70) and (2.77) yield:

$$\underline{\psi}(n) = \underline{v}(n)^T(\nabla_k A(n))\underline{g}(n-1) + f_N(n)\underline{v}(n) \tag{2.79}$$

Where $\underline{v}(n)$ is the vector of $v_i(n)$ coefficients.

From equation (2.79), the expressions for successive gradient terms are given by:

$$\left.\begin{aligned} \psi_1(n) &= v_i(n)(2f_0(n) - f_1(n)) - g_0(n-1)\Delta_0(n) \\ &\vdots \\ \psi_N(n) &= v_N(n)(2f_{N-1}(n) - f_N(n)) - g_{N-1}(n-1)\Delta_{N-1}(n) \end{aligned}\right\} \quad (2.80)$$

Where:

$$\begin{aligned} \Delta_0(n) &= v_0(n)k_0(n) \\ \Delta_1(n) &= v_1(n)k_1(n) + \Delta_0(n) \\ &\vdots \\ \Delta_N(n) &= v_N(n)k_N(n) + \Delta_{N-1}(n) \end{aligned}$$

For the Ayala algorithm, the process for calculating the $\psi_i(n)$ terms in equation (2.68) requires only the current residuals and lattice coefficients at the ith stage, and a recursive update term in equation (2.80). This is considerably simpler than the formulation proposed in[47].

2.22 Cascading Second-Order Sections

An alternative method for ensuring filter stability is to realize the recursive part of the filter as a cascade of second order sections. Maintaining stability is achieved by constraining the two coefficients from each recursive section (b_1, b_2) within the triangular region shown in figure 1.6.

An IIR LMS algorithm for adapting a cascade of biquadratic sections was proposed by David in[6]. The algorithm was modified in[50] for a filter structure consisting of a Direct Form All-Zero Section followed by a cascade of second order All-Pole sections. These algorithms and filter structures are considered in detail in Chapter 3.

2.23 Total Adaptive Filter

In an attempt to overcome the problem of local minima in the IIR filter performance surface, Soderstrand[52] proposed the Total Adaptive Filter (TAF). The TAF utilises the fact that implementation of an adaptive filter using a microprocessor enables the filter structure to be changed, as well as the coefficients. For a number of filter structures, the coefficients are adapted and the minimum m-s-e value is recorded. The filter producing the lowest m-s-e value is then implemented. Generally, convergence times from this type of approach will be slow.

2.3 OVERVIEW OF IIR FILTER ADAPTATION ALGORITHMS

Analysis of the adaptive IIR filtering algorithms proposed to date, reveals a lack of robustness which explains the predominance of adaptive FIR filters. Four major problems face prospective Adaptive IIR FIlter users,viz:

1) Will the filter adapt to, and converge on the global minimum ?
2) Will the filter remain stable ?
3) How much computation will the filter structure and adaptation algorithm require ?
4) Is the speed of adaptation fast enough ?

The above points are discussed in the Sections (2.31 → 2.34) based on the available mathematical proofs and experimental results. Conclusions are drawn in Section 2.35.

2.31 Adaptation to the Global Minimum

It was stated in[18] that the HARF algorithm, and hence SHARF employing a small adaptation constant, is guaranteed convergent to the

global minimum provided that the G(z) transfer function is S.P.R., and the order of the adaptive filter is at least as great as that of the target plant. However, in many signal processing applications these two conditions are difficult to ensure. Choice of the $\{c_i\}$ coefficients which specify the S.P.R. region (equation (2.49)) is governed by some a-priori knowledge of where the target poles lie. This information may not be available and a poor choice of $\{c_i\}$ coefficients may result in slow convergence or instability. The MHARF algorithm avoids the problem by adaptation of the $\{c_i\}$ coefficients. However, this algorithm has stability problems which will be considered in the next section. Furthermore, it is known that hyperstability is an over-restrictive criterion. For example, the Feintuch algorithm has been shown to converge to plant poles lying outside the S.P.R. region[40].

A second difficulty with Hyperstable adaptive filter algorithms arises if the adaptive filter is of insufficient size to model the target. It was shown in[44] that, in general, the adaptive filter will not converge to either a local or the global minimum of the m.s.e. surface. In fact, both HARF and SHARF adapt to the same position as would Feintuch LMS. This was confirmed using the bimodal error surface example presented in[29]. For many signal processing tasks, the order of the adaptive filter will be less than the target, and in such cases the hyperstable algorithms will not necessarily adapt to a minimum in the m.s.e. surface.

Although gradient algorithms have been proven convergent only under very limited conditions (ie., White noise input, sufficient order adaptive filter) experimentation has suggested that they will converge under less favourable input conditions[12]. Apart from the Feintuch LMS algorithm, the gradient algorithms have the advantage over hyperstable algorithms of adaptation to a minimum in the m.s.e. surface.

While the Genetic algorithm is too complex for real-time operation, it will find the global minimum of the m.s.e. surface with a high percentage of success, and otherwise will adapt to a local minimum.

2.32 Adaptive Filter Stability

Maintaining filter stability during adaptation is a great problem for all adaptive IIR filter algorithms using a Direct Form filter structure. For real-time applications, gradient-based algorithms are likely to go unstable as a complicated stability check cannot be implemented. The HARF and SHARF algorithms are guaranteed stable provided that the G(z) transfer function is S.P.R.. However, a poor choice of $\{c_i\}$ coefficients may result in instability. In addition, the choice of μ parameters is still empirical. MHARF runs into stability problems due to the C(z) error filter. Transfer function poles may adapt outside the unit circle whilst being masked by the zeros of C(z). This can lead to a stall condition in the adaptation[36,44]. To avoid the problem, a stability check on the adaptive filter pole positions must be incorporated into the algorithm. As previously mentioned, this is not practical for the Direct Form filter structure. The lattice and cascade filter structures, on the other hand, are easily kept stable.

2.33 Computation Costs

A major reason for the development of adaptive IIR filters is the computation savings offered when compared with a very long adaptive FIR filter. To achieve this aim, the IIR algorithm needs to be as simple as possible - especially for dedicated hardware applications. Reviewing the range of algorithms, the SER, HARF, MHARF and the Genetic algorithms have a high degree of complexity. Algorithms such as the White algorithm and the Recursive Lattice algorithms are moderately complex and may be too complicated for real-time implementation. Soderstrand[36] reviewed

the cost and performance of a number of algorithms including the White and Feintuch algorithms. Using several examples, it was demonstrated that the FIR ladder filter using the LMS algorithm can give a better performance than many IIR filtering algorithms for a fixed computation cost.

2.34 Speed of Adaptation

Whilst being of great importance in many practical applications such as echo cancelling, line equalization, etc., the problem of speed of adaptation tends to be ignored by the great majority of publications concerned with adaptive IIR filtering. Although a low order adaptive IIR filter may achieve the necessary reduction in m.s.e. to replace a high order adaptive FIR filter, the time taken for each to adapt to the optimum filter may be a significant factor.

The speed of adaptation of an FIR filter using the Widrow LMS algorithm is governed by the length of the filter and the eigenvalue spread of the input signal[8]. However, the error surface is quadratic in shape and adaptation is generally fast. In the IIR case, the function to be minimized is non-uniform[12] and adaptation times will therefore be slower in general. For the computationally simple algorithms, the step size μ must be kept small to retain stability of the gradient algorithms, and to keep the hyperstable feature of SHARF. This unfortunately leads to very slow convergence times, generally much slower than FIR filters.

2.35 Conclusions

The results presented in Sections 2.31 → 2.34 suggest that the hyperstable algorithms are only suitable for system identification problems where the adaptive filter is of sufficient order to fully model the unknown system. Hence, the hyperstable algorithms are not suited to

the majority of signal processing applications, which fall into the signal identification and signal correction categories. For these applications, the adaptive filter is generally insufficient in order and the filter performance surface may be multimodal.

The problem of local minima affects all algorithms except those taking a random search approach. However, the Etter algorithms are too complicated to implement for real-time applications. While being simple to implement, the gradient algorithms will adapt to a minimum in the error surface provided that the values of the μ parameters are correctly chosen. Hence, gradient algorithms can always reduce the m.s.e. value to a minimum in the error surface, which is not the case for the hyperstable algorithms. In addition, for some applications, the error reduction obtained from a local minimum position in the error surface may be sufficient.

One of the major problems of gradient algorithms is maintaining filter stability during adaptation. This problem can be circumvented by implementing the poles using either a lattice filter, or a cascade of second-order recursive sections. By incorporating a filter stability check into a gradient algorithm, the size of μ adaptation parameters may be greatly increased in size compared with the values needed to retain filter stability. This will result in much faster convergence.

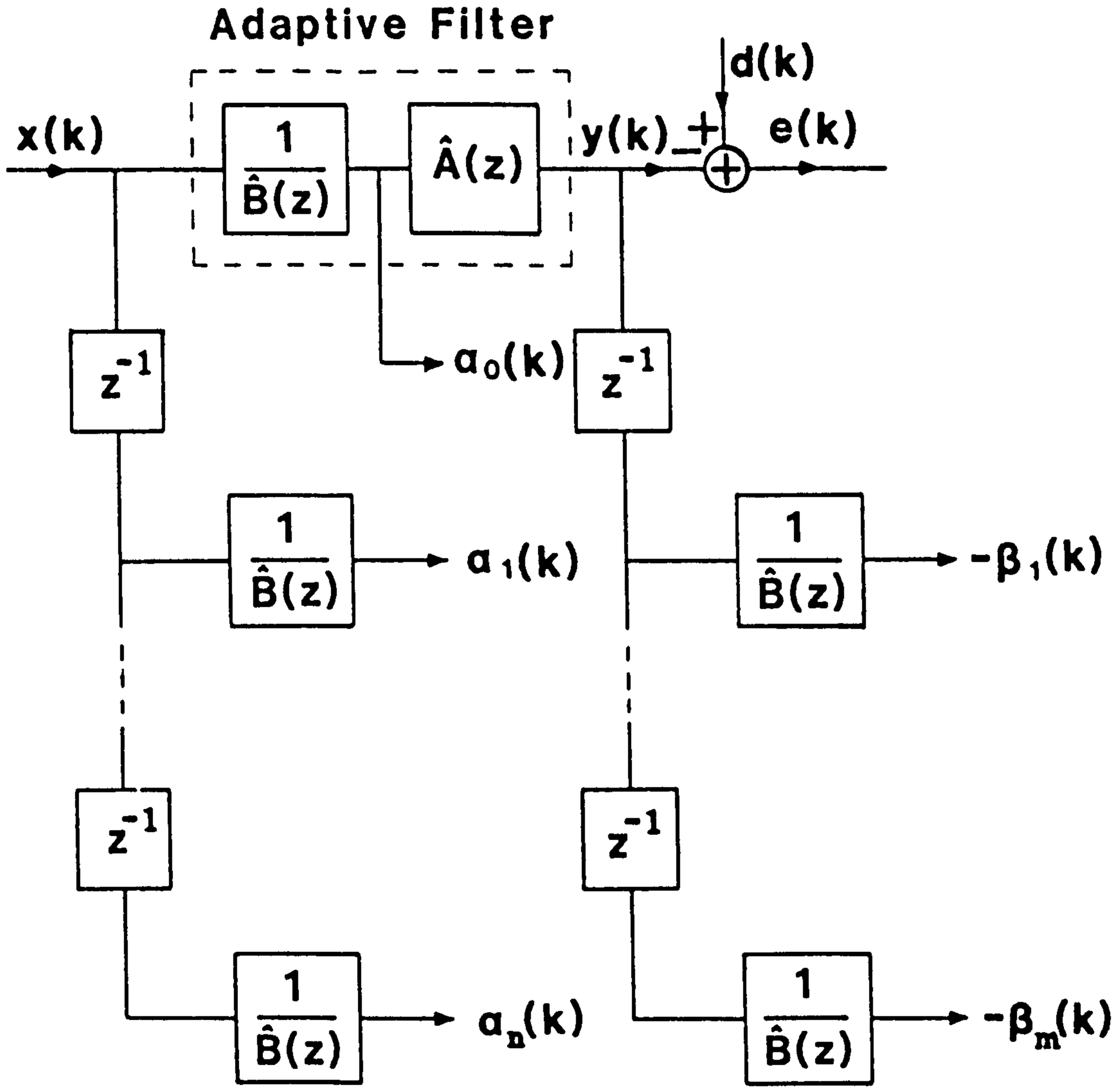
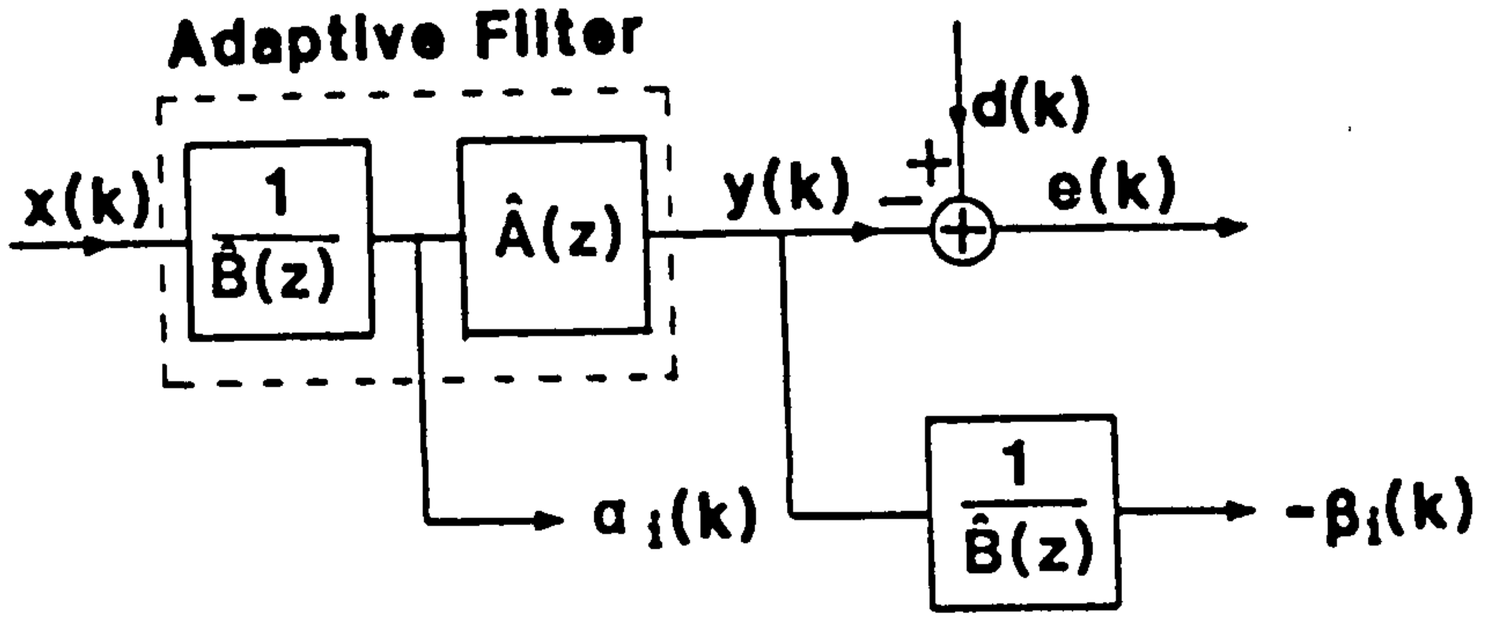
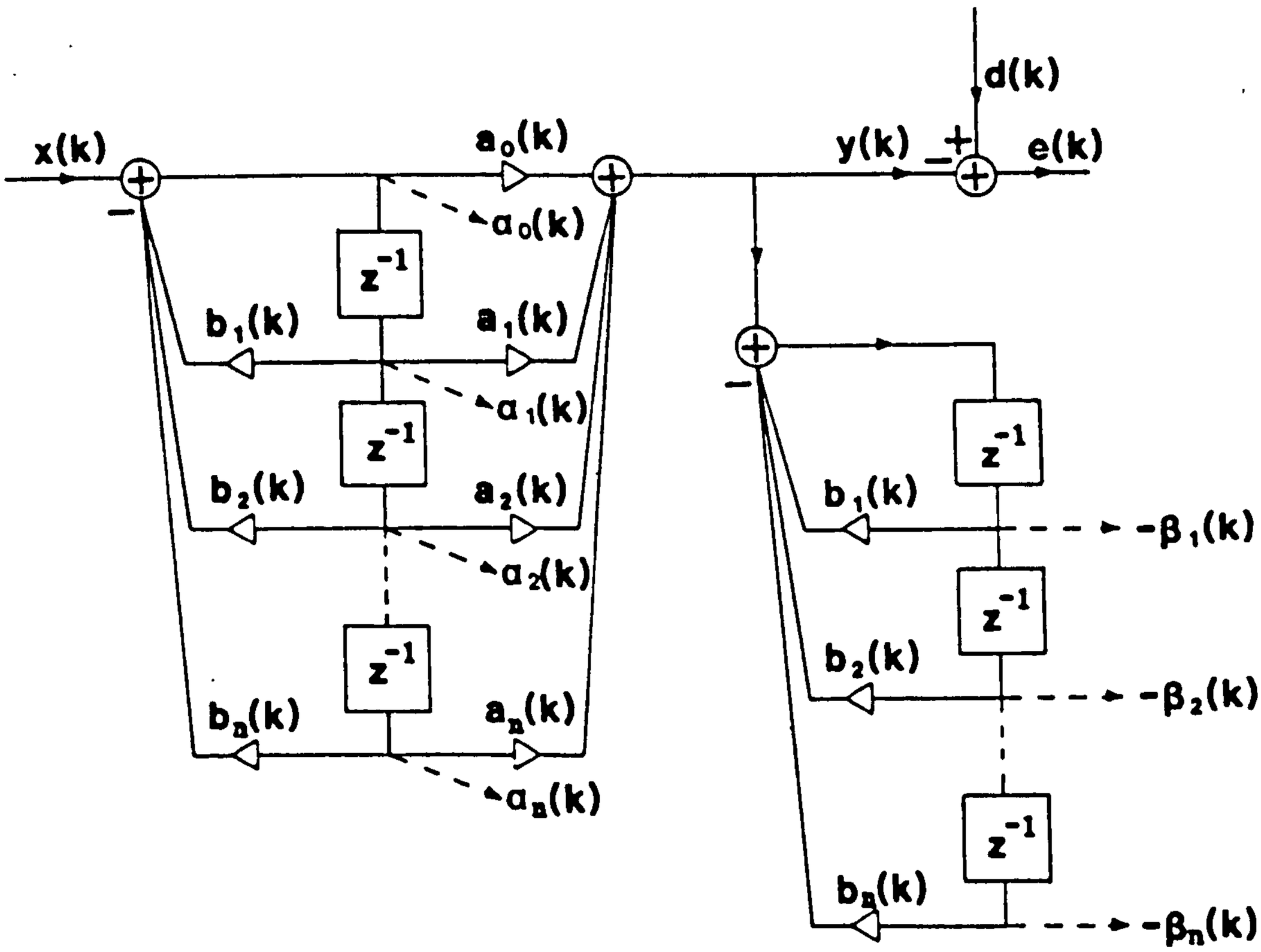


Figure 2.1 Filter Structure for the White/Stearns Algorithm.



Block Diagram.



Filter Implementation.

Figure 2.2 Filter Structure for the Hsia/Horvath Algorithm.

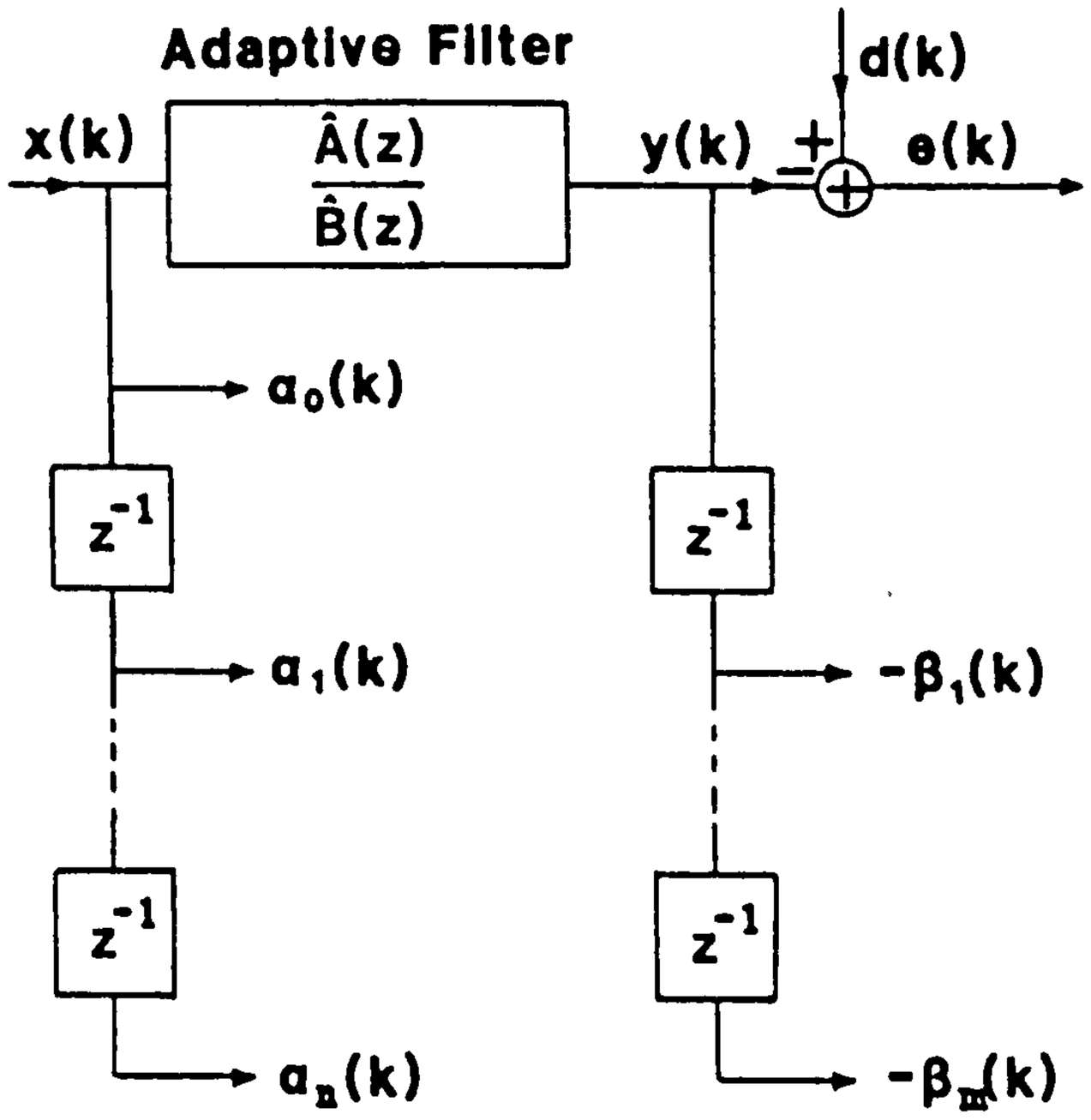


Figure 2.3 Filter Structure for the Feintuch Algorithm.

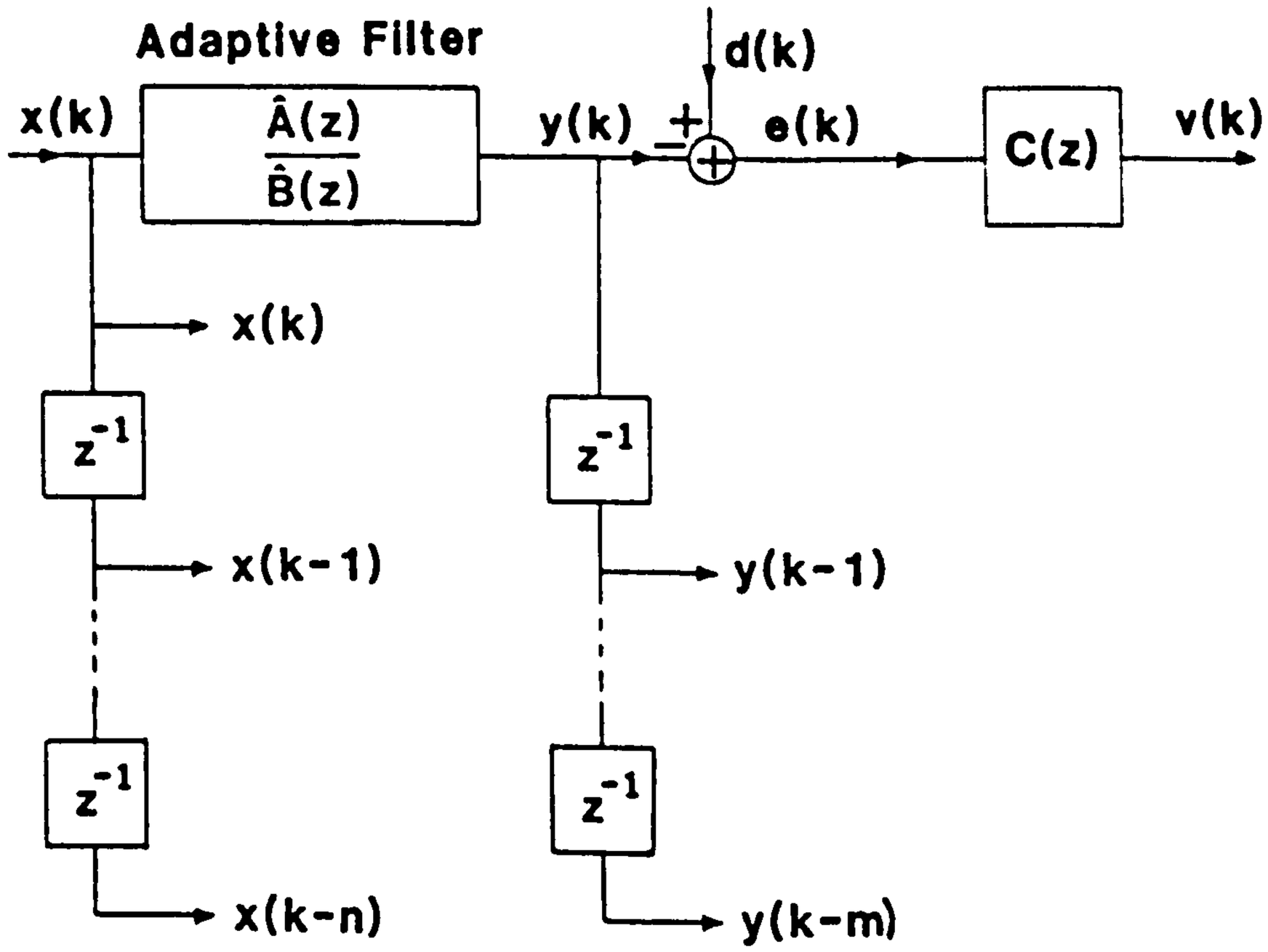


Figure 2.4 Filter Structure for the SHARF algorithm.

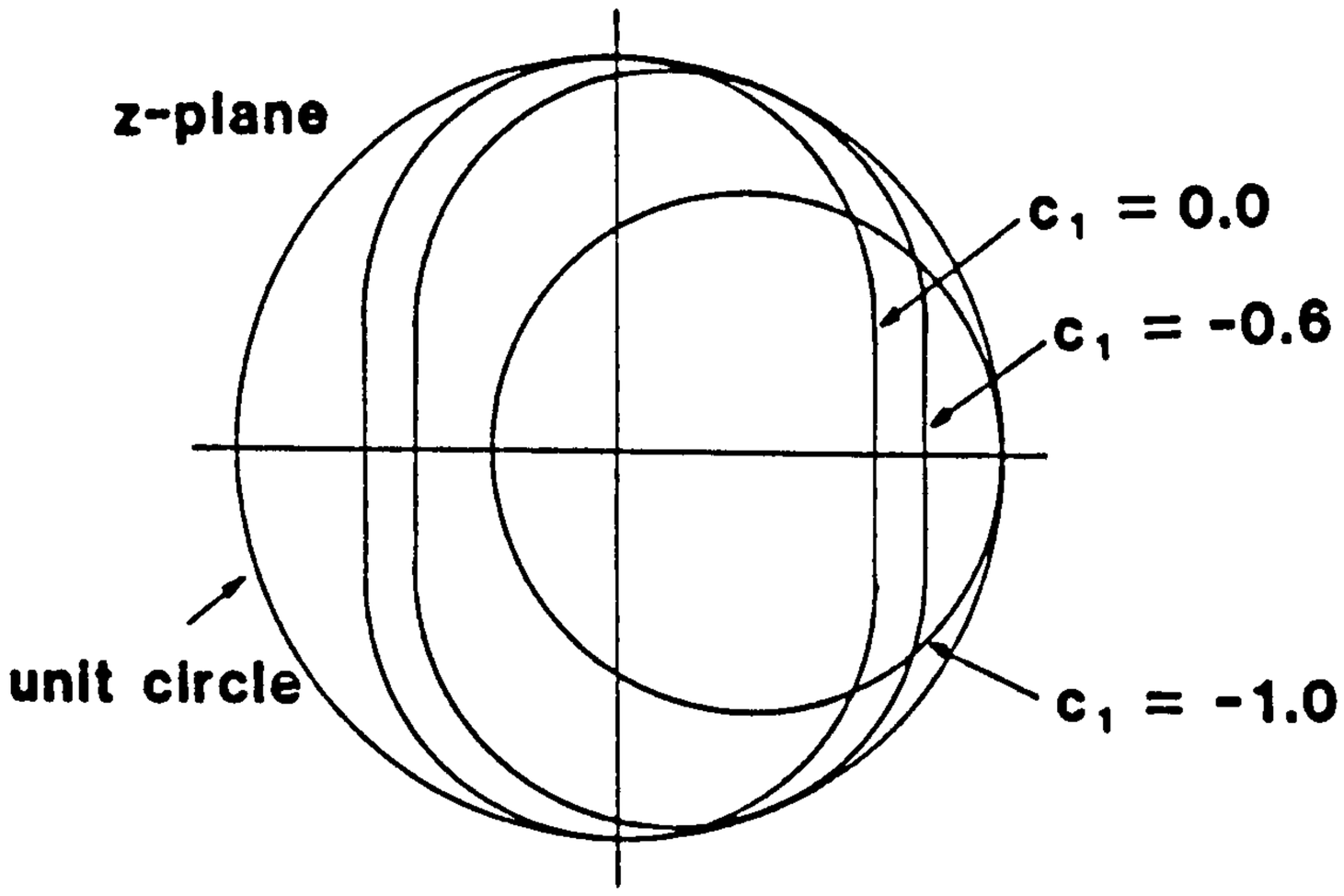


Figure 2.5 SPR Regions for $C(z) = 1 + c_1 z^{-1}$.

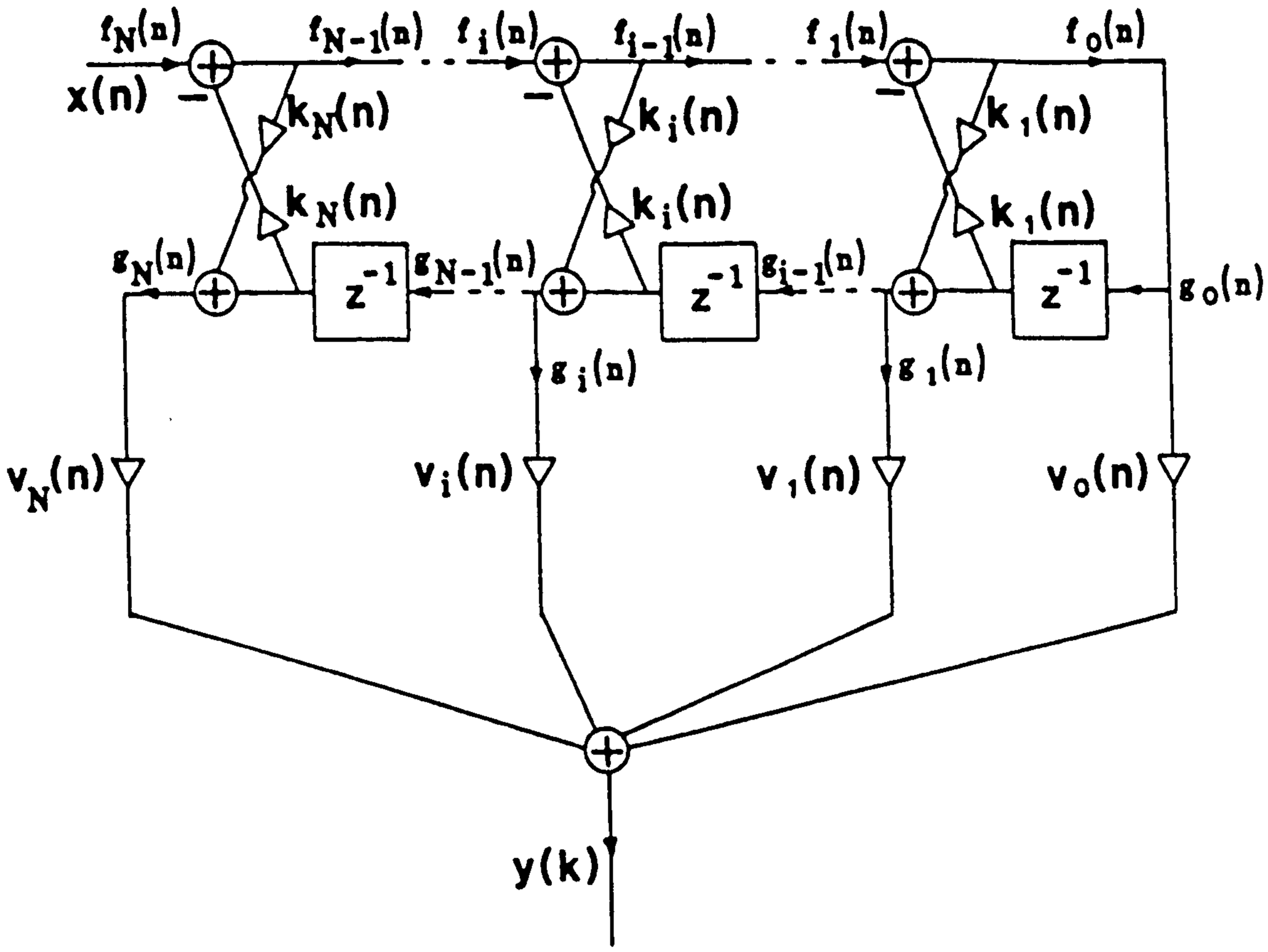


Figure 2.6 IIR Lattice Filter Structure.

CHAPTER 3

3.0 A CLASS OF STABLE GRADIENT-BASED ADAPTIVE IIR FILTERS

In this chapter, simple gradient-based adaptive IIR filtering algorithms are developed for a set of filter structures which are easily constrained to be stable. Three filter structures are proposed; a cascade of biquadratic pole and zero sections, a modified cascade with the zeros implemented in direct form, and a parallel arrangement of biquadratic sections. Filter stability is maintained by implementing the poles of the IIR filter transfer function in cascade and parallel arrangements of second order recursive sections, and then restricting the coefficients from each section to lie within the triangular region shown in figure 1.6, Chapter 1. Gradient algorithms are derived to adapt the structures based on the method of steepest descent. Approximations made in the derivations of the algorithms result in very little extra computation being required to generate the gradient terms. The proposed algorithms require considerably less computation than previously published algorithms by David[6,50] for cascade and modified cascade filter structures.

3.1 FILTER STABILITY

One of the major disadvantages of implementing an adaptive IIR filter as a direct form structure, is the absence of a simple stability check (See Section 2.2). The algorithms proposed in this chapter circumvent the problem by implementing the filter poles in combinations of second order recursive sections. Filter stability is maintained by constraining the (b_1,b_2) recursive coefficients from each section to lie within the 'stability triangle', ie. the triangular region shown in figure 1.6, which is a mapping of the z-plane unit circle.

Following a filter coefficient update, two inequalities are tested for each filter section, viz:

$$|b_2| < 1 \tag{3.1}$$

$$|b_1| < b_2+1 \tag{3.2}$$

If the point (b_1,b_2) lies on or outside the triangle then either one or both the inequalities will not be met, and the coefficient update is ignored. By repeating the process for all recursive sections in the filter, overall filter stability is guaranteed. It should be noted, however, that because the filter is adaptive, a bounded input signal will not necessarily produce a bounded output signal, even though the poles always lie within the unit circle. This problem should not arise in practice, however, as adaptive IIR filtering algorithms require the coefficient updates to be small[23,26,50].

3.2 ALGORITHM DERIVATIONS

In this section, three adaptive IIR filter structures which incorporate second order recursive sections for pole implementation are considered. The adaptation algorithms for the filter coefficients are based on the method of steepest descent for computational simplicity, and, therefore, applicability to hardware implementation. In Section 3.2.1, an adaptation algorithm for a cascade structure of biquadratic pole and zero sections is derived. The algorithm is modified in Section 3.2.2 for a filter structure consisting of a cascade of second order all-pole sections and a direct form all-zero section. Finally, in Section 3.2.3, an algorithm is derived for a parallel structure of biquadratic sections.

3.2.1 Adaptive Cascade Structure

The Adaptive Cascade Structure of biquadratic sections is shown in figure 3.1. The individual section transfer functions are given by:

$$\hat{A}_j(z) = 1 + a_{1j}z^{-1} + a_{2j}z^{-2} \qquad (3.3)$$

$$j=1,2,..,n-1$$

$$\hat{A}_n(z) = a_{0n} + a_{1n}z^{-1} + a_{2n}z^{-2} \qquad (3.4)$$

$$\hat{B}_j(z) = 1 + b_{1j}z^{-1} + b_{2j}z^{-2} \qquad (3.5)$$

$$j=1,2,..,n$$

The overall filter transfer function $\hat{H}(z)$ is obtained from:

$$\hat{H}(z) = \hat{H}_1(z).\hat{H}_2(z)...\hat{H}_n(z) \qquad (3.6)$$

where:

$$\hat{H}_J(z) = \frac{\hat{A}_J(z)}{\hat{B}_J(z)} \qquad (3.7)$$

The modularity of the Cascade structure is highly beneficial for hardware implementation because of the multiplexing possibilities, and, in addition, because the cascade structure coefficients are less sensitive to coefficient quantization than direct form coefficients[64].

Applying the method of steepest descent, which was described in Section 2.11, the filter coefficients are updated using:

$$a_{ij}(k+1) = a_{ij}(k) + \mu_a.(-\nabla_{a_{ij}(k)}) \qquad (3.8)$$

$$b_{ij}(k+1) = b_{ij}(k) + \mu_b.(-\nabla_{b_{ij}(k)}) \qquad (3.9)$$

where:

$$\nabla_{a_{ij}(k)} = \frac{\partial \overline{e^2(k)}}{\partial a_{ij}(k)} \tag{3.10}$$

$$\nabla_{b_{ij}(k)} = \frac{\partial \overline{e^2(k)}}{\partial b_{ij}(k)} \tag{3.11}$$

As $\overline{e^2(k)}$ is not readily available, the instantaneous squared error $e^2(k)$ is used as a local estimate. From figure 3.1, the error signal e(k) is obtained from:

$$e(k) = d(k) - y(k) \tag{3.12}$$

Hence:

$$e^2(k) = d^2(k) + y^2(k) - 2d(k)y(k) \tag{3.13}$$

Taking partial derivatives of equation (3.13) with respect to the filter coefficients, equations (3.10) and (3.11) yield:

$$\nabla_{a_{ij}(k)} = -2.e(k).\alpha_{ij}(k) \tag{3.14}$$

$$\nabla_{b_{ij}(k)} = -2.e(k).\beta_{ij}(k) \tag{3.15}$$

Where:

$$\alpha_{ij}(k) \triangleq \frac{\partial y(k)}{\partial a_{ij}(k)} \tag{3.16}$$

$$\beta_{ij}(k) \triangleq \frac{\partial y(k)}{\partial b_{ij}(k)} \tag{3.17}$$

Generation of the $\nabla_{a_{ij}}(k)$ and $\nabla_{b_{ij}}(k)$ coefficient update terms (equations (3.14),(3.15)) requires calculation of the partial derivatives of the adaptive filter output y(k) with respect to the filter coefficients. However, as for the Direct Form structure (See

Section 2.111), the Adaptive Cascade Filter output will contain transients from past coefficient changes, making calculation of the derivatives difficult. By only allowing small coefficient updates at each iteration, the adaptive filter transfer function will remain approximately constant over a number of iterations, and y(k) can be assumed to be the output of a fixed filter. This assumption is perhaps more justified for the cascade structure than the direct form structure, because of the lower sensitivity of the cascade filter coefficients[64]. However, coefficient changes in an early section of the cascade filter will create undesirable transient effects which will propagate through the filter. These effects are ignored in the following analysis and it is assumed that small coefficient changes in one section will not affect the inputs to the other sections.

From equations (3.6) and (3.7), the z-transform of the adaptive filter output is obtained from:

$$Y(z) = X(z) \prod_{j=1}^{n} \frac{\hat{A}_j(z)}{\hat{B}_j(z)} \qquad (3.18)$$

where X(z) is the z-transform of the filter input, and $\hat{A}_j(z)$, $\hat{B}_j(z)$ are obtained from equations (3.3), (3.4), and (3.5). Taking the partial derivatives of equation (3.18) with respect to the filter coefficients yields:

$$\alpha_{ij}(z) = \frac{Y(z).z^{-i}}{\hat{A}_j(z)} \qquad \begin{cases} j=1,2,n-1 \\ i=1,2 \end{cases} \qquad (3.19)$$

$$\begin{cases} j=n \\ i=0,1,2 \end{cases}$$

and:

$$\beta_{ij}(z) = \frac{-Y(z).z^{-i}}{\hat{B}_j(z)} \qquad \begin{cases} j=1,n \\ i=1,2 \end{cases} \tag{3.20}$$

where $\alpha_{ij}(z)$ and $\beta_{ij}(z)$ are obtained by taking the z-transform of equations (3.16) and (3.17).

Inspection of equations (3.19) and (3.20) reveals that the $\alpha_{ij}(z)$ and $\beta_{ij}(z)$ terms can be generated by a series of time-varying filters which are driven by the adaptive filter output Y(z). Strictly, because the $\hat{A}_j(z)$ and $\hat{B}_j(z)$ filter coefficients are constantly changing in time, 2n+1 extra second order recursive sections are required to generate the $\alpha_{ij}(z)$ terms, with a further 2n recursive sections needed for the $\beta_{ij}(z)$ terms. However,as only small coefficients updates are made at each iteration, $\hat{B}_j(z)$ and $\hat{A}_j(z)$ at time t = kT can be approximated by $\hat{B}_j(z)$ and $\hat{A}_j(z)$ respectively at time t = (k-i)T. This approximation was employed in the Hsia/Horvath algorithm[26,27] as a simplification of the White/Stearns algorithm[23,24]. The simplification enables the $\alpha_{ij}(z)$ and $\beta_{ij}(z)$ terms to be generated using only 2n second order recursive filters. For the final filter section, the $\alpha_{in}(z)$ terms are generated from:

$$\alpha_{in}(z) = \frac{Y(z).z^{-i}}{\hat{A}_n(z)} \qquad i=0,1,2 \tag{3.21}$$

These expressions can be rewritten as:

$$\alpha_{in}(z) = X(z).\frac{1}{\hat{B}_n(z)}.z^{-i}.\prod_{j=1}^{n-1}\frac{\hat{A}_j(z)}{\hat{B}_j(z)} \qquad i=0,1,2 \tag{3.22}$$

Close inspection of equation (3.22) reveals that the $\alpha_{in}(z)$ terms are already generated within the delay line of the final filter section, reducing the number of extra filters required to 2n-1 second order recursive sections. The steepest descent coefficient update equations (3.8) and (3.9) therefore become:

$$a_{ij}(k+1) = a_{ij}(k) + \mu_a.e(k).\alpha_{ij}(k) \quad \begin{cases} j=1,n-1 & , \quad i=1,2 \\ j=n & , \quad i=0,1,2 \end{cases} \tag{3.23}$$

$$b_{ij}(k+1) = b_{ij}(k) + \mu_b.e(k).\beta_{ij}(k) \quad \begin{cases} j=1,n \\ i=1,2 \end{cases} \tag{3.24}$$

with the $\alpha_{ij}(k)$ and $\beta_{ij}(k)$ terms generated from a set of filters driven by the adaptive filter output, y(k). The complete filter structure required to implement the Recursive Least Mean Squares (RLMS) Cascade Filter algorithm is shown in figure 3.2a.

As n-1 of the $\alpha_{ij}(z)$ gradient term generating filters are recursive filters, the transfer function zeros of the corresponding sections from the adaptive filter are restricted to lie within the z-plane unit circle and hence can only model minimum phase zeros. For the nth (final) filter section, the $\alpha_{in}(z)$ terms are generated using a non-recursive filter, and the zeros of that particular section have the freedom to adapt anywhere in the z-plane.

As the $\alpha_{ij}(z)$ and $\beta_{ij}(z)$ calculators are fed with the same input signal (Y(z)), no two pole pairs (or zero pairs) may be started from the same initial coefficients. If this occurs, the coefficient updates for the sections involved will be identical, and hence the filter coefficients will be unable to separate. As an alternative, sections could start adapting at different times, or with different values of adaptation constants (the second alternative would, however, result in an algorithm that was not strictly steepest descent).

Following each coefficient update, the stability of the

adaptive filter and of the $\alpha_{ij}(k)$ and $\beta_{ij}(k)$ term calculating filters are tested using equations (3.1) and (3.2). This involves the checking of the n recursive sections of the main filter, and n-1 $\alpha_{ij}(k)$ generating filters. If a section has adapted to coefficients which produce an unstable filter, the coefficient updates for that particular section are ignored. The true gradient vector will always point to a location within the unit circle for the $\{b_i\}$ filter coeffficients[50]. If the vector points outside then either:

a) μ_b is too large to allow convergence to the minimum.

b) Noise in the adaptation algorithm gives an incorrect gradient vector.

Reducing the size of μ_b is one alternative when stability limits are encountered. Methods for the adaptation of μ_b are considered in Chapter 4.

It is important to note that for the $\alpha_{ij}(k)$ calculating filters, it is possible that the gradient vector will correctly point outside the stability triangle during filter adaptation. However, apart from those zeros implemented in the final zero section, the remaining zeros must be restricted within the triangle to ensure the $\alpha_{ij}(k)$ calculating filters do not become unstable.

A considerably more complicated RLMS cascade structure was proposed by David in[6] in which the zeros were not confined within the unit circle. This was achieved at the expense of much greater computation. A block diagram of the David Structure is shown figure 3.2b for a comparison.

3.2.2 Modified Cascade Structure

The major disadvantage with the cascade structure of the previous section is that the transfer function zeros are constrained to lie within the z-plane unit circle, which therefore restricts the generality of the filter. To eliminate the constraint on the zeros, a modified cascade structure is proposed consisting of second order All-Pole sections and a Direct Form All-Zero section. A block diagram of the filter structure is shown in figure 3.3. Despite the loss in the modularity of the biquadratic cascade structure, the addition of the gradient calculating filters to the two structures results in a simpler overall filter structure for the modified cascade structure.

In addition, implementation of the IIR transfer function numerator as a ladder filter yields a quadratic error surface with respect to the $\{a_i\}$ filter coefficients. While this is in general beneficial, it may result in slower adaptation to sensitive zero positions. However, because the error surface with respect to the denominator coefficients is possibly very irregular, it is expected that the rate of adaptation of the poles will control the overall rate of adaptation of the filter. Hence, the sensitivity of the zero positions is of much less importance than the sensitivity of the poles.

From figure 3.3, the individual filter sections have transfer functions:

$$\hat{A}(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_n z^{-n} \qquad (3.25)$$

$$\hat{B}_j(z) = 1 + b_{1j} z^{-1} + b_{2j} z^{-2} \qquad j=1,2,\ldots,m \qquad (3.26)$$

The overall adaptive filter transfer function $\hat{H}(z)$ is given by:

$$\hat{H}(z) = \frac{\hat{A}(z)}{\hat{B}_1(z).\hat{B}_2(z)..\hat{B}_m(z)} \qquad (3.27)$$

Using the method of steepest descent, the derivation of the coefficient adaptation algorithm follows that for the biquadratic cascade in the previous section, for equations (3.8) → (3.17), but differs in the calculation of the $\alpha_i(k)$ gradient terms. For the modified cascade structure, the filter output is given by:

$$Y(z) = X(z).\hat{H}(z) \tag{3.28}$$

where H(z) is given by equation (3.27).

For the coefficient update equations (3.23) and (3.24), the partial derivatives of Y(z) with respect to the filter coefficients $\alpha_i(k)$, $\beta_{ij}(k)$ must be calculated. Making the same assumptions regarding the adaptive filter transfer function as made for the biquadratic cascade structure (Section 3.2.1), the $\alpha_i(z)$ and $\beta_{ij}(z)$ terms are obtained from equations (3.16),(3.17) and (3.28), viz:

$$\alpha_i(z) = \frac{X(z)z^{-i}}{\hat{B}_1(z).\hat{B}_2(z)...\hat{B}_m(z)} \qquad i=0,1,..,n \tag{3.29}$$

$$\beta_{ij}(z) = \frac{-Y(z).z^{-i}}{\hat{B}_j(z)} \qquad \begin{cases} i=1,2 \\ j=1,2,..,m \end{cases} \tag{3.30}$$

Generation of the $\alpha_i(z)$ and $\beta_{ij}(z)$ terms again requires an additional set of time-varying filters. By allowing only small coefficient updates, the $\beta_{ij}(k)$ terms can be generated using m second-order recursive sections, as for the biquadratic section cascade. However, the implementation of the zeros in direct form for the modified cascade structure enables the $\alpha_i(z)$ terms to be generated within the delay line of the $\hat{A}(z)$ filter, and hence require no extra processing or hardware.

The full filter structure required for implementation of the modified cascade adaptation algorithm is shown in figure 3.4. The filter coefficients are updated using:

$$a_i(k+1) = a_i(k) + \mu_a.e(k).\alpha_i(k) \qquad i=0,1,..,n \tag{3.31}$$

$$b_{ij}(k+1) = b_{ij}(k) + \mu_b.e(k).\beta_{ij}(k) \qquad \begin{cases} i=1,2 \\ j=1,2,....,m \end{cases} \tag{3.32}$$

A gradient-based coefficient adaptation algorithm for the modified cascade structure has been previously published by David[50]. However, the generation of the $\alpha_i(k)$ and $\beta_{ij}(k)$ terms is considerably simpler for the filter structure proposed here than for the David structure.

3.2.3 Parallel Structure

By implementing an IIR filter from second order biquadratic pole and zero sections connected in parallel[64], a further stable adaptive IIR filter structure can be constructed. A block diagram of the Parallel structure is shown in figure 3.5, with the filter section transfer functions given by:

$$\hat{A}_j(z) = a_{0j} + a_{1j}z^{-1} \tag{3.33}$$

$$\hat{B}_j(z) = 1 + b_{1j}z^{-1} + b_{2j}z^{-2} \qquad j=1,2,....,n \tag{3.34}$$

The overall filter transfer function $\hat{H}(z)$ is obtained from:

$$\hat{H}(z) = c + \sum_{j=1}^{n} \hat{H}_j(z) \tag{3.35}$$

Where:

$$\hat{H}_j(z) = \frac{\hat{A}_j(z)}{\hat{B}_j(z)} \tag{3.36}$$

Using the method of steepest descent, the derivation of the coefficient adaptation is initially the same as the derivation for the biquadratic cascade structure (equations (3.8) → (3.17)). The structures differ in the calculation of the partial derivatives of the filter output y(k) with respect to the coefficients. For the parallel structure these derivatives are defined as:

$$\alpha_{ij}(k) = \frac{\partial y(k)}{\partial a_{ij}(k)} \tag{3.37}$$

$$\beta_{ij}(k) = \frac{\partial y(k)}{\partial b_{ij}(k)} \tag{3.38}$$

$$\gamma(k) = \frac{\partial y(k)}{\partial c(k)} \tag{3.39}$$

With the filter output y(k) obtained from:

$$Y(z) = X(z).(c + \sum_{j=1}^{n} \hat{H}_j(z)) \tag{3.40}$$

Assuming that by taking only small coefficient updates, the filter output y(k) approximates the output from a fixed filter over a small number of iterations, then the $\alpha_{ij}(k)$, $\beta_{ij}(k)$ and $\gamma(k)$ terms are obtained from equations (3.33) → (3.40), viz:

$$\alpha_{ij}(z) = \frac{X(z).z^{-i}}{\hat{B}_j(z)} \qquad \begin{cases} i=0,1 \\ j=1,n \end{cases} \tag{3.41}$$

$$\beta_{ij}(z) = \frac{-Y_j(z).z^{-i}}{\hat{B}_j(z)} \qquad \begin{cases} i=1,2 \\ j=1,n \end{cases} \tag{3.42}$$

$$\gamma(z) = X(z) \tag{3.43}$$

As for the biquadratic and modified cascade structures, it can be assumed that with small coefficient steps, $\hat{B}_j(z)$ at time t = kT can be approximated by $\hat{B}_j(z)$ at time t = (k-i)T for i=1,2. This allows the $\alpha_{ij}(z)$ terms to be generated within the delay line of the $\hat{A}_j(z)$ filter sections. $\gamma(k)$ has the value of the present input x(k) (from equation (3.43)) and does not require any computation. Generation of the $\beta_{ij}(z)$ terms requires n extra second order All-Pole sections driven by the outputs of the n filter sections ($Y_j(z)$, j=1,2,..,n). The full adaptive parallel filter structure is shown in Figure 3.6. The filter coefficients are updated every sample using:

$$a_{ij}(k+1) = a_{ij}(k) + \mu_a.e(k).\alpha_{ij}(k) \quad \begin{cases} i=0,1 \\ j=1,n \end{cases} \tag{3.44}$$

$$b_{ij}(k+1) = b_{ij}(k) + \mu_b.e(k).\beta_{ij}(k) \quad \begin{cases} i=1,2 \\ j=1,n \end{cases} \tag{3.45}$$

$$c(k+1) = c(k) + \mu_c.e(k).\gamma(k) \tag{3.46}$$

The individual filter sections must be started with different initial coefficients, otherwise the sections will have identical coefficient updates and hence be unable to separate. Similarly, coherence of filter sections during adaptation must be avoided.

The parallel structure differs from the cascade structures in that the overall zero positions of the parallel structure transfer function are dependent on the pole positions, in addition to the zero positions, of the individual filter sections. Also, each section of the parallel filter is driven by the overall filter input X(z), whereas the cascade filter section input characteristics change from section to section.

3.3 PERFORMANCE ASSESSMENT

To assess the performance of the three adaptive filter structures derived in the previous section, a series of system modelling tests have been performed using the configuration shown in figure 1.2, Chapter 1. In the tests, a known plant H(z) and the adaptive filter $\hat{H}(z)$ are both excited by the same white Gaussian noise input X(z) of unity power. The task of the adaptive filter is to adapt its coefficients to form a model of the plant. To provide a performance comparison with the filter structures proposed in this chapter, a Direct Form IIR Filter Structure adapted using the White/Stearns steepest descent algorithm was also included in the tests.

In this section, three examples are presented to illustrate advantages and disadvantages of the various filter structures. In all examples, the adaptive filter was of sufficient order to model the plant, and hence could reduce the power in the error signal e(k) in figure 1.2 to zero[12]. Unlike the Direct Form Structure, the Biquadratic Cascade, Modified Cascade and Parallel Structures do not have a unique minimum in the performance surface, as the order of the filter sections can be permutated in various ways. However, the overall transfer function resulting in each case will be the same. To give an indication of speed of convergence of the filter structures, a performance measure used in echo cancellation applications was utilized. The criterion is called Echo Return Loss Enhancement (ERLE)[53] and is numerically given by:

$$\text{ERLE (dB)} = -10 \log_{10} \frac{\overline{e^2(k)}}{\overline{d^2(k)}} \qquad (3.47)$$

For an echo canceller, the error signal e(k) will appear as noise to the receiver and the value of ERLE increases as the adaptive filter lowers the error power. The problem of Echo cancellation is

discussed in detail in Chapter 5. For the examples presented in this section, convergence was said to have occurred when a value of 30dB ERLE was reached. To achieve this figure, the adaptive filter must form an accurate model of the plant.

In each of the examples, fixed values of adaptation constants μ_a and μ_b were used, with the values of each chosen to give the fastest convergence times. Less noisy convergence may be obtained by reducing the values of μ_a and μ_b. Any coefficient update that would take a recursive filter section unstable was ignored for that particular section. The true gradient vector will always point to a location within the unit circle[55], but the instantaneous gradient estimate may take filter sections unstable, even though suitable values for μ have been chosen for overall filter convergence.

Examples (1) and (2) are 4th order All-Pole examples and provide a comparison of the convergence rates of the direct form, cascade and parallel recursive filter sections. For the Parallel structure, the $\{a_{ij}\}$ filter coefficients are also adapted, in order to cancel the zeros produced by a parallel implementation of poles. For each example, a number of different white noise sequence inputs were used, and the convergence times quoted were the slowest obtained.

3.3.1 Example (1)

A plant consisting of two pairs of closely spaced poles near $z = -1$ in the z-plane was implemented. The plant had transfer function H(z) given by:

$$H(z) = \frac{0.05}{(1 + 1.2z^{-1} + 0.8z^{-2})(1 + 1.5z^{-1} + 0.7z^{-2})} \qquad (3.48)$$

The complex pole pairs of H(z) have radii of 0.90 and 0.85 with corresponding arguments 132°and 153°. The Direct Form and Cascade

Adaptive Filter structures were also fourth order All-Pole filters with a non-adaptive a_0 coefficient set to 0.05 (the value of the a_0 coefficient from the plant). The denominator coefficients of the Direct Form Structure were all initially set to zero, but the corresponding Cascade and Parallel recursive section coefficients must be adapted from different initial values. Therefore, the two sections were adapted from (b_{11},b_{21}) = (-0.2,0.0) and (b_{12},b_{22}) = (0.2,0.0). To obtain a sufficient sized Parallel adaptive filter structure, the $\{a_{ij}\}$ coefficients of both filter sections must also be adaptive. These coefficients were initially set to zero.

Convergence plots showing the attenuation of the squared error for the three filter structures are shown in figure 3.7a,b,c. The Cascade Structure achieved the fastest convergence obtaining 30dB ERLE in about 7,000 iterations. This compares with 80,000 iterations for the Parallel Structure, and almost 700,000 iterations for the Direct Form Structure to reach the 30dB limit.

These results were obtained with values of the adaptation constant μ_b chosen to given the fastest convergence time for each structure. The high sensitivity of the Direct Form coefficients to closely spaced poles[64], coupled with possible instability, necessitated a very small value of μ_b to produce convergence. This results in the very slow convergence for the Direct Form Structure. For the two structures composed of second-order All-Pole sections, guaranteed stability enabled the use of larger values of μ_b. However, the parallel structure has problems due to the adaptation of the $\{b_{ij}\}$ filter coefficients, which create zeros in the overall transfer function. The $\{a_{ij}\}$ coefficients must adapt to cancel these zeros for an All-pole target plant. For this example with closely spaced poles, the parallel structure did not perform very well and adaptation was slow. From the convergence plot shown in figure 3.7b, undesirable large spikes

of noise appear during the adaptation which suggests that the coefficients are sensitive to small changes. With its relatively insensitive coefficients, the cascade structure was able to achieve considerably faster convergence than the other two structures.

3.3.2 Example (2)

A plant consisting of two pairs of poles widely spaced in the unit circle was implemented. The plant had transfer function H(z) given by:

$$H(z) = \frac{1}{(1 + 1.5z^{-1} + 0.8z^{-2})(1 - 1.5z^{-1} + 0.7z^{-2})} \qquad (3.49)$$

The complex conjugate pole pairs have radii of 0.90 and 0.85 with respective arguments of 147°and 26°. The Direct Form, Cascade and Parallel filter structures had the same initial adaptive filter coefficients as used in Example (1). Squared error convergence plots for the three adaptive filter structures are shown in figure 3.8a,b,c.

In direct contrast to example (1), the Direct Form achieved the fastest convergence with 30dB ERLE obtained in only 200 iterations. The Parallel Structure was the next quickest taking 8000 iterations. The Cascade Structure, however, was unable to model the plant and only obtained 5 dB ERLE when adapted. The inability of the Cascade Structure to model the plant is caused by a local minimum in the filter coefficient performance surface, with both sections "locking up" at $(b_1,b_2) = (-0.15,-0.15)$. The pole positions corresponding to these filter coefficients are shown in figure 3.9. The coefficient values give rise to two pairs of split real poles, each section contributing one pole at E (z = -0.32), and one pole at F (z = 0.47). The plant poles are situated at C and D.

From figure 1.2, during adaptation the $\hat{B}_j(z)$ sections adapt to

minimize $\overline{e^2(k)}$ obtained from:

$$E(z) = X(z).\left(\frac{1}{B(z)} - \frac{1}{\hat{B}_1(z)\hat{B}_2(z)}\right) \tag{3.50}$$

Where $H(z) = 1/B(z)$ is the transfer function of the plant. The individual filter sections are adapting such that:

$$\hat{B}_1(z) \rightarrow \frac{B(z)}{\hat{B}_2(z)} \tag{3.51}$$

$$\hat{B}_2(z) \rightarrow \frac{B(z)}{\hat{B}_1(z)} \tag{3.52}$$

If the poles of $\hat{B}_1(z)$ and $\hat{B}_2(z)$ lie close to the centre of the unit circle, it appears to each section that it is modelling the plant alone.

If a single second-order pole section is used to model this plant, then the minimum mean-squared-error value is obtained with split real poles placed near E and F. Although an adaptive filter consisting of two second-order pole sections is of sufficient order to model the plant, if the initial pole positions are not split far enough apart, then the error surface forces the filter into the local minimum with each recursive section contributing one pole at E and one at F. The constraints of the cascade structure prevent the formation of complex conjugate pole pairs from this position, and hence adaptation is halted. To enable the cascade structure to converge to the solution, then either;

1) The poles have to have initial starting positions which are split far enough apart in the unit circle to avoid the problem, or;
2) By allowing the filter to lock up, and then replacing each split real pole pair by a suitable complex conjugate pole pair.

The first method requires prior knowledge of the plant pole

locations, as the initial separation distance required for the adaptive filter poles is dependent on plant characteristics. Furthermore, if the plant poles were closely spaced and the adaptive filter poles had initial positions that were a large distance apart, then adaptation times could be increased considerably.

Using the second method of pole replacement, with $\mu_b = 0.04$ the cascade filter structure coefficients cohered such that the b_{1j} and b_{2j} coefficients from each section were separated by less than 0.01 in 450 iterations. The two split real pole pairs are then replaced with two complex conjugate pairs using coefficient manipulation.

When the sections lock-up, $b_{11} = b_{12}$ and $b_{21} = b_{22}$. This corresponds to pole positions z_1 and z_2 given by:

$$z_1 = \frac{-b_{11} - \sqrt{(b_{11}^2 - 4b_{21})}}{2.0} \tag{3.53}$$

$$z_2 = \frac{-b_{11} + \sqrt{(b_{11}^2 - 4b_{21})}}{2.0} \tag{3.54}$$

The new coefficients corresponding to two complex conjugate pairs are obtained from:

$$b_{11} = 2.0 * -z_1 \tag{3.55}$$

$$b_{21} = z_1^2 \tag{3.56}$$

$$b_{12} = 2.0 * -z_2 \tag{3.57}$$

$$b_{22} = z_2^2 \tag{3.58}$$

Convergence to 30dB ERLE was then obtained in a further 2,050 iterations with $\mu_b = 0.005$, resulting in a total of 2,500 iterations for convergence. A convergence plot showing the reduction in squared error is shown in figure 3.10a.

It is interesting to note that the parallel structure of second order sections is not trapped in a similar local minimum to that which

traps the cascade structure, when both structures have the same initial denominator coefficients, viz, (b_{11},b_{21}) = (-0.2,0.0), (b_{12},b_{22}) = (0.2,0.0). Investigation revealed that the parallel structure can fall into the local minimum if the adaptive filter poles are initially situated very close together. For example, with μ_a = 0.05, μ_b = 0.05 and initial transfer function:

$$\hat{H}(z) = \frac{0.0 + 0.0z^{-1}}{(1 + 0.05z^{-1} + 0.0z^{-2})} + \frac{0.0 + 0.0z^{-1}}{(1 - 0.05z^{-1} + 0.0z^{-2})} \qquad (3.59)$$

the coefficients lock up in 200 iterations. Following coefficient replacement, the filter obtained 30dB ERLE convergence in a further 1,050 iterations using μ_a = 0.005 and μ_b = 0.008. Figure 3.10b shows a convergence plot for the squared error. The 1,250 iterations required for convergence is considerably less than the 8,000 iterations obtained with gradient adaptation from initial coefficients (b_{11},b_{21}) = (-0.2,0.0) and (b_{12},b_{22}) = (0.2,0.0).

Inspection of cascade and parallel recursive coefficients during adaptation shows that, in general, they follow similar paths in the coefficient planes. However, the major difference between the cascade and parallel structures for an all-pole plant, is that the parallel structure requires adaptation of the $\{a_{ij}\}$ filter coefficients to match the plant transfer function. The adaptation of the $\{a_{ij}\}$ coefficients enables the parallel poles to avoid the "lock-up" problem. If the parallel structure poles and zeros are started initially from the constrained local minimum position, no further adaptation occurs. Conversely, if the cascade filter is adapted using $\{a_{ij}\}$ coefficients in addition to the $\{b_{ij}\}$ coefficients, the local minimum is avoided and the poles adapt to the plant pole postions with the transfer function zeros converging to the origin of the unit circle in the final stages of the adaptation.

3.3.3 EXAMPLE (3)

A fourth order plant was implemented consisting of two pairs of complex conjugate poles closely spaced near z = -1 in the z-plane, and two pairs of zeros closely spaced near z = 1. The plant had the transfer function:

$$H(z) = \frac{0.01(1 - 1.2z^{-1} + 0.8z^{-2})(1 - 1.5z^{-1} + 0.7z^{-2})}{(1 + 1.2z^{-1} + 0.8z^{-2}) (1 + 1.5z^{-1} + 0.7z^{-2})} \qquad (3.60)$$

The poles are those implemented in example (1) and have radii of 0.9 and 0.85 with arguments 132°and 153° respectively. The zeros have radii 0.9 and 0.85 with corresponding arguments 48°and 27°. Modelling this plant is a very difficult task as the poles are close together near the unit circle, similarly the zeros, and the poles are placed well away from the zeros.

In this example, each adaptive filter had the same order numerator and denominator as the plant. With transfer function zeros being modelled, both the Biquadratic Cascade and the Modified Cascade structures are compared with the Parallel and Direct Form structures. For the cascade and parallel filter structures, the denominator coefficients were initially set to $(b_{11},b_{12}) = (-0.2,0.0)$ and $(b_{12},b_{22}) = (0.2,0.0)$. The $\{a_i\}$ coefficients for the Direct Form and Modified Cascade structures, and the $\{a_{ij}\}$ coefficients for the Parallel structure were initially set to zero. The Biquadratic Cascade filter structure consisted of two second-order biquadratic sections and a final filter section consisting of the a_{03} gain coefficient alone. The $\{a_{ij}\}$ coefficients for the two biquadratic sections must be adapted from different initial coefficients and were preset to $(a_{11},a_{21}) = (-0.2,0.0)$ and $(a_{12},a_{22}) = (0.2,0.0)$.

It was found that the Direct Form structure was unable to adapt to this plant within 1,000,000 iterations due to the sensitivity of the

coefficients to closely spaced poles and closely spaced zeros. Convergence plots for the Biquadratic, Modified Cascade and Parallel structures are given in figures 3.11a,b,c respectively. Of the three structures, the Parallel structure achieved the fastest convergence taking 42,000 iterations to reach 30dB ERLE. The Modified Cascade structure was second fastest taking 52,000 iterations. For the Biquadratic Cascade Structure, it was necessary to use a smaller value of μ to adapt the a_{03} gain coefficient than was required for the other $\{a_{ij}\}$ filter coefficients (for the two biquadratic sections $\mu_a = 1.0$, for the $a_{03}(k)$ gain coefficient $\mu_g = 0.00005$). The different values of μ are necessary because of the very large power difference between the $\alpha_{03}(k)$ terms and the other $\alpha_{ij}(k)$ terms. This is a great problem for all adaptive cascade filters. Convergence times for the Biquadratic Cascade structure varied greatly between 40,000 and 60,000 iterations. It is seen that the convergence plot for this structure is very noisy when compared with the plots for the Modified Cascade and Parallel structures.

Inspection of the Modified Cascade Structure coefficients during filter adaptation showed that whilst the plant zeros lie inside the unit circle, the adaptive filter zeros move outside the unit circle and then return back in when following a steepest descent path. For the Biquadratic Cascade structure, the coefficient adaptation algorithm limits the zeros to lie within the unit circle. Hence the $\{a_{ij}\}$ coefficients from the two filter sections adapt as close to the unit circle as numerical accuracy allows, causing problems with the stability of the $\alpha_{ij}(k)$ gradient term generating filters. This is very undesirable from a hardware implementation viewpoint. The restriction on the location of the zeros introduces noise into the adaptation which results in large variations in convergence times. Although the Biquadratic Cascade Filter Structure can allow the zeros of the final filter section

to adapt outside the unit circle, convergence times for this example were found to be slower with the a_{03} gain coefficient integrated into the second filter section.

3.4 CONCLUSIONS

Three simple adaptive IIR filter structures have been proposed which use gradient techniques to minimize the mean-square-error performance criterion. Examples have been given to compare the performance of the proposed algorithms and the Direct Form structure adapted using the standard gradient algorithm. The results show that in terms of speed of convergence, the best choice of filter structure is very dependent on the plant pole positions. The Direct Form structure adapted quickly to a plant with widely separated poles (example (2)), but was very slow when adapting to plants with closely spaced poles near the unit circle (examples (1) and (3)). The Parallel structure can also have difficulty with closely spaced poles due to the adaptation of the $\{b_{ij}\}$ coefficients affecting overall transfer function zero positions (example (1)). However, in example (3), the Parallel structure converged faster than the other structures when the plant consisted of the poles of example (1) implemented with two pairs of close zeros.

Both of the cascade filter structures, and also the Parallel structure, can have difficulties when adapting to poles with locations which are widely spaced in the unit circle, as illustrated in example (2). By replacing the 'locked' coefficients with complex conjugate pole pairs, it was demonstrated that relatively fast convergence to the solution could be obtained. It should be noted that the existence of the constrained local minimum is not necessary a major disadvantage as the local minimum will lie towards, and often close to, the global minimum.

For the two cascade structures, it was expected that the Biquadratic Section Cascade would be less sensitive than the Modified

Cascade to closely spaced zeros in the target plant. However, example (3) demonstrated that the Biquadratic Cascade structure can have problems during adaptation as the zeros may wish to move outside the unit circle when following a steepest descent path, and are restricted from doing so by the constraints on the gradient term calculating filters. Although the Biquadratic Section Cascade can allow the zeros of one filter section to adapt outside the unit circle, this was not found to be advantageous for the example presented.

The Biquadratic Cascade Structure proposed by David[6] enables all the zeros to adapt outside the unit circle, but at a greatly increased computational cost. A further disadvantage of implementing the zeros in cascade form is that they are susceptible to the constrained local minimum problem of example (2), when adapting to widely spaced zeros. The existence of a possible lock-up condition for an All-Zero cascade filter was noticed by Ching and Goodyear[57], and was verified by tests to exist under similar conditions as those for the all-pole sections.

As the order of the adaptive filter is increased, adaptation times will in general also be increased. It can be envisaged that the irregularities and non-uniformity of the performance surface with respect to the Direct Form coefficients will severely affect adaptation times of Direct Form filter algorithms. The filter structures proposed in this chapter are easily kept stable and hence offer much faster convergence through the adaptation of the μ parameters.

From the results presented, the Modified Cascade structure offers the best overall performance of the filter structures considered. However, the optimum values of μ parameters used in the examples were found by experimentation. For each filter structure, the choice of μ_a and μ_b parameters is an empirical process which, as the results demonstrated, is highly plant dependent.

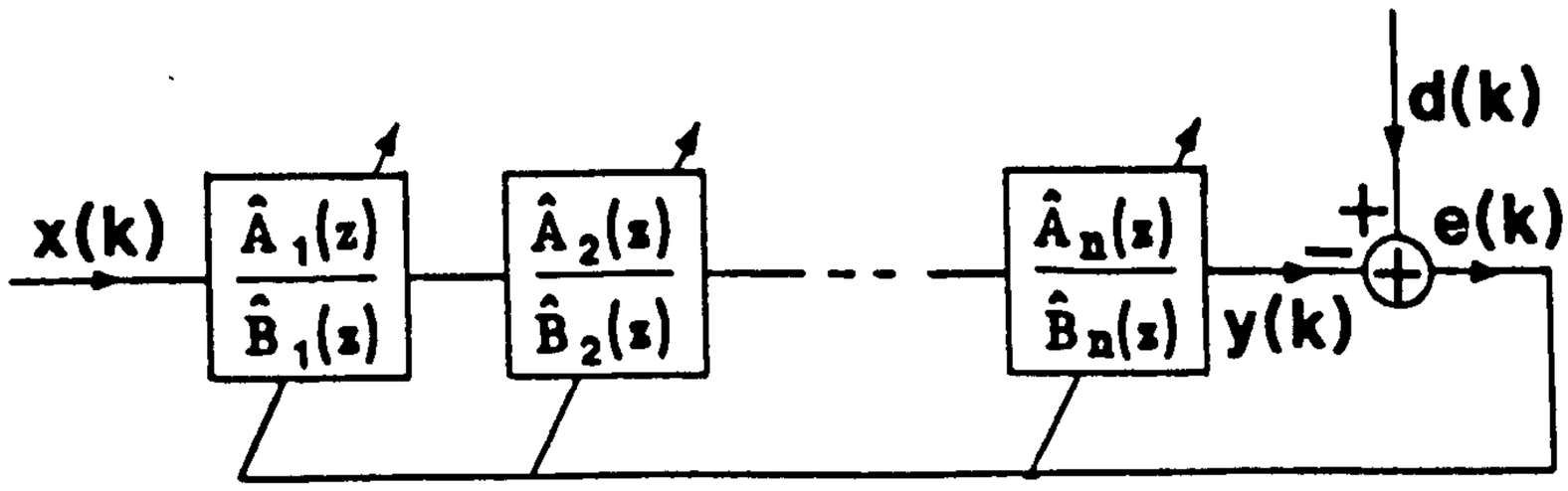
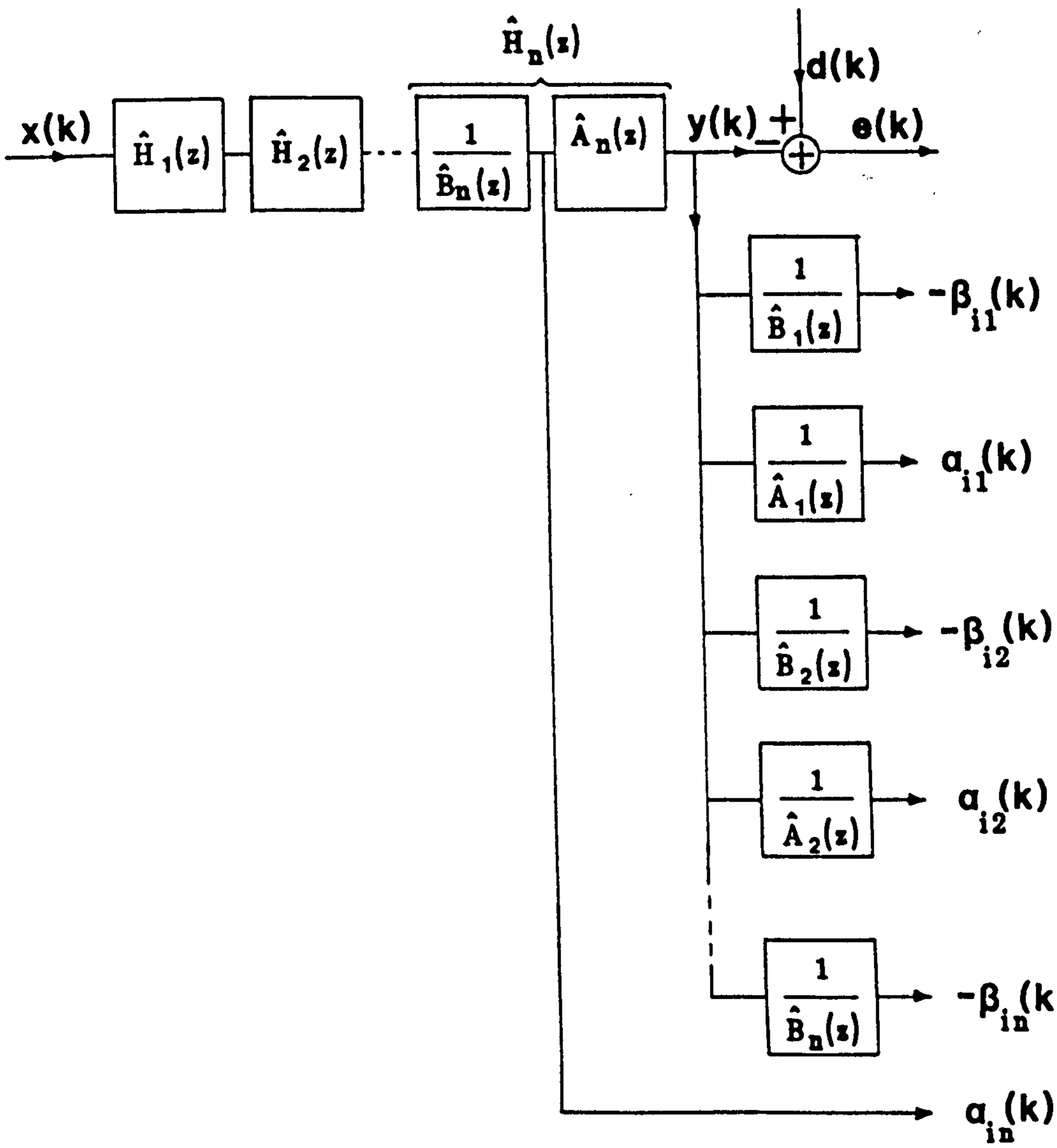
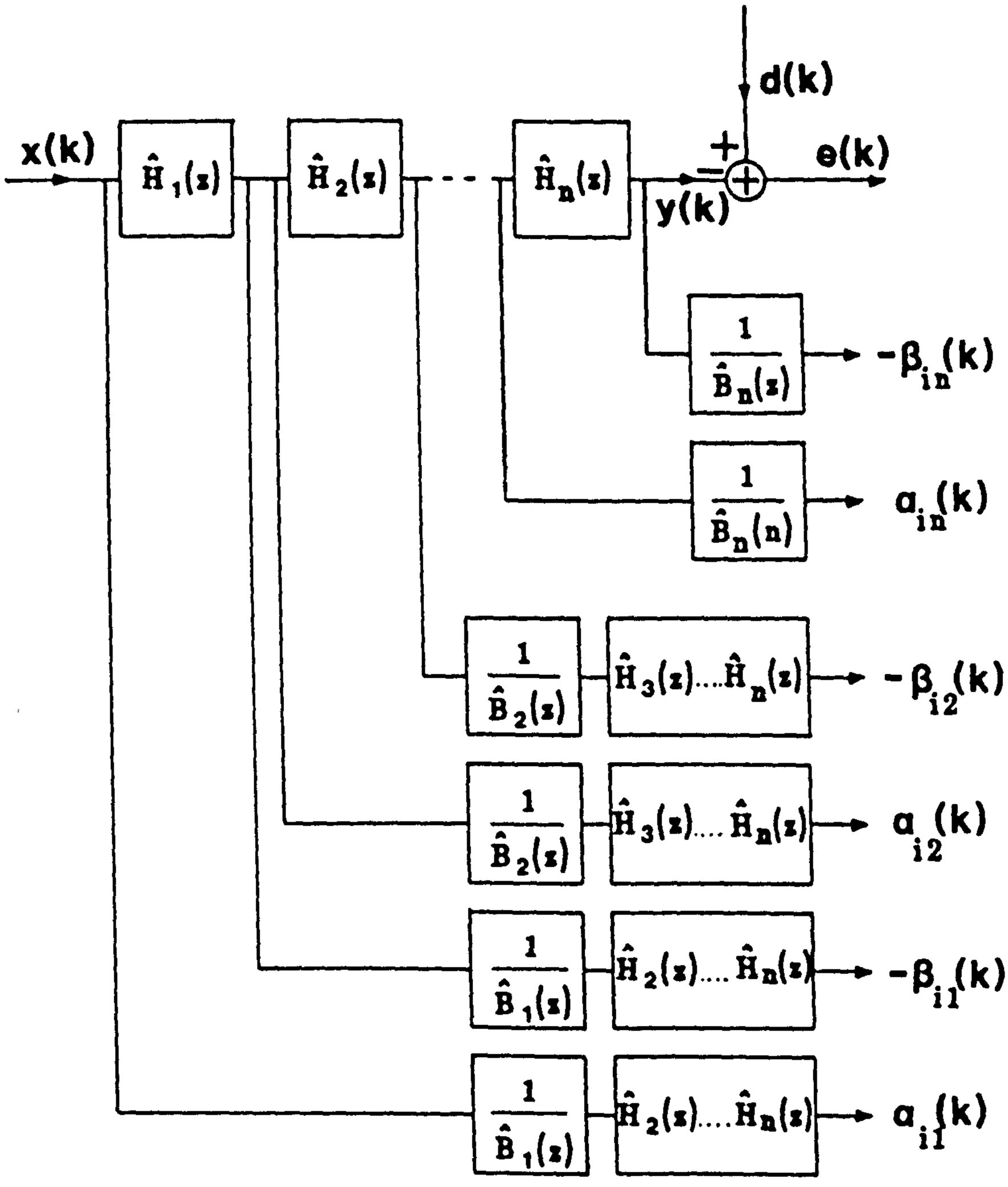


Figure 3.1 Adaptive Cascade of Biquadratic Filter Sections.



a) Cascade Filter Structure implementing Simplified RLMS Adaptation Algorithm.

Figure 3.2 Adaptive Biquadratic Cascade Filter Structures.



b) David CRLMS Filter Structure

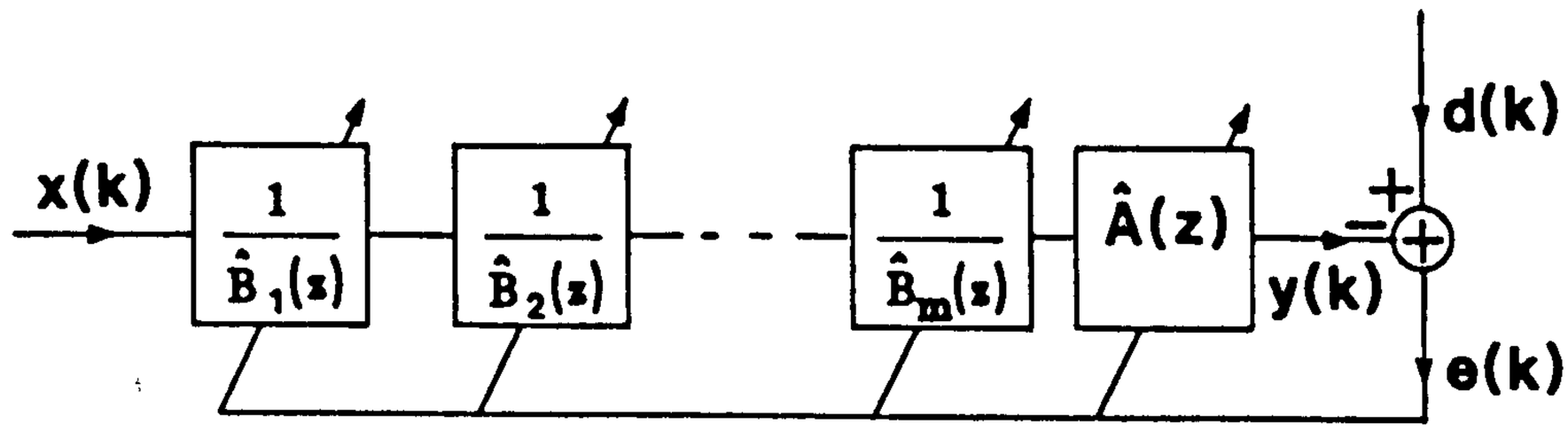


Figure 3.3 Adaptive Modified Cascade Filter Structure.

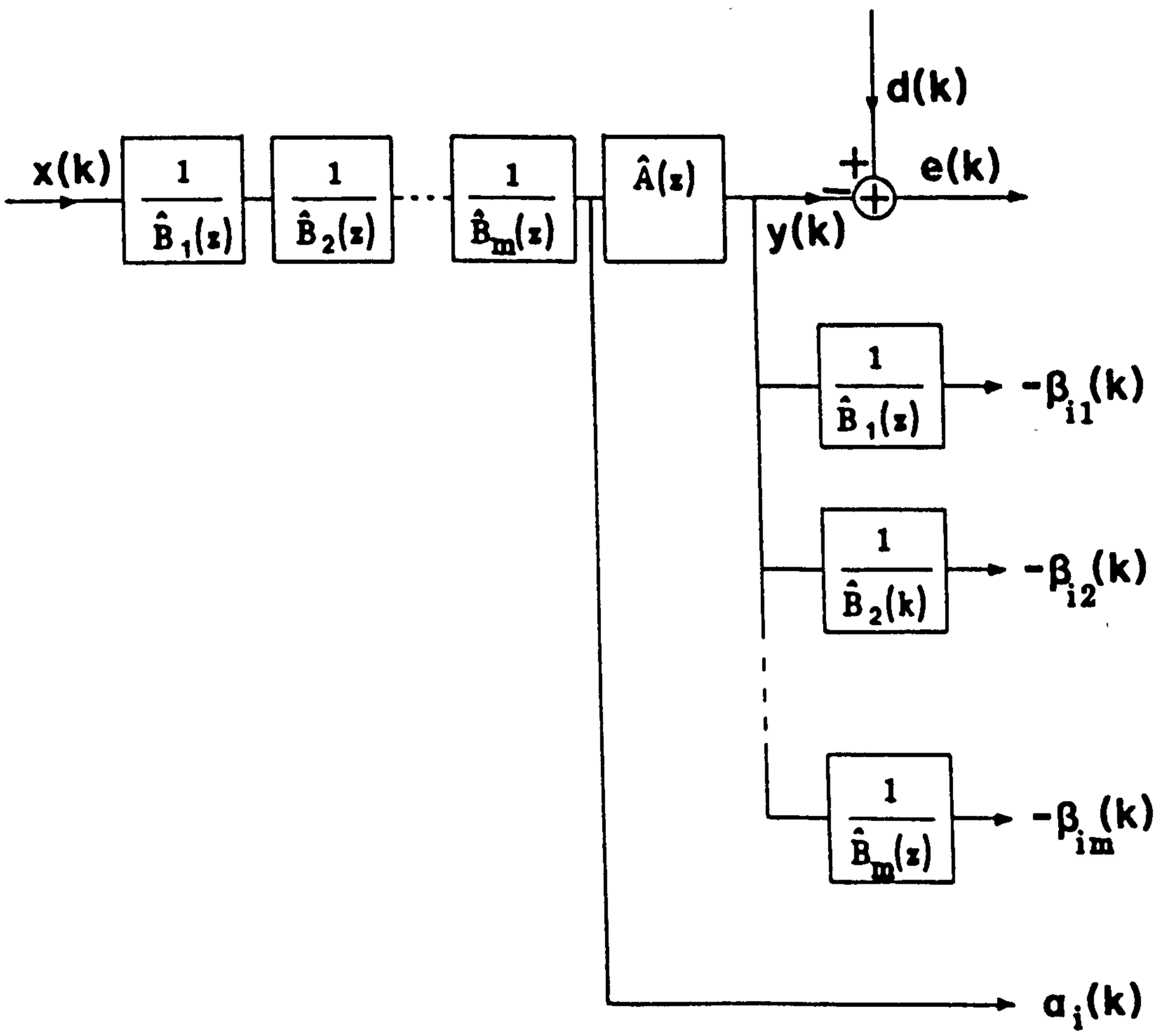


Figure 3.4 Modified Cascade Structure implementing Simplified RLMS Adaptation Algorithm.

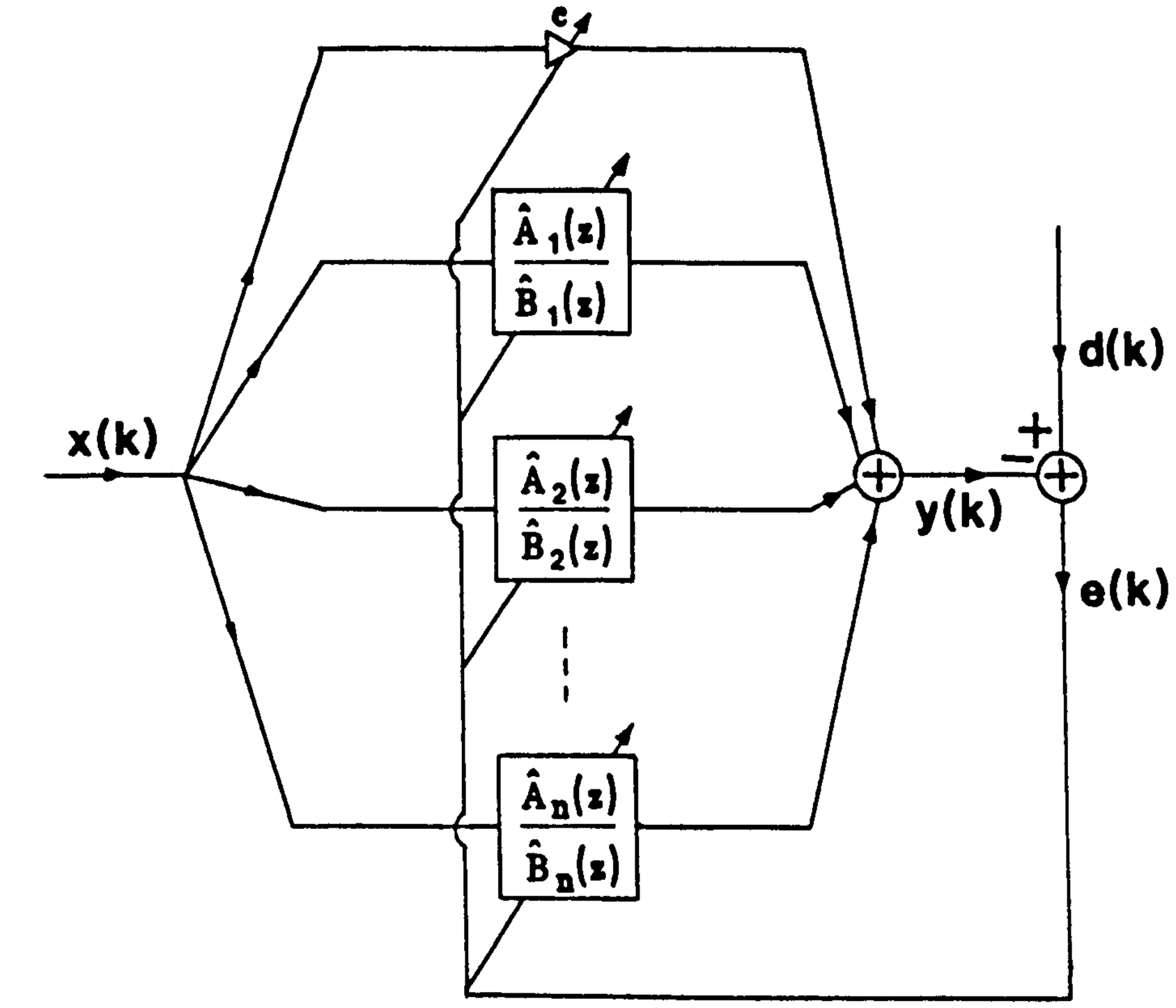


Figure 3.5 Adaptive Parallel Filter Structure.

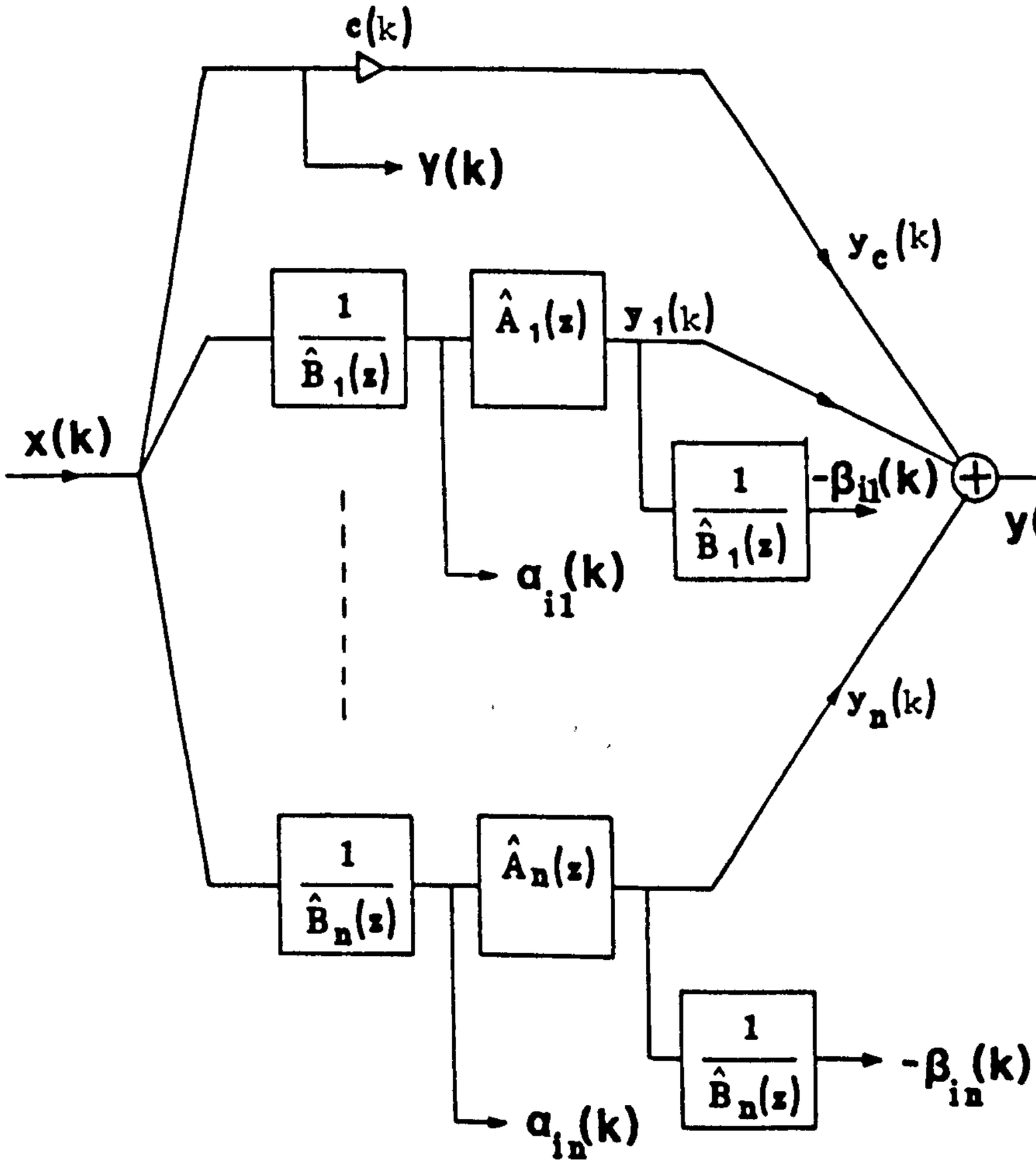
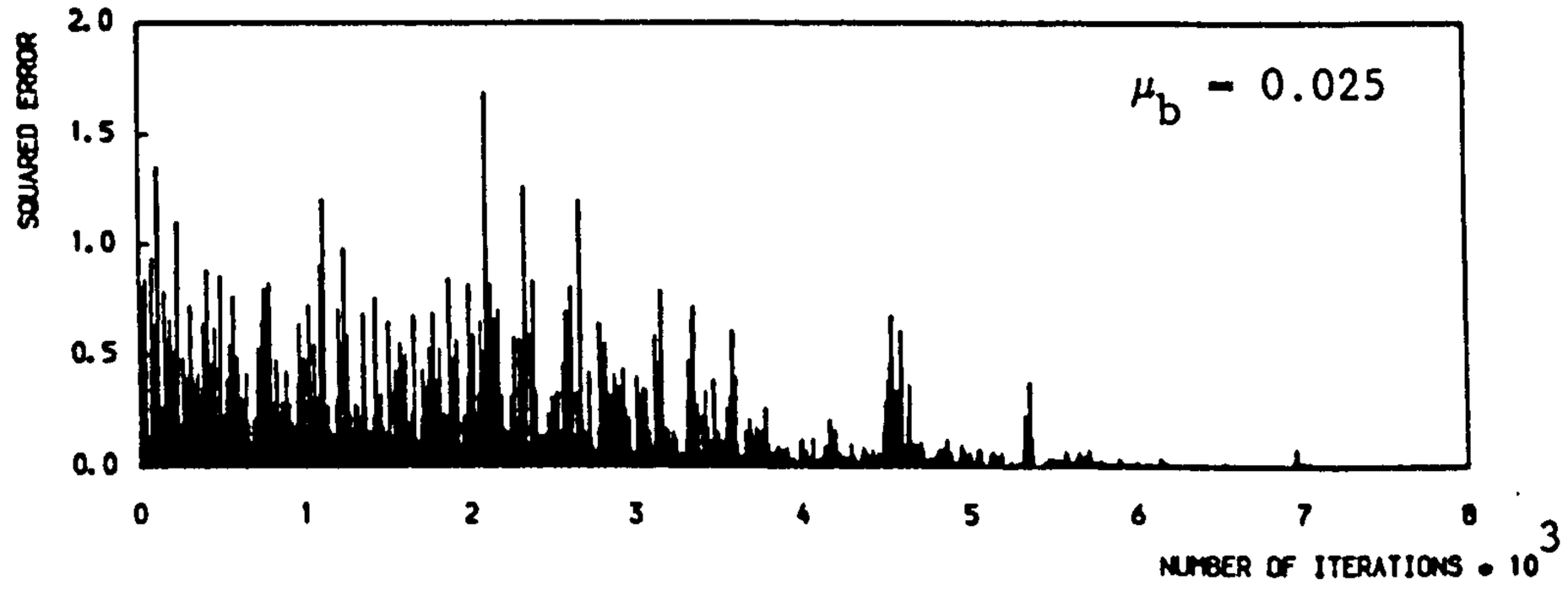
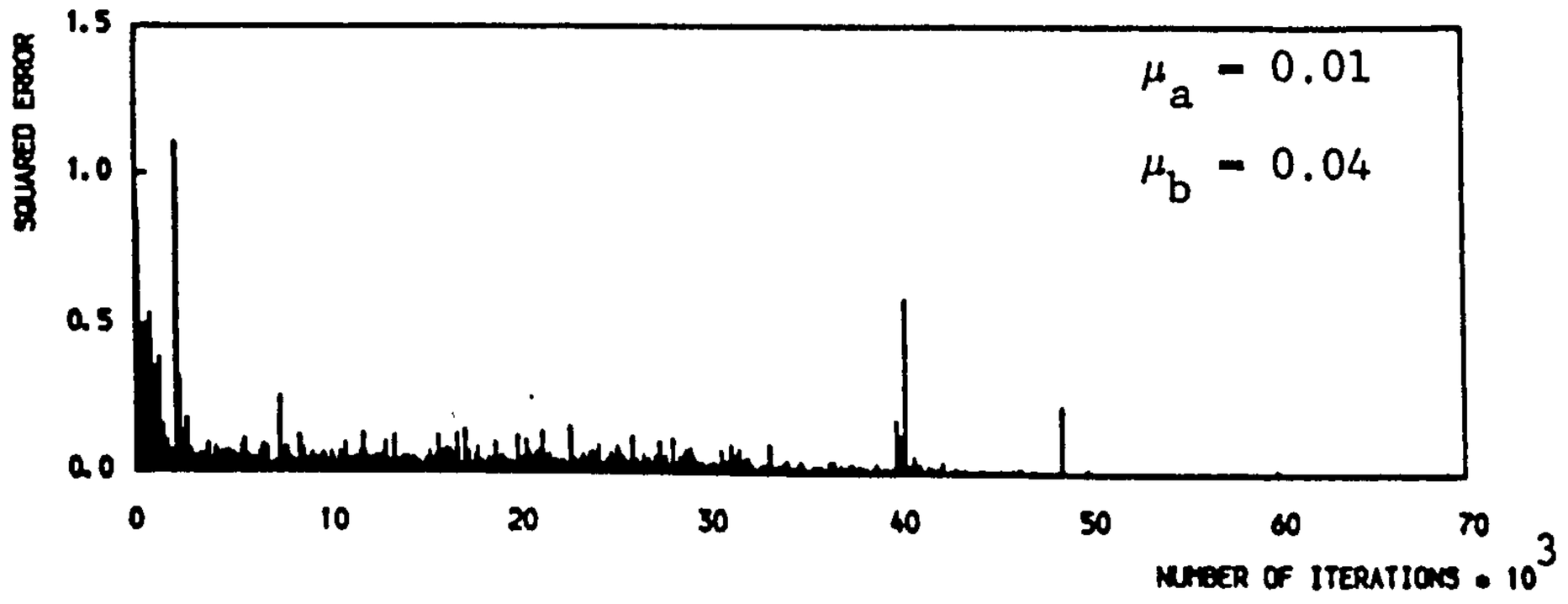


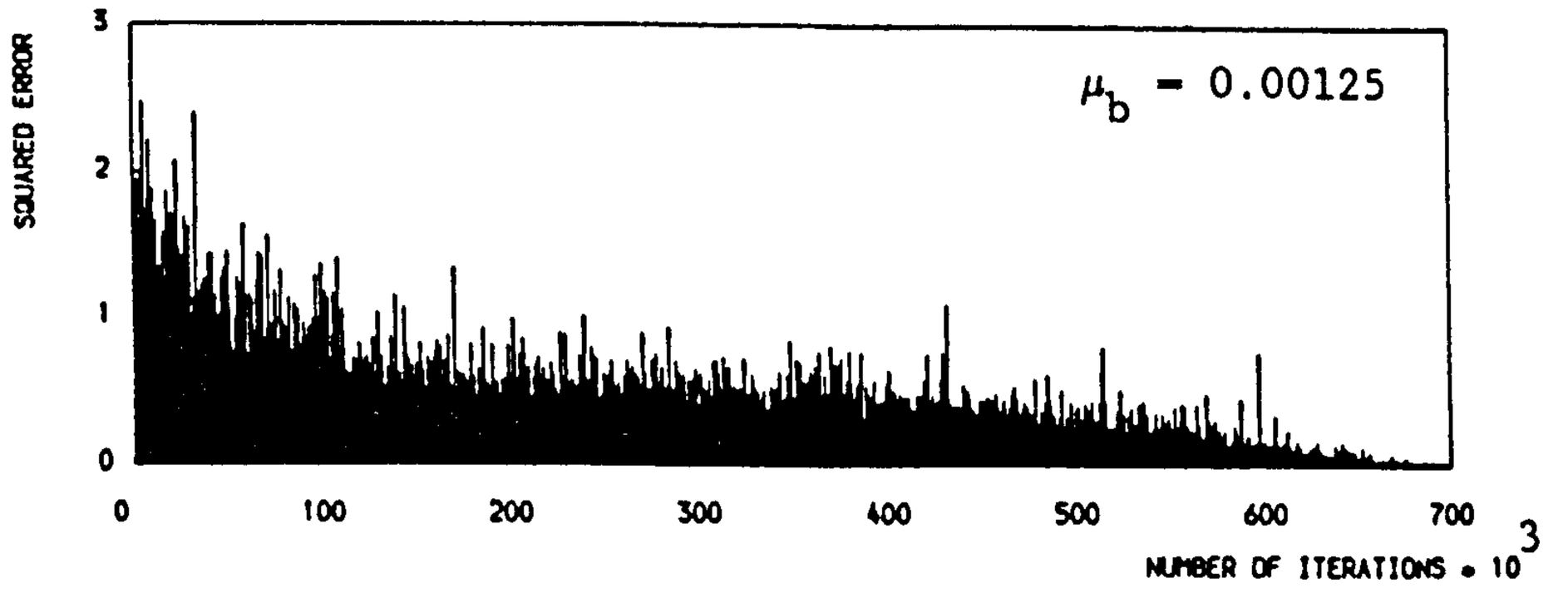
Figure 3.6 Parallel Filter Structure implementing Simplified RLMS Adaptation Algorithm.



a) Cascade Filter Structure.

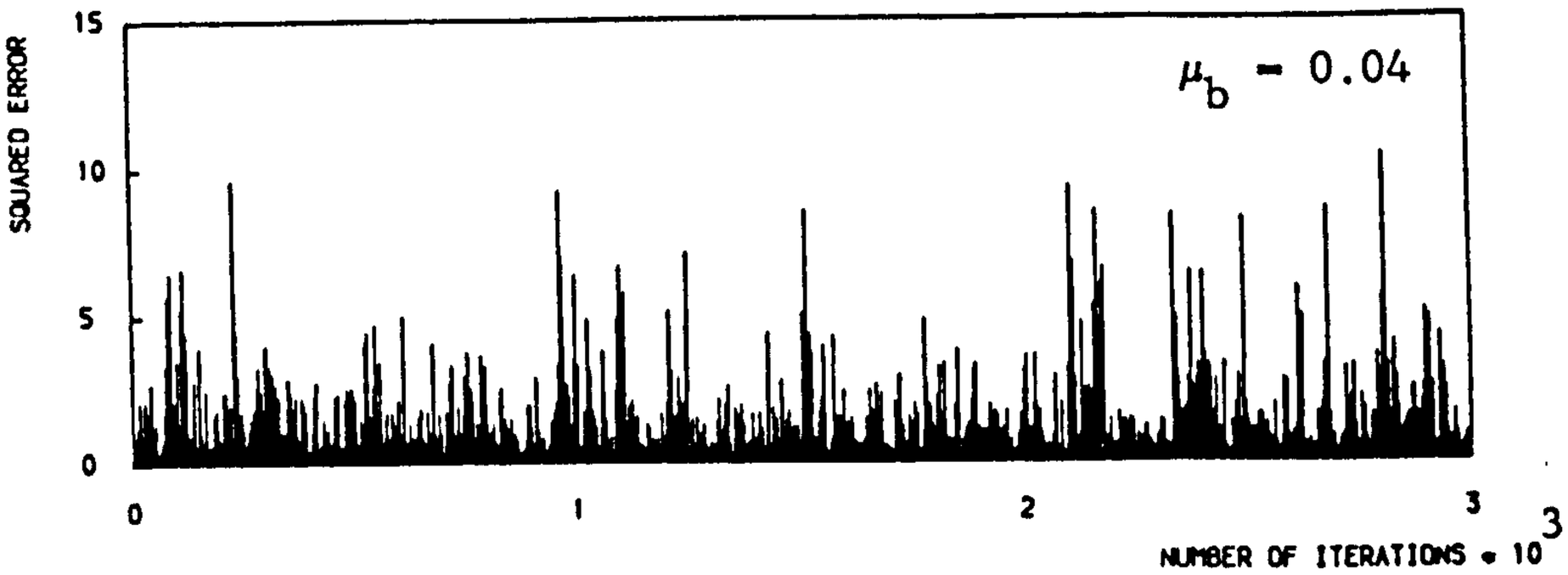


b) Parallel Filter Structure.



c) Direct Form Filter Structure.

Figure 3.7 Convergence Plots for Example (1).



a) Cascade Filter Structure.

b) Parallel Filter Structure.

c) Direct Form Filter Structure.

Figure 3.8 Convergence Plots for Example (2).

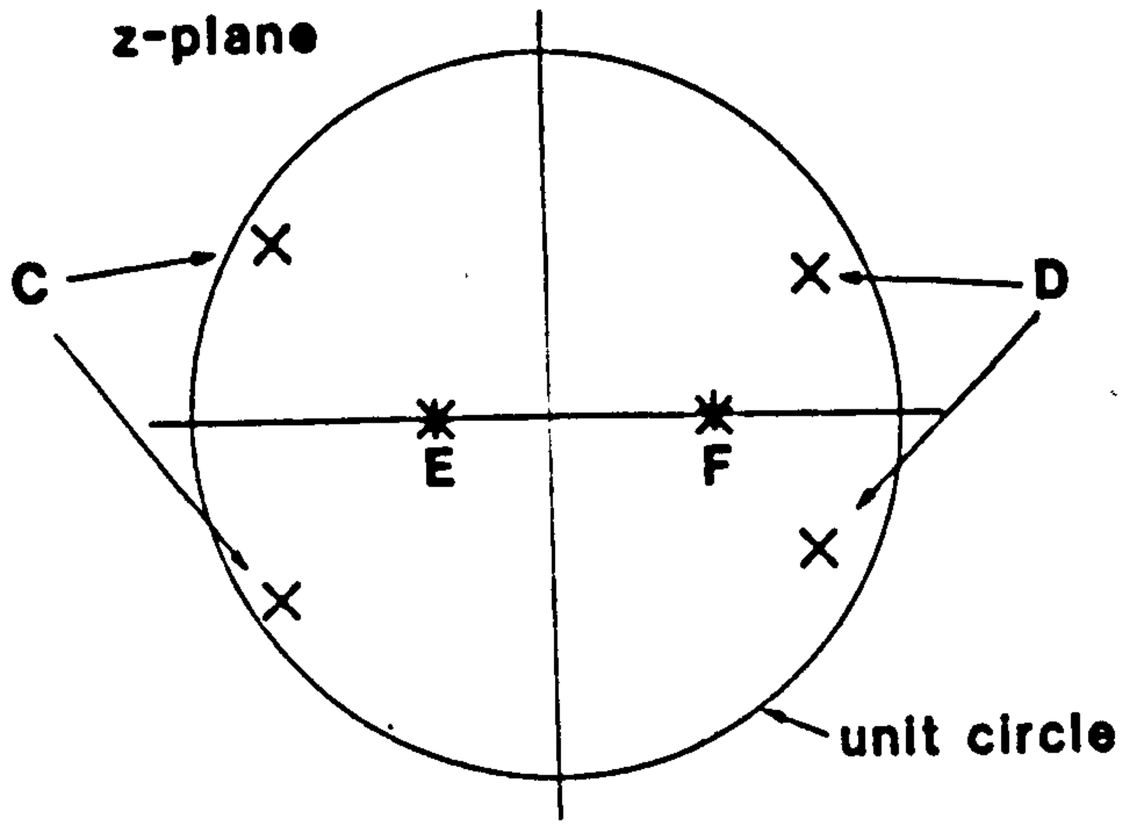
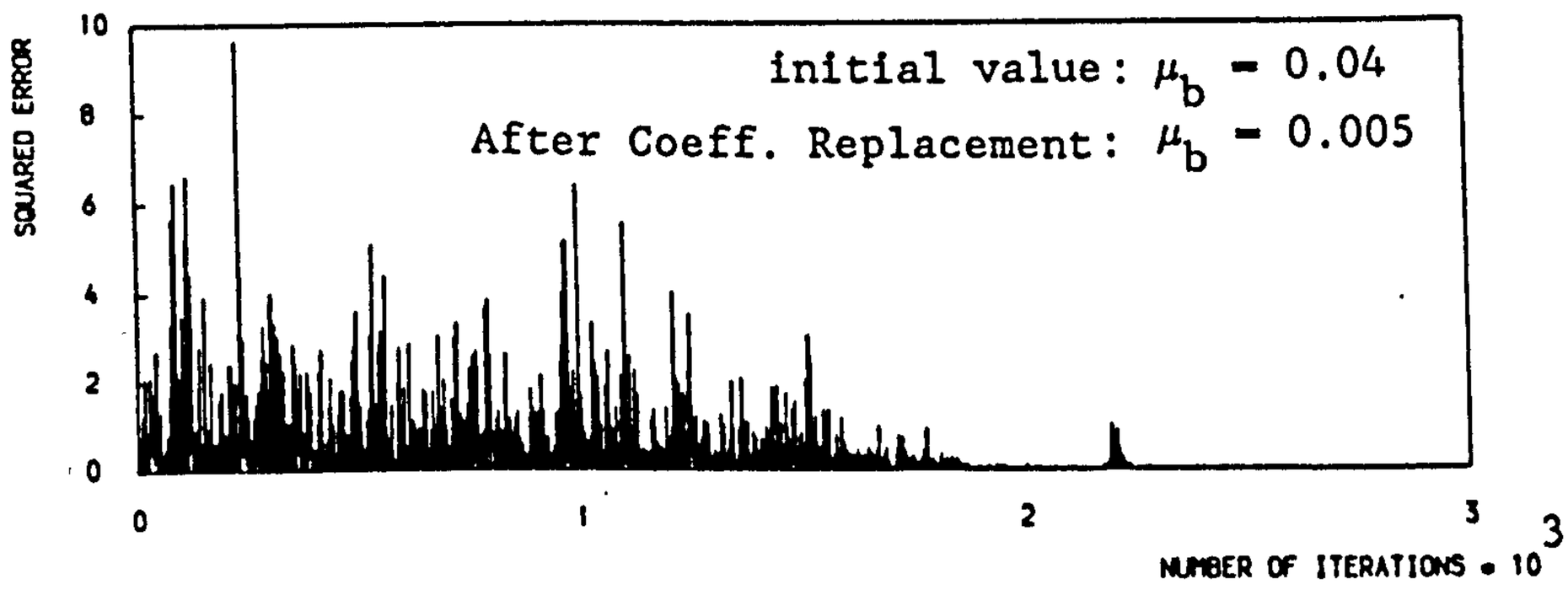
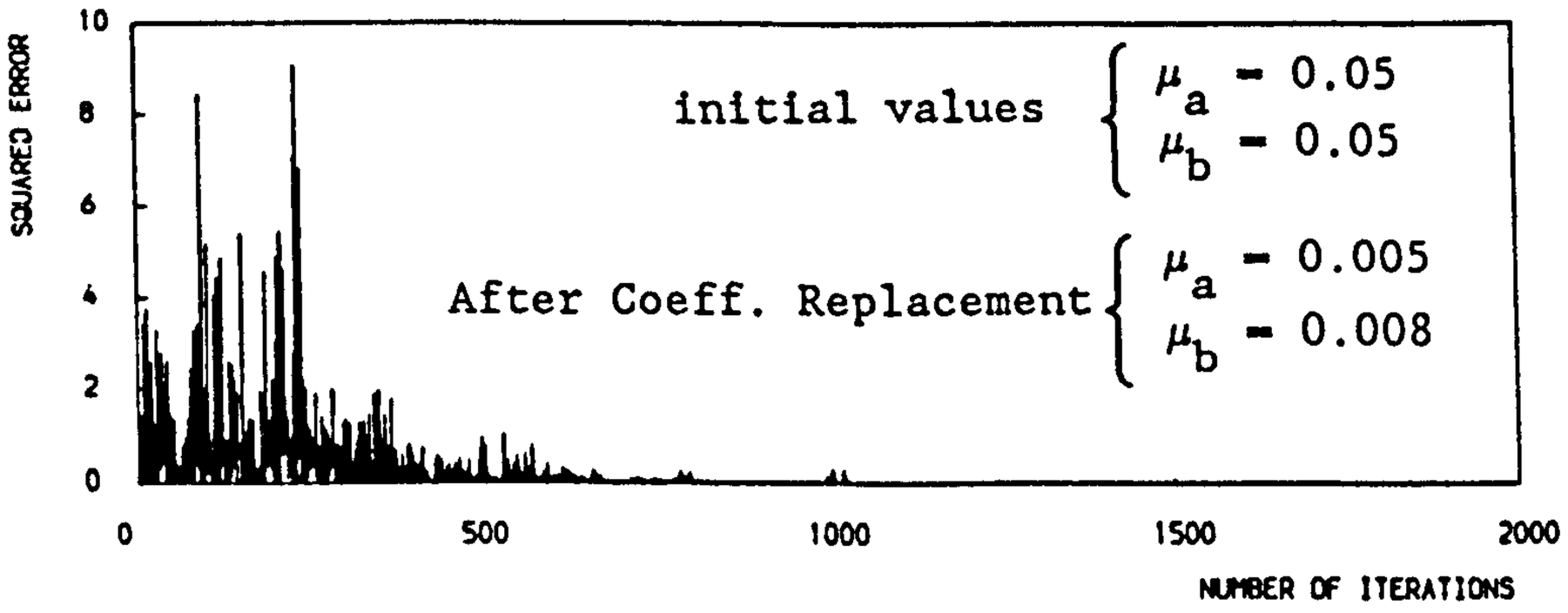


Figure 3.9 Local Minimum Pole Positions.

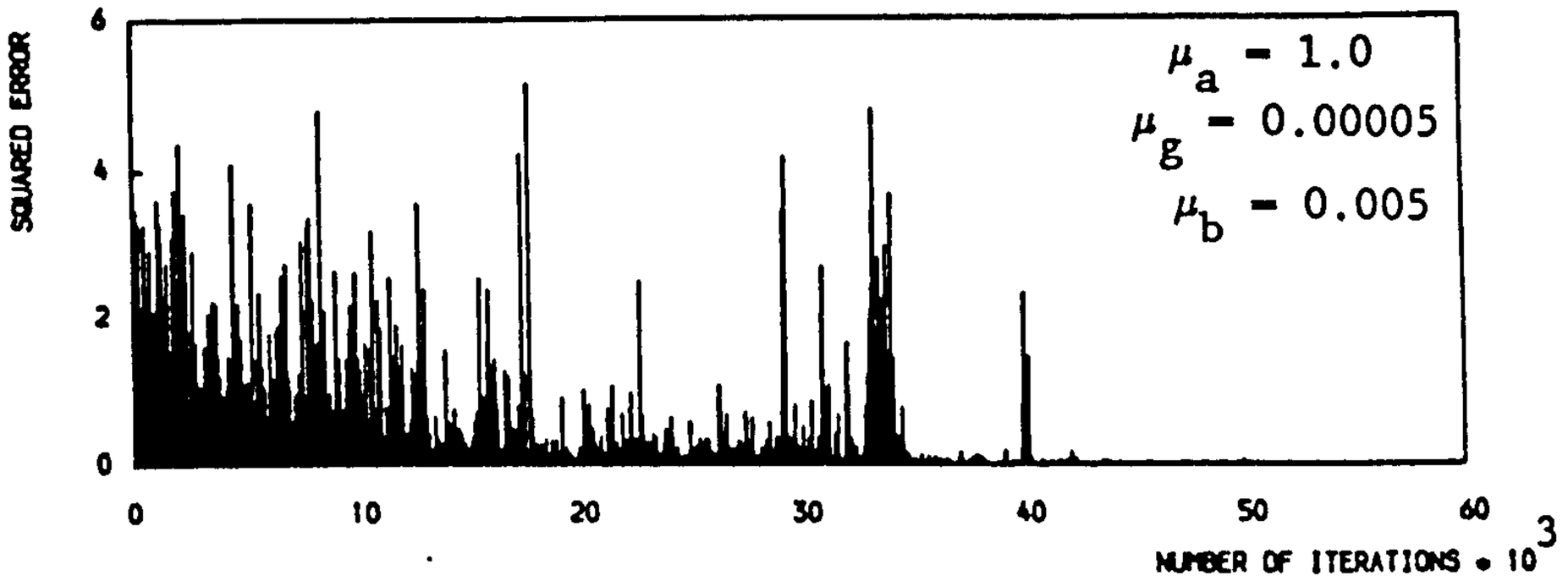


a) Cascade Filter Structure.

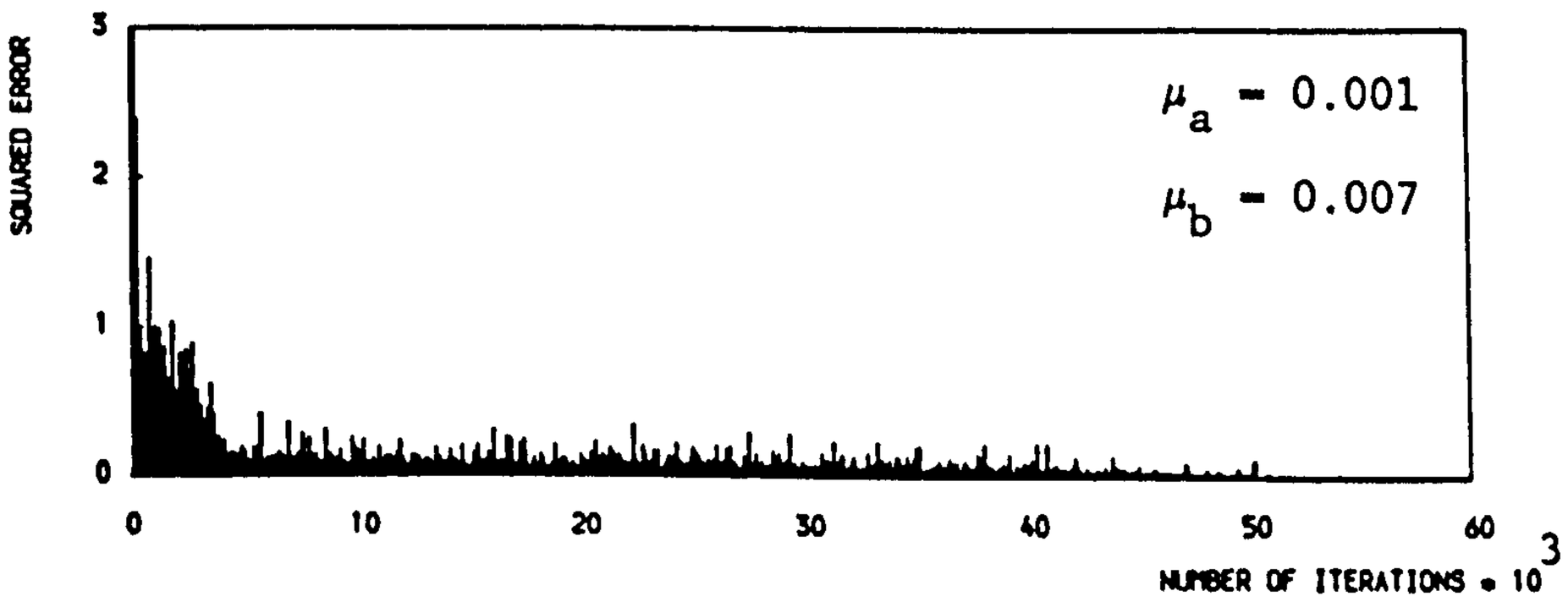


b) Parallel Filter Structure.

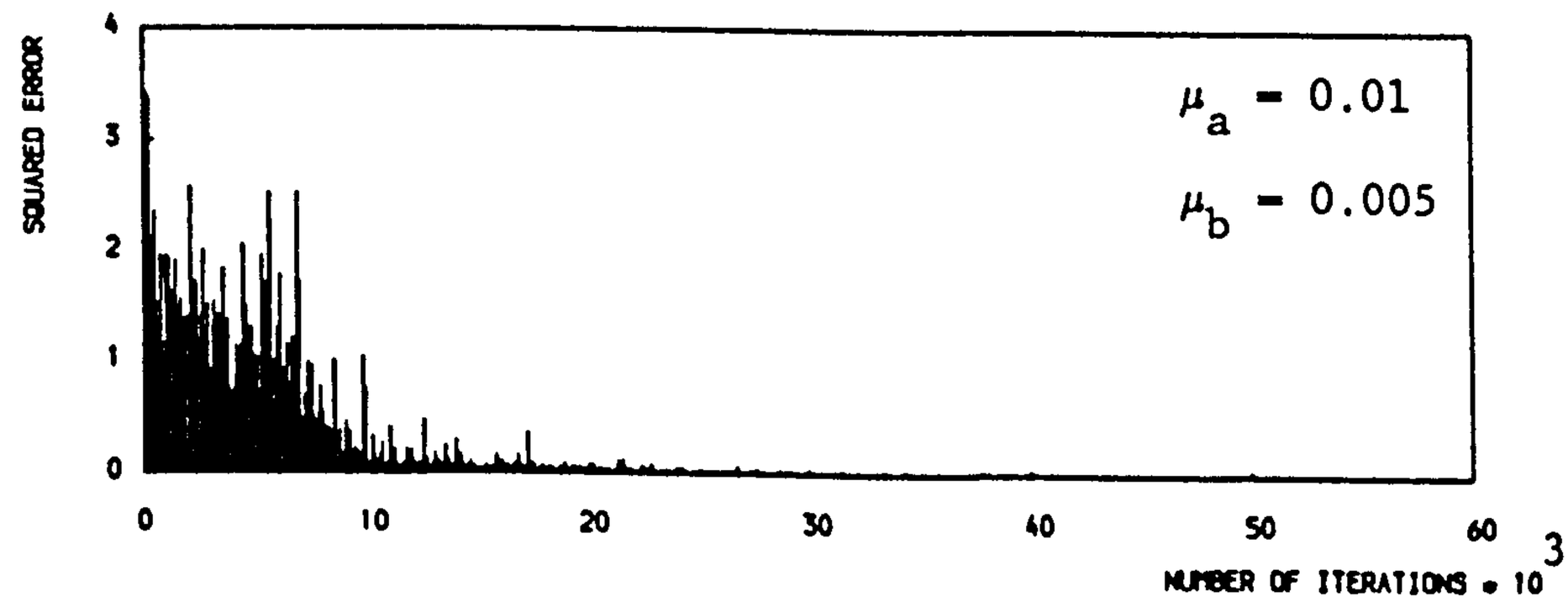
Figure 3.10 Pole Replacement Convergence Plots.



a) Biquadratic Cascade Filter Structure.



b) Modified Cascade Filter Structure.



c) Parallel Filter Structure.

Figure 3.11 Convergence Plots for Example (3).

CHAPTER 4

4.0 INCREASING THE CONVERGENCE RATES OF STEEPEST DESCENT ADAPTIVE IIR FILTERS

In Chapter 3, the rates of convergence of four gradient search Adaptive IIR Filter Structures were compared in a series of system identification experiments. Steepest descent algorithms were used to adapt the filters, and it was found that even with optimum values of the adaptation parameters μ_a and μ_b (the choice of which is an empirical process), convergence times for the different structures varied greatly and were highly dependent upon target plant characteristics. In this chapter, it is shown that the large disparity in convergence times can be attributed to the non-uniform nature of the filters' performance surfaces. A study of error surface characteristics leads to the proposal of adaptation techniques which give improved rates of convergence.

It is demonstrated in Section 4.1 that characteristics of the IIR filter performance surface make it ill-suited for adaptation using fixed values of parameters μ_a and μ_b. In Section 4.2, previously published algorithms concerned with the problem are assessed. In Sections 4.3 and 4.4, new algorithms for adapting the Modified Cascade Structure are proposed and are shown to offer faster convergence times than the steepest descent algorithm proposed in Section 3.2.2. The new algorithms are the "Valley-Search" Algorithm which employs a pattern search technique, and the "R-Cosθ" Algorithm in which the radius and angle of the adaptive filter transfer function poles are adapted independently.

4.1 Second Order All-Pole Section Adaptation.

It was shown in[12] that the performance surface with respect to the IIR Direct Form Filter coefficients is quadratic with respect to the (a_i) numerator coefficients, and irregular and possibly multimodal with respect to the (b_i) denominator coefficients. Ideally, adaptation

algorithms employing gradient techniques require a uniform performance surface, and hence adaptation of the $\{b_i\}$ coefficients will hinder the overall filter adaptation. In this section, the adaptation of a second order All-Pole filter using gradient techniques is investigated.

Typical examples of second-order adaptive filter error surfaces with respect to the $\{b_i\}$ coefficients were shown in figure 1.7, Chapter 1. Consider coefficient adaptation from an initial position of $(b_1,b_2) = (0.0,0.0)$ on each of the four performance surfaces. For the surface shown in figure 1.7a, filter adaptation using a steepest descent algorithm will give a straight line coefficient track to the minimum. However, the surfaces shown in figures 4.1b,c,d create difficulties for steepest descent algorithms as each of the surfaces is characterised by a steep valley.

To illustrate the problems created by the presence of the valleys, coefficient tracks for the (b_1,b_2) denominator coefficients modelling three of the target plants featured in figure 1.7 are given in figure 4.1a,b,c. The adaptive filter was implemented in the system identification configuration shown in figure 1.2. The input signal X(z) was gaussian white noise of unity power. In addition to the two denominator coefficients, an a_0 gain coefficient was included in the adaptive filter transfer function. The White/Stearns Adaptation Algorithm was used to adapt the filter coefficients and values of μ_a and μ_b were chosen in each example to give the fastest convergence to 30dB ERLE. The coefficient tracks shown are very noisy and in a non-ideal environment smaller values of μ would be necessary for filter convergence. In example (a), convergence to the plant was achieved in only 200 iterations and it is seen from figure 4.1a that the coefficient track to the minimum follows a straight path. The influence of the valleys is seen in the tracks shown in figures 4.1b,c. From the initial coefficients, $(b_1,b_2) = (0.0,0.0)$, the gradient vector points to the

floor of the valley well away from the minimum position. Adaptation along the valley to the minimum is slow due to the main component of the gradient vector pointing to the valley floor rather than towards the minimum. This results in the coefficients following a path which "bounces" from side-to-side across the valley. Smaller values of μ_b are required to negotiate the valleys, and convergence times are very much slower for this type of plant. The problem is magnified in the case of example (c) as the valley is narrower and steeper. Convergence to 30dB ERLE was achieved in 3300 iterations for example (b) and 170,000 iterations for example (c).

With a suitable choice of μ parameters, convergence to the error surface minimum was obtained for each example. However, for an IIR filter, the choice of μ values is empirical, unlike the choice of μ values for an adaptive FIR filter[8]. In addition, fixed values of μ parameters are not ideal for the IIR performance surface. For $\{b_i\}$ coefficient adaptation, a large distance from the error surface minimum the gradient is small, and ideally a large value of μ_b is needed. However, adaptation down the valley requires a small value of μ_b to reduce the lateral perturbations. On the valley floor the adaptation could again be slow due to the small gradients, and hence a larger value of μ_b is required[6].

A fixed value for the μ_a adaptation parameter is also undesirable. Consider, as an example, the Modified Cascade Structure proposed in Section 3.2 (the problem, however, is a general one for adaptive IIR filter structures). In the initial stages of adaptation, the poles of the adaptive filter transfer function will be situated close to the centre of the unit circle, and the All-Zero ladder filter section will have a low-power input signal and a low eigenvalue ratio in its input correlation matrix. Hence a large value of μ_a is required for the $\{a_i\}$ coefficient adaptation[8]. However, if the poles adapt to

positions near the z-plane unit circle, then the All-Zero filter section may have a high energy input signal and a high eigenvalue ratio in its input correlation matrix. A small value of μ_a is now required to adapt the $\{a_i\}$ coefficients.

The irregular nature of adaptive IIR filter performance surfaces suggest that steepest descent coefficient adaptation algorithms ideally require adaptive μ values. However, the non-uniformity of the surfaces (which depends upon the target plant characteristics) makes this a very difficult task.

4.2 MODIFIED GRADIENT ALGORITHMS

The unsuitability of the IIR filter performance surface for steepest descent coefficient adaptation has led to modified steepest descent algorithms being proposed by David/Stearns[6,55], and David[6]. In this section, an assessment of these algorithms is made with a view to implementation using the Modified Cascade Structure proposed in Chapter 3.

A series of system modelling experiments were conducted to study characteristics of the error surface with respect to the coefficients of cascaded second-order recursive sections from the Modified Cascade Structure. The results obtained suggest that the coefficient tracks for individual All-Pole sections from the cascade tend to follow paths similar to the second-order coefficient tracks shown in figure 4.1. For example, in figure 4.2 the coefficient tracks are shown for the two recursive sections in the fourth-order Cascade Filter example presented in Section 3.3.1. The coefficient tracks show the presence of valleys for both All-Pole sections when the target poles lie close to $z = -1$ in the z-plane. In Sections 4.2.1 and 4.2.2 the suitability of two algorithms for adaptation of the Modified Cascade Filter Coefficients is considered.

4.2.1 Smoothed Gradient Directional Algorithm

For an adaptive filter, it has been shown[55] that the instantaneous squared error ($e^2(k)$) is an unbiased estimate of the mean-square-error. Therefore, by averaging gradient vectors over a number of iterations, a more accurate estimate of the true gradient is obtained. David and Stearns utilized this factor with a smoothed gradient directional algorithm in[6,55] which used a fixed stepsize in the direction of the averaged gradient to adapt the filter coefficients of a Second-Order Direct Form Filter. Hence, the algorithm does not require the μ_a and μ_b adaptation parameters.

The coefficient updates can be expressed as:

$$a_i(k+N) = a_i(k) + \frac{\Delta.(e\alpha_i)_{AV}}{\sqrt{(\lambda_{AV}\ \lambda_{AV}^T)}} \qquad i = 0,1,2 \tag{4.1}$$

$$b_i(k+N) = b_i(k) + \frac{\Delta.(e\beta_i)_{AV}}{\sqrt{(\lambda_{AV}\ \lambda_{AV}^T)}} \qquad i = 1,2 \tag{4.2}$$

Where the α_i and β_i terms are generated using the Hsia/Horvath algorithm,

$$\lambda_{AV} = [(e\alpha_0)_{AV},(e\alpha_1)_{AV},(e\alpha_2)_{AV},(e\beta_1)_{AV},(e\beta_2)_{AV}] \tag{4.3}$$

$$(e\gamma_i)_{AV} = \frac{1}{N}\sum_{j=k+1}^{k+N} e(j)\gamma_i(j) \qquad \gamma = \alpha,\beta \tag{4.4}$$

and Δ determines the stepsize. In a simple version of the algorithm, a fixed value of stepsize is chosen. Alternatively, the value of Δ may be changed from a large initial value, to a small final value for

convergence to the minimum position. The value of Δ chosen for convergence will depend upon whether the target system is stationary, and also the sensitivity of the adaptive filter coefficient values.

The gradient vector used in the coefficient updates is found by averaging the gradients over N iterations (equation (4.4)). A large value of N will give smoother coefficient tracks but slower adaptation. In addition to the stepsize Δ, the value of N may also be varied during adaptation. In the early stages of adaptation the coefficients are varying rapidly and a large value of N is required. As the coefficients near convergence, the coefficient update steps contain less noise and N can be reduced in size.

David published a number of second-order adaptive filter examples which showed that the Smoothed Gradient Directional Algorithm offers faster convergence than the White/Stearns algorithm. However, problems occur when the $\{b_i(k)\}$ coefficients are adapting to pole positions near either $z = 1$ or $z = -1$ in the z-plane. If the adaptive filter coefficients give rise to a point situated on the side of a steep valley in the error surface, averaging gradients will not be particularly beneficial for filter adaptation. The predominant gradient vector will point to the floor of the valley, rather than towards the minimum position. By averaging gradients and taking a large step in the direction calculated, the filter will jump to the opposite side of the valley. The process will then repeat itself.

Hence, adaptation times for the algorithm will be slowed considerably when the error surface contains a steep valley. In addition, while the Smoothed Gradient Directional Algorithm removes the necessity for a choice of μ parameters, the choice of stepsize Δ is itself an empirical decision governed by the target system characteristics.

4.2.2 Normalized RLMS, CRLMS Algorithms

David[6] proposed a modification for steepest descent algorithms using a simplified version of the second-order Newton Method. The algorithm uses a normalizing factor to take small adaptation steps when the gradient is changing rapidly. The modification was applied to the Hsia/Horvath RLMS algorithm and to the David Cascade RLMS (CRLMS) algorithm[6], giving rise to the NRLMS and NCRLMS algorithms.

The LMS version of Newton's Algorithm is given by:

$$\underline{W}_{k+1} = \underline{W}_k + \underline{\mu}.H^{-1}.\underline{g} \tag{4.5}$$

Where $\underline{W}_k$ and $\underline{\mu}$ are defined for equation (2.1), and H and $\underline{g}$ are obtained from:

$$H = \frac{\partial^2 \overline{e^2(k)}}{\partial \underline{W}_k^{\,2}} \tag{4.6}$$

$$\underline{g} = \frac{\partial\ \overline{e^2(k)}}{\partial \underline{W}_k} \tag{4.7}$$

$\underline{g}$ is the vector of first-order derivatives utilized by steepest descent algorithms and H is a square matrix of second-order derivatives. David used the Hsia/Horvath algorithm to calculate the terms in $\underline{g}$, and a scalar estimate of the trace of H as a crude approximation to the power contained in the matrix. The estimate is obtained from[6]:

$$\mathrm{Tr}\ H \simeq \psi_k^T\ \psi \tag{4.8}$$

Generation of the ψ_k term differs for the NRLMS and NCRLMS algorithms.

a) For the Normalized Hsia/Horvath algorithm, ψ_k is obtained from:

$$\psi_k^T = [\alpha_0(k), \ldots, \alpha_n(k), \beta_1(k), \ldots, \beta_m(k)] \tag{4.9}$$

Where the $\alpha_i(k)$ and $\beta_i(k)$ terms are generated using equations (2.15) and (2.16). The coefficient updates for the NRLMS algorithm are given by:

$$a_i(k+1) = a_i(k) + \frac{\mu_a}{\psi_k^T \psi_k} .e(k).\alpha_i(k) \tag{4.10}$$

$$i = 0,1,..,n$$

$$b_i(k+1) = b_i(k) + \frac{\mu_b}{\psi_k^T \psi_k} .e(k).\beta_i(k) \tag{4.11}$$

$$i = 1,2,..,m$$

b) For the Normalised Cascade RLMS algorithm, ψ_k is obtained from:

$$\psi_k^T = ([\alpha^1,\beta^1],[\alpha^2,\beta^2],.....,[\alpha^n,\beta^n]) \tag{4.12}$$

Where:

$$[\alpha^j,\beta^j] = [\alpha_{1j}(k),\alpha_{2j}(k),\beta_{1j}(k),\beta_{2j}(k)] \tag{4.13}$$

$$j = 1,n-1$$

$$[\alpha^n,\beta^n] = [\alpha_{0n}(k),\alpha_{1n}(k),\alpha_{2n}(k),\beta_{1n}(k),\beta_{2n}(k)] \tag{4.14}$$

Using equation (4.12), the coefficient updates for the NCRLMS algorithm are calculated using:

$$a_{ij}(k+1) = a_{ij}(k) + \frac{\mu_a}{\psi_k^T \psi_k} .e(k).\alpha_{ij}(k) \tag{4.15}$$

$$\begin{cases} j=1,n-1 \text{ , } i=1,2 \\ j=n \text{ , } i=0,1,2 \end{cases}$$

$$b_{ij}(k+1) = b_{ij}(k) + \frac{\mu_b}{\psi_k^T \psi_k} .e(k).\beta_{ij}(k) \tag{4.16}$$

$$\begin{cases} i=1,2 \\ j=1,n \end{cases}$$

The values of $\alpha_{ij}(k)$ and $\beta_{ij}(k)$ can be generated using equations (3.19),(3.20) and (3.22).

The NCRLMS Algorithm can be applied to the Modified Cascade Structure by setting:

$$\alpha^j = 0 \qquad j = 1, m-1 \tag{4.17}$$

$$[\alpha^m, \beta^m] = [\alpha_0(k), \ldots, \alpha_n(k), \beta_{1m}(k), \beta_{2m}(k)] \tag{4.18}$$

in equation (4.12). The $\alpha_i(k)$ and $\beta_{ij}(k)$ terms are generated using equations (3.29) and (3.30).

Examples were presented in[6] to demonstrate that the Normalized algorithms achieve faster convergence times than Steepest Descent algorithms. Figures 4.3a and 4.3b show coefficient tracks for a second-order adaptive All-Pole section adapting to the target plants from examples (b) and (c) in Section 4.1. Convergence to 30dB ERLE was obtained in 800 iterations for example (b), and 25,000 iterations for example (c). Comparing convergence times with the White RLMS Algorithm (figures 4.1b and c), it is seen that with optimum choice of μ parameters in each case, the NRLMS algorithm gives much faster convergence in both examples. However, convergence times for the NRLMS algorithm were found to be very erratic due to the large coefficient changes in the early stages of the adaptation. This occurs because the error surface gradients are small and are only varying slowly. In fact, it is necessary to include a test in the algorithm for divide-by-zero errors which may occasionally occur. Comparing the coefficient tracks shown in figures 4.1 and 4.3, it is seen that the tracks for the RLMS algorithm are smooth in the earlier stages of adaptation and noisy near the error surface minimum, whereas the NRLMS algorithm tracks are smooth near the minimum and noisy at the start of the adaptation.

The optimum choice of μ values for the NRLMS algorithm is empirical, although a smaller range of μ values is required than is needed for RLMS algorithms. Figure 4.4 shows a coefficient track for the NRLMS algorithm adapting to the target plant in example (b), Section 4.1, with the μ values used in example (c). Convergence is now considerably slower than that obtained with optimum μ values and 30dB ERLE was obtained in 9000 iterations. The coefficient track is much smoother, however.

From the results presented for both RLMS and NRLMS algorithms, it is seen that in a large number of iterations, the adaptive filter coefficients jump from side-to-side across valleys in the error surface, rather than moving towards the surface minimum position. The Smoothed Gradient Directional Algorithm discussed in Section 4.2.1 also suffers from the same problem. Ideally, the trend of the valleys should be recognized by the adaptation algorithm and coefficient steps taken along the valley floor.

4.3 VALLEY-SEARCH ALGORITHM

The adaptation algorithms considered in Sections 4.1 and 4.2 demonstrate the slower convergence of gradient-based algorithms to target poles located near either $z = 1$ or $z = -1$ in the z-plane. Such pole positions will have error surfaces characterised by steep valleys. The Valley-Search Algorithm employs a pattern search technique which initially locates the centre of the valley, and then takes a normalized coefficient step in the direction of the minimum. The algorithm consists of three modes of adaptation; Gradient-Search, Cross-Sectional Search, and a Normalized Coefficient Step. The three modes of the adaptation for a single second-order recursive filter section are described on the following page.

(1) Gradient-Search Adaptation

In this mode, the adaptive filter uses a gradient algorithm to modify the filter coefficients.

(2) Cross-Sectional Search

By adapting the b_1 filter coefficient alone (keeping the b_2 coefficient stationary), a gradient algorithm is used to locate the floor of the error surface valley.

(3) Normalized Coefficient Step

A Normalized Coefficient Step is made in the direction of the minimum by finding the trend of the valley using steps (1) and (2).

The adaptation of a second-order recursive filter section using the Valley-Search Algorithm is illustrated in figure 4.5. In the first stage of the adaptation, the filter is adapted from initial coefficients $(b_1,b_2) = (0.0,0.0)$ using step (1) of the adaptation algorithm (A → B in figure 4.5). The second phase of the adaptation involves holding the b_2 coefficient stationary and locating the floor of the valley by adapting the b_1 coefficient in a Cross-Sectional Search (B → C). The coefficients of point C are stored, and will be used later to direct the Normalized Coefficient Step. From point C, the filter is adapted using step (1) of the algorithm to find in which direction the error surface minimum lies with respect to the b_2 coefficient (C → D).

It is seen from figures 1.7 and 4.1 that the error surface valleys lie in directions which approximately radiate from the bottom point of the stability triangle ($(b_1,b_2) = (0.0,-1.0)$). This point (E) is used as an initial point for directing the first Normalized Coefficient Step. The coefficient step is taken in the direction E → C from point C, moving the filter coefficients to point F. If the minimum

lies in a negative direction with respect to the b_2 coefficient, then the step taken is halved in size, a Cross-Sectional Search is made using step (2), and then the filter is adapted using step (1) until convergence is achieved.

If point F lies in a positive direction with respect to the b_2 coefficient, a Cross-Sectional Search is again made using step (2) (F → G), and the direction of the minimum found by step (1), (G → H). A Normalized Coefficient Step is then made in the direction (C → G) from point G, taking the filter to point I. The algorithm makes use of the fact that points C and G each lie on the floor of the valley, and can therefore give a good indication of the direction of the error surface minimum for a stationary, or slowly varying target system.

The filter adaptation progresses with the three steps being ordered (2), (1), (3). The coefficients resulting from the current and previous adaptation cycle's Cross-Sectional Search Phases are used to direct the Normalized Coefficient Step. Adaptation continues until the Normalized Coefficient Step (Step (3)) of the adaptation cycle is in a negative direction with respect to the b_2 coefficient. When this occurs, the stepsize is halved, a Cross-Sectional Search made, and the filter is adapted until convergence is achieved.

In computer simulations of the algorithm, the transients caused by the Normalized Coefficient Step was ignored as the following phase of adaptation was a Cross-Sectional Search and noise introduced into the algorithm was not found to be significant. The NRLMS gradient algorithm was used for adaptation modes (1) and (2) in preference to a RLMS algorithm. However, the NRLMS algorithm adapts more slowly as the filter negotiates an error surface valley. This is due to the error signal being reduced in power and the $\beta_i(k)$ terms varying rapidly. Hence, it is necessary to increase the number of iterations used for step (1) of the algorithm for each successive adaptation cycle. If the Normalized

Coefficient Step takes the filter unstable, the stepsize can be reduced in size until a stable filter is reached.

Convergence plots of the filter coefficients modelling the two target plants used in Section 4.1, examples (b) and (c) are given in figures 4.6a,b. The adaptive filter and the target plants were second-order All-pole filters in each example. In addition to the (b_1,b_2) coefficients, an a_0 gain coefficient was also included in the adaptive filter transfer function. This coefficient was adapted using the NRLMS algorithm for steps (1) and (2) of the adaptation algorithm.

In the examples presented, 750 iterations were used for step (2) of the algorithm and a Normalized Coefficient Step of size 0.5. 400 iterations were used initially for step (1), with the number of iterations increasing by 100 each adaptation cycle. The RLMS algorithm was used for the final run for convergence of the filter using the optimum values of μ from examples (b) and (c), Section 4.1. The RLMS algorithm is used because of the poor rates of convergence of the NRLMS algorithm when close to the error surface minimum. The slow convergence is due to the low power of the error signal, and the rapidly varying gradient vector. With the size of Normalized Coefficient Step used for the examples, after five Cross-Sectional Search adaptation phases, the adaptive filter will be very close to the upper corners of the stability triangle. If this occurs, the filter must be close to the error surface minimum and is run for convergence using step (1).

The Valley-Step algorithm achieved convergence to 30dB ERLE in 6,400 and 10,000 iterations for the two examples. Comparing these results with those for the NRLMS algorithm in Section 4.2, it is seen that the Valley-Search algorithm obtained the fastest convergence for the second example, but not the first. This is as expected as the advantages to be gained using this type of algorithm will be greater for target pole positions in the upper two corners of the stability

triangle. The values of μ chosen for the NRLMS algorithm were the same for each example. The size of the Normalized Coefficient Step controls the speed of adaptation and the NRLMS algorithm is only used to find the direction for the step. Therefore, the choice of μ_b values can be made independently of the target system characteristics, unlike the algorithms considered previously in this chapter.

The Valley-Search algorithm is limited in the type of applications for which it can be used, as it is not suited for filter adaptation in a non-stationary environment and is computationally demanding for higher order filters. In addition, error surfaces arising from insufficient order adaptive filters have not been considered. However, by using the NRLMS algorithm to find the direction in which to take the Normalized Coefficient Step, the Valley-Search algorithm offers fast convergence to the approximate location of a stationary minimum without a target plant-dependent choice of μ_b (the value of μ_b will depend only on properties of the input signal). Unfortunately, the optimum choice of μ_a value is still empirical. In addition, the choice of μ parameters required for convergence to the minimum is empirical for both NRLMS and RLMS algorithms, and hence also for the Valley-Search algorithm.

4.4 R-COSθ ALGORITHM

An alternative approach which may be taken in order to reduce adaptation times for cascaded All-pole second-order filter sections, is to modify the structure of each section,so allowing the radius and angle of the poles to be adapted independently. The filter structure of a R-Cosθ Second Order All-Pole section is shown in figure 4.7. For the Direct Form second-order pole section, the relationship between the filter coefficients and the (r,θ) pole positions in the unit circle is given by equations (1.20) and (1.21) for complex conjugate pole pairs.

For the R-Cosθ filter section, the relationship is:

$$b'_1 = -2\cos\theta \tag{4.19}$$

$$b'_2 = r \tag{4.20}$$

Hence, adaptation of the b'_1 coefficient will change $\cos\theta$, and adaptation of b'_2 will change the radius r of the poles. A cascade structure of R-Cosθ recursive filter sections is therefore only able to synthesize complex conjugate pole pairs and not split real poles. Near z = 1 and z = -1 in the z-plane, small changes in Cosθ give large changes in θ, and hence decoupling r and $\cos\theta$ should offer better convergence performance to pole positions in these areas. One additional advantage of using a cascade structure of recursive R-Cosθ second-order filter sections is the ease of maintaining filter stability. It is only necessary to check that $|b'_2| < 1$ and $|b'_1| < 2$ for each filter section, which is a simpler test than that required for the general second-order section filters proposed in Chapter 3. A Steepest Descent algorithm for adaptation of an R-Cosθ All-Zero cascade filter was shown to give better performance than the general second-order filter cascade structure when used for speech prediction in[56,57]. A Steepest Descent algorithm for adaptation of a Modified Cascade IIR Filter incorporating R-Cosθ recursive filter sections and a Direct-Form All-Zero section is proposed in the following section.

4.41 R-Cosθ Adaptation Algorithm Derivation.

The modified cascade filter structure is shown in figure 3.8. The filter has transfer function $\hat{H}(z)$ where:

$$\hat{H}(z) = \frac{\hat{A}(z)}{\hat{B}'_1(z).\hat{B}'_2(z)\ldots\hat{B}'_m(z)} \tag{4.21}$$

Incorporating R-Cosθ recursive sections, the individual filter transfer functions are given by:

$$\hat{A}(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_n z^{-n} \tag{4.22}$$

$$\hat{B}'_j(z) = 1 + b'_{1j} b'_{2j} z^{-1} + b'_{2j} z^{-2} \tag{4.23}$$

For a Steepest Descent algorithm, the coefficients are updated using equations (3.31) and (3.32) where the $\alpha_i(k)$ and $\beta_{ij}(k)$ terms are defined in equations (3.16) and (3.17). The $\alpha_i(k)$ terms for the coefficient updates are obtained from equation (3.29). As for the Modified Cascade Structure from Chapter 3, the $\alpha_i(k)$ terms are generated within the delay line of the All-Zero Section, and require no further computation. Using equations (4.21), (4.22) and (4.23), the $\beta_{ij}(k)$ terms are generated from:

$$\beta_{1j}(z) = \frac{-Y(z).b'_{2j} z^{-1}}{\hat{B}'_j(z)} \tag{4.24}$$

$$\beta_{2j}(z) = \frac{-Y(z).(b'_{1j} z^{-1} + 2b'_{2j} z^{-2})}{\hat{B}'_j(z)} \tag{4.25}$$

$$j = 1,2,..,m$$

The complete adaptive filter structure filter structure is shown in figure 4.8. To retain filter stability, it is necessary to restrict $|b'_{2j}| < 1$ and $|b'_{1j}| < 2$. As the $\beta_{ij}(k)$ gradient term calculating filters are driven with the same input signal, the All-Pole filter sections must be adapted from different initial coefficients otherwise filter sections will remain locked. In addition, the b'_1 and b'_2 coefficients from an individual section should not both be started from zero, otherwise the $\beta_{ij}(k)$ terms will also be zero and the section will not adapt.

4.42 Performance Assessment

To assess the performance of the R-Cosθ algorithm, the System Identification configuration shown in figure 1.2 was used. Two second-order All-Pole adaptive filter examples and a fourth-order adaptive filter example are presented. Coefficient tracks for the two second-order examples are given in figure 4.9a,b. The $\acute{b}_1,\acute{b}_2$ R-Cosθ filter coefficients are converted into the corresponding Direct Form structure b_1,b_2 coefficients for the plots. The relationship between the coefficients is given by:

$$b_1 = \acute{b}_1.\acute{b}_2 \tag{4.26}$$

$$b_2 = \acute{b}_2^{\,2} \tag{4.27}$$

The target models were those used to assess the other algorithms in this chapter, ie. examples (b) and (c) in Section 4.1. Convergence to 30dB ERLE was obtained in 800 iterations for the plant from example (b), and 34,000 iterations for the plant from example (c). These results were obtained using optimum values of μ parameters and the initial denominator coefficients were set at $(\acute{b}_1,\acute{b}_2) = (2.0,0.0)$. Comparing the coefficient tracks in figure 4.9 with figures 1.6 and 4.1, it is seen that the tracks for the R-Cosθ algorithm follow the parabola of real pole pairs until the error surface valley is reached, and then adapt along the valley to the minimum.

For a fourth order example, the target plant from Example 3, Section 3.3.3 was used. The transfer function of the plant is given in equation 3.60 and the R-Cosθ Filter was of sufficient size to fully model the plant. A Convergence Plot for the algorithm is shown in figure 4.10. The R-Cosθ modified cascade filter took 19,000 iterations for convergence to 30dB ERLE. The initial coefficients of the recursive sections were $(\acute{b}_{11},\acute{b}_{21}) = (-2.0,0.1)$, $(\acute{b}_{12},\acute{b}_{22}) = (2.0,0.1)$. The Modified Cascade Structure of Chapter 3 took 52000 iterations to adapt to this plant. Hence, the R-Cosθ algorithm shows a large improvement.

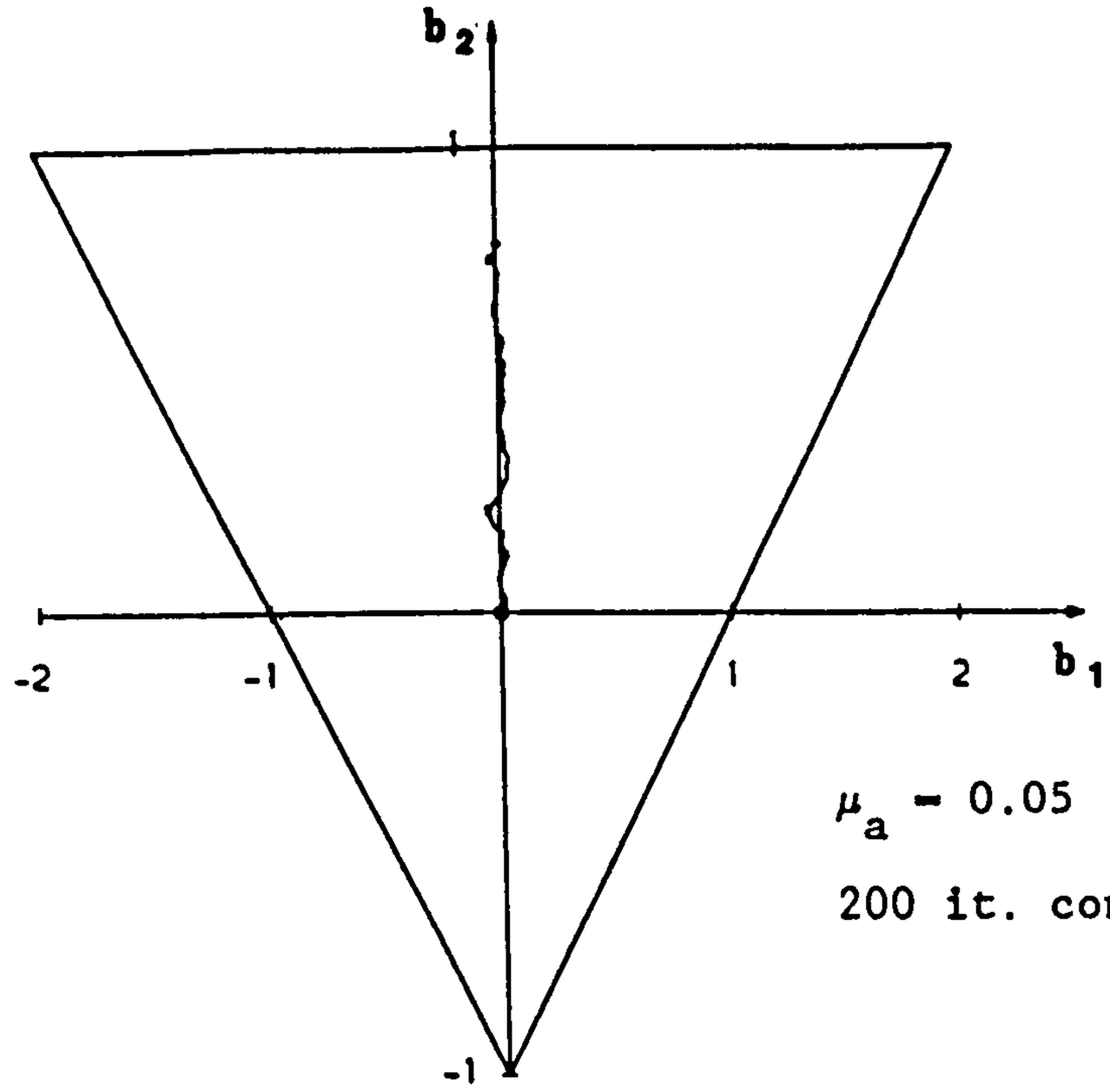
4.5 CONCLUSIONS

In this chapter, characteristics of the IIR filter performance surface have been investigated with a view to improving the rates of convergence of Gradient-based adaptation algorithms. Two previously published algorithms were investigated (Smoothed Gradient Directional and Normalized RLMS Algorithms), and two new gradient algorithms have been proposed for the adaptation of the Modified Cascade Filter structure proposed in Chapter 3 (Valley-Search and R-Cosθ algorithms).

All the algorithms investigated have problems with the choice of μ values, or the size of Δ in the case of the Smoothed Gradient Directional Algorithm. The size of adaptation parameters required is governed by the sensitivity of the final coefficient positions and characteristics of the target system. This is a major disadvantage for adaptive IIR Filter algorithms. The NRLMS and CRLMS algorithms have adaptive μ values and require a smaller variation in μ values than RLMS algorithms. However, although the optimum adaptation times achieved by the NRLMS algorithm are much faster than the corresponding RLMS algorithm results, convergence times can still vary considerably.

Using the Valley-Search algorithm, results have shown that convergence close to the error surface minimum could be obtained with a choice of μ values which is independent of the target system. However, the choice of μ required for convergence is still empirical and depends on the target system characteristics. In addition, the algorithm requires a considerable amount of computation and implementation is difficult for higher order filters.

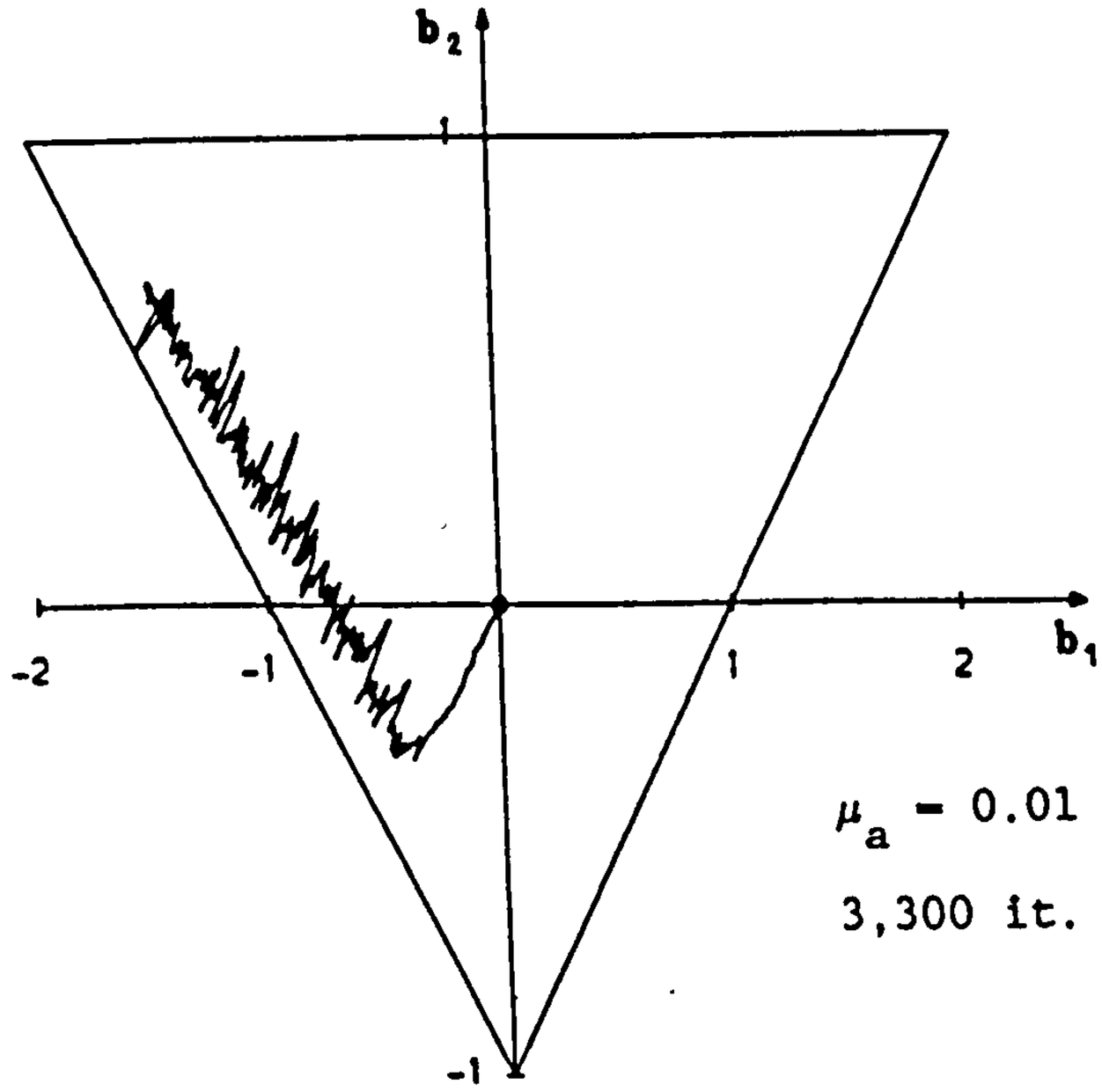
The R-Cosθ algorithm results were very encouraging when compared with the steepest descent algorithms of Chapter 3. However, split real poles cannot be synthesized by the filter structure, which reduces the generality of this approach.



μ_a - 0.05 μ_b - 0.01

200 it. convergence.

a) Target Plant Coefficients: (a_0,b_1,b_2) - (1.0,0.0,0.8)

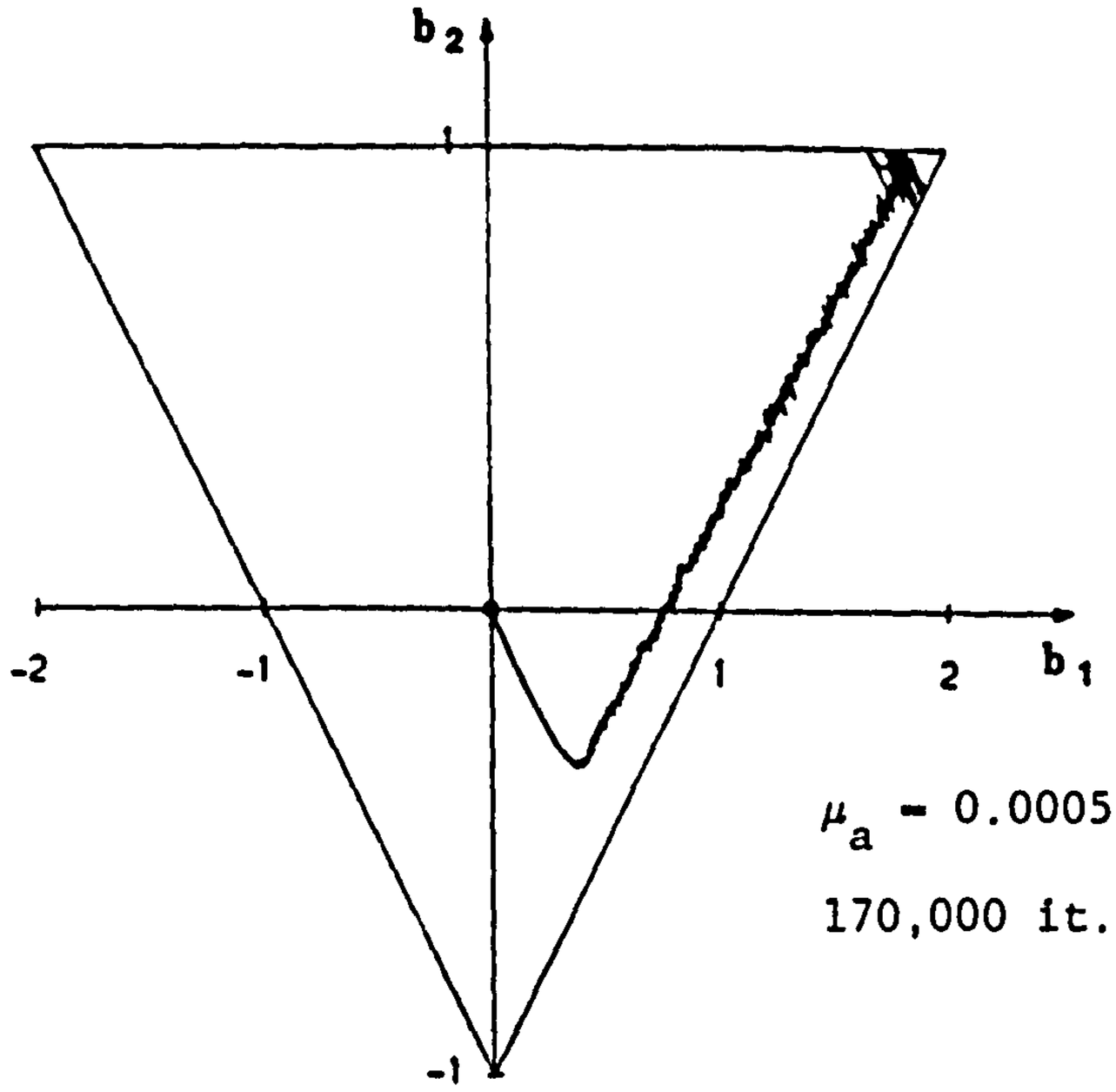


μ_a - 0.01 μ_b - 0.0005

3,300 it. convergence

b) Target Plant Coefficients: (a_0,b_1,b_2) - (1.0,-1.5,0.7)

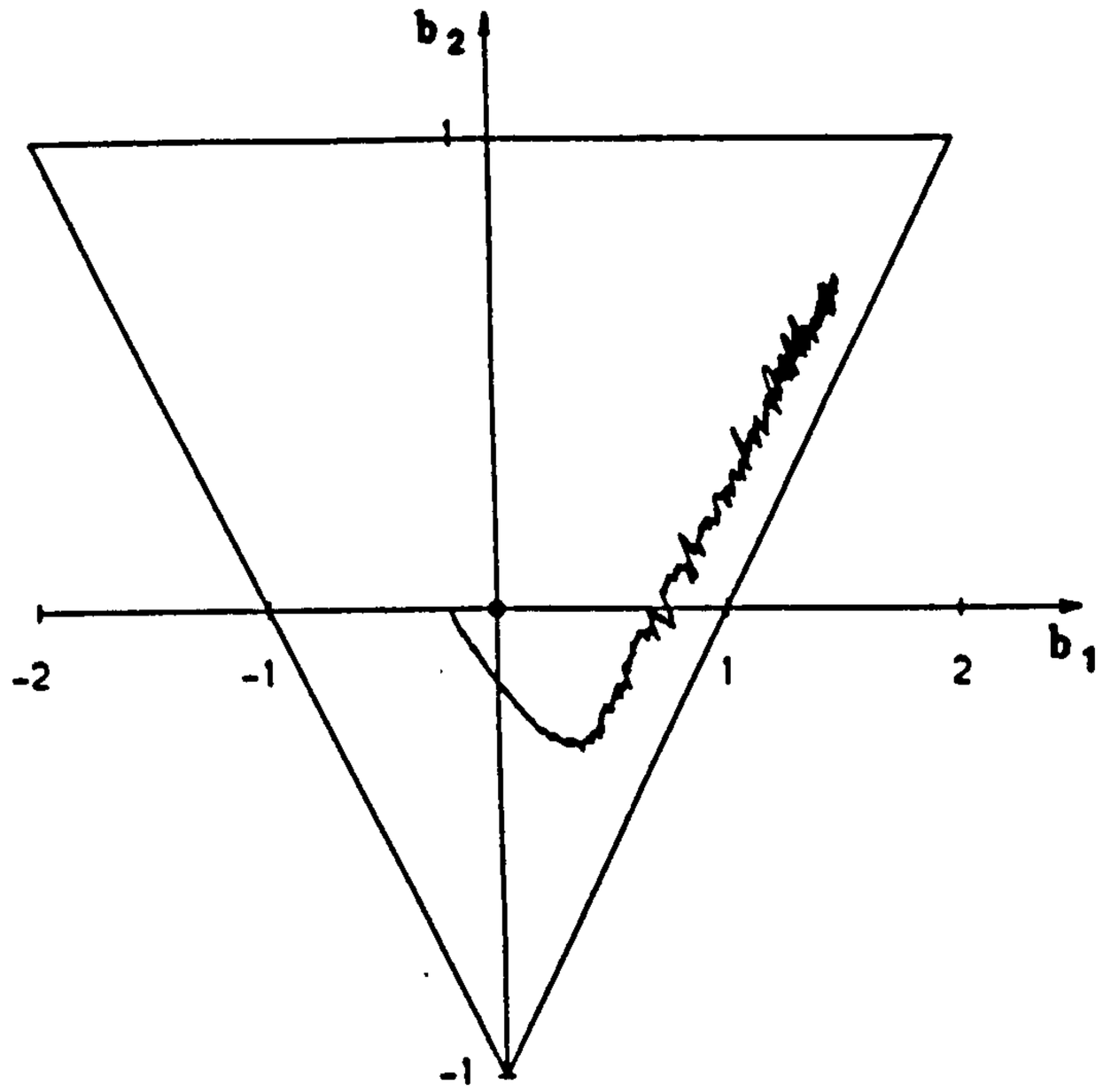
Figure 4.1 (b_1,b_2) Coefficient Tracks for the White/Stearns Algorithm.
Initial Filter Coefficients: (a_0,b_1,b_2) - (0.0,0.0,0.0)



μ_a = 0.0005 μ_b = 0.000003

170,000 it. convergence

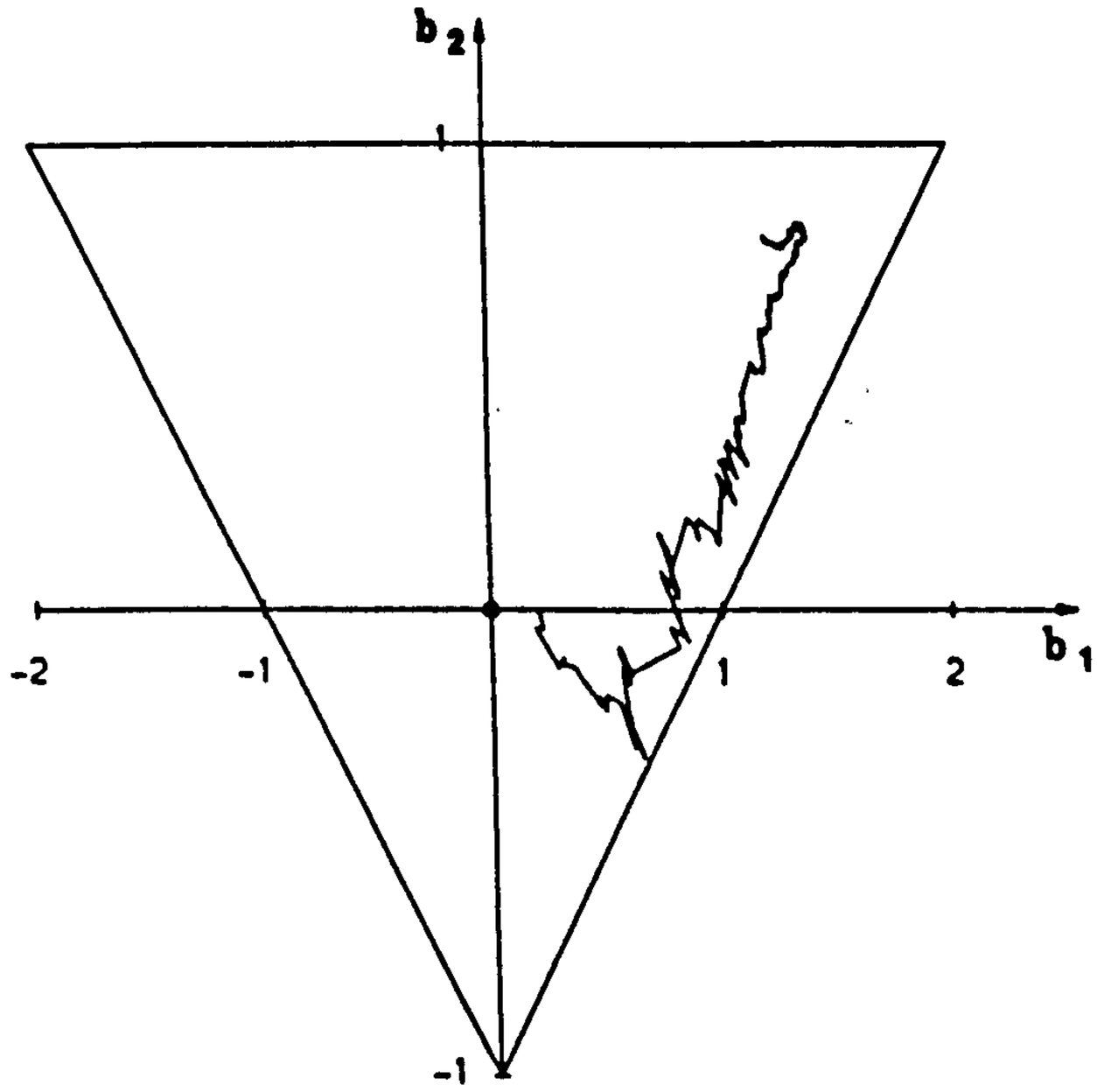
c) Target Plant Coefficients: (a_0, b_1, b_2) = (1.0, 1.8, 0.95)



Filter Section (1)

Initial Coefficients (b_{11},b_{21}) = (-0.2,0.0)

Target Plant Coefficients: (b_1,b_2) = (1.5,0.7)

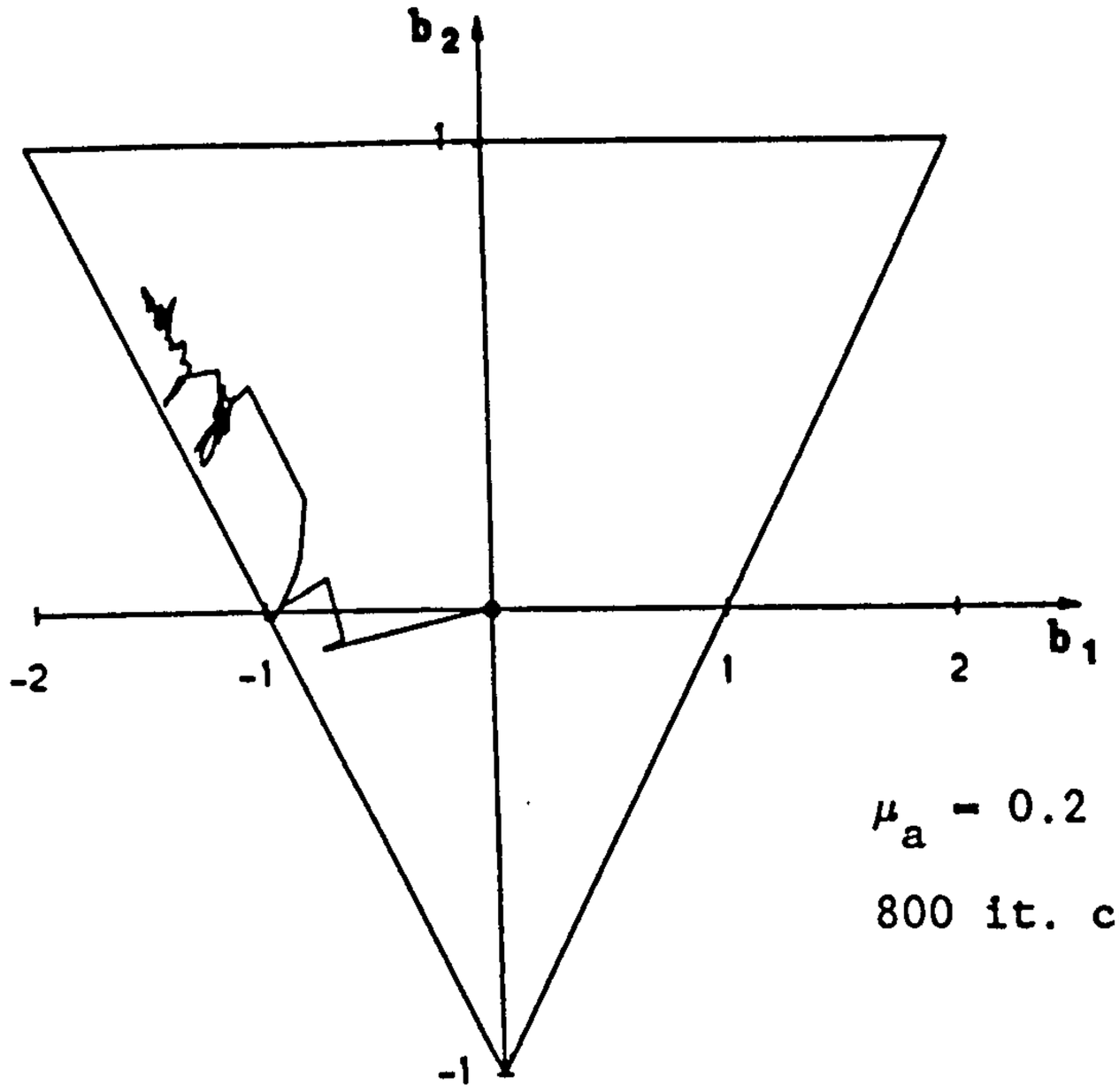


Filter Section (2)

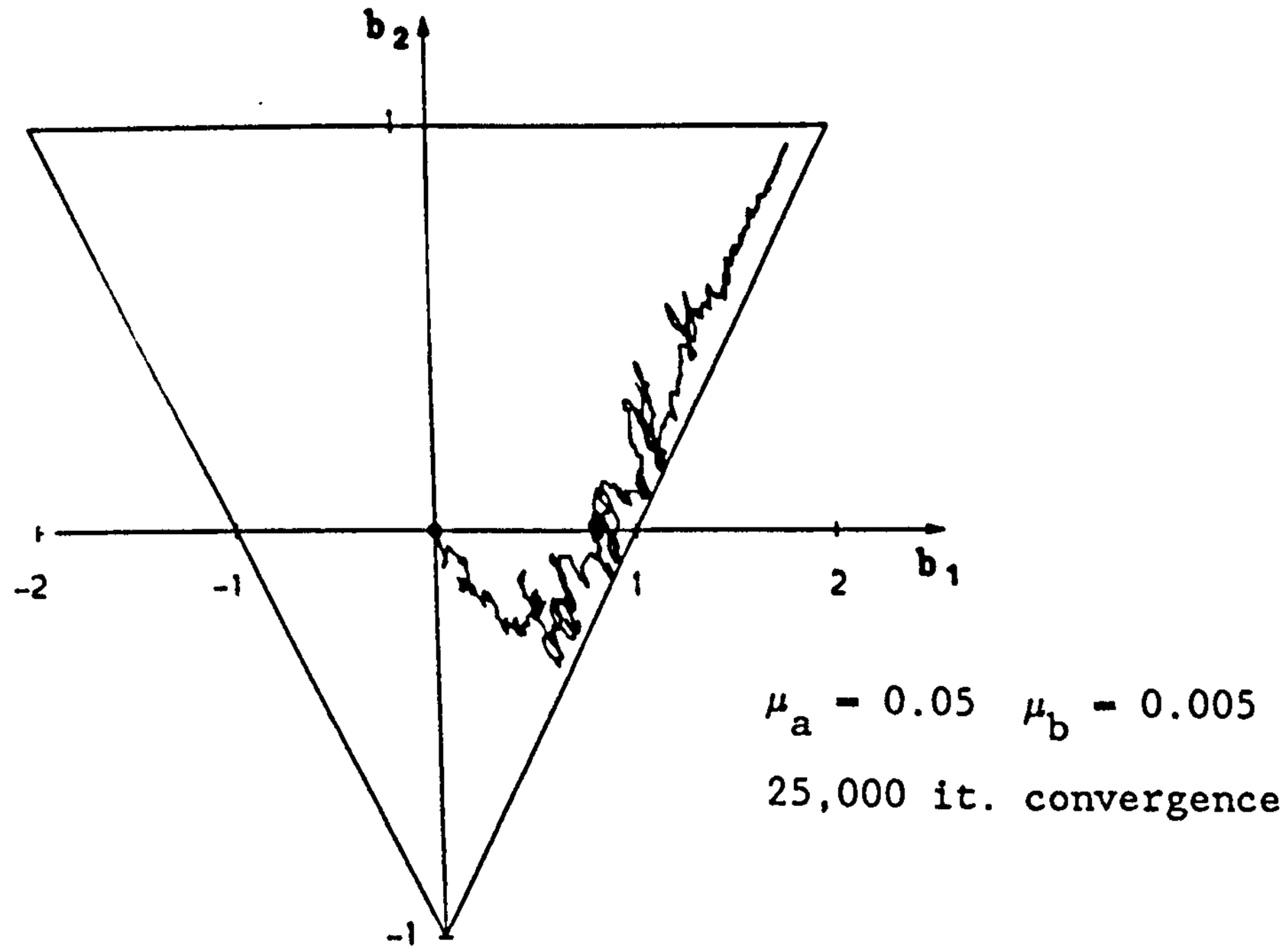
Initial Coefficients (b_{12},b_{22}) = (0.2,0.0)

Target Plant Coefficients: (b_1,b_2) = (1.2,0.8)

Figure 4.2 (b_{1j},b_{2j}) Coefficient Tracks for Cascaded All-Pole Filter Sections adapting to the fourth-order plant from Section 3.3.1.



a) Target Plant Coefficients: $(a_0,b_1,b_2) = (1.0,-1.5,0.7)$



b) Target Plant Coefficients: $(a_0,b_1,b_2) = (1.0,1.8,0.95)$

Figure 4.3 (b_1,b_2) Coefficient Tracks for the NRLMS Algorithm.
Initial Filter Coefficients: $(a_0,b_1,b_2) = (0.0,0.0,0.0)$

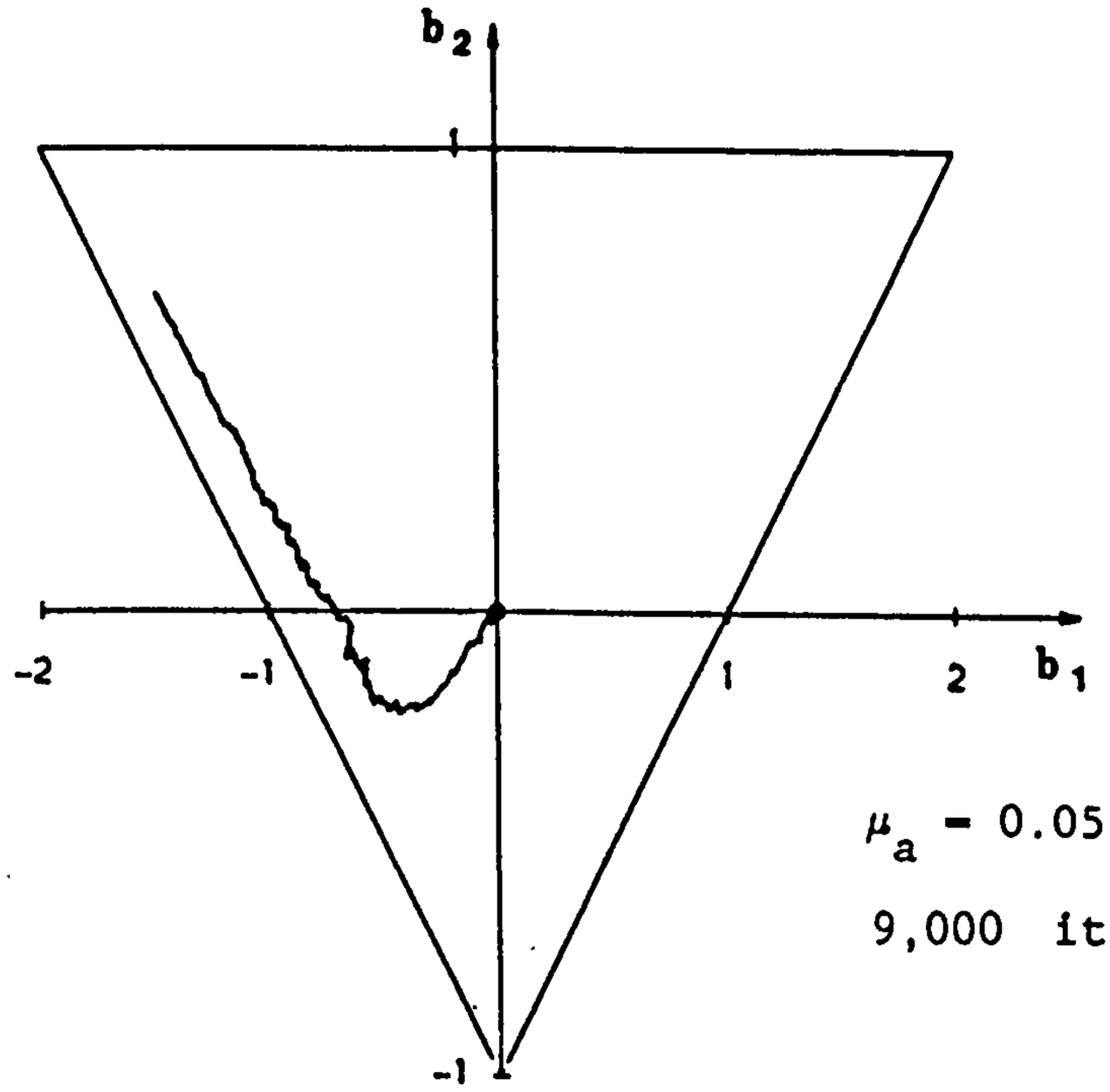


Figure 4.4 (b_1,b_2) Coefficient Track for the NRLMS Algorithm with non-optimum μ_b value.

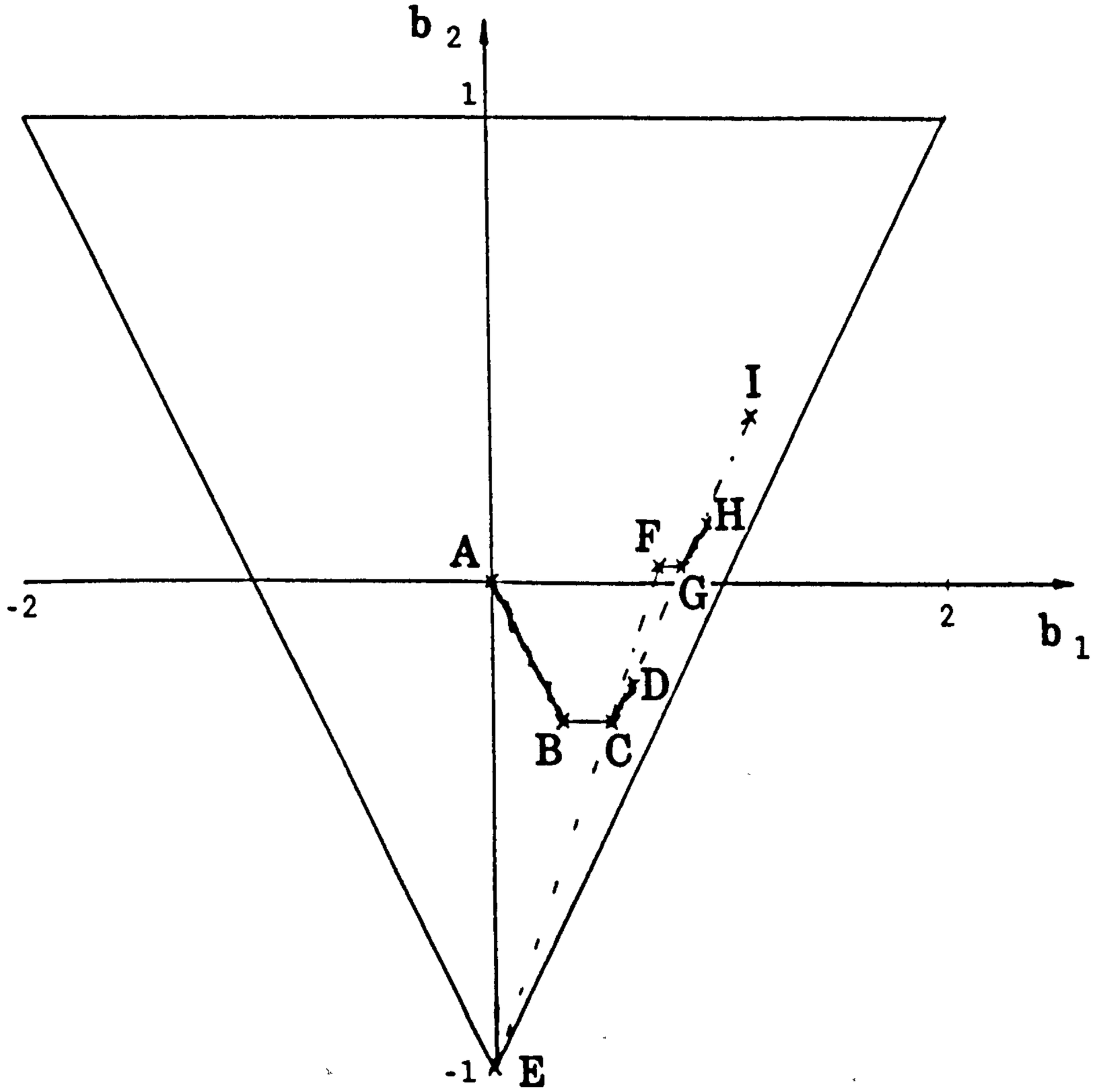
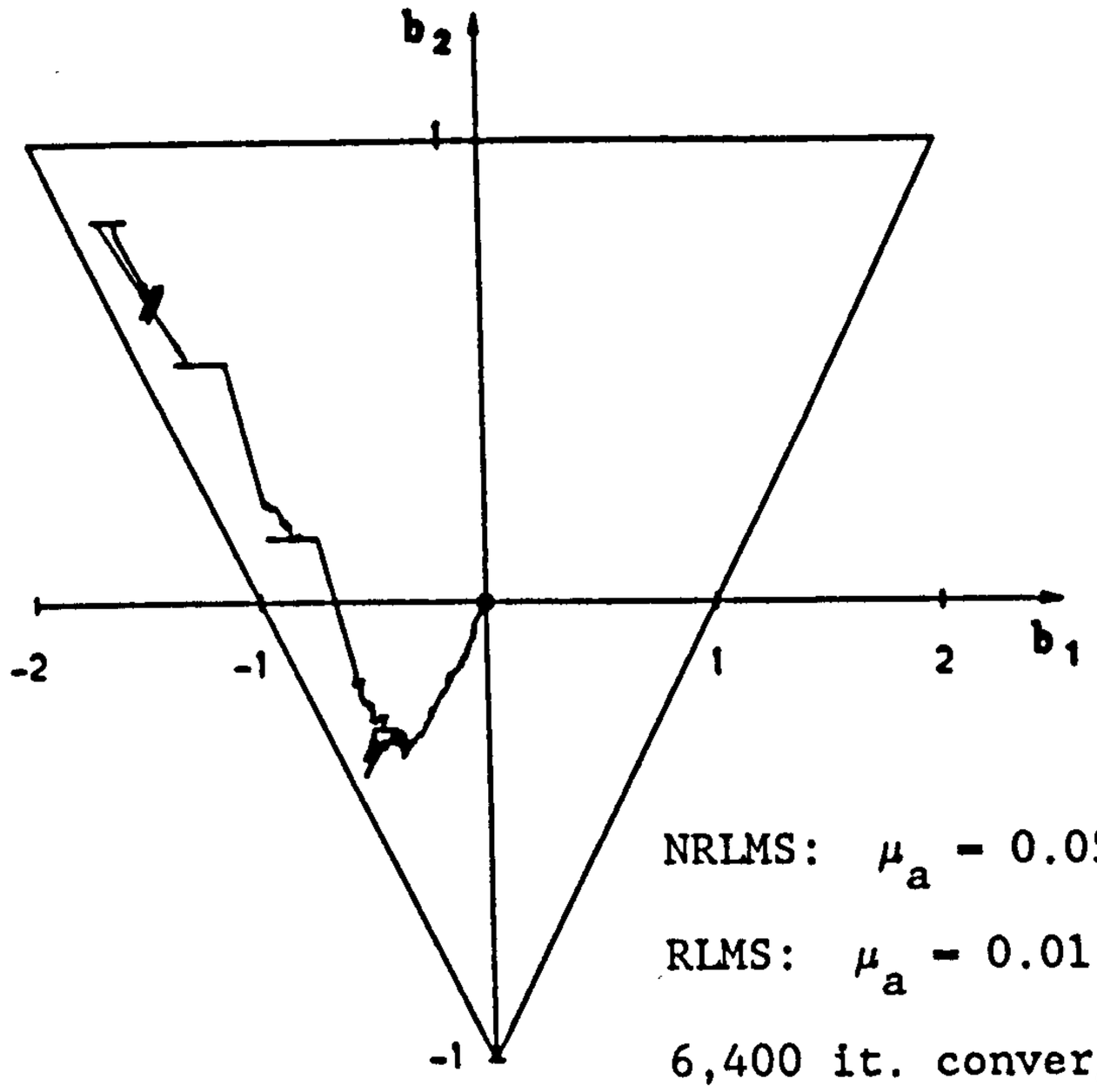
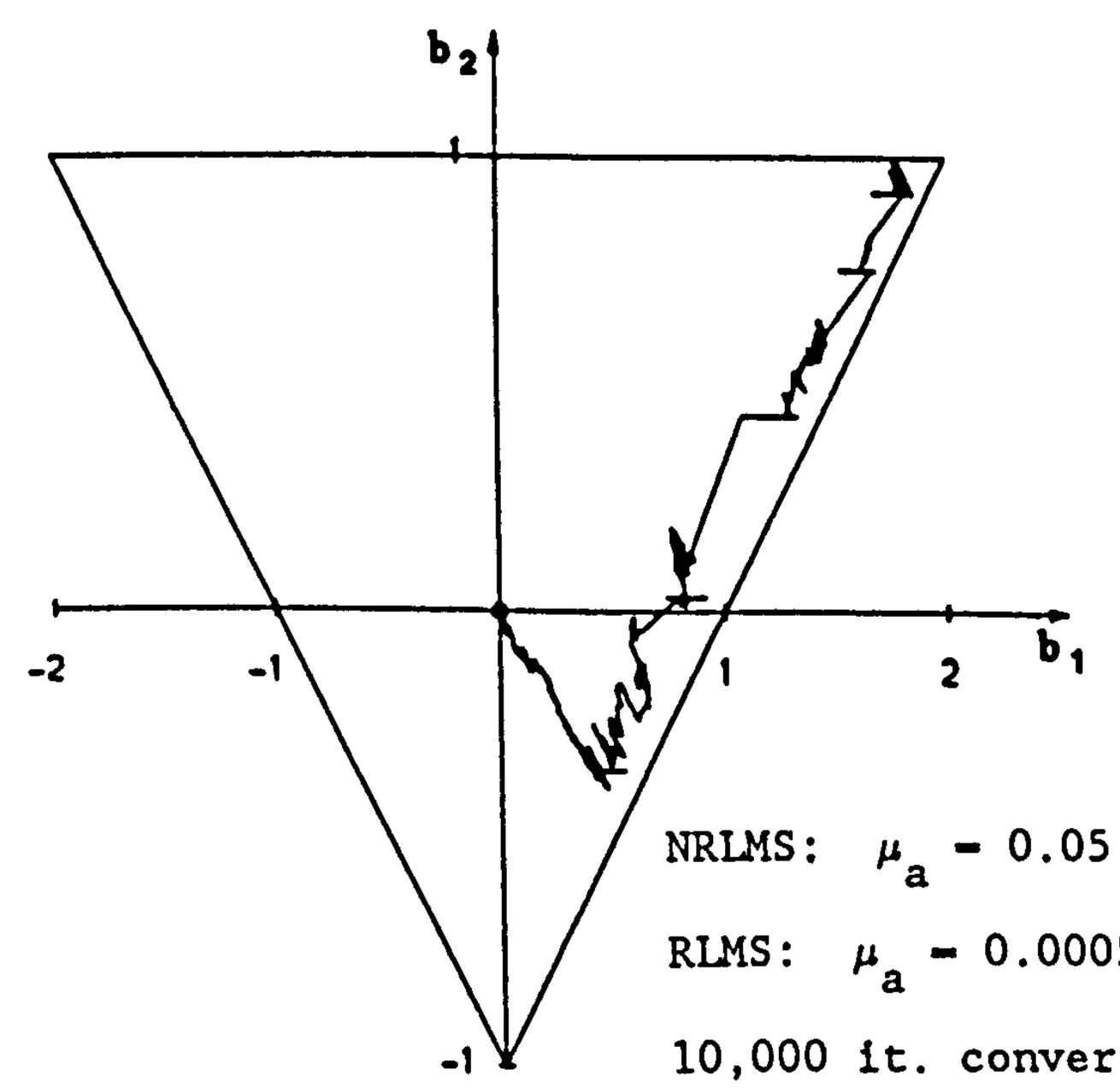


Figure 4.5 Valley-Search Algorithm Adaptation.



a) Target Plant Coefficients: $(a_0,b_1,b_2) = (1.0,-1.5,0.7)$



b) Target Plant Coefficients: $(a_0,b_1,b_2) = (1.0,1.8,0.95)$

Figure 4.6 (b_1,b_2) Coefficient Tracks for the Valley-Search Algorithm.
Initial Filter Coefficients $(a_0,b_1,b_2) = (0.0,0.0,0.0)$

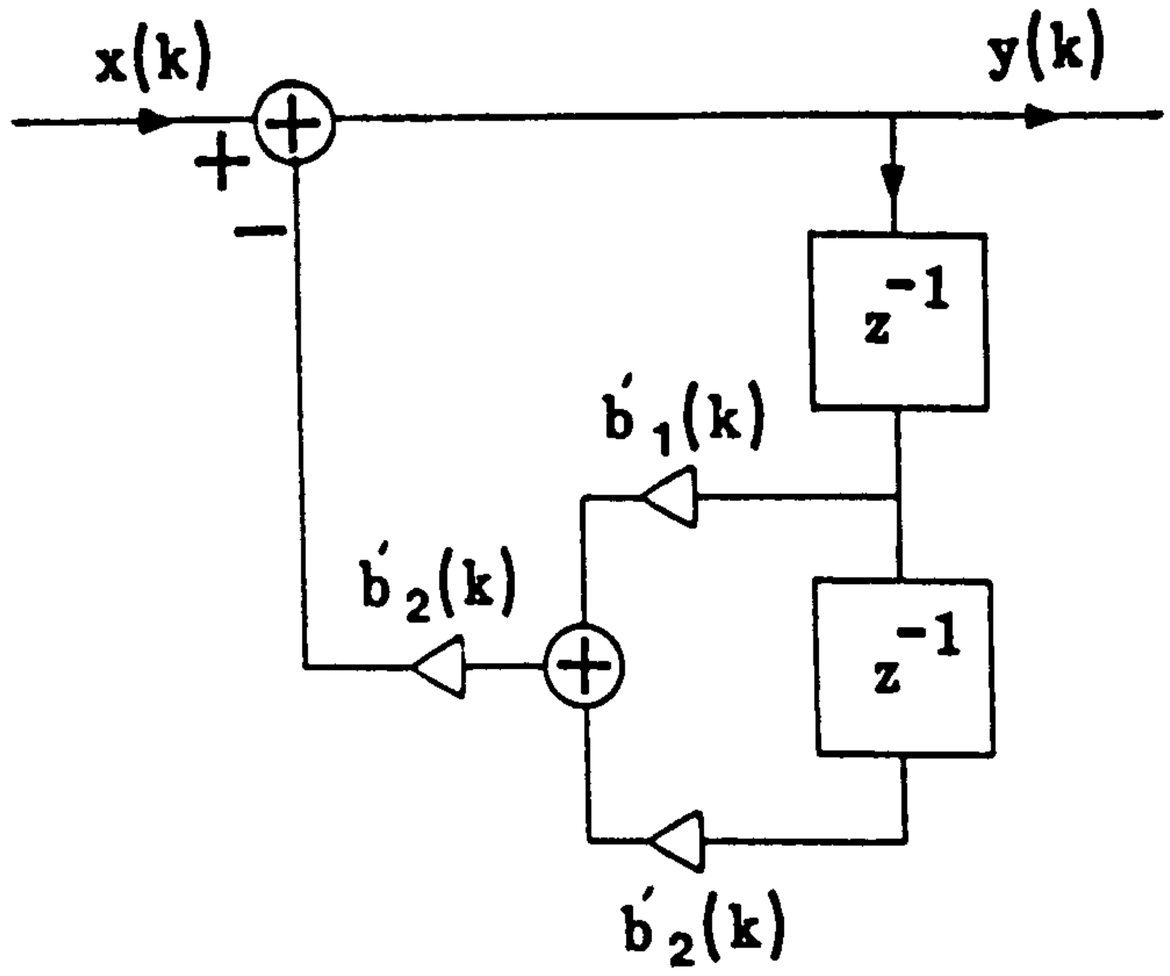


Figure 4.7 R-Cosθ Recursive Filter Section.

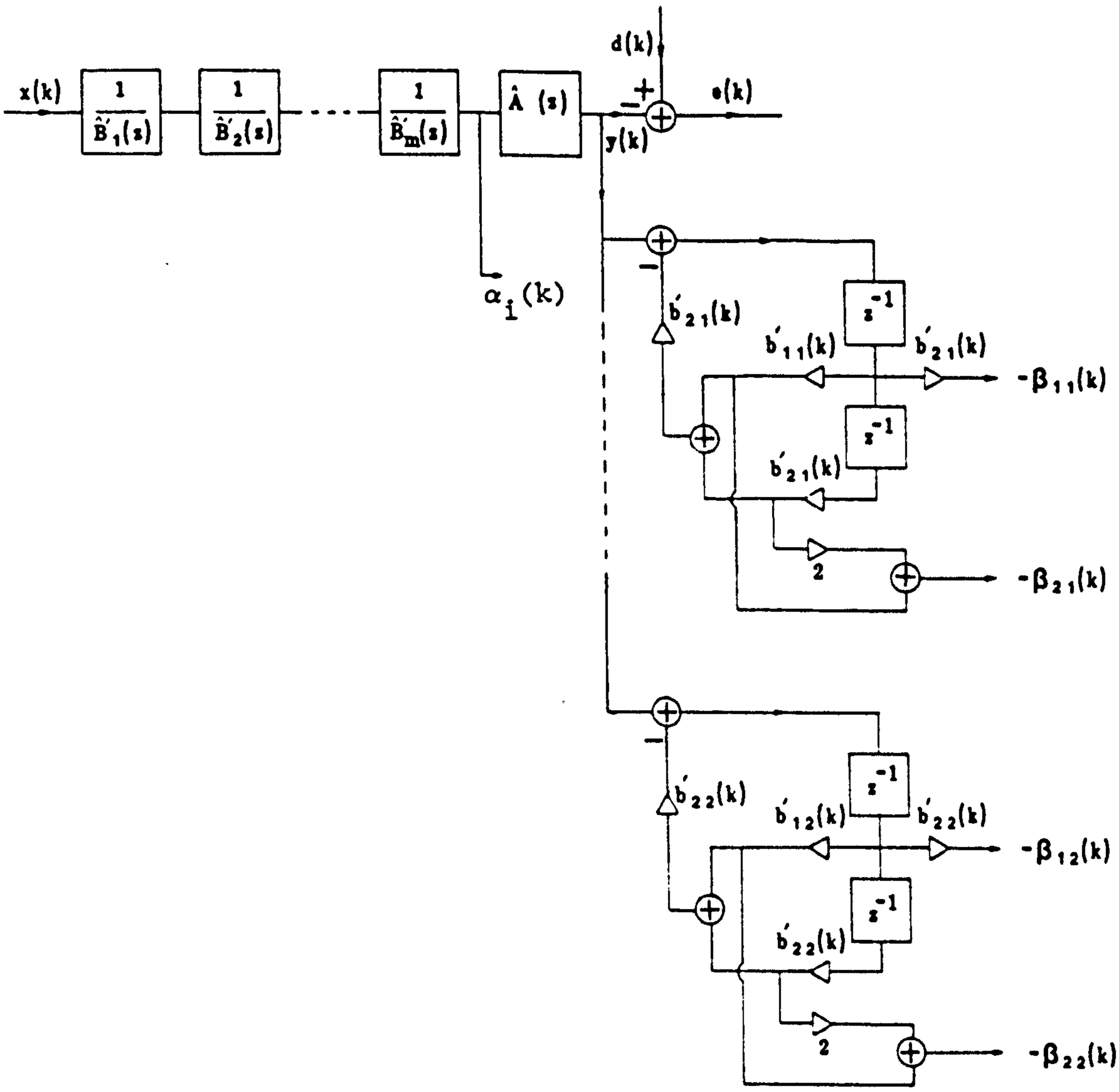
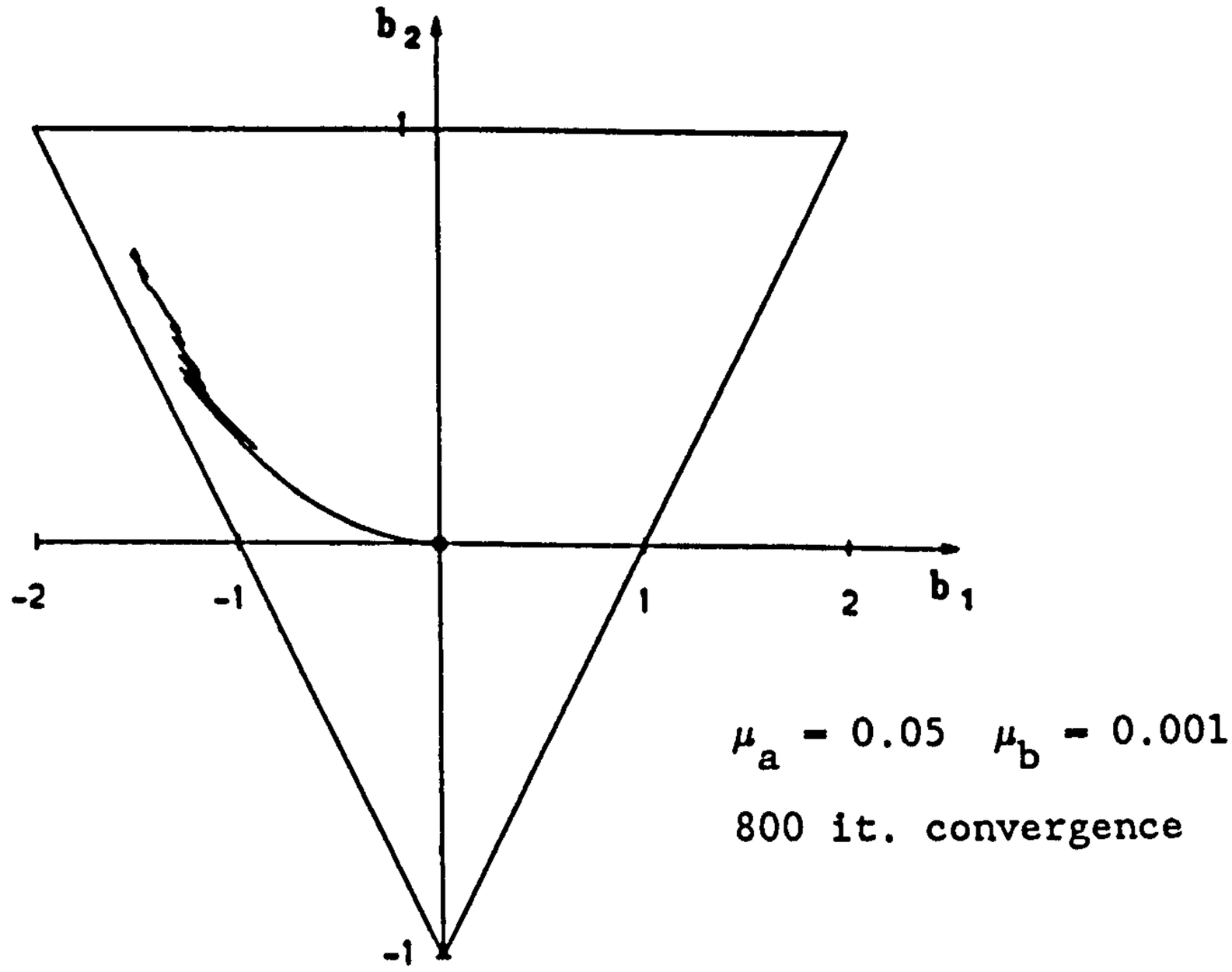
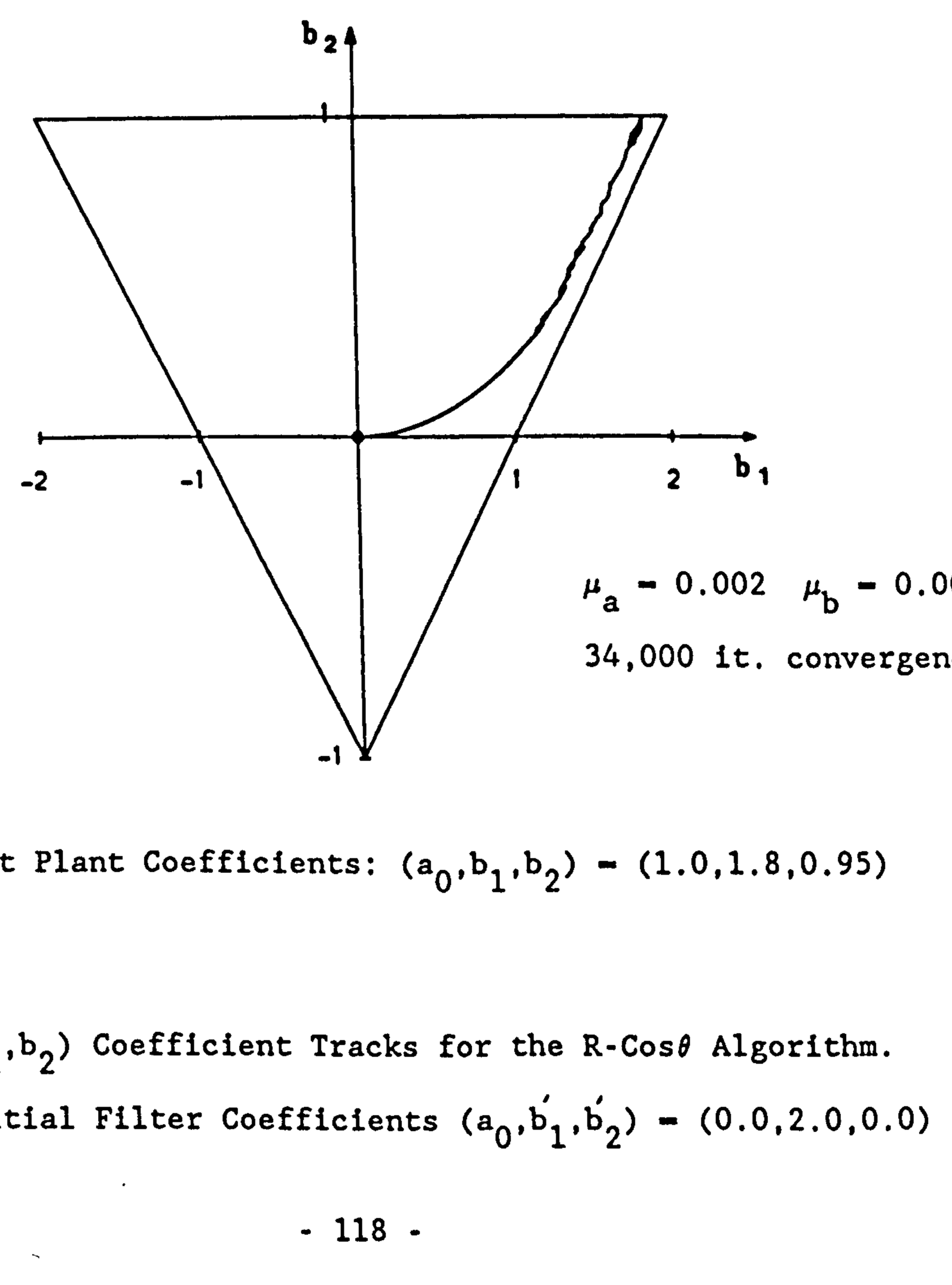


Figure 4.8 R-Cosθ Adaptive Cascade Structure.



a) Target Plant Coefficients: $(a_0,b_1,b_2) = (1.0,-1.5,0.7)$



b) Target Plant Coefficients: $(a_0,b_1,b_2) = (1.0,1.8,0.95)$

Figure 4.9 (b_1,b_2) Coefficient Tracks for the R-Cosθ Algorithm.
Initial Filter Coefficients $(a_0,b_1',b_2') = (0.0,2.0,0.0)$

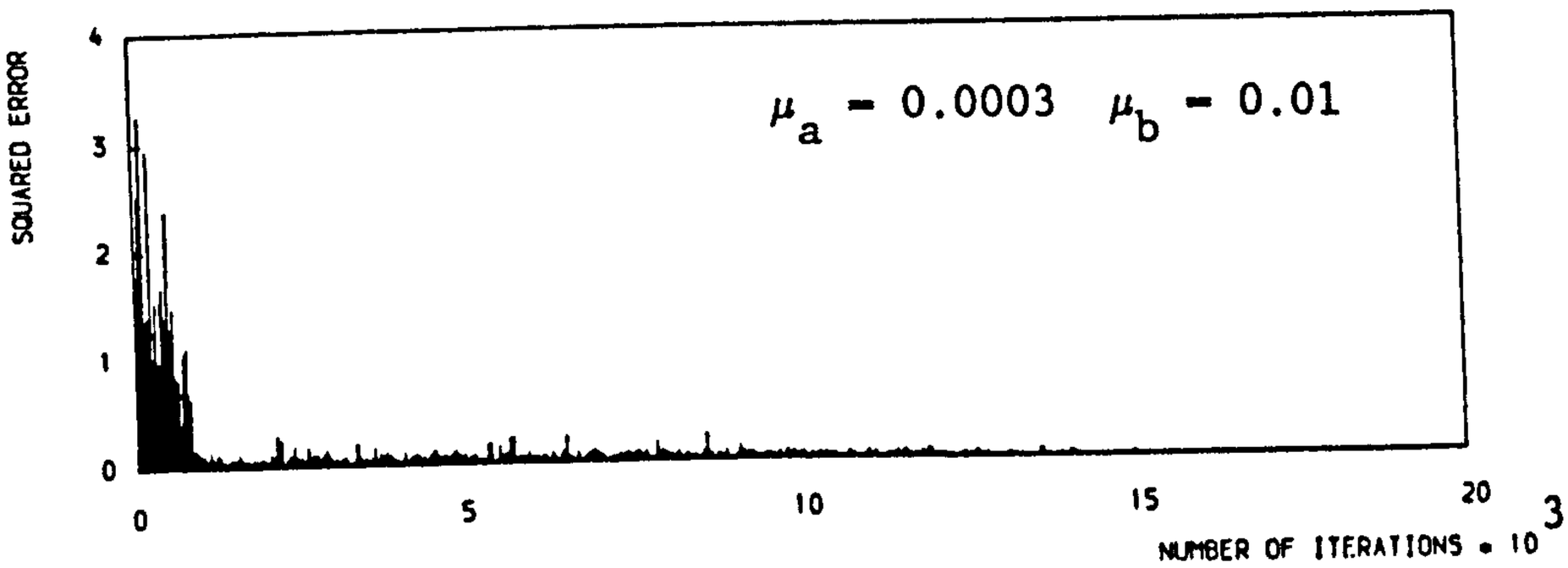


Figure 4.10 R-Cosθ Algorithm Convergence Plot for Fourth-Order Example.

CHAPTER 5

5.0 AN INVESTIGATION INTO ECHO CANCELLATION FOR A DUPLEX DIGITAL TRANSMISSION SYSTEM

In this chapter, the feasibility of using adaptive IIR filters for echo cancellation in a two-wire 144 kbit/s duplex data transmission system is investigated. Echo responses for varying line lengths and line transformers are studied, and the results used to design an IIR echo cancelling structure and an algorithm for the filter adaptation. Computer simulations are then used to show that the performance of the IIR filter is much superior to that of the standard non-recursive filter, and as a result offers possible hardware savings.

5.1 INTRODUCTION

Currently, telephone systems world-wide are being converted from the old analogue networks to modern digital transmission systems. One advantage of a digital telephone network is that both speech and data can be transmitted simultaneously on a single channel. A CCITT recommendation has proposed a channel capacity of 144 kbit/s for transmission on the two-wire link between subscriber and exchange[58]. The 144 kbit/s data rate consists of two 64 kbit/s speech channels and a 16 kbit/s data channel. One of the major considerations in the design of the transmission system is the removal of the locally transmitted signal echo from the receive path. This problem is illustrated in figure 5.1.

Consider the case of the subscriber, say, (the problem is identical at the exchange end) when both the subscriber and exchange are transmitting. Due to imperfections in the hybrid A, part of the transmitted signal from Tx_1 will leak into the receiver Rx_1. This leakage signal, or 'echo' will be -8 to -15 dB below the level of the transmitted signal. However, due to line attenuation, the incoming signal from Tx_2 could have a loss of -40 dB. The wanted signal, therefore, is perhaps 32 dB below the level of the echo of the locally

transmitted signal. Two methods for the removal of this echo have been proposed.

Time Division multiplexing of the two transmitted signals will eliminate the problem, but at the expense of a doubling in the bandwidth for a given baudrate, and hence increased line attenuation with associated problems. The second, and more attractive solution, is that of echo cancellation using adaptive digital filters. This approach is illustrated in figure 5.2.

The filter $\hat{H}(z)$ is adapted to minimize $\overline{E^2(z)}$ (the mean square error), at which point the output Y(z) will form a best least squares approximation to D(z). The echo cancellation process belongs to the signal estimation category of Section 1.2. The output from $\hat{H}(z)$ must be a sufficiently accurate copy of D(z) to achieve at least 60 dB attenuation of the echo. This will enable the far end signal to be recovered with an acceptable error rate. The cancellation should be obtained with the least possible outlay in hardware to minimize on cost.

Most standard echo cancellers rely on the simple, adaptive FIR ladder structure. Convergence behaviour of the LMS algorithm used for the filter adaptation has been studied and shown to be robust[8]. However, when the echo channel impulse response contains significant energy outside the time span of the FIR filter, the cancellation obtainable may not be sufficient. If this energy is distributed over a long time period, then considerable lengthening of the adaptive filter will only result in a small improvement in performance for a large outlay in computation and hardware.

Hentsche[58,59] studied echo channel impulse responses for a 144 kbit/s system and found that the energy distribution of the echo could result in a FIR filter being impractical for the necessary echo cancellation. He proposed an adaptive filter structure consisting of a low order recursive section (having an Infinite Impulse Response) in

parallel with a FIR ladder filter to minimize the hardware outlay for the echo canceller.

5.2 DESCRIPTION OF ECHO CHANNEL RESPONSES

Using a hardware model of the complete system, echo channel responses were generated for a range of line lengths and line transformers. The pulses transmitted by the system were coded using a (1,-1) multiple response code to reduce the level of intersymbol interference created by the channel. The shape of the transmitted pulses is shown in figure 5.3. The formation of the echo from the hybrid is illustrated in figure 5.4. The hybrid should act as a balanced bridge with the outgoing pulse being removed from the receive path. However, as the transformer and line impedances are frequency dependent, complete cancellation using resistor R_2 is not possible. The mismatch leads to the echo appearing at the receiver.

It was found that the echo channel response to a (1,-1) pulse consisted of a precursor, two main lobes which contained most of the energy, and an exponentially decaying tail. A typical echo is shown in diagrammatic form in figure 5.5. The relative amplitudes of each of the component parts is a function of the line length and the transformer impedance. For a particular line transformer, the line length was varied over the range of possible distances between the exchange and subscriber. It was found that the echo reponse tails decayed approximately monotonically and that the rate of decay was only dependent on the inductance of the line transformer and not on the length of the line. A high inductance transformer has a low amplitude, slowly decaying tail, while a low inductance transformer has a large amplitude, quickly decaying tail. The region of the response affected by the changing line length was the area around the zero crossing marked 'Z' in figure (5.5). For a short line, the zero crossing occurred

earlier in time than for a long line, and, in addition, the rise in the response amplitude following the zero crossing was less for a short line than for a long line.

In[58,59] Hentschke suggested that ringing may occur in the echo tails, and designed his echo canceller to allow complex conjugate pole pairs. For the echo reponses generated in these tests, no ringing was found and a different echo cancelling structure is proposed.

5.3 ADAPTIVE IIR ECHO CANCELLER STRUCTURE

The characteristics of the echo channel responses suggests the use of an FIR filter to cancel the high amplitude main lobes, and a low order adaptive IIR filter to cancel the long tail. However, it is not clear where the FIR filter span should end to give minimization of hardware for the required 60 dB echo attenuation. This problem is further complicated by IIR filter adaptation problems (See Sections (2.3), (4.1)). The non-uniform nature of the IIR filter performance surface could prove very disadvantageous by slowing the rate of adaptation considerably, and possibly driving the filter into a local minimum. Hence, the order of the IIR filter section should ideally be kept to a minimum.

Figures 5.6a,b and 5.7a,b show plots of the main sections and tails for echo responses from two different lines lengths, a 200m line and a 5km line. As these line lengths are at opposite extremes of possible line lengths, in general, echo responses will lie between the reponses shown in figures 5.6 and 5.7. A 32 coefficient FIR ladder filter was implemented in the transmission system model with the line transformer giving rise to the echoes shown in figures 5.6 and 5.7. It was found that the filter was unable to obtain the echo cancellation necessary to allow the system to function satisfactorily, because of the long echo tails. The 32T limit of the FIR echo canceller is marked on

figures 5.6b and 5.7b. It is seen that the echo tails have a very slow decay rate, and hence considerable lengthening of the FIR echo canceller will not reduce the echo cancellation significantly.

To obtain the necessary cancellation of the echo tails, an Adaptive IIR Filter Section was added to the end of the 32 coefficient FIR filter. Inspection of the echo responses reveals two types of tails to be modelled. In figure 5.6b the echo decays approximately monotonically following the 32T limit, whereas the echo in figure 5.7b rises quickly for a further 24T before decaying. In both examples the tail amplitudes remained positive and the final rates of decay are approximately 0.99^{nT}.

The non-oscillatory nature of the response tails suggests that they can be modelled using real poles (ignoring complex conjugate pairs). A real pole has the transfer function:

$$H(z) = \frac{1}{(1 - b_1 z^{-1})} \qquad (5.1)$$

The impulse response of this filter is an infinite geometric series, viz:

$$Y(z) = 1 + b_1 z^{-1} + b_1^2 z^{-2} + b_1^3 z^{-3} + \ldots \qquad (5.2)$$

Where $|b_1| < 1$ for the filter to be stable. The response of two real poles at $z = b_1$ and $z = b_2$ is the convolution of their individual impulse responses. From figure 5.6b, it is seen that the tail of the echo for a short line should be adequately modelled using a first order pole section having a positive coefficient. However, the 5km tail requires a second pole in order to give the initial rise required in the response (See figure 5.7b). Both the b_1 and b_2 coefficients will have positive values in this second case.

Although the two responses can be approximately modelled with second order IIR filters, close inspection reveals that they are in fact responses from higher order systems. This implies that the performance surface which drives the adaptation of the coefficients will possibly contain local minima[12]. Local minima will create difficulties for the filter adaptation if the coefficients are adapted from arbitrary starting values. By keeping selected coefficients stationary at predetermined values, certain features of the performance surface can be exploited to obtain an optimum echo canceller.

5.31 Choice of Structure for the IIR Filter Section

In Chapter 1 it was shown that implementation of the $\{a_i\}$ numerator coefficients from the filter transfer function as a Direct Form filter section produced a quadratic performance surface with respect to those coefficients. The proposed adaptation algorithm for the IIR echo canceller utilizes this uniformly shaped error surface to avoid local minima in the corresponding surface for the filter $\{b_i\}$ denominator coefficients.

As the decay rate of the echo tails is governed only by the inductance of the line transformer, keeping the b_1, b_2 denominator coefficients (giving rise to the poles) stationary at suitable coefficient values will allow the $\{a_i\}$ numerator coefficients to adapt to best match the poles' impulse response to the echo tail. When this has been achieved, the $\{a_i\}$, $\{b_i\}$ IIR Section coefficients and the Main FIR ladder filter coefficients can be adapted to find the optimum echo canceller. Adapting the filter using this method will avoid any local minimum present in the error surface.

The proposed IIR section of the echo canceller has the transfer function:

$$\hat{H}_I(z) = \frac{a_0 + a_1 z^{-1}}{(1 - b_1 z^{-1})(1 - b_2 z^{-1})} \tag{5.3}$$

The poles are implemented as two first order sections to constrain them to be real. For stability, $|b_1|$ and $|b_2|$ must be kept less than unity. In addition, the characteristics of the echo tails means that b_1 and b_2 can be further constrained to be positive.

From figures 5.6 and 5.7, as the length of the line between the subscriber and the exchange is varied, it is known that the final exponential decay of the echo response tails will be approximately 0.99^{nT}. The b_1 coefficient is preset to a value of 0.99 to model the echo tail decay. A value of $b_2 = 0.4$ will provide the rise in the reponse required for a long-line echo. By fixing b_1 and b_2 at these values, the a_0 and a_1 coefficients can adapt to a short line echo by positioning a transfer function zero to cancel the pole generated from the b_2 coefficient. For longer line echoes the $\{a_i\}$ coefficients shape the all-pole response to give a best least-squares fit to the echo tail.

5.32 Derivation of the Adaptation Algorithm

The echo canceller structure proposed in Section (5.31) can be implemented using a FIR ladder filter followed by two first order All -pole sections and an All-zero section , as shown in figure 5.8. The $\{a_i\}$, $\{b_i\}$ and $\{c_i\}$ coefficients are adapted to minimize $\overline{e^2(k)}$. As this is not readily available, $e^2(k)$ is taken as a local estimate. From figure 5.8:

$$e^2(k) = d^2(k) + y^2(k) - 2d(k)y(k) \tag{5.4}$$

The method of steepest descent, which was described in Section 2.1, is used to adapt the filter coefficients. To generate the coefficient update equations (See equation (2.1), partial derivatives of $\overline{e^2(k)}$ with respect to $\{a_i\}$, $\{b_i\}$ and $\{c_i\}$ have to be calculated. From figure 5.8, the output of the adaptive filter at time t = kT is given by:

$$y(k) = y_1(k) + y_2(k) \tag{5.5}$$

Taking the z-transform of equation (5.5) and substituting the filter transfer functions from figure 5.8 yields:

$$Y(z) = X(z)\left[\hat{C}(z) + \frac{\hat{A}(z).z^{-(n+1)}}{\hat{B}(z)}\right] \tag{5.6}$$

Where:

$$\hat{A}(z) = a_0 + a_1 z^{-1}$$

$$\hat{B}(z) = (1 - b_1 z^{-1})(1 - b_2 z^{-1})$$

$$\hat{C}(z) = c_1 z^{-1} + c_2 z^{-2} + \ldots + c_n z^{-n}$$

From equation 5.4:

$$\frac{\partial \overline{e^2(k)}}{\partial a_i} = -2.e(k).\alpha_i(k) \tag{5.7}$$

$$\frac{\partial \overline{e^2(k)}}{\partial b_i} = -2.e(k).\beta_i(k) \tag{5.8}$$

$$\frac{\partial \overline{e^2(k)}}{\partial c_i} = -2.e(k).\gamma_i(k) \tag{5.9}$$

Where:

$$\alpha_i(k) = \frac{\partial y(k)}{\partial a_i(k)} \quad , \quad \beta_i(k) = \frac{\partial y(k)}{\partial b_i(k)} \quad \text{and} \quad \gamma_i(k) = \frac{\partial y(k)}{\partial c_i(k)}$$

From equation (5.6):

$$\gamma_i(z) = X(z).z^{-i} \qquad i = 1,..,n \tag{5.10}$$

The $\gamma_i(k)$ terms are therefore generated in the delay line of the $\hat{C}(z)$ filter and do not require any further computation. To calculate the $\alpha_i(k)$ and $\beta_i(k)$ terms, certain assumptions and simplifications need to be made to reduce the hardware outlay to an acceptable level for real time operation. Problems arise because the IIR section of the filter is time-varying and recursive, and hence $y_2(k)$ at a particular time reflects the past variations in the coefficients. By allowing only small coefficient updates at each iteration, the filter transfer function remains approximately constant over a number of iterations, and $y_2(k)$ can be assumed to be the output from a fixed filter. Therefore, from equation (5.6):

$$\alpha_i(z) = \frac{X(z).z^{-(n+1+i)}}{\hat{B}(z)} \qquad i=0,1 \tag{5.11}$$

$$\beta_i(z) = \frac{Y_2(z).z^{-1}}{(1 - b_i z^{-1})} \qquad i=1,2 \tag{5.12}$$

Hence, the $\alpha_i(k)$ and $\beta_i(k)$ terms can be generated by a series of time-varying IIR filters. As the coefficient changes are small, the output from the two recursive filter sections will be approximately constant over a few iterations, and the $\alpha_i(k)$ terms can be generated within the delay line of the $\hat{A}(z)$ filter and require no additional hardware. Two single-order recursive sections are required to calculate the $\beta_i(k)$ terms. The complete echo cancelling structure is shown in

figure 5.9. The coefficients are updated using:

$$a_i(k+1) = a_i(k) + \mu_a.e(k).\alpha_i(k) \qquad i = 0,1 \tag{5.13}$$

$$b_i(k+1) = b_i(k) + \mu_b.e(k).\beta_i(k) \qquad i = 1,2 \tag{5.14}$$

$$c_i(k+1) = c_i(k) + \mu_c.e(k).\gamma_i(k) \qquad i = 1,n \tag{5.15}$$

5.33 Sequence for Coefficient Adaptation

The coefficients of the echo canceller are adapted in the sequence given below. In addition to avoiding local minima in the filter performance surface, the adaptation sequence enables the adaptive filter to converge very quickly to the optimum echo canceller.

1) The $\{c_i\}$ coefficients of the main FIR ladder section are adapted for 1000 iterations to cancel the major part of the echo.

2) While still adapting the $\{c_i\}$ coefficients, the a_0, a_1 coefficients are adapted from (0.0,0.0) with the b_1 and b_2 coefficients fixed at suitable values for the line transformer inductance. The $\{a_i(k)\}$ coefficients match the fixed poles' response to the echo tail. The filter is adapted in this mode for 500 iterations.

3) Finally, the $\{a_i\}$, $\{b_i\}$, and $\{c_i\}$ coefficients are all adapted to find the nearest minimum in the error surface (which should be the global minimum). If adequate echo cancellation is achieved with the first two stages, this step can be ignored.

5.4 COMPUTER SIMULATIONS

To assess the performance of the recursive echo canceller, a computer simulation of the required sections of the system was used. A block diagram of echo channel modelling is shown in figure 5.10. During

the initial training sequence for the echo canceller, the far end signal is not transmitted and the input sequence X(z) is generated using a pseudo random noise sequence. The echo channel model was constructed from echo data obtained from a number of lines. The Analogue to Digital Converter (A.D.C.) was included in the model to measure its effect on the level of echo cancellation. For the adaptive filter structure, 32 (c_i) coefficients were used and values of the b_1, b_2 coefficients were initially set at 0.99 and 0.4 respectively.

For the echo responses considered in Section 5.3, the 32 tap FIR filter was unable to achieve the 60dB echo-cancellation required for a working transmission system. A convergence plot for the FIR echo canceller adapting to a 5km line echo is given in figure 5.11. An A.D.C. of 2^{11} levels limits the lower limit of cancellation to 66 dB. With the addition of the IIR filter section, the level of echo cancellation is greatly improved and drops below the 60dB threshold. Figures 5.12 and 5.13 show convergence plots for the recursive echo canceller adapting to a 5km echo and a 200m echo respectively. The adaptation sequence given in Section 5.33 was used for the adaptation and it was found that step (3) was not required as sufficient cancellation was obtained from the first two stages. For each line length, the number of A.D.C. levels was varied to see the effect on convergence.

It is seen from figures 5.12 and 5.13 that with the addition of the recursive filter section, a working system is easily achieved. The amount of echo cancellation obtained is in fact limited by the accuracy of the A.D.C.. Without an A.D.C. in the simulation to introduce quantization noise, greater cancellation is obtained for the 200m line than for the 5km line. This is expected because the short line echo tail is simpler to model. For the inductance of the line transformer used in this example, it was found to be unnecessary to adapt the poles from their initial values. With the (b_1,b_2) coefficients fixed, the extra

hardware for generating the $\{\beta_i(k)\}$ terms is not required and the echo canceller structure is simplified further.

From equations (5.10) and (5.11), it is seen that the $\{a_i\}$ feedforward coefficients and the $\{c_i\}$ feedforward coefficients require different values of μ parameters for the update equations (5.13) and (5.15). The μ_a adaptation parameter will have a smaller value than μ_c because the input signal for the $\{a_i\}$ coefficients is generated by the feedback via the $\{b_i\}$ coefficients, in addition to the filter input X(z). The power in the input signal to the $\{a_i\}$ filter coefficients will depend on the fixed values of the $\{b_i\}$ coefficients. For the line transformer used in this example, suitable values for the adaptation parameters were found to be $\mu_c = 2^{-9}$ and $\mu_a = 2^{-14}$.

5.41 Adaptation of the B(z) Filter Coefficients

Figure 5.14 shows a convergence plot for the complete recursive echo canceller adapting to a 200m line echo. The initial $\{b_i\}$ coefficients were fixed at (0.9,0.0) and started adapting after 1500 iterations. Problems arise in the adaptation due to the size of μ_b. When the $\{b_i\}$ coefficients start adapting, 60dB echo cancellation will already have been achieved. The error signal E(z) will contain very little power, the $a_0(k)$, $a_1(k)$ coefficients will be small values (because the tail amplitude is small), and hence $\beta_1(k)$ and $\beta_2(k)$ will be small values also. To give a significant coefficient change, μ_b must be very large. The great difference in size between μ_b and both e(k) and $\{\beta_i(k)\}$ will prove troublesome for the coefficient updates (equation 5.14) in a hardware realization.

A solution to this problem is to use SIGN $(\beta_i(k))$ rather than $\beta_i(k)$ for the $\{b_i\}$ coefficient updates. This method was used for the convergence plot given in figure 5.14. The (b_1,b_2) coefficients adapted to (0.985,0.2) and the echo cancellation limit set by the ADC was

reached after 2500 iterations

5.5 CONCLUSIONS

Echo channel responses for a 144 kbit/s duplex data transmission system were examined, and it was found that the energy distribution of the echoes is dependent on the line transformer inductance, and to a lesser degree, the length of the line. The energy contained in the echo could extend over a long time period making adaptive FIR cancellers an impractical proposition. In this chapter, a low order adaptive IIR filter was added to a main FIR filter as a means of reducing the cost for the echo cancelling section of the system.

A study of the echo tail characteristics showed that the tails were approximately monotonically decaying and did not oscillate. These characteristics allowed an IIR filter structure to be chosen which only implemented real poles, and not complex conjugate pairs. While this choice of structure simplifies the nature of the filter performance surface, local minima may still exist because of the insufficient order of the adaptive filter. However, the fact that the decay rate of the echo tail is dependent on the inductance of the line transformer enables an initial choice of pole coefficients to be made which guarantees a large improvement in echo cancellation. This is achieved by adapting the (a_i) coefficients with the (b_i) coefficients fixed at suitable predetermined values, thereby avoiding local minima in the error surface. It is noted that the IIR filter section chosen is a simplified version of the Modified Cascade Structure proposed in Section 3.2.2.

The IIR echo cancelling structure was tested and the results showed a significant improvement when compared with the 32 tap FIR echo canceller. For the example presented, a small addition in the amount of hardware necessary for the FIR filter, resulted in an extra 20 dB echo cancellation being obtained with the simplified IIR structure alone.

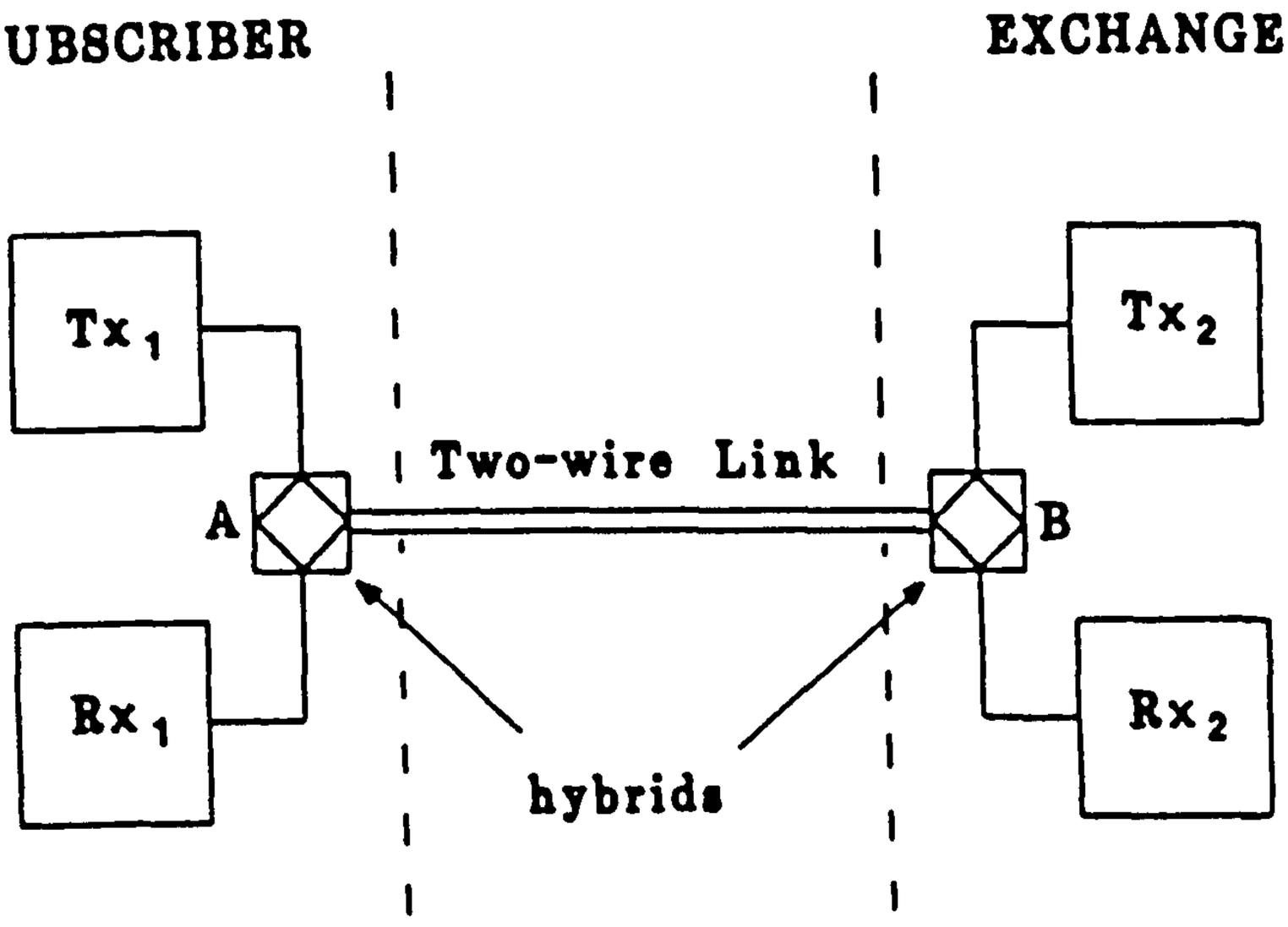


Figure 5.1 Duplex Data Transmission System.

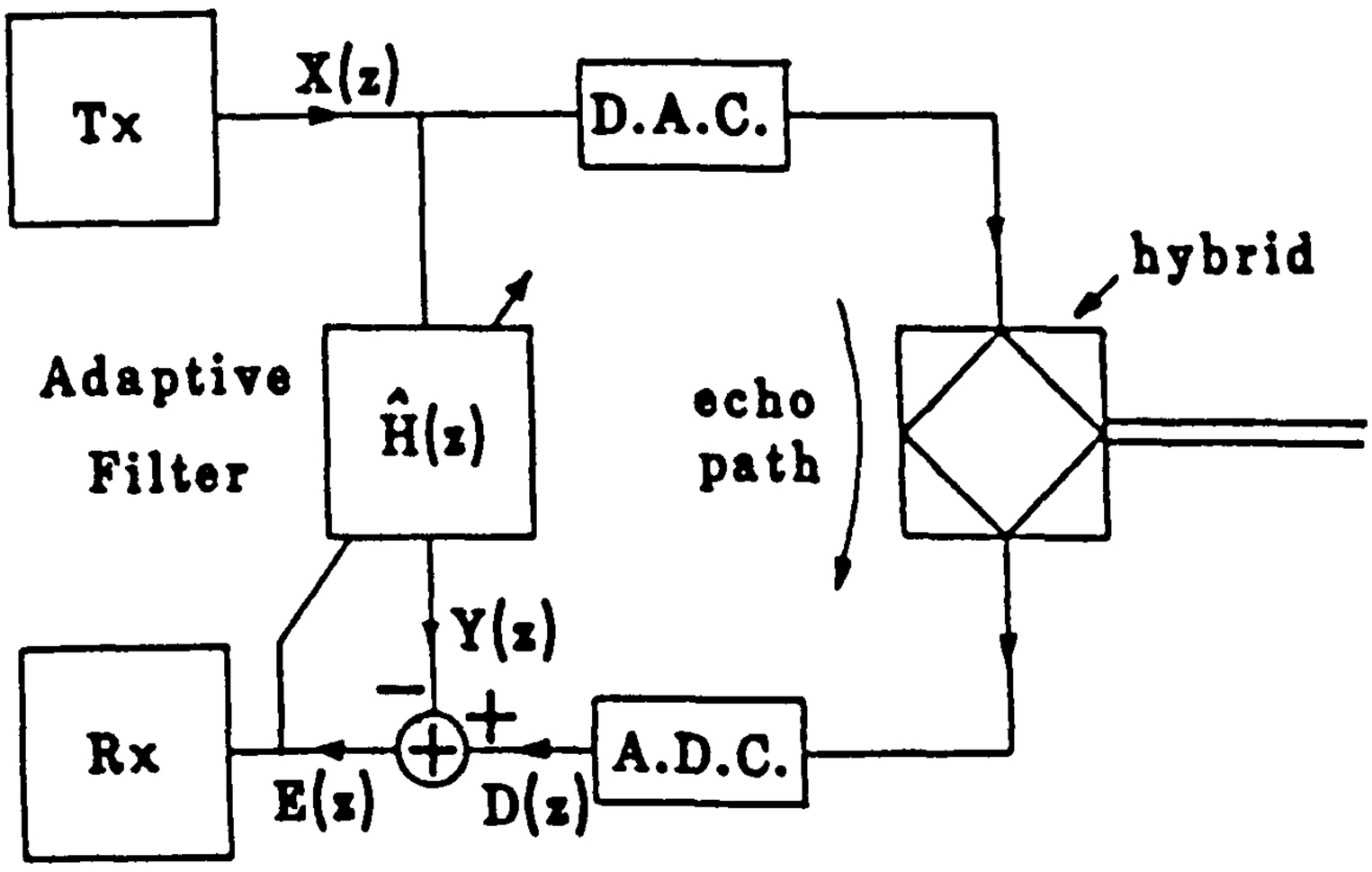


Figure 5.2 Adaptive Echo Canceller.

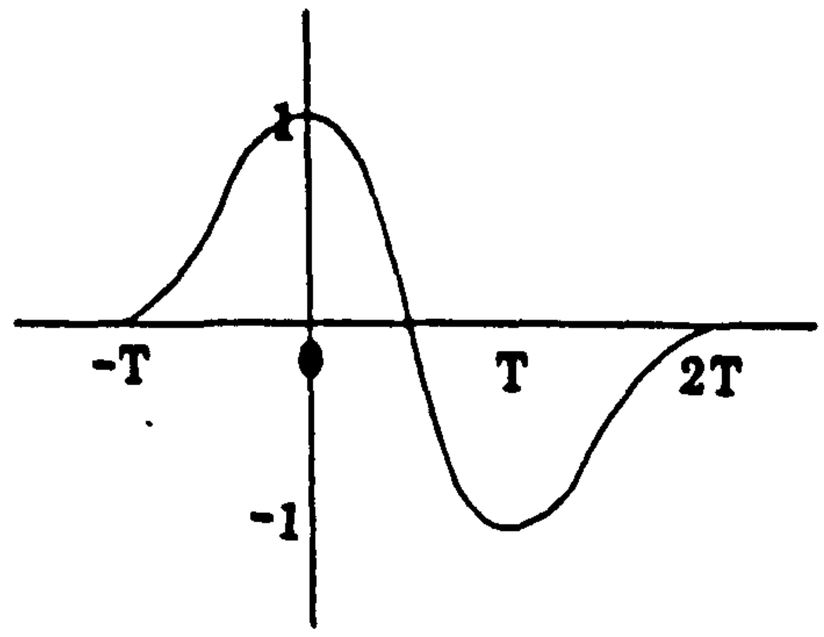


Figure 5.3 Transmit Pulse Shape

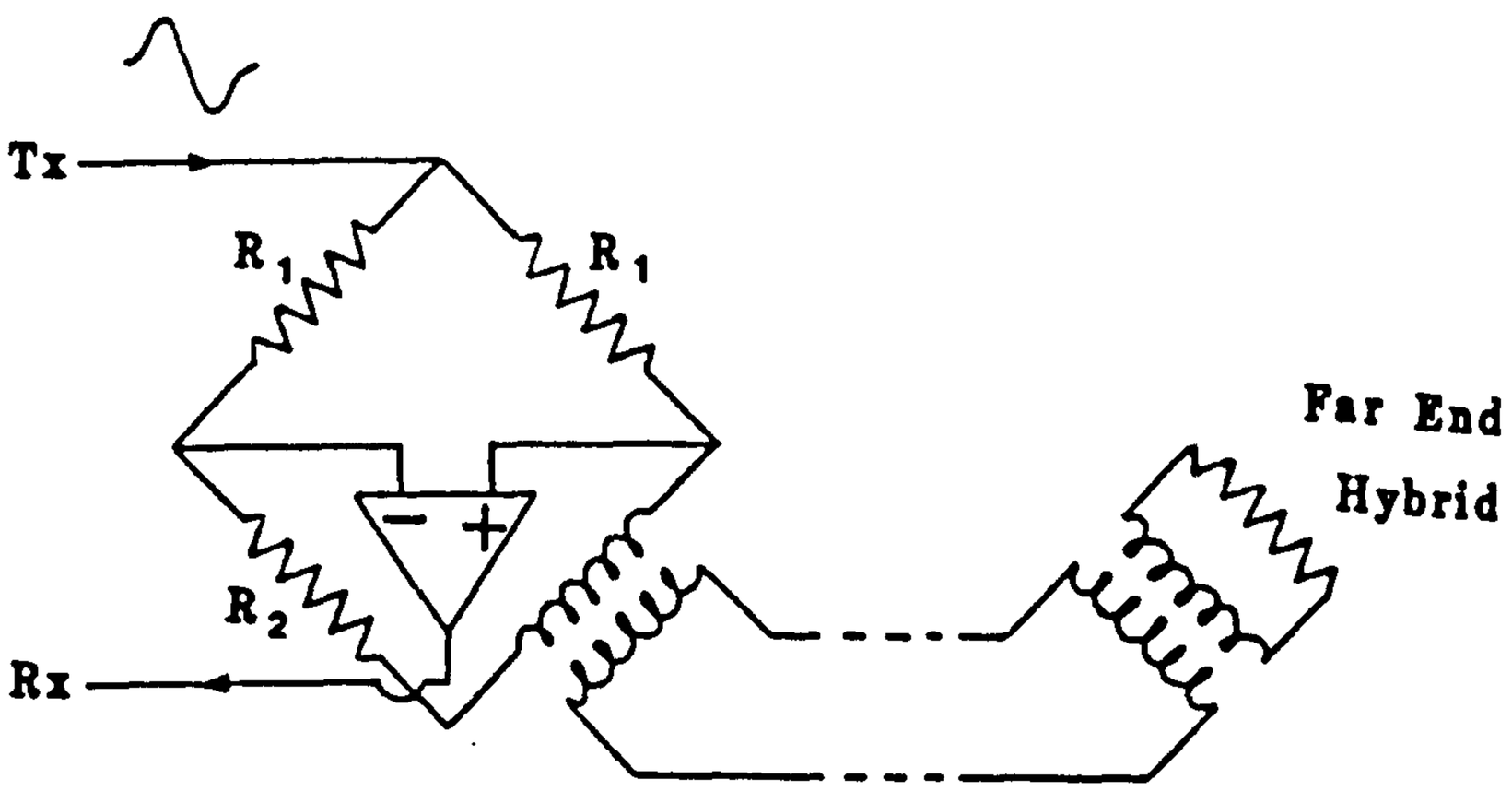


Figure 5.4 Echo Generation within Hybrid.

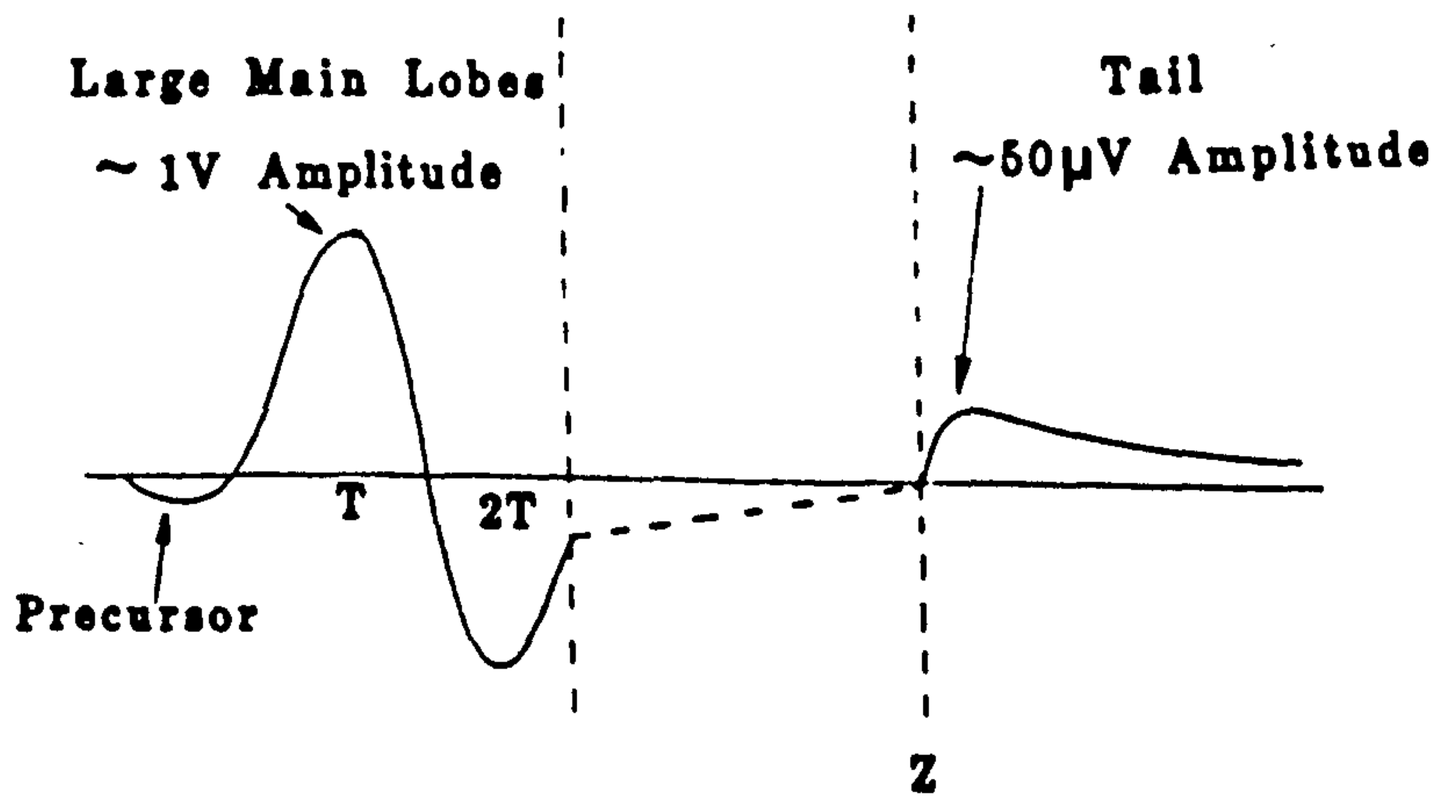
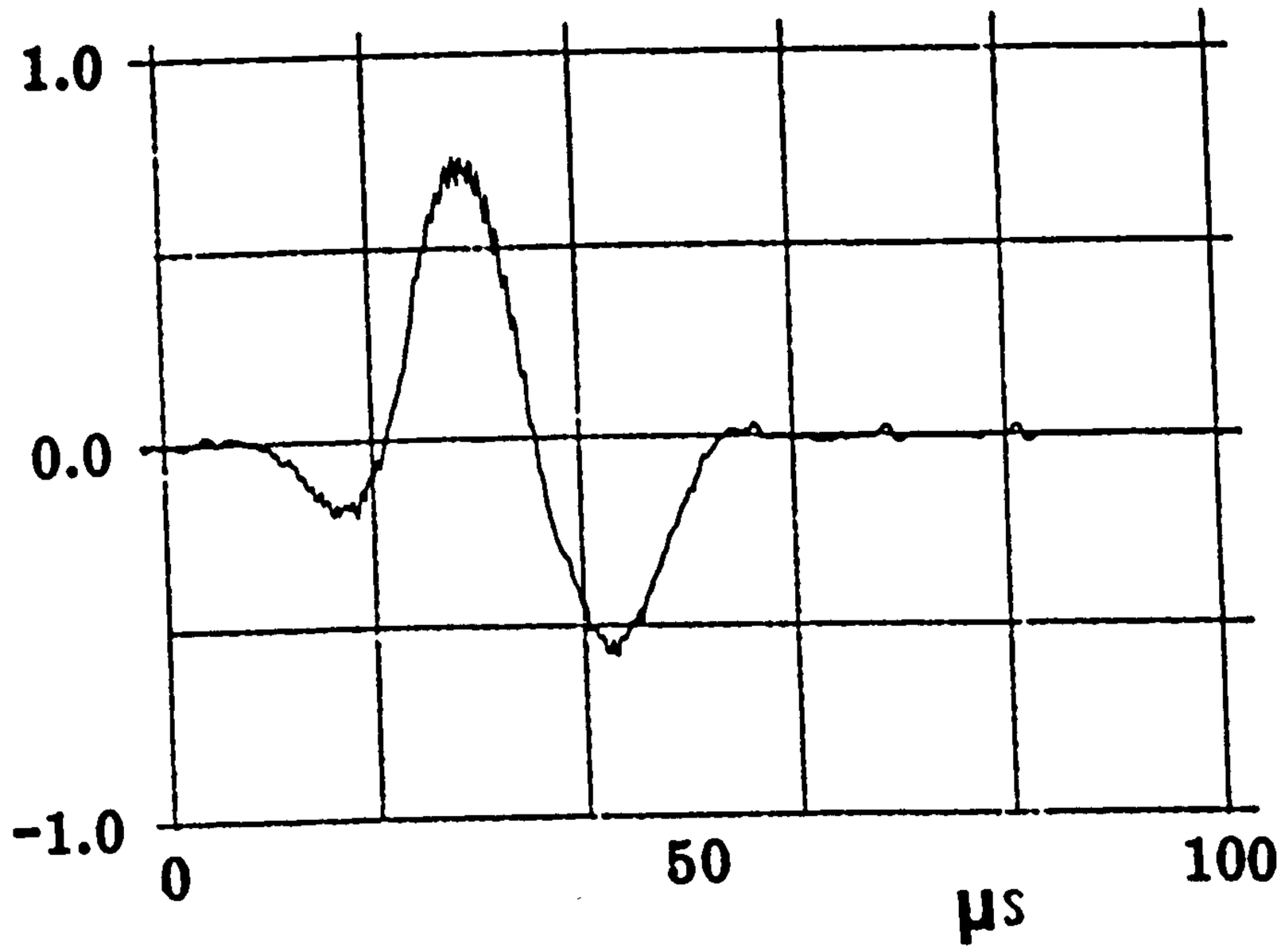
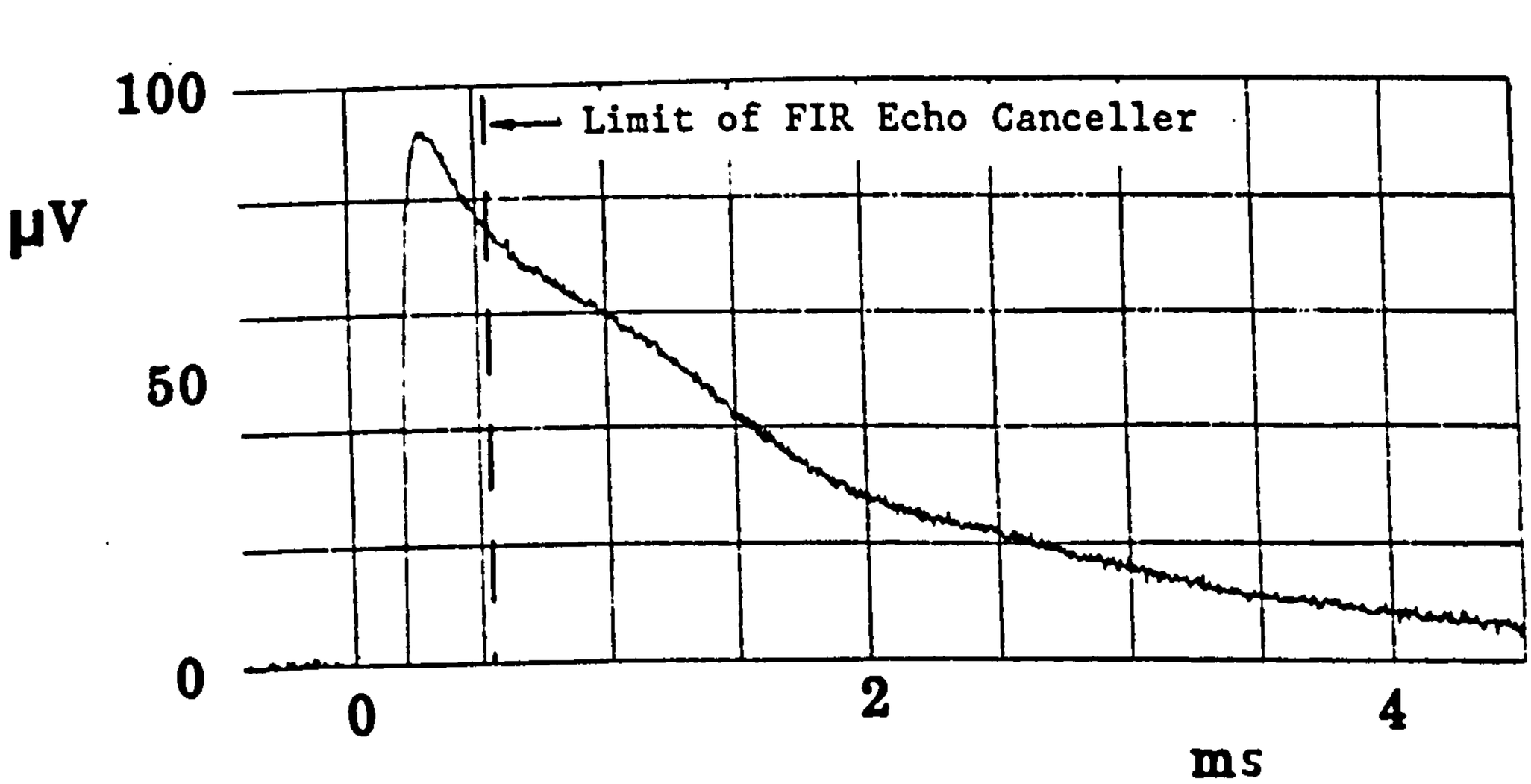


Figure 5.5 General Echo Response.

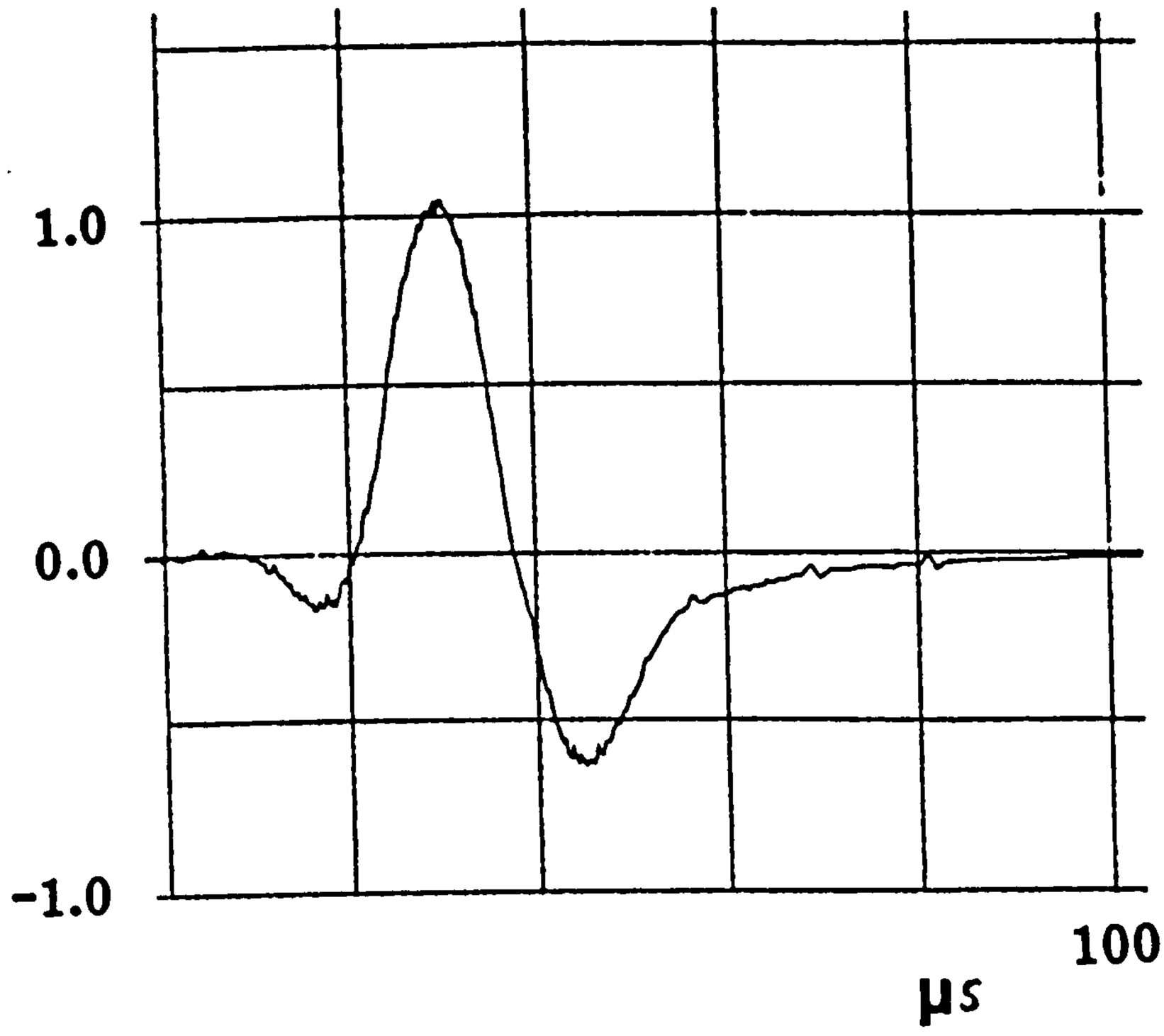


(a) Main Section of Echo

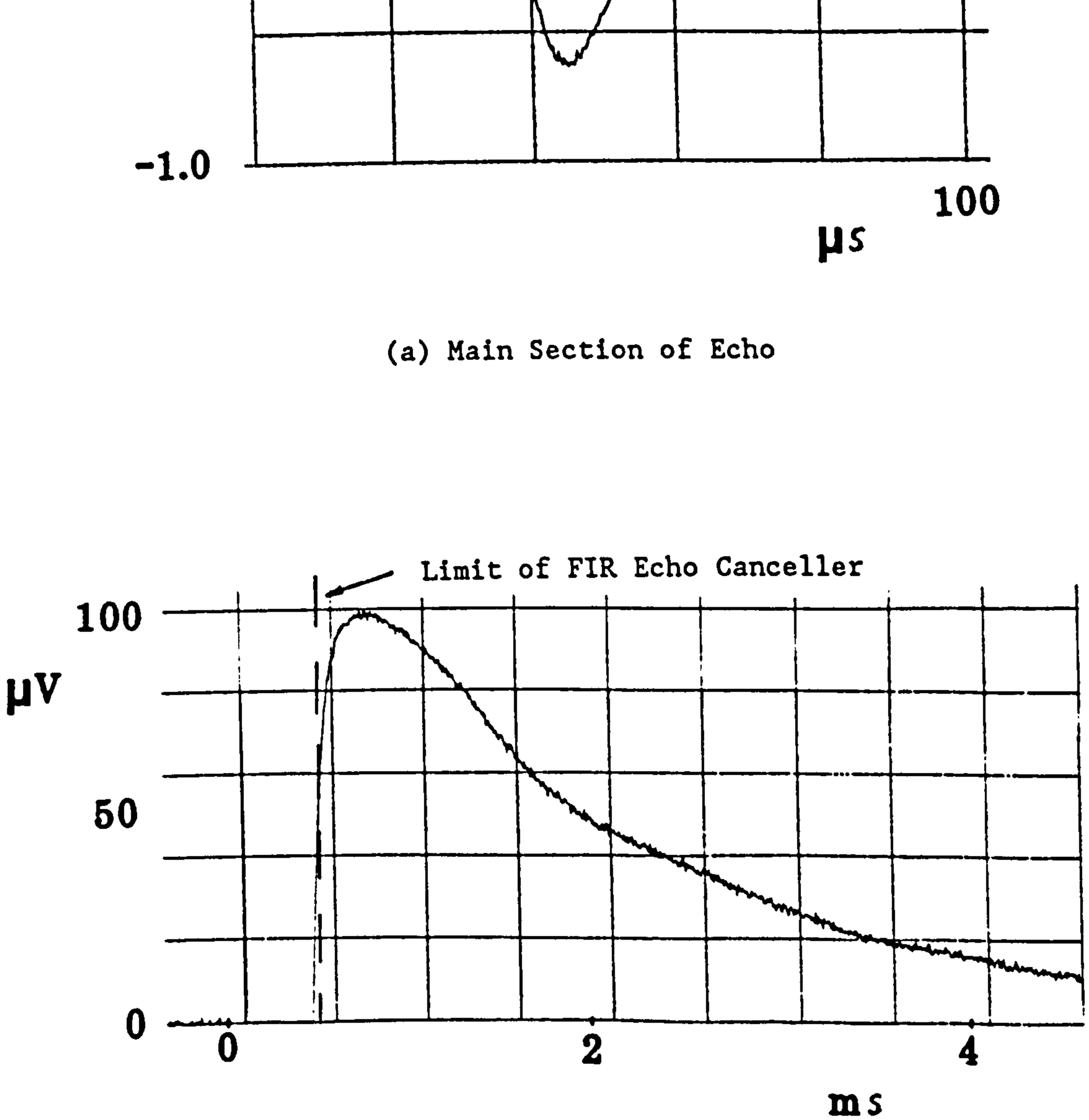


(b) Echo Tail

Figure 5.6 Echo Response from a 200m Line.



(a) Main Section of Echo



(b) Echo Tail

Figure 5.7 Echo Response from a 5km Line.

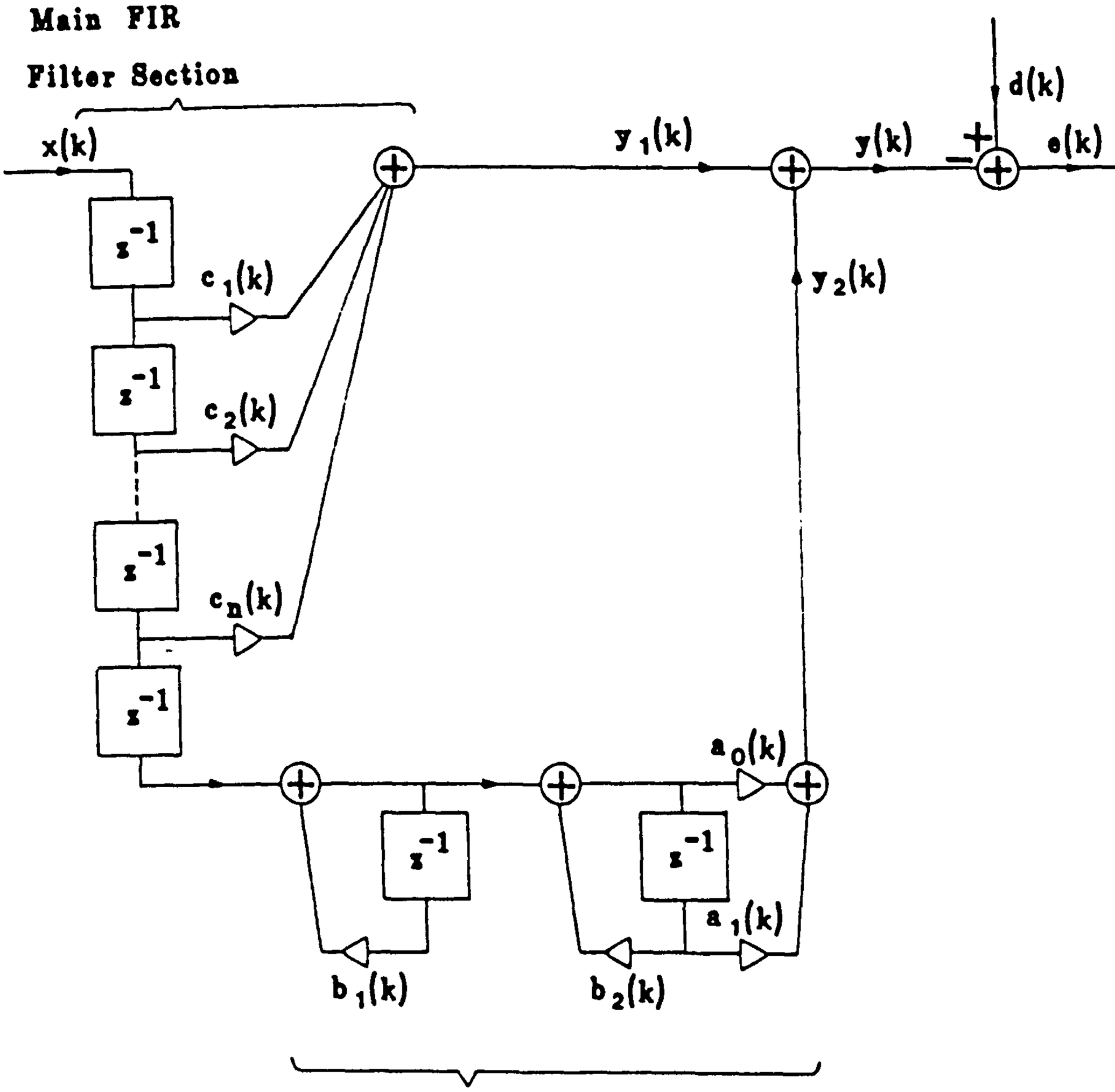


Figure 5.8 Filter Structure for Echo Canceller.

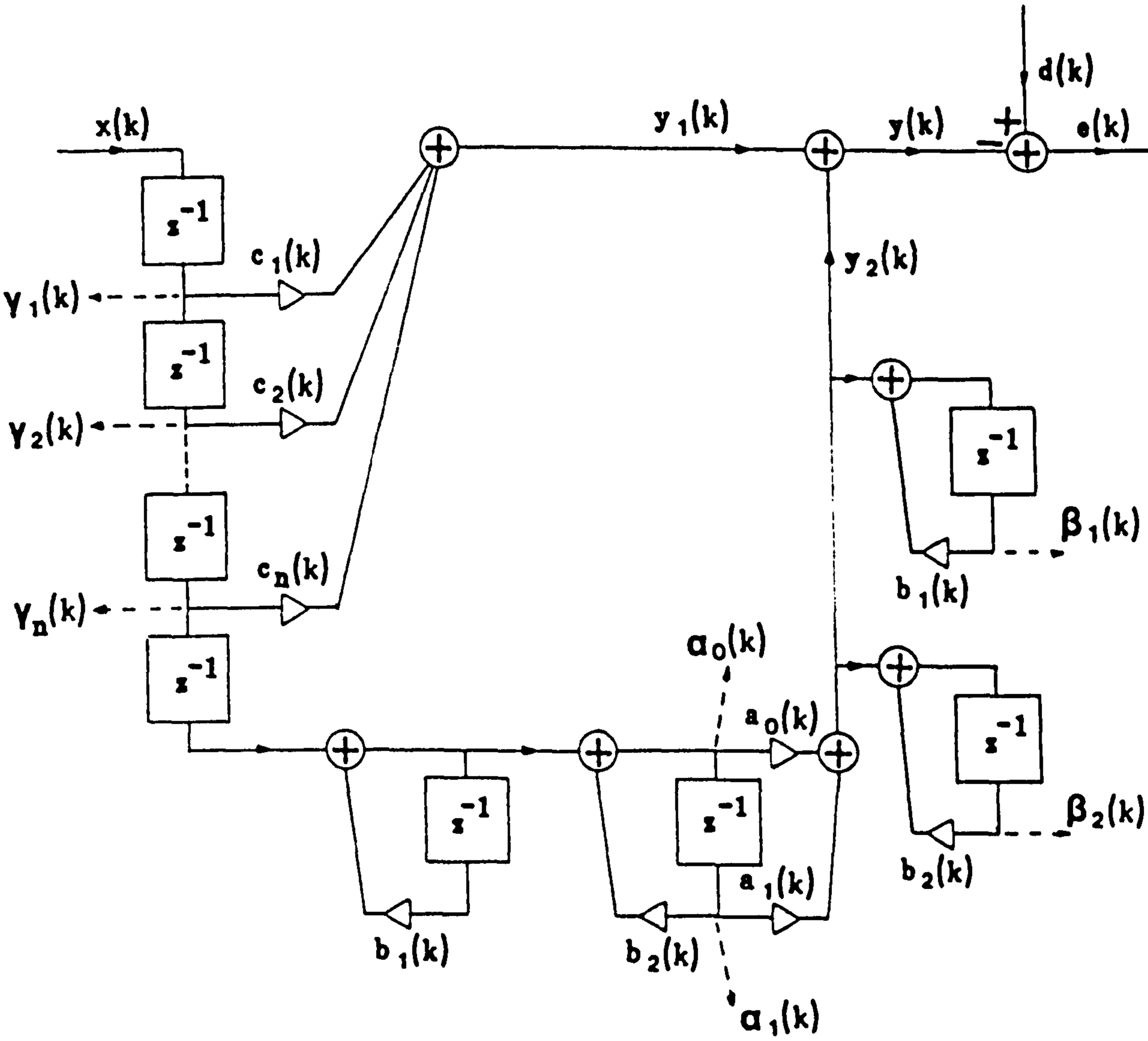


Figure 5.9 Adaptive IIR Echo Canceller.

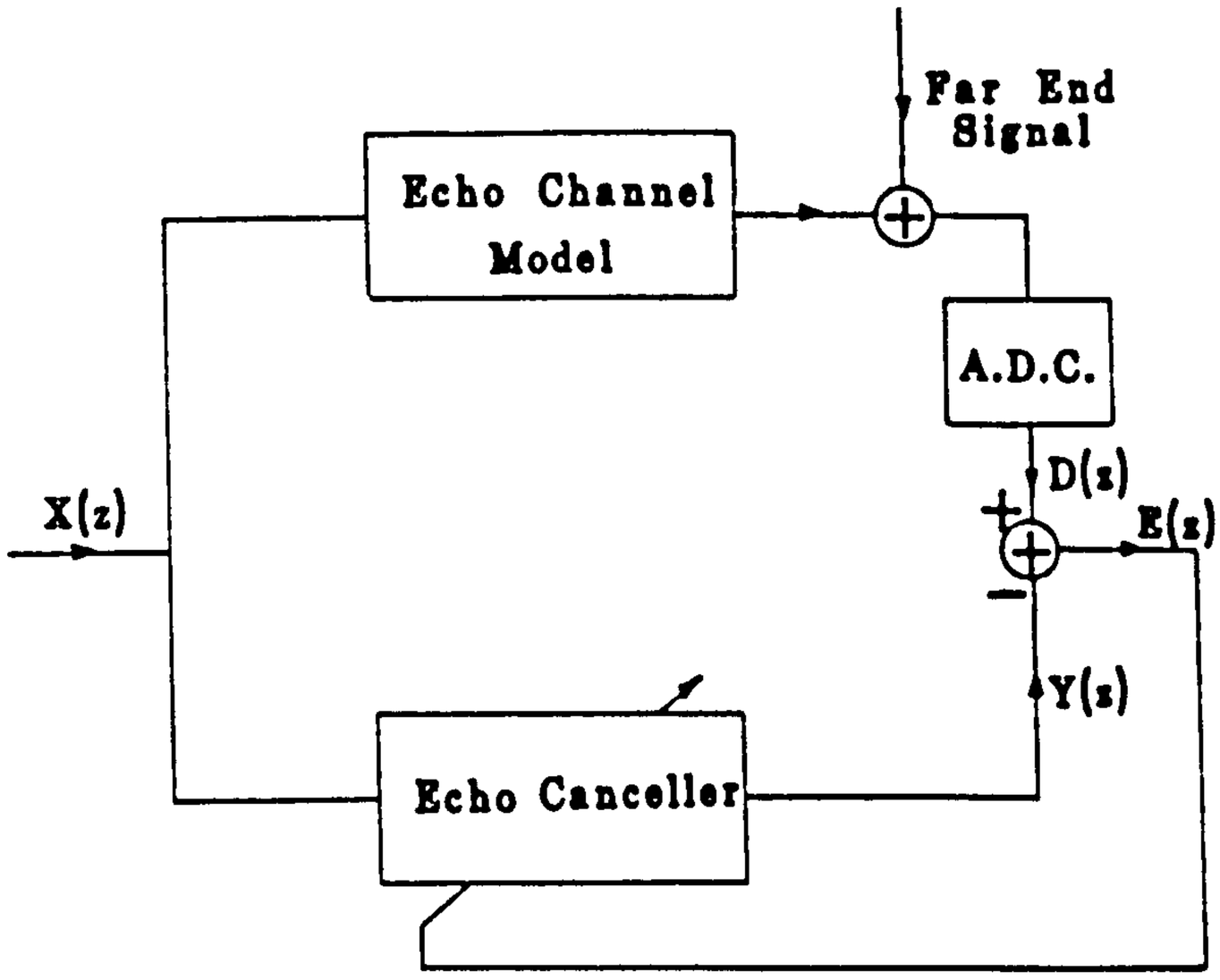


Figure 5.10 Echo Response Modelling.

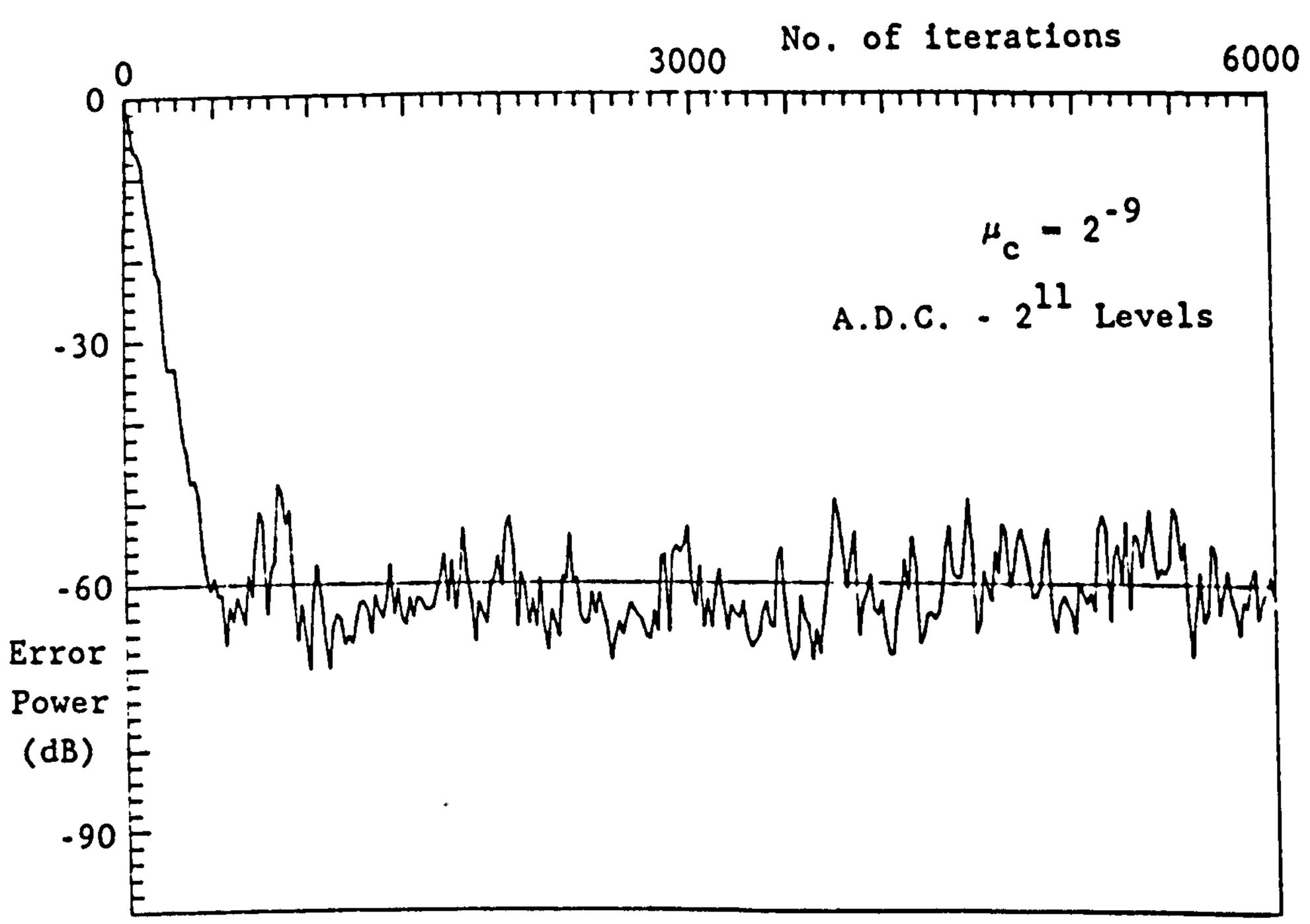


Figure 5.11 Convergence Plot for FIR Echo Canceller.

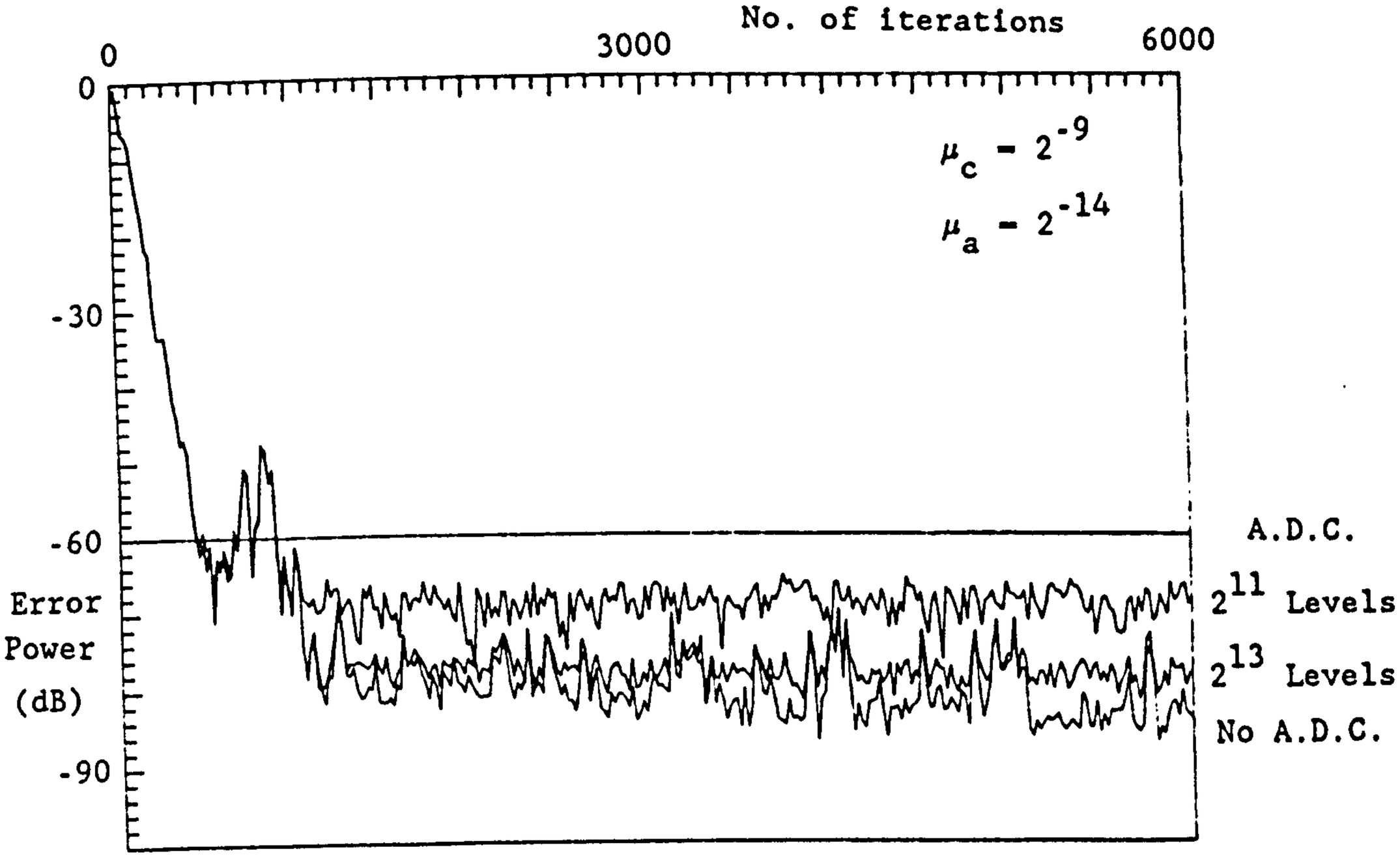


Figure 5.12 Convergence Plot for 5km Line.

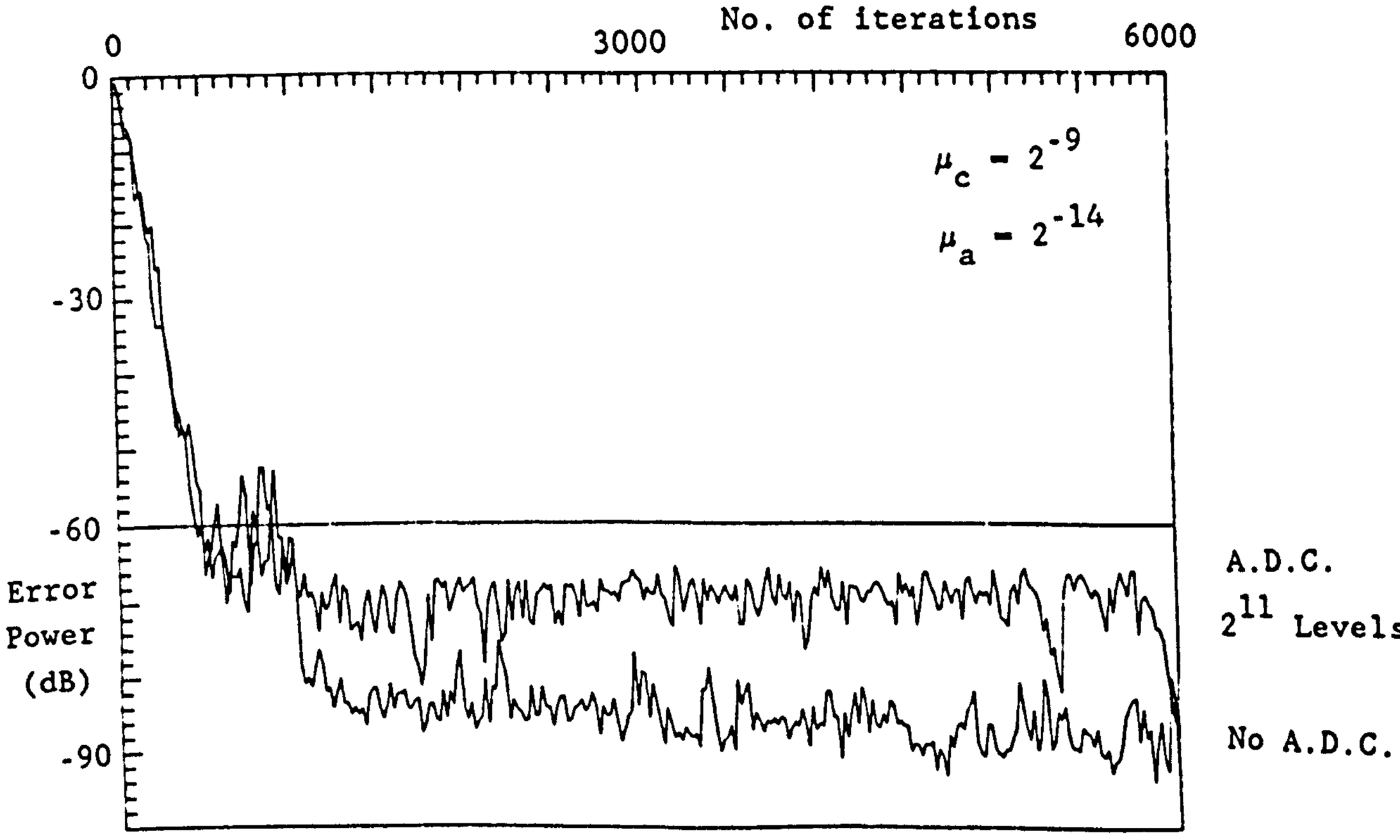


Figure 5.13 Convergence Plot for 200m Line.

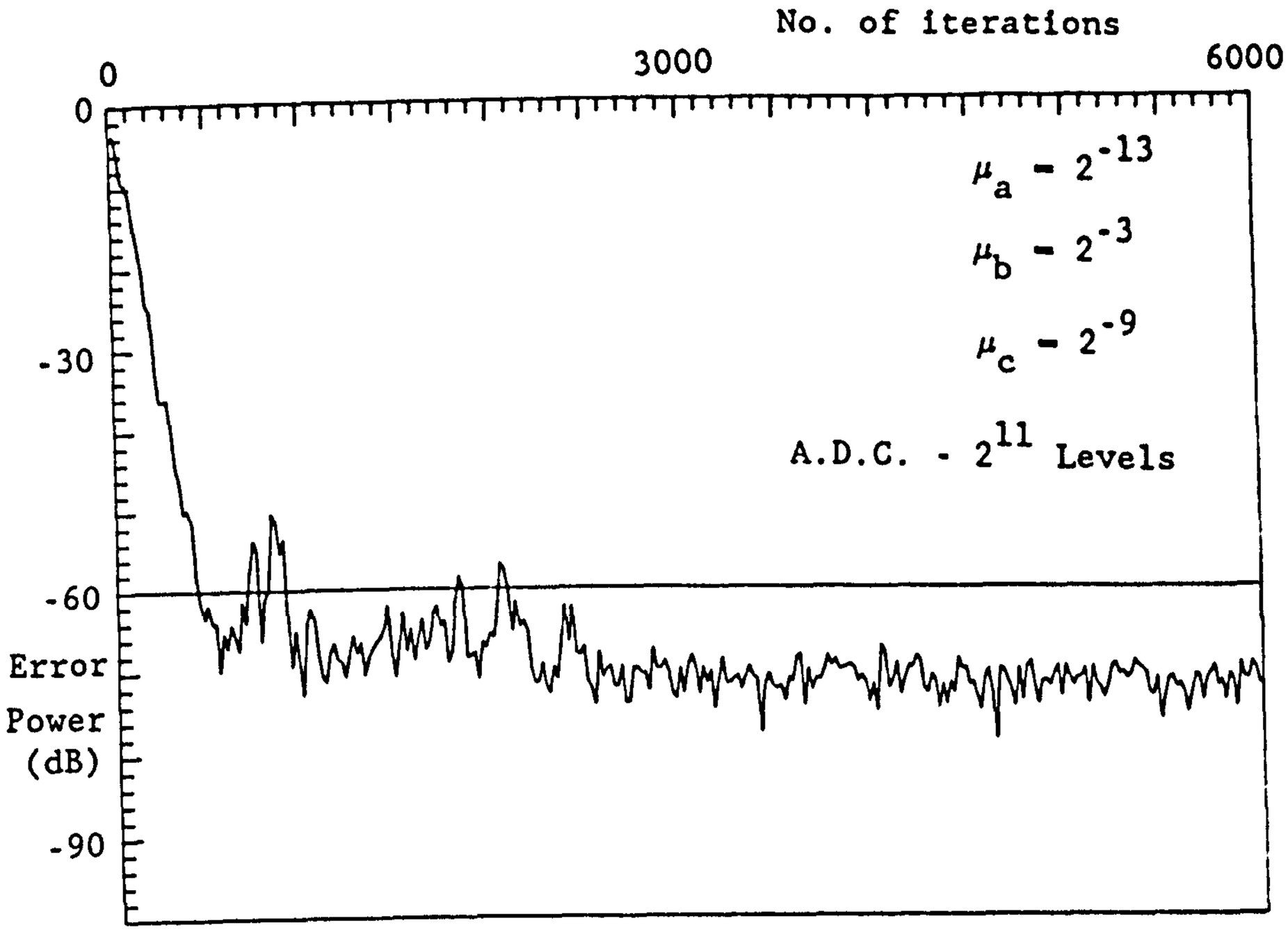


Figure 5.14 Convergence Plot for the complete IIR Echo Canceller.

CHAPTER 6

6.0 THE MASTER-SLAVE ADAPTATION ALGORITHM: A NEW APPROACH TO GRADIENT SEARCH ADAPTIVE IIR FILTERING

In Chapter 3, simple adaptive IIR filtering structures which use gradient search techniques to minimize the mean-square-error were developed. Methods for increasing rates of convergence of the structures were studied in Chapter 4 and shown to offer some improvement. However, the irregular nature of the IIR performance surface severely limits the reductions in convergence times that may be achieved. In addition, gradient search output error minimizing algorithms do not address the problem of locating the global minimum of a possibly multimodal performance surface.

In this chapter, an adaptive IIR filter algorithm is introduced which uses adaptive FIR filters for the adaptation of both the feedforward and feedback coefficients. It is shown that the resulting performance surface is more suitable for gradient search techniques than the corresponding surface for conventional adaptive IIR filters. In addition, results suggest the possible existence of a unique minimum for the filter as a whole. The new algorithm is termed "Master-Slave" as it incorporates an auxilliary adaptive process to the main "Master" filter adaptation using an additional filter (the "Slave"). The two filters are adapted using different adaptation strategies. The Slave filter adapts to minimize the usual output error criterion, while the Master Filter adapts to minimize an equation error criterion. The equation error minimization process is considered in the following section.

6.1 EQUATION ERROR MINIMIZATION

The IIR filter adaptation algorithms discussed in the preceding chapters were all designed to minimize the mean square error obtained from the general adaptive filter structure shown in figure 1.1 (Chapter 1). The related performance surface for IIR filter coefficients

leads to the problems discussed previously in Section 4.1. One approach to avoid these problems is by adapting the filter to minimize a different error criterion. The equation error criterion has been proposed as a possible solution to the output error IIR performance surface problem[60], as the corresponding equation error surface has been shown to be quadratic with respect to all filter coefficients, and hence has a unique minimum for the filter as a whole.

Figure 6.1 shows the adaptive filter configuration for minimization of the equation error. The poles and zeros of the adaptive filter are implemented using two FIR filters, $\hat{A}(z)$ implements the zeros and $\hat{B}(z)$ the poles. The equation error is obtained from :

$$e_q(k) = d(k) - y(k) \tag{6.1}$$

where d(k) is the desired response at time t = kT and y(k) is the adaptive filter output. Without any loss in generality, y(k) can be obtained from:

$$y(k) = \sum_{i=0}^{n} a_i(k)x(k-i) + \sum_{i=1}^{m} b_i(k)d(k-i) \tag{6.2}$$

where $\{a_i(k)\}$ and $\{b_i(k)\}$ are the coefficients of ladder filter implementations of $\hat{A}(z)$ and $\hat{B}(z)$ respectively.

A comparison between equation (6.2) and equation (1.2) shows that the output of the equation error adaptive filter differs from that of the output error adaptive filter by the substitution of the actual desired response d(k) in place of the adaptive filter output y(k) in the right-hand side term. The equation error configuration therefore uses a series/parallel model instead of the parallel, output error minimizing

adaptive filter. The substitution of d(k) for y(k) means that the equation error adaptive filter does not contain any feedback, and hence the structure itself poses no stability problems.

6.2 A SIMPLIFIED CASCADE STRUCTURE FOR EQUATION ERROR MINIMIZATION

The simplest structure for implementation of the equation error adaptive filter is obtained by using FIR ladder filters for both the $\hat{A}(z)$ and $\hat{B}(z)$ filters. Mantey[60] showed that the resulting m.s.e. performance surface is quadratic with respect to all the filter coefficients. Whilst the equation error adaptive filter does not suffer any inherent stability problems, in many applications the transfer function $\hat{A}(z)/\hat{B}(z)$ is the target for the filter adaptation. In such cases, it may prove necessary to restrict the zeros of $\hat{B}(z)$ to remain within the unit circle in the z-plane during adaptation. However, as was shown in Section 2.2, this is not a simple task when the $\hat{B}(z)$ polynomial is realized in direct form.

One method of providing a simple check on the position of the $\hat{B}(z)$ zeros is to implement $\hat{B}(z)$ as a cascade of second order All-Zero sections. The (b_1,b_2) coefficients from each section are confined within the stability triangle shown in figure 1.6 in a similar fashion to the All-Pole sections of the algorithms of Chapter 3. David[61] proposed a cascade structure for equation error minimization. Using simplifications suggested in Chapter 3, a simpler structure is now proposed.

The cascade structure for equation error minimization is shown in figure 6.2. The adaptive filter transfer functions are given by:

$$\hat{A}(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_n z^{-n} \tag{6.3}$$

$$\hat{B}(z) = 1 + b_{1j} z^{-1} + b_{2j} z^{-2} \qquad j=1,2,..,m \tag{6.4}$$

The filter structure shown in figure 6.1 can be redrawn as the structure shown in figure 6.2 by feeding d(k) directly into the $\hat{B}_j(z)$ filter sections.

The realization of the $\hat{B}(z)$ filter as a cascade of second order All-Zero sections changes the shape of the performance surface with respect to the filter coefficients, losing the quadratic error surface of the direct form filter coefficients. However, the cascade structure coefficients will be less sensitive than the direct form structure coefficients when adapting to closely spaced poles in the target model[64]. In addition, the performance surface with respect to the $\{b_{ij}(k)\}$ cascade filter coefficients is much more uniform for FIR filter sections than for the corresponding output error performance surface for recursive $\hat{B}(z)$ sections.

The coefficient adaptation algorithm is a gradient algorithm which uses the method of steepest descent described in Section 2.11. The coefficient updates are expressed as:

$$a_i(k+1) = a_i(k) + \mu_a.(-\nabla_{a_i(k)}) \qquad i=0,1,..,n \tag{6.5}$$

$$b_{ij}(k+1) = b_{ij}(k) + \mu_b.(-\nabla_{b_{ij}(k)}) \qquad \begin{cases} i=1,2 \\ j=1,2,..,m \end{cases} \tag{6.6}$$

where $\nabla_{a_i(k)}$ and $\nabla_{b_{ij}(k)}$ are defined by:

$$\nabla_{a_i(k)} \triangleq \frac{\partial \overline{e_q^2(k)}}{\partial a_i(k)} \tag{6.7}$$

$$\nabla_{b_{ij}(k)} \triangleq \frac{\partial \overline{e_q^2(k)}}{\partial b_{ij}(k)} \tag{6.8}$$

and from figure (6.2), $e_q^2(k)$ is given by:

$$e_q^2(k) = (d'(k) - y'(k))^2 \tag{6.9}$$

Taking $e_q^2(k)$ as a local estimate of $\overline{e_q^2(k)}$, partial differentiation of equation (6.9) with respect to the adaptive filter coefficients yields:

$$\nabla_{a_i(k)} = -2\, e_q(k)\, \alpha_i(k) \tag{6.10}$$

$$\nabla_{b_{ij}(k)} = 2\, e_q(k)\, \beta_{ij}(k) \tag{6.11}$$

where:

$$\alpha_i(k) \triangleq \frac{\partial y'(k)}{\partial a_i(k)} \tag{6.12}$$

$$\beta_{ij}(k) \triangleq \frac{\partial d'(k)}{\partial b_{ij}(k)} \tag{6.13}$$

For the equation error adaptive filter, $y'(k)$ is a function of $\hat{A}(z)$ and $d'(k)$ is a function of $\hat{B}(z)$. Transferring to the z-domain, $D'(z)$ and $Y'(z)$ are obtained from:

$$D'(z) = D(z).\prod_{j=1}^{m}(1 + b_{1j}z^{-1} + b_{2j}z^{-1}) \tag{6.14}$$

$$Y'(z) = X(z).\hat{A}(z) \tag{6.15}$$

Assuming that the $\hat{B}_j(z)$ filter sections remain approximately constant over a small number of iterations by only allowing small coefficient updates, equations (6.12) → (6.15) yield:

$$\alpha_i(z) = X(z).z^{-i} \tag{6.16}$$

$$\beta_{ij}(z) = \frac{D'(z).z^{-i}}{\hat{B}_j(z)} \tag{6.17}$$

The $\alpha_i(k)$ and $\beta_i(k)$ terms can again be generated as the outputs of a set

of time varying filters. The $\alpha_i(k)$ terms do not require any extra computation as they are generated within the delay line of the $\hat{A}(z)$ filter. Using $\hat{B}(z)$ at time $t = (k-i)T$ to approximate $\hat{B}(z)$ at time $t = kT$ allows the $\beta_{ij}(k)$ terms to be generated using m second order All-Pole filters, as in the case of the algorithms derived in Chapter 3. The $\beta_{im}(z)$ terms are generated from :

$$\beta_{im}(z) = D(z).\prod_{j=1}^{m} \hat{B}_j(z) \; \frac{z^{-i}}{\hat{B}_m(z)} \tag{6.18}$$

Hence:

$$\beta_{im}(z) = D(z) \prod_{j=1}^{m-1} (\hat{B}_j(z)) \; .z^{-i} \tag{6.19}$$

The $\beta_{im}(z)$ terms are already generated within the delay line of the $\hat{B}_m(z)$ filter section. Therefore, the number of extra filters required for the generation of the $\beta_{ij}(k)$ gradient terms is reduced to m-1 second-order sections. The complete filter structure is shown in figure 6.3a. The coefficients are updated every sample using:

$$a_i(k+1) = a_i(k) + \mu_a.e(k).\alpha_i(k) \qquad i = 0,1,..,n \tag{6.20}$$

$$b_{ij}(k+1) = b_{ij}(k) + \mu_b.e(k).\beta_{ij}(k) \qquad \begin{cases} i=1,2 \\ j=1,2,..,m \end{cases} \tag{6.21}$$

A block diagram of the cascade structure for equation error minimization proposed by David[61] is given in figure 6.3b. A comparison between figures 6.3a and b shows that the David structure is considerably more complicated than the structure derived in this section.

6.3 PERFORMANCE ASSESSMENT OF THE EQUATION ERROR ADAPTIVE FILTER

Equation error adaptive filters have been compared with output error adaptive filters in the literature[61,62,53,51], and shown to perform very favourably under restricted conditions. To study the performance of the Cascade Equation Error Adaptive Filter, the system identification configuration shown in figure 1.2, Chapter 1 is modified to that shown in figure 6.4.

From figure 6.4 it is seen that the mean squared equation error is obtained from:

$$E_q(z) = X(z).\left[\frac{A(z)\hat{B}(z)}{B(z)} - \hat{A}(z)\right] \tag{6.22}$$

For an output error adaptive filter used in the system identification configuration, the mean square output error is obtained from:

$$E(z) = X(z).\left[\frac{A(z)}{B(z)} - \frac{\hat{A}(z)}{\hat{B}(z)}\right] \tag{6.23}$$

If the adaptive filter is of sufficient order to fully model the plant, both output error minimization and equation error minimization result in the same solution: $\hat{A}(z)$ adapts to cancel $A(z)$, $\hat{B}(z)$ adapts to cancel $B(z)$, and both the output error and the equation error are reduced to zero. The equation error performance surface is, however, much better suited to iterative gradient adaptive techniques than the corresponding output error surface. Consequently, convergence times are considerably faster for the equation error adaptive filter[61,62,51].

Tests using the Cascade Equation Error Adaptive Filter with fixed values of μ_a and μ_b demonstrated that adaptation times were generally much faster than those obtained by the improved adaptation techniques considered in Chapter 4. Figure 6.5 shows a convergence plot

for the Cascade Equation Error minimizing filter adapting to the 4th order plant example in Section 3.3.3. The plant has transfer function:

$$H(z) = \frac{0.01(1 - 1.2z^{-1} + 0.8z^{-2})(1 - 1.5z^{-1} + 0.7z^{-2})}{(1 + 1.2z^{-1} + 0.8z^{-2})(1 + 1.5z^{-1} + 0.7z^{-2})} \qquad (6.24)$$

To obtain a valid comparison with the algorithms of Chapter 3, the error plotted on the vertical scale is that which is generated by substituting $\hat{A}(z)$ and $\hat{B}(z)$ in the output error minimizing configuration shown in figure 1.2. Convergence to 30 dB ERLE was obtained in 4,200 iterations, which compares with 19,000 iterations for the R-Cosθ Cascade output error minimizing adaptive filter in Chapter 4.

The general shapes of the convergence plots for the two adaptive filters are very different in character. The output error minimizing filter tends to reduce the m.s.e. very slowly for a large proportion of the adaptation time, and then the error falls quickly when the filter is near the target. This is a direct result of the error surface shape (See Section 4.1). For the equation error filter, the error drops quickly in the early stages of the adaptation, and then the filter slowly locks in on the target. In general, this is more desirable. However, as the equation error minimizing filter uses a different performance surface to the output error minimizing filter, it is possible that the output error generated by using $\hat{A}(z)/\hat{B}(z)$ constructed from the equation error filter sections will rise in the initial stages of the adaptation.

When the Cascade equation error filter is used to model the widely spaced poles plant of Example 2, Section 3.3, it is found that the filter falls into the same local minimum as the output error cascade filter, with two pairs of split real zeros in $\hat{B}(z)$. By adopting the pole replacement strategy proposed in Chapter 3 for the output error cascade,

the two pairs of split real zeros in $\hat{B}(z)$ are replaced by two complex conjugate zero pairs, thereby removing the local minimum.

While the equation error minimizing filter can be shown to offer faster convergence than the output error minimizing filter in a noise-free, sufficient order modelling situation, there are several disadvantages with the structure which may limit its usefulness. The first problem arises when the desired response is measured in the presence of noise, as in general the coefficient estimates obtained will be biased[63].

Consider the problem of system identification when the desired response D(z) is corrupted by a zero mean, white noise source N(z). The mean square error value, previously obtained from equation (6.22), is now obtained from:

$$E_q(z) = X(z)\left[\frac{A(z)\hat{B}(z)}{B(z)} - \hat{A}(z)\right] + \hat{B}(z)N(z) \qquad (6.25)$$

Equation (6.25) differs from equation (6.22) in that the adaptive coefficients also have to adapt to cancel the $\hat{B}(z)N(z)$ term, which leads to a biased result. This problem may exclude the equation error filter from such applications as echo cancellation, as the far end signal will appear as N(z) to the near end canceller (See Chapter 5) and lead to a non-optimum solution.

A further disadvantage occurs when the equation error adaptive filter is of insufficient order to fully model the target plant. In such a case, a single minimum exists in the performance surface with respect to the equation error filter coefficients[61], while the output error performance surface may contain multiple minima[12]. However, the optimum coefficients for the equation error filter will not, in general, match those of the optimum output error filter. Hence, if the $\hat{A}(z)$ and $\hat{B}(z)$ sections of the optimum equation error filter are implemented as

$\hat{A}(z)/\hat{B}(z)$ in an output error configuration, the resulting filter will not, in general, lie at the global minimum of the output error surface. The problem arises from the difference between equation (6.22) and (6.23). The two expressions generally have different minima when $\hat{A}(z)$ is unable to match A(z) and/or $\hat{B}(z)$ is unable to match B(z).

With the majority of adaptive filtering applications requiring minimization of the output error criterion using an unbiased adaptive filter, opportunities for the use of the equation error adaptive filter are limited.

6.4 MASTER-SLAVE CONFIGURATION

Development of the Master-Slave adaptation algorithm arose from a wish to combine the fast adaptation and more preferable error surface of the equation error minimizing filter, with the unbiased adaptation of the output error minimizing filter. A block diagram of the Master-Slave filter structure is shown in figure 6.6. The filter structure can be divided into two parts, a main filter section consisting of the IIR Master filter Am(z)/Bm(z) and its two adaptive filter sections $\hat{A}(z)$ and $\hat{B}(z)$, and a second section consisting of the auxilliary Slave Filter $\hat{C}(z)$.

When viewed externally,the Master-Slave Filter appears as an output error minimimizing adaptive filter. However, only the slave filter $\hat{C}(z)$ is adapted to minimize the mean square output error, $\overline{e^2(k)}$. The Master Filter is adapted by using the $\hat{A}(z)$ and $\hat{B}(z)$ filters to minimize an internal error, and then updating the Am(z) and Bm(z) filter coefficients with the $\hat{A}(z)$ and $\hat{B}(z)$ filter coefficients respectively.

The Master-Slave Filter is adapted in two stages. Initially, the Master Filter Am(z)/Bm(z) has its coefficients fixed and the Slave Filter is adapted to minimize $\overline{e^2(k)}$. The $\hat{C}(z)$ filter is an FIR ladder Filter and has a quadratic performance surface for the adaptation of its

coefficients. After a number of iterations, the $\hat{C}(z)$ filter coefficients are fixed. $Y_2(z)$ is now a better estimate of D(z) than $Y_1(z)$, and (Am(z)C(z))/Bm(z) gives a unbiased approximation to the unknown system.

The second stage of the adaptation involves adapting $\hat{A}(z)$ and $\hat{B}(z)$ to minimize $\overline{e_q^2(k)}$. During this phase the Am(z)/Bm(z) and $\hat{C}(z)$ filters are fixed, and $\hat{A}(z)/\hat{B}(z)$ forms a model of Am(z)C(z)/Bm(z) (the plant approximation) using equation error minimization. The cascade equation error minimization structure proposed in the previous section is used to adapt the $\hat{A}(z)$ and $\hat{B}(z)$ filters as stability of the $\hat{A}(z)/\hat{B}(z)$ transfer function must be guaranteed. Hence, $\hat{A}(z)$ is implemented as a FIR ladder filter and $\hat{B}(z)$ as a cascade of second order All-Zero sections. The Master Filter Am(z)/Bm(z) is updated after the second adaptation stage using the $\hat{A}(z)$ and $\hat{B}(z)$ filter coefficients. The Master filter structure must be compatible with $\hat{A}(z)/\hat{B}(z)$ for the coefficient transfer, and therefore consists of a cascade of second order All-Pole sections and a direct form All-Zero section. This is the modified cascade structure used in Section 3.2.2.

Following the updating of the Master Filter coefficients, the adaptation algorithm reverts to the first stage of adaptation and the Slave Filter is adapted again. Each additional adaptation cycle produces a Master Filter which is a more accurate representation of the target plant. The Slave Filter, therefore, 'leads' the Master towards its target.

The Master-Slave filter structure enables an IIR filter to be adapted using three adaptive FIR filters. Two of the filters ($\hat{A}(z)$,$\hat{C}(z)$) have quadratic performance surfaces. Implementation of the zeros of $\hat{B}(z)$ as a cascade of second order All-Zero sections means that the performance surface with respect to the $\hat{B}(z)$ filter coefficients is no longer quadratic. However, as results from the previous section illustrated, the surface is still considerably more regular than that

for IIR filter denominator coefficients.

The new algorithm, therefore, offers considerably faster convergence using gradient search techniques than that obtained for a direct output error minimizing IIR filter alone. In addition, a major disadvantage of equation error minimization is that noise in the desired response results in biased coefficient estimates. For the Master-Slave algorithm, use of the Slave filter $\hat{C}(z)$ allows the Master to adapt to an unbiased solution, thereby combining the advantages of equation error and output error minimization.

6.5 ALGORITHM DERIVATION

The derivation of the Master-Slave filter adaptation algorithm is broken down into the two adaptive processes. Section 6.5.1 considers adaptation of the $\hat{C}(z)$ Slave Filter, and Section 6.5.2 adaptation of the $\hat{A}(z)$ and $\hat{B}(z)$ filters. All filters are adapted using the method of steepest descent.

6.5.1 Slave Filter Adaptation

A block diagram of the $\hat{C}(z)$ filter adaptation is shown in figure 6.7. The slave filter has the transfer function:

$$\hat{C}(z) = c_0 + c_1 z^{-1} + c_2 z^{-2} + ..+ c_p z^{-p} \qquad (6.26)$$

Using the method of steepest descent, the filter is adapted to minimize $\overline{e^2(k)}$. The coefficients are updated using:

$$c_i(k+1) = c_i(k) + \mu_c . \left(\frac{-\partial \overline{e^2(k)}}{\partial c_i(k)} \right) \qquad (6.27)$$

From figure 6.7 the error signal e(k) is obtained from:

$$e(k) = d(k) - y_2(k) \tag{6.28}$$

Taking $e^2(k)$ as a local estimate of $\overline{e^2(k)}$, $\partial\overline{e^2(k)}/\partial c_i(k)$ is obtained by taking the square of both sides of equation (6.28), and then differentiating with respect to $\{c_i\}$ giving:

$$\frac{\partial e^2(k)}{\partial c_i(k)} = -2.e(k).\gamma_i(k) \tag{6.29}$$

where:

$$\gamma_i(k) \triangleq \frac{\partial y_2(k)}{\partial c_i(k)} \tag{6.30}$$

From figure 6.7, $Y_2(z)$ can be written as:

$$Y_2(z) = Y_1(z).\hat{C}(z) \tag{6.31}$$

Hence, $\gamma_i(k)$ is given by:

$$\gamma_i(k) = y_1(k-i) \tag{6.32}$$

The positioning of the slave filter after the master filter enables the $\gamma_i(k)$ terms to be generated within the delay line of the $\hat{C}(z)$ filter and not require any extra computation. If the slave filter and master filter positions were exchanged, this would no longer be the case.

The $\{c_i\}$ filter coefficients are updated every sample using:

$$c_i(k+1) = c_i(k) + \mu_c.e(k).\gamma_i(k) \qquad i = 1,p \tag{6.33}$$

where $e(k)$ and $\gamma_i(k)$ are obtained from equations (6.28) and (6.32) and μ_c is the adaptation constant.

6.5.2 Master Filter Adaptation

A block diagram for the $\hat{A}(z)$ and $\hat{B}(z)$ filter adaptation is shown in figure 6.8. The $\hat{A}(z)$ and $\hat{B}(z)$ filters are adapted to minimize $\overline{e_q^2(k)}$ using the cascade equation error minimization structure proposed in section 6.2. The two filters have transfer functions:

$$\hat{A}(z) = a_0 + a_1 z^{-1} + ,\ldots, + a_n z^{-n} \tag{6.34}$$

$$\hat{B}(z) = \prod_{j=1}^{m} (1 + b_{1j} z^{-1} + b_{2j} z^{-2}) \tag{6.35}$$

A full derivation of the adaptation algorithm for the filter coefficients is given in section 6.2. The coefficients are updated using:

$$a_i(k+1) = a_i(k) + \mu_a.e_q(k).\alpha_i(k) \qquad i = 0,1,..,n \tag{6.36}$$

$$b_{ij}(k+1) = b_{ij}(k) + \mu_b.e_q(k).\beta_{ij}(k) \qquad \begin{cases} i = 1,2 \\ j = 1,2,..,m \end{cases} \tag{6.37}$$

The $\alpha_i(k)$ terms are generated within the $\hat{A}(z)$ filter and the $\beta_{ij}(k)$ terms require m-1 extra second order All-Pole sections. When the $\hat{A}(z)$ and $\hat{B}(z)$ filters have adapted, their coefficients are duplicated in the Master Filter. The complete filter structure is shown in figure 6.9.

6.6 PERFORMANCE ASSESSMENT OF THE MASTER-SLAVE ADAPTIVE FILTER

The performance of the Master-Slave Adaptive filter was studied using the system identification structure shown in figure 1.2. The Adaptive Filter output is given by:

$$Y_2(z) = X(z)\left[\frac{Am(z)\hat{C}(z)}{Bm(z)}\right] \tag{6.38}$$

Hence if the target plant has transfer function H(z) where:

$$H(z) = \frac{A(z)}{B(z)} \tag{6.39}$$

the error signal e(k) is obtained from:

$$E(z) = X(z)\left[\frac{A(z)}{B(z)} - \frac{Am(z)\hat{C}(z)}{Bm(z)}\right] \tag{6.40}$$

To minimize the mean square value of e(k), the slave filter $\hat{C}(z)$ adapts such that:

$$\hat{C}(z) \rightarrow \frac{A(z)Bm(z)}{B(z)Am(z)} \tag{6.41}$$

If the Master filter Am(z)/Bm(z) is of sufficient order to fully match the plant, $\hat{C}(z) \rightarrow 1.0$ as the filter converges.

To keep the adaptation of the slave filter stable as the power level in the output of the Master Filter varies, a low pass filtering operation is used to normalize the μ_c adaptation parameters. μ_c is calculated from [9]:

$$\mu_c = \frac{\mu_c^f}{p.r(k)} \tag{6.42}$$

Where μ_c^f is a constant, p is the order of $\hat{C}(z)$ and r(k) is an estimate

of the power in $Y_1(z)$, the input to $\hat{C}(z)$. r(k) is obtained from:

$$r(k) = \rho.r(k-1) + (1-\rho).y_1^2(k) \qquad (6.43)$$

ρ is a 'forgetting' factor which determines the bandwidth of the low pass filter operation.

When the $\hat{C}(z)$ filter adaptation is started, transients will exist in $Y_1(z)$ caused by the sudden change in the Master Filter coefficients as they are updated. It was decided to ignore the transients and use a small number of iterations for each adaptation cycle, thereby reducing the coefficient step. An alternative approach would be to allow time for the transients to settle before $\hat{C}(z)$ is adapted. When the Master Filter coefficients are updated, the $\hat{C}(z)$ filter may either be adapted from the final $\{c_i(k)\}$ coefficient values of the previous adaptation, or alternatively, the coefficients can be reset to $c_1 = 1.0$, $c_{2,3..p} = 0.0$. Adaptation was found to be quicker using the second method.

Using the system identification configuration, examples are presented in the following sections to demonstrate the advantages the Master Slave Adaptive Filter has over the output error minimizing adaptive IIR filters.

6.6.1 Sufficient Order Adaptive Filter Examples

(1) Second Order Example

The plant chosen has the transfer function:

$$H(z) = \frac{1.0}{(1 + 1.8z^{-1} + 0.95z^{-2})} \qquad (6.44)$$

This is a second order All-Pole plant with poles of radius 0.975 and

angles +/- 157.4°. The plant has been previously used to assess the output error minimizing algorithms of Chapter 4. The results obtained showed it to be a very difficult plant to model. The performance surface for the output error adaptive filter coefficients is shown in figure 1.7d. The steep valley that characterises the surface causes severe problems to gradient-based adaptation algorithms. The Valley-Search algorithm achieved the fastest convergence of the Chapter 4 algorithms taking 10,000 iterations for 30 dB ERLE convergence (Section 4.3). However, the computation required for the algorithm is considerable.

Error convergence plots for the Master-Slave filter adapted using 16 (c_i) coefficients are shown in figure 6.10 a,b. The Master and Slave sections were each adapted for blocks of 200 iterations at a time (ie. one complete adaptation cycle took 400 iterations). Figure 6.10a shows the squared output error for the Master-Slave Filter. Figure 6.10b shows the squared output error obtained by the Master Filter alone.

Convergence to 30 dB ERLE was achieved by the Master-Slave Filter in 5,500 iterations. This is considerably faster than all the Chapter 4 algorithms. In addition, it is seen from figure 6.10a that the initial error is reduced in amplitude very quickly. Figure 6.10c shows a convergence plot for the White/Stearns algorithm adapting to the same plant. A comparison of figures 6.10a and c shows the effect of the differing error surfaces. The fast initial adaptation of the Master-Slave Filter is due to the equation error performance surface for the Master Filter. This is shown in figure 6.10b. The output error for the Master Filter is seen to drop quickly. As the filter is updated in two stages, the error plots for the Master and the Slave filter outputs rise suddenly when the master filter coefficients are updated.

The coefficient track for the (b_1,b_2) filter coefficients is shown in figure 6.11. It is seen that the valley which restricted the output error IIR adaptive filters (figure 4.1c) is again in evidence

with the coefficient track bouncing from side to side. However, the coefficient steps taken are larger, the valley is less steep and convergence is consequently much faster for the Master-Slave Filter.

(2) Fourth Order Example

To assess the performance of the Master-Slave Filter in a higher order adaptive filter example, the fourth-order pole/zero plant of Section 3.3.3 was used. With the transfer function zeros implemented in direct form, the plant transfer function H(z) is given by:

$$H(z) = \frac{0.01 - 0.027z^{-1} + 0.033z^{-2} - 0.0207z^{-3} + 0.0056z^{-4}}{(1.0 + 1.2z^{-1} + 0.8z^{-2})(1.0 + 1.5z^{-1} + 0.7z^{-2})} \qquad (6.45)$$

As in the previous example, the Master and Slave Filters were adapted in 200 iteration blocks and 16 $\{c_i\}$ coefficients were used.

Figures 6.12a,b show error convergence plots for the output of the Slave filter and Master filter respectively. For this example it was found that the output error of the Master-Slave Filter fell quickly in the early stages of the adaptation reaching 20 dB ERLE in 3000 iterations and 30 dB ERLE in about 5000 iterations, at which point the Master filter transfer function was:

$$\hat{H}(z) = \frac{0.010 - 0.021z^{-1} + 0.037z^{-2} - 0.022z^{-3} + 0.003z^{-4}}{(1 + 1.234z^{-1} + 0.810z^{-2})(1 + 1.441z^{-1} + 0.655z^{-2})} \qquad (6.46)$$

It is seen that the Master filter coefficient values are very close to the plant coefficient values (equation (6.45)). However, as the adaptation continued, the $\hat{A}(z)$ and $\hat{B}(z)$ filter coefficients were unable to converge on their targets very quickly and it took 76,000 iterations to obtain 30 dB ERLE from the Master Filter. The overall convergence of

the Master and Slave therefore also increased very slowly reaching 40 dB ERLE convergence in 80,000 iterations.

One method for improving the rate of convergence is to use the Master-Slave Filter for 5,000 iterations to adapt quickly near the solution, and then switching to normal output error minimization using the Master Filter alone (discarding the Slave). The error convergence plots from Chapter 3 show that the RLMS IIR Filter rapidly lowers the squared error when near the solution.

Switching in the output error minimization after 5,000 iterations using the modified cascade structure of Chapter 3 results in 6,000 iterations convergence to 30dB ERLE, with 40dB ERLE obtained in 8,000 iterations and 50dB in 10,000 iterations. The fastest previous convergence to this plant was 19,000 iterations for 30dB ERLE obtained by the R-Cosθ algorithm in section 4.4.

6.6.2 Insufficient Adaptive Filter Examples

If the Master Filter is of sufficient order to fully model the plant, tests have shown that the Master-Slave adaptation algorithm drives the Master Filter towards the minimum that an unbiased equation error minimizing filter would adapt to. As shown in Section 6.3, this could lead to an increase in output error from the Master filter output. However, with the Slave Filter providing extra filtering power, it is probable that a smaller m.s.e. signal will result from a Master Slave Filter than a parallel RLMS IIR filter, which may become trapped in a local minimum. To examine the insufficient adaptive filter problem, two previously published examples from the literature have been studied.

(1) Example 1

Johnson[29] published an insufficient adaptive filter example to demonstrate the non-gradient nature of the Feintuch

Algorithm. The example has since been used to assess the performance of several other algorithms[31,44,68]. Using the System Identification Configuration, a first-order plant with transfer function $\hat{H}(z)$, given by:

$$\hat{H}(z) = \frac{a_0}{1 + b_1 z^{-1}} \tag{6.47}$$

is used to model a second order plant with transfer function:

$$H(z) = \frac{0.05 - 0.4z^{-1}}{1 - 1.1314z^{-1} + 0.25z^{-2}} \tag{6.48}$$

The performance surface with respect to the two adaptive coefficients in a RLMS filter is shown in figure 6.13. It is seen that the surface is bimodal, and hence an RLMS filter will adapt to the minimum closest to the initial filter coefficients. The global minimum denoted by * is situated at $(a_0,b_1) = (-0.311,-0.906)$, and the local minimum denoted by + is at $(a_0,b_1) = (0.114,0.519)$. Using an adaptive filter to minimize the equation error criterion, the performance surface is quadratic with respect to all the filter coefficients and the minimum was found to be at $(a_0,b_1) = (0.04,-0.88)$.

A series of modelling tests were carried out using different initial coefficients for the Master Filter. In each case the Master-Slave algorithm drove the filter coefficients to the equation error surface minimum. A coefficient track for the Master Filter initially adapted from the output error surface local minimum position is shown in figure 6.14.

(2) Example 2

This example was used by Stearns[11] and Etter[12,13]. The adaptive filter is a second-order recursive section, and the plant being

modelled consists of a single zero. The plant transfer function H(z) is:

$$H(z) = 1 + 10z^{-1} \tag{6.49}$$

and the adaptive filter transfer function $\hat{H}(z)$:

$$\hat{H}(z) = \frac{a_0}{(1 + b_1 z^{-1} + b_2 z^{-2})} \tag{6.50}$$

The performance surface with respect to the (b_1, b_2) adaptive filter coefficients was shown in figure 1.8. It is seen that the surface is again bimodal, with the global minimum situated at:

$$\hat{H}(z) = \frac{4.5}{1 - 1.0z^{-1} + 0.5z^{-2}} \tag{6.51}$$

and the local minimum at:

$$\hat{H}(z) = \frac{-3.4}{1 + 1.1z^{-1} + 0.5z^{-2}} \tag{6.52}$$

For an equation error adaptive filter, the performance surface minimum was found to be situated at:

$$\hat{H}_{eq}(z) = \frac{1.05}{1 - 0.1z^{-1} + 0.0z^{-2})} \tag{6.53}$$

A series of modelling tests were carried out using different initial Master Filter Coefficients. As with the previous example, the Master Filter was driven to the equation error minimum in every case. Figure 6.15 shows a coefficient track for the Master Filter adapting from the output error local minimum position.

6.2.3 Assessment of Insufficient Order Examples

In both examples (1) and (2), the output error obtained by the Master Filter alone rose when the filter adapted. This can be seen in figures 6.14 and 6.15. The Master filter coefficients adapt from the local minimum to a higher m.s.e. value. The reduction in error power achieved by the Master-Slave Filter as a whole, therefore is dependent on the error reduction obtainable from the Slave when the Master Filter has adapted to the equation error minimum position. This in turn will depend upon the length of the Slave Filter, and the nature of the plant. For example, the plant transfer function in example (2) consists of a single zero which the All-Pole Master filter will model poorly. However, the addition of the All-Zero Slave Filter to the Master Filter will lower the output error significantly.

As the Master-Slave algorithm will drive the Master Filter to the equation error surface minimum, the algorithm may prove unsuitable for uses in cases where the Master filter is insufficient, and the filter coefficients are the parameters required for the adaptation. However, by initially using the Master-Slave algorithm, and then adapting the Master Filter to locate the nearest local minimum in the output error performance surface, a more optimum set of filter coefficients are obtained. For examples (1) and (2), adapting the Master filter using this strategy will result in the coefficients finding the global minimum of the output error performance surface. This will occur irrespectively of the choice of initial Master filter coefficients. Adaptation to a number of other bimodal error surface plants using this adaptation strategy proved successful with the Master Filter adapting to the global minimum in each case. Use of this method, therefore, will possibly locate the global minimum of a multimodal performance surface.

6.7 CONCLUSIONS

In this chapter, the equation error criterion has been considered as an alternative to the output error criterion for the adaptation of IIR filters. The advantages and disadvantages of using the equation error criterion have been highlighted and a new, simple IIR adaptation algorithm has been proposed for equation error minimization. The performance of the new algorithm has been assessed using the system identification configuration.

A new adaptive filter algorithm has been proposed which incorporates the advantages of output error minimization and equation error minimization processes. By using FIR filters for the adaptation of both the filter feedforward and feedback coefficients, results have demonstrated that the Master-Slave Filter offers faster convergence than conventional output error minimizing RLMS adaptive filters. In addition, for cases in which the Master Filter is of insufficient order to fully model the unknown system, the adaptation algorithm drives the Master Filter to a unique minimum in the equation error surface. From this position, adaptation of the Master Filter coefficients using output error minimization enables the filter to adapt to the output error surface global minimum in the examples presented. The algorithm developed in this chapter is therefore a possible method of avoiding minima in the output error performance surface.

In the introduction to Chapter 1, the potential advantages of using adaptive IIR filters in preference to adaptive FIR filters were discussed. The Master-Slave filter offers a viable alternative to FIR filters with its speed of adaptation and simple, stable filter structure. In Section 6.1.1, Example 1, a system identification problem was presented using a target plant with poles very near $z = 1$ in the z-plane. This plant has a very long, slowly decaying impulse response. The second-order Master-Slave Filter Structure used for this example is

shown in figure 6.9. In terms of computation, the Master-Slave Filter is considerably simpler than a 64 coefficient FIR ladder filter. When an adaptive 64 coefficient FIR filter was used to model this plant, only 12dB ERLE could be achieved when the optimum filter had been reached. This compares with 170,000 iterations convergence to 30dB ERLE by the standard output error minimizing RLMS filter (figure 4.1c). For a similar order of computation, the Master-Slave filter achieves 30dB ERLE convergence in 5,500 iterations, illustrating the advantages of the algorithm and the filter structure.

With its less troublesome error surface the Master-Slave algorithm will facilitate adaptation for high order adaptive filters, for which the output error surface for RLMS filters will probably prohibit the use of direct gradient techniques.

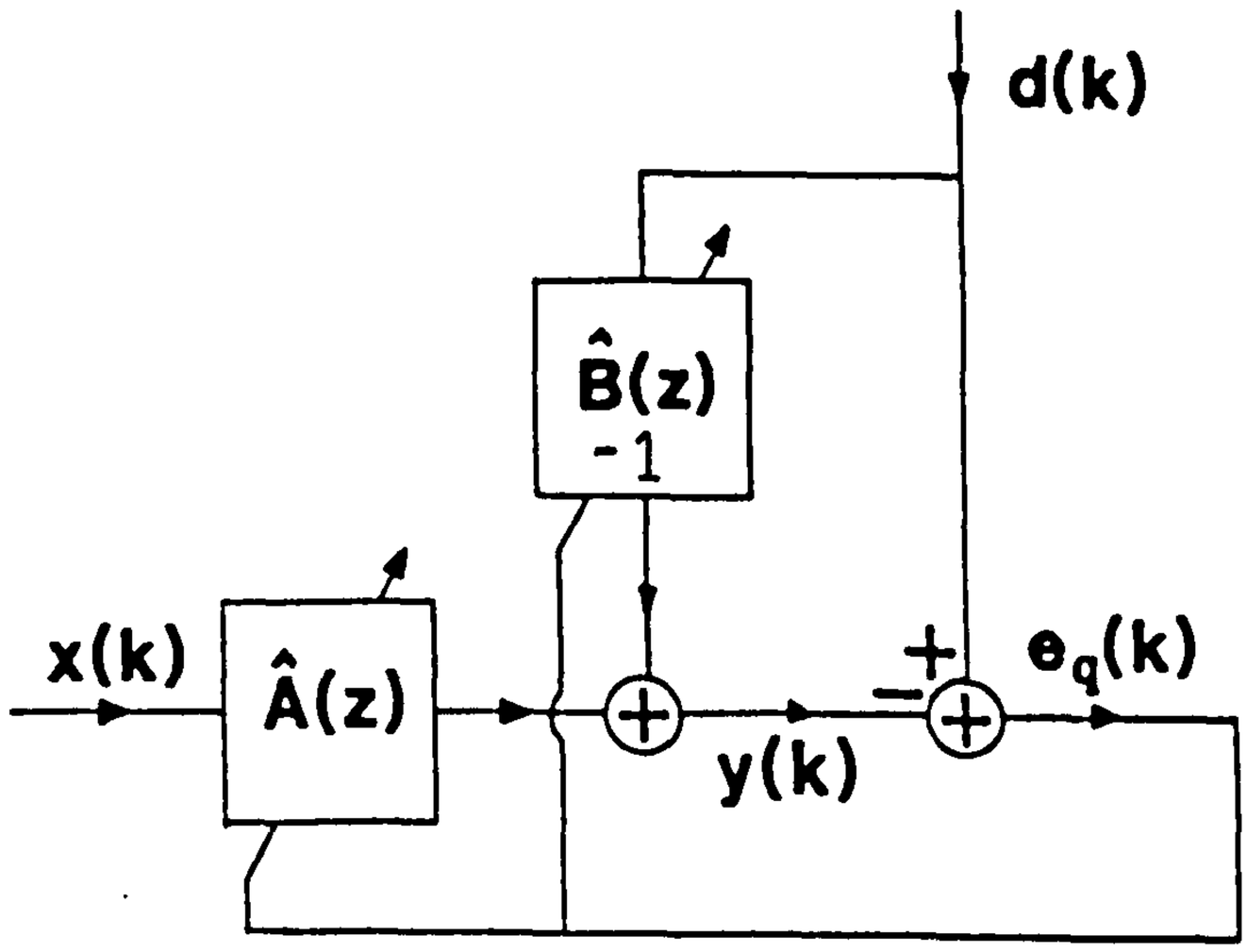


Figure 6.1 Equation Error Adaptive Filter.

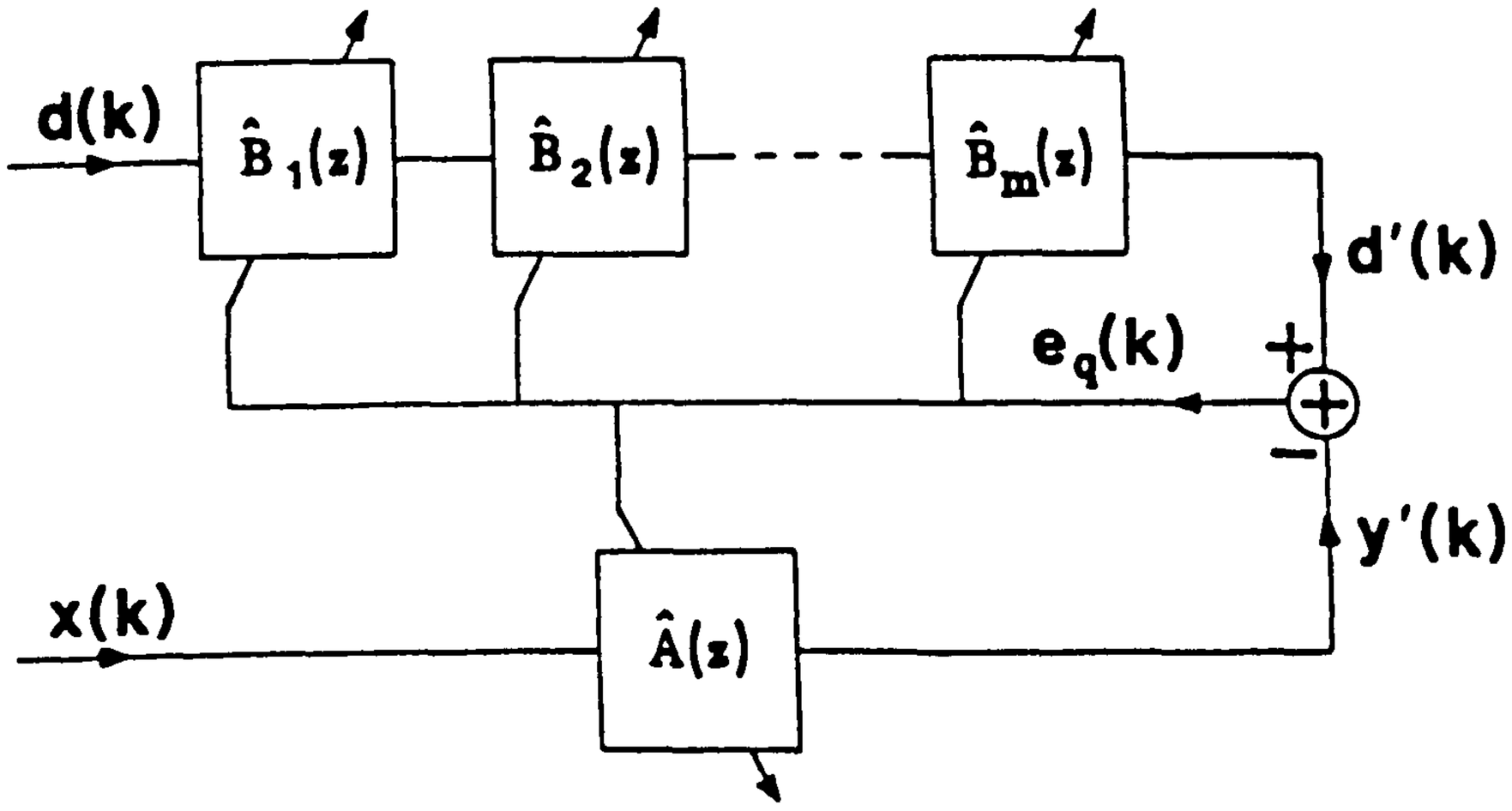
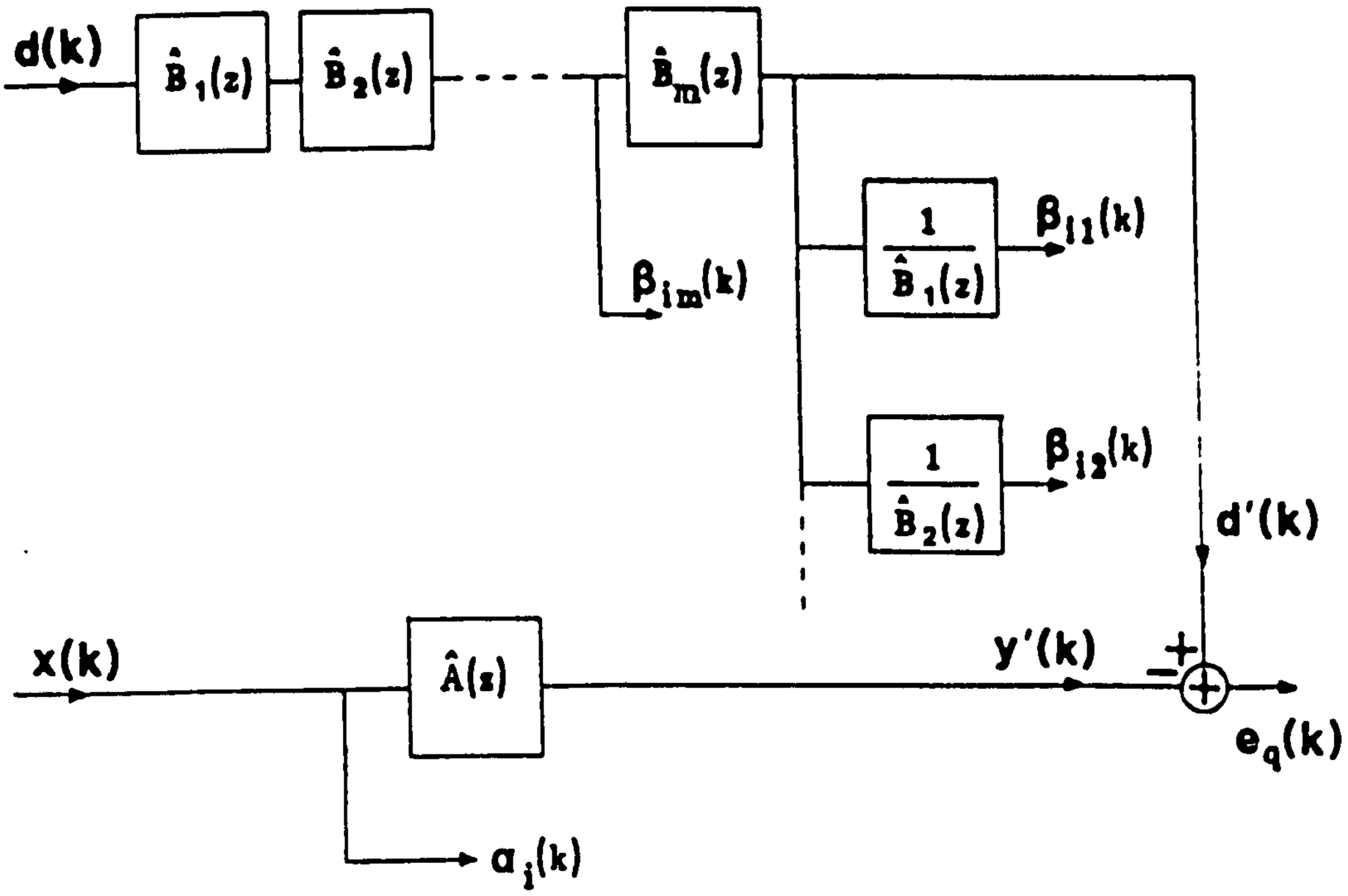
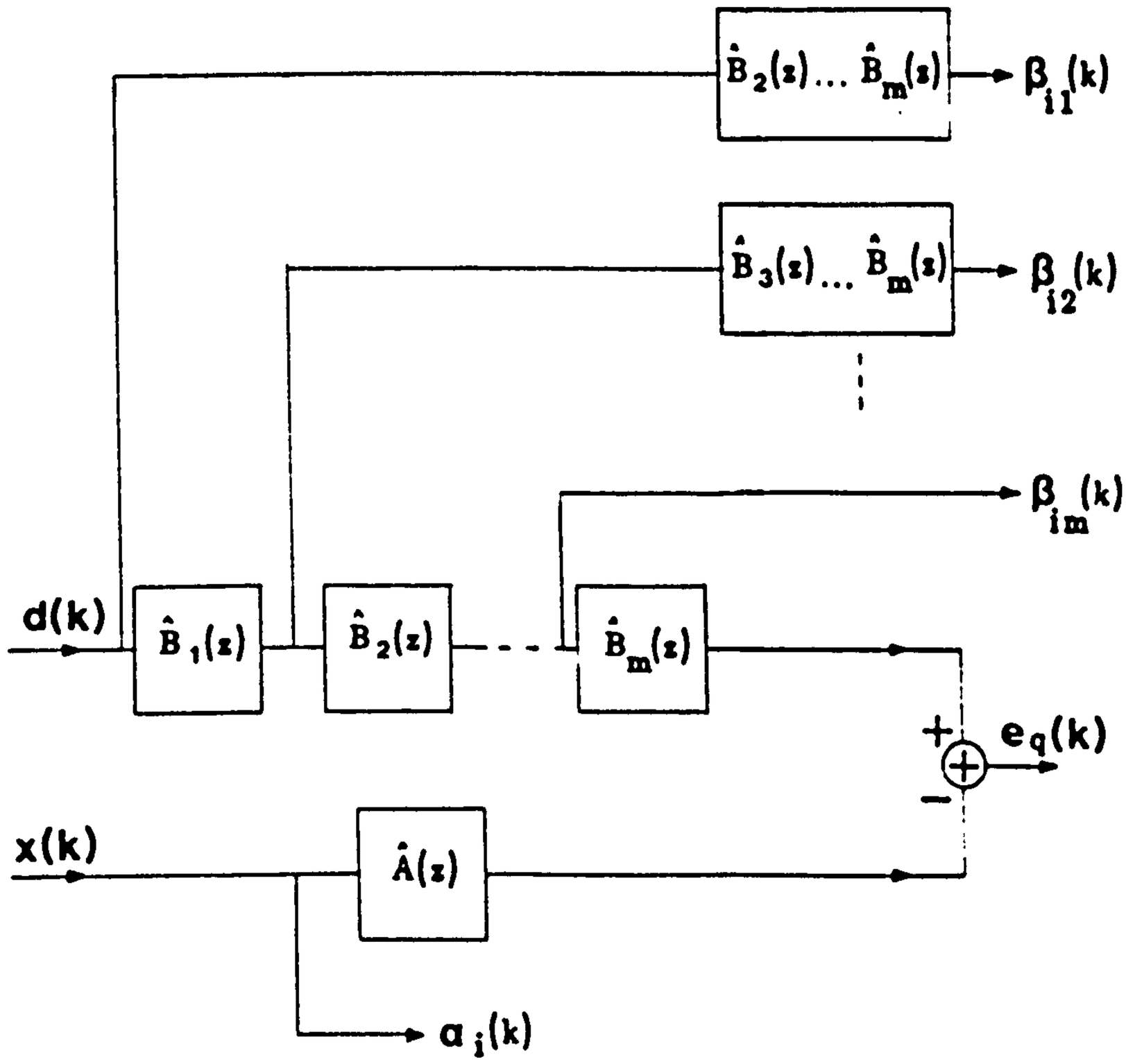


Figure 6.2 Adaptive Cascade Filter for Equation Error Minimization.



a) Simplified Cascade Structure for Equation Error Minimization.



b) David Structure.

Figure 6.3 Adaptive Cascade Filter Structures.

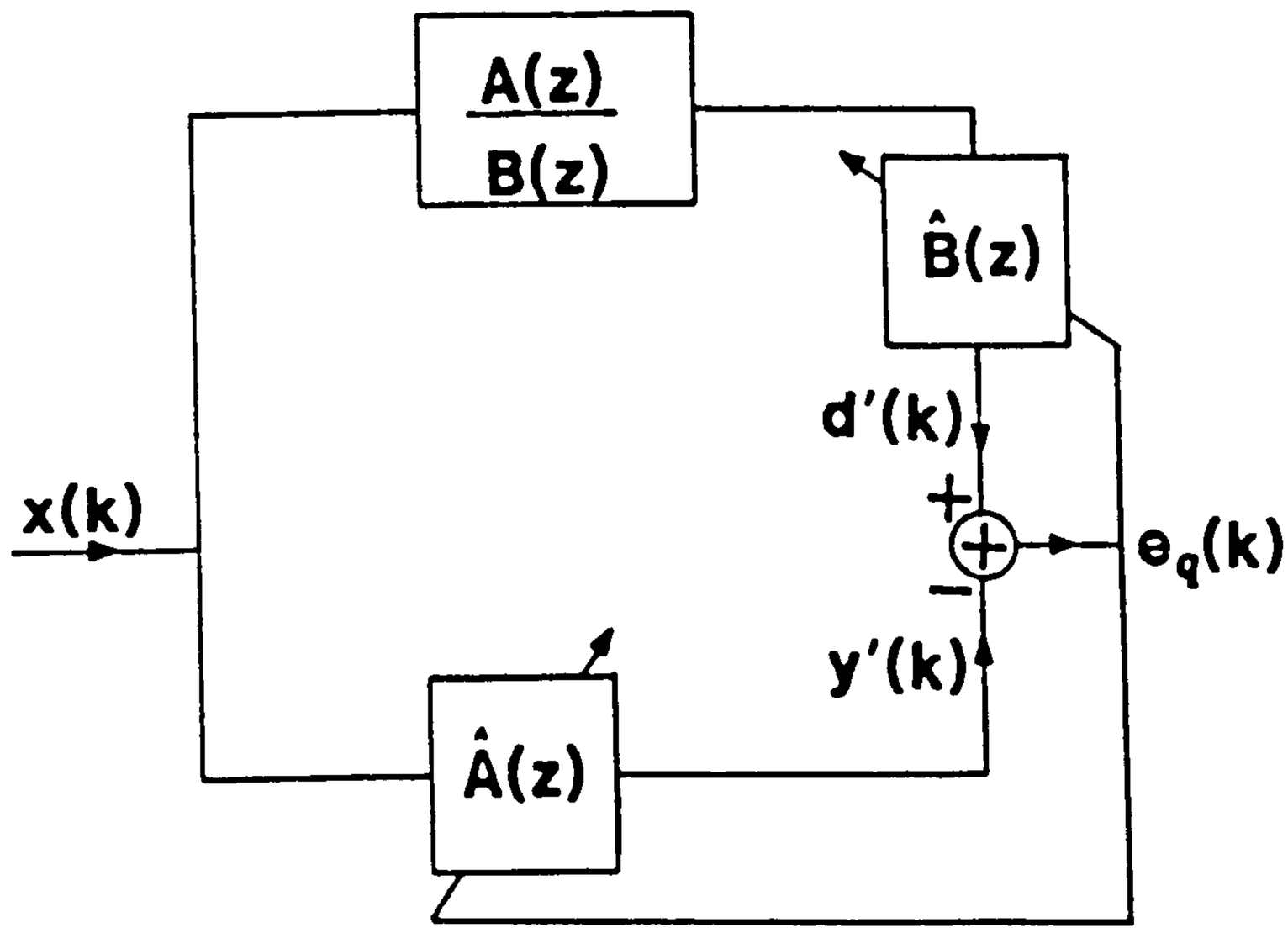


Figure 6.4 System Identification via Equation Error Minimization.

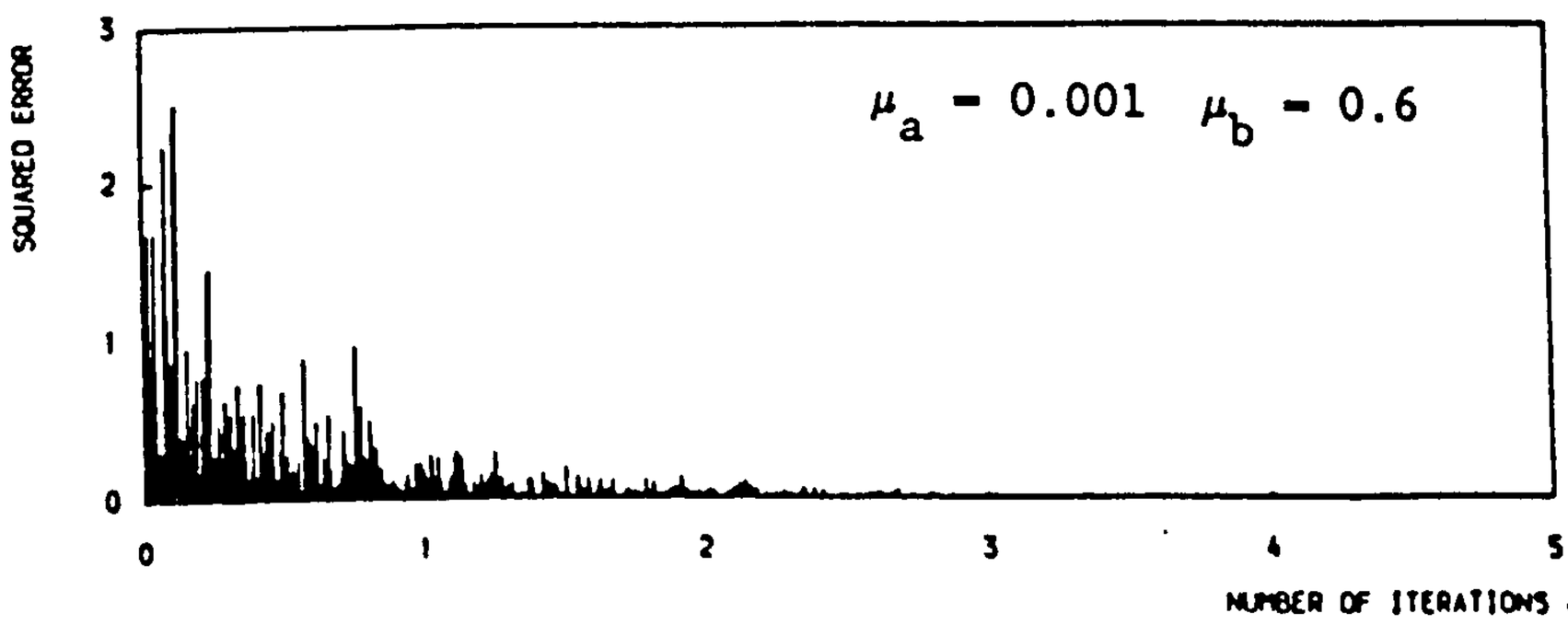


Figure 6.5 Output-Error Convergence Plot for Fourth Order Example.

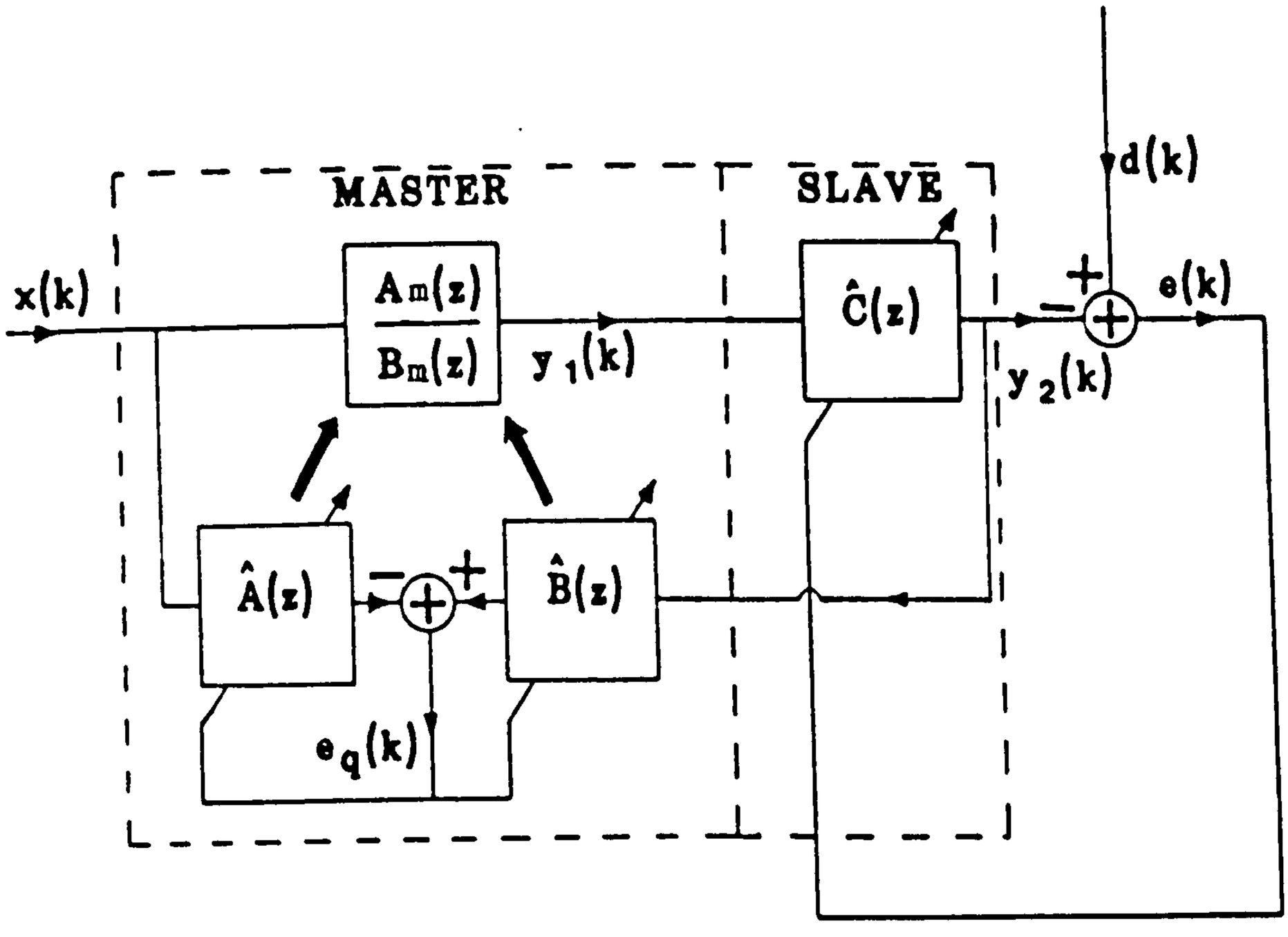


Figure 6.6 Master-Slave Adaptive Filter.

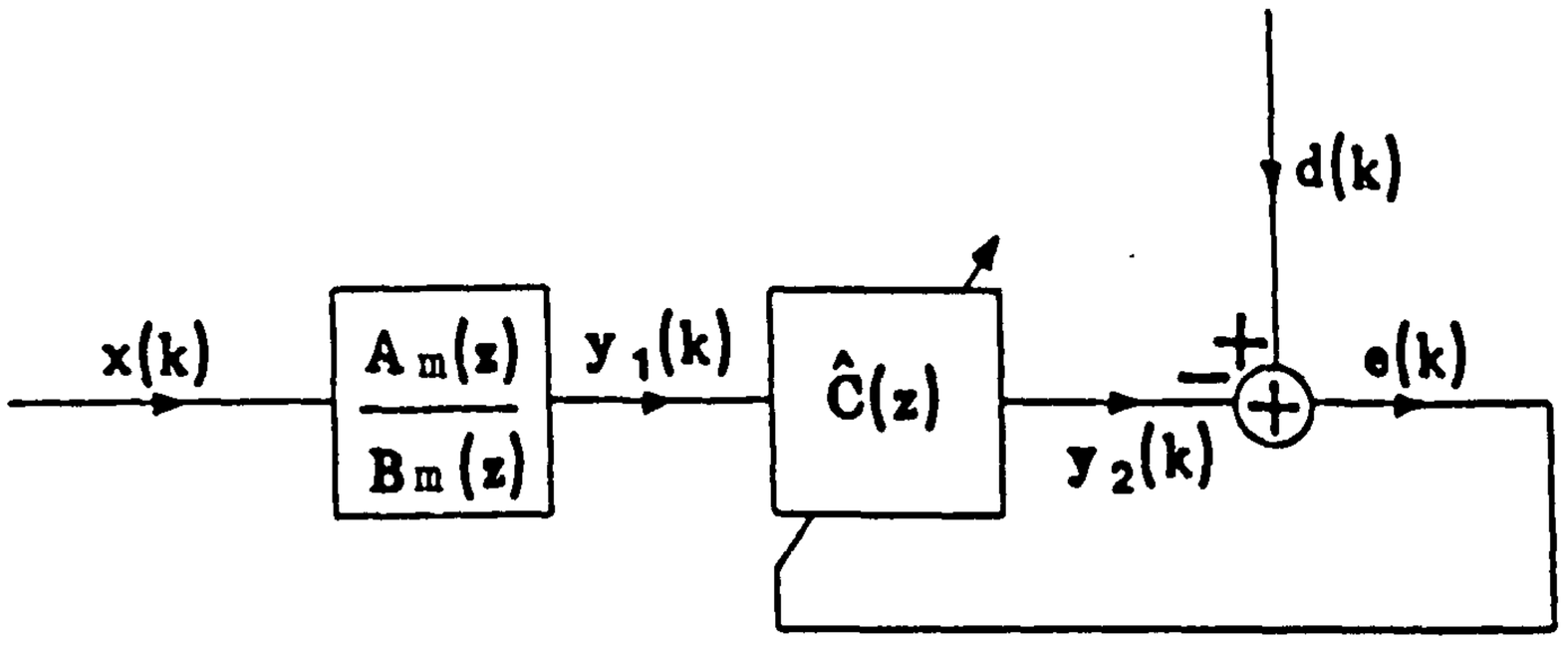


Figure 6.7 Slave Filter Adaptation.

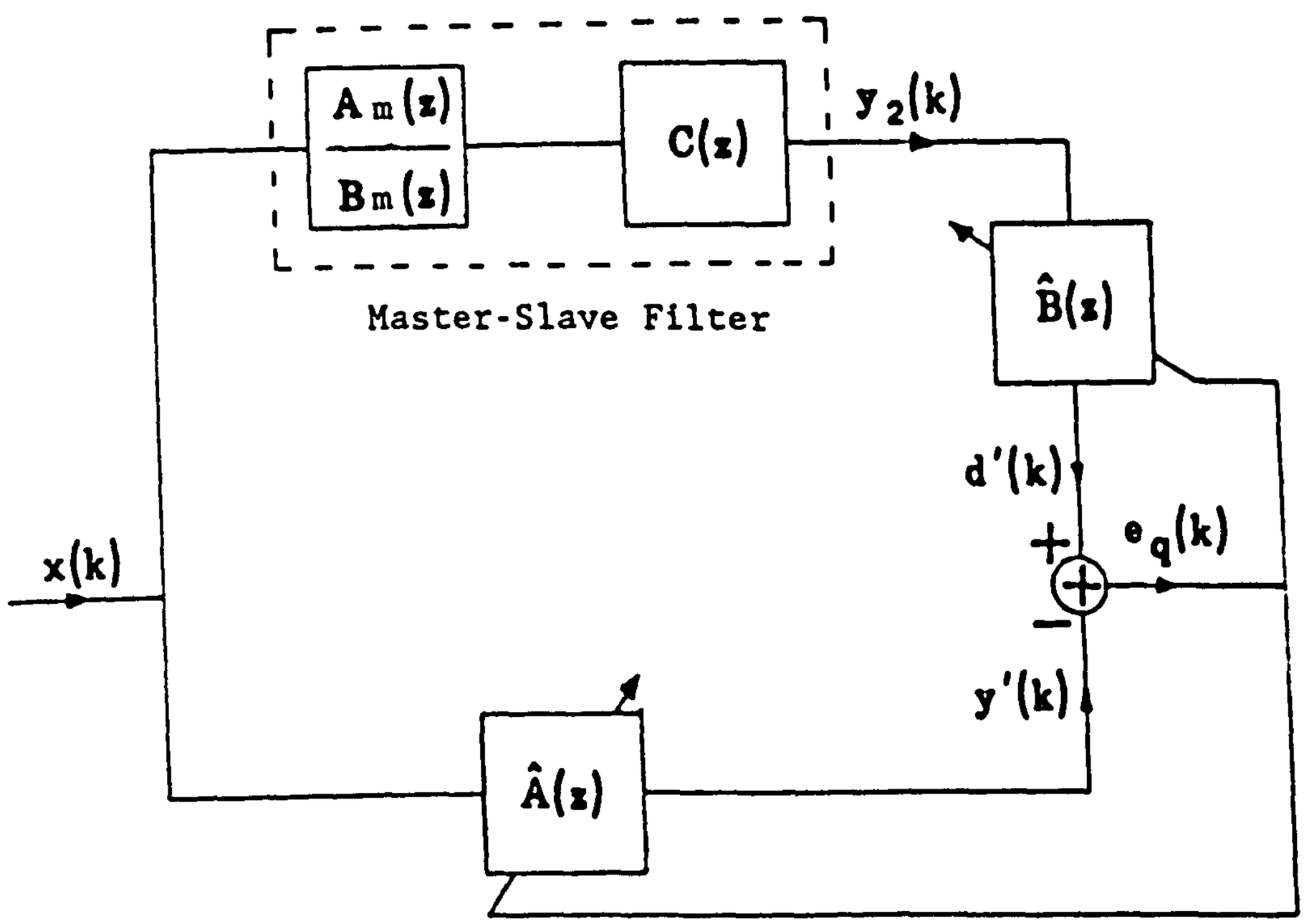


Figure 6.8 Master Filter Adaptation.

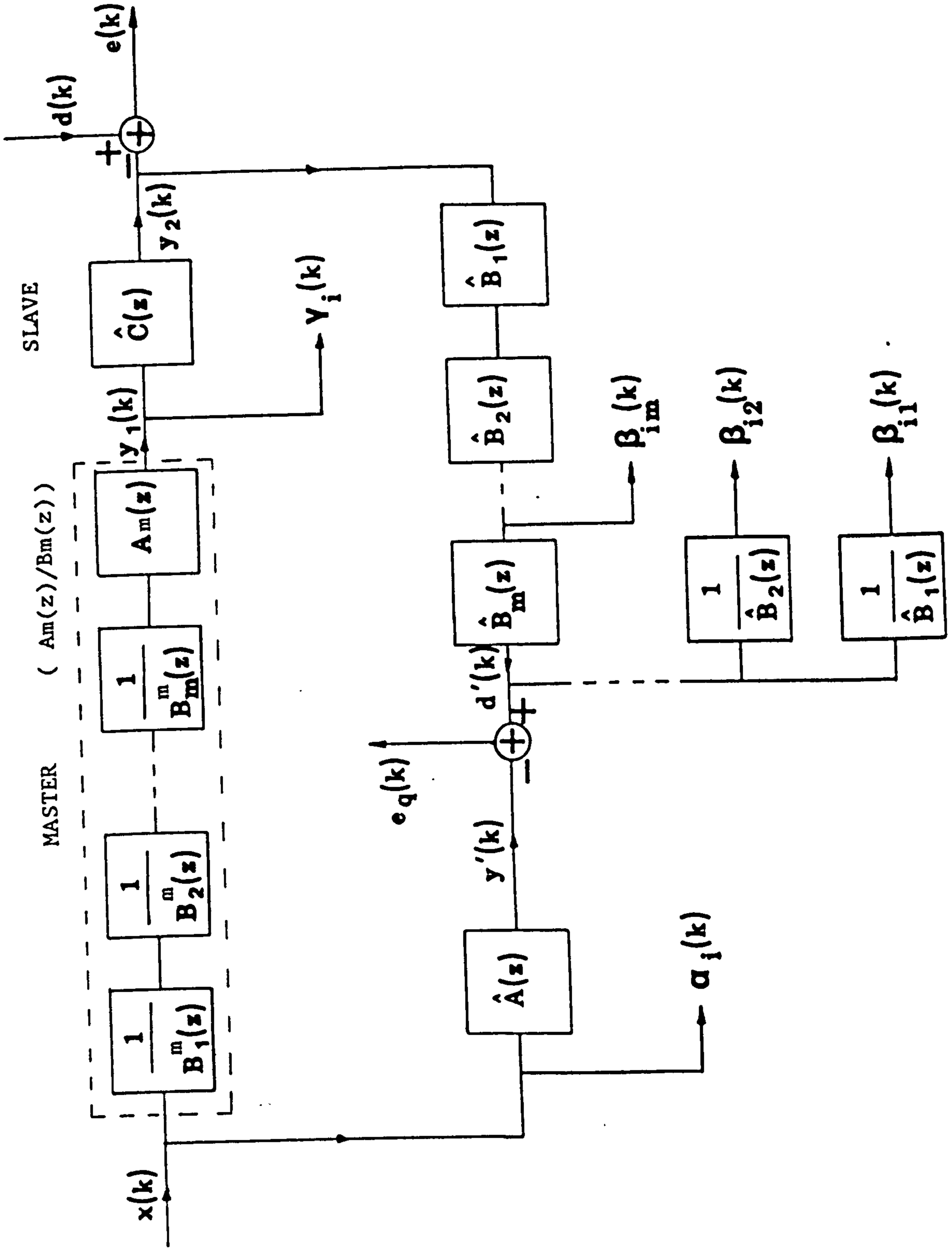
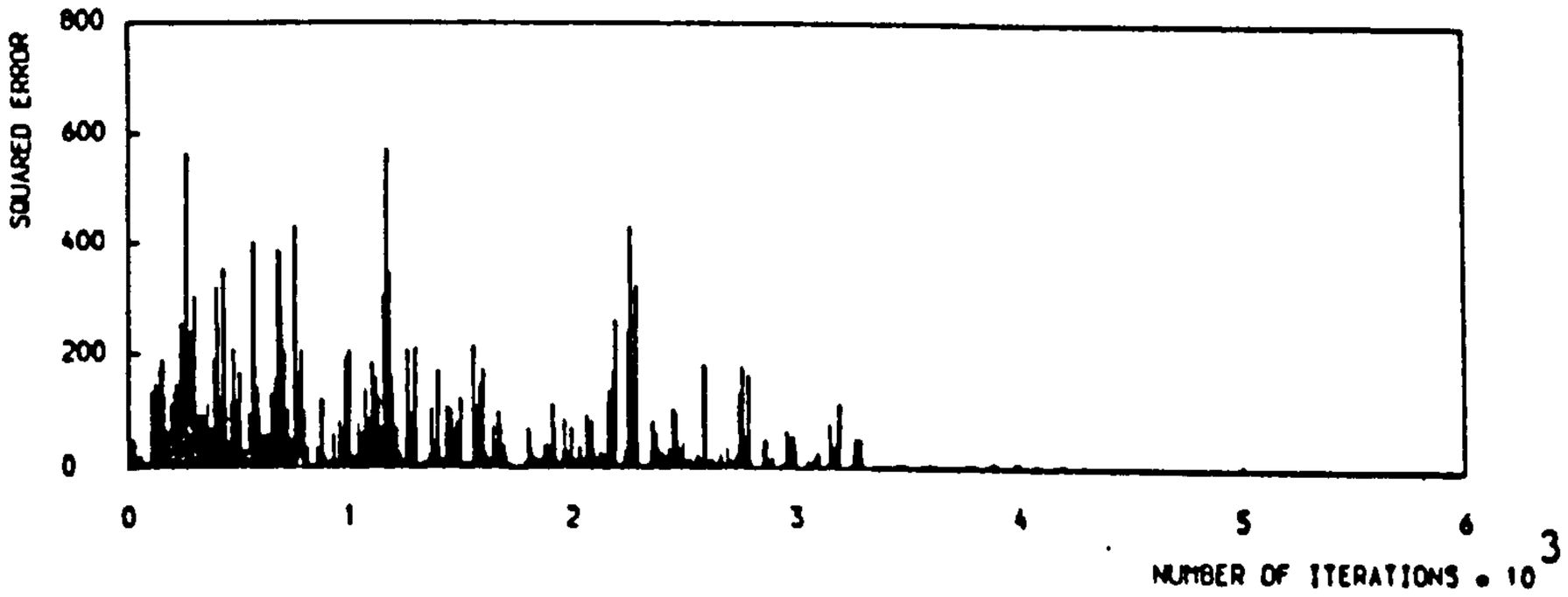


Figure 6.9 Master-Slave Filter Structure.

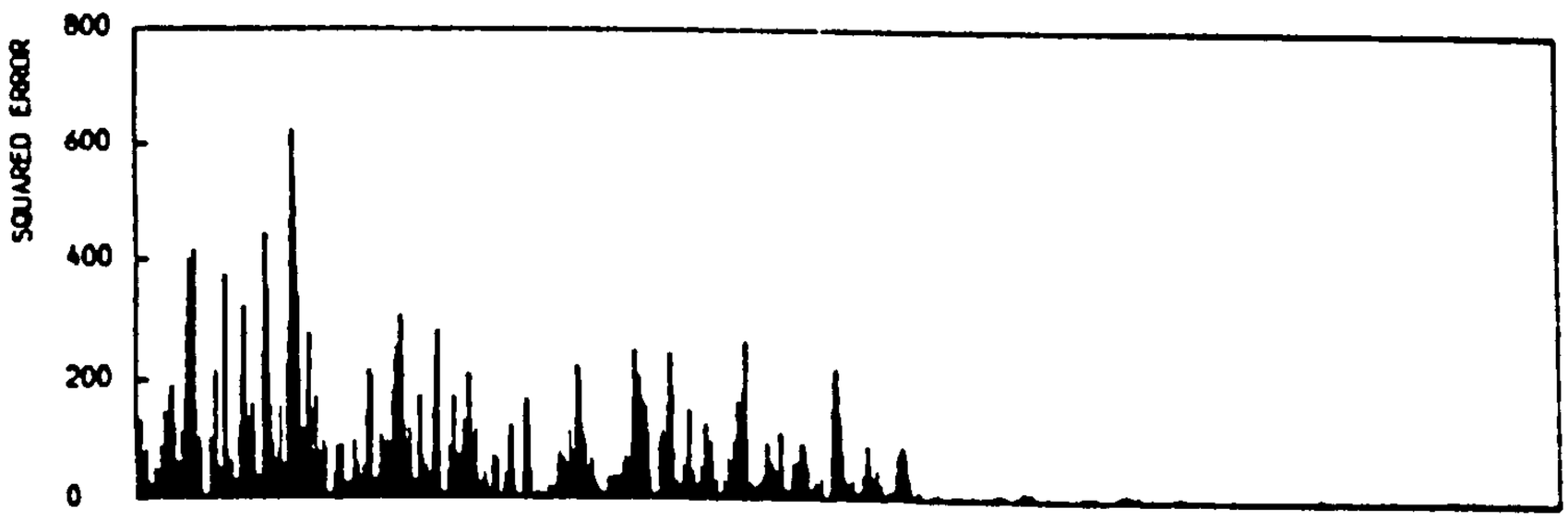


a) Output-Error Convergence Plot for Master-Slave Filter.

$\mu_a = 0.002 \quad \mu_b = 0.0006$

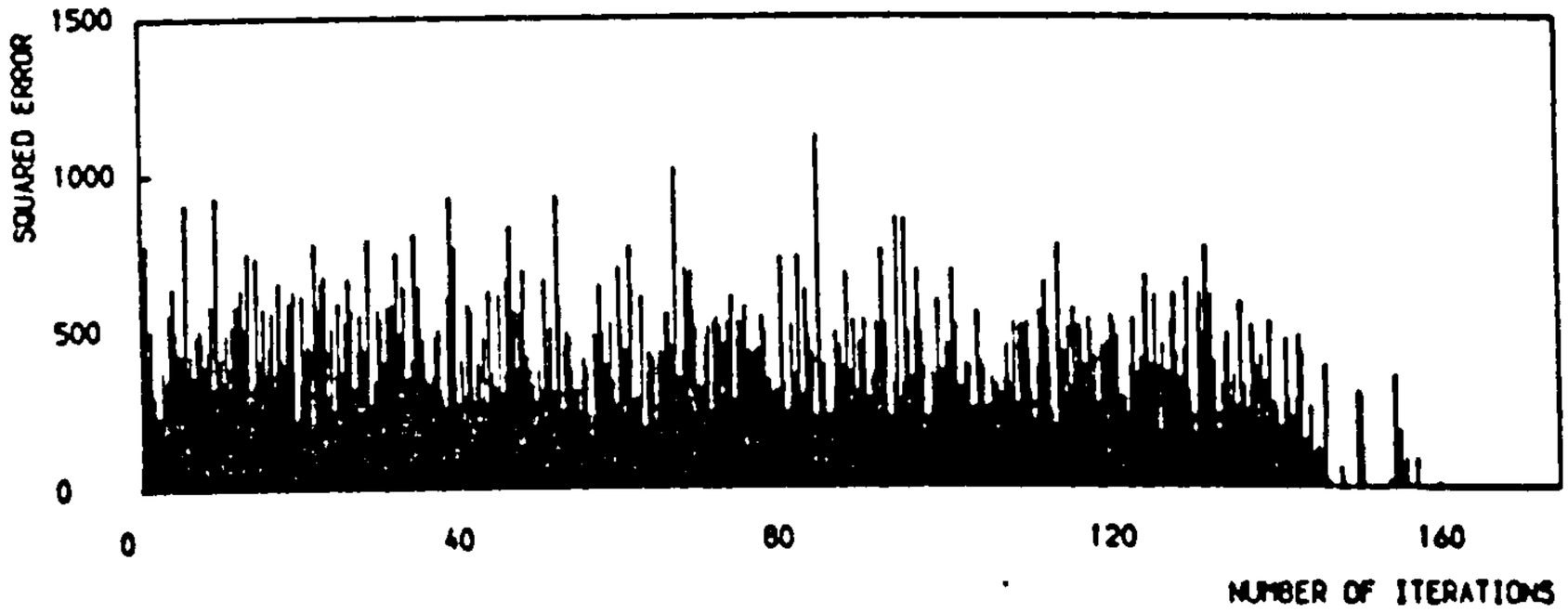
$\mu_c^f = 1.5 \quad \rho = 0.9$

initial coefficients: $(a_0, b_1, b_2) = (1.0, 0.0, 0.0)$



b) Output-Error Convergence Plot for Master Filter.

Figure 6.10 Master-Slave Filter: Second-Order Example.



c) Output-Error Convergence Plot for White/Stearns Algorithm.

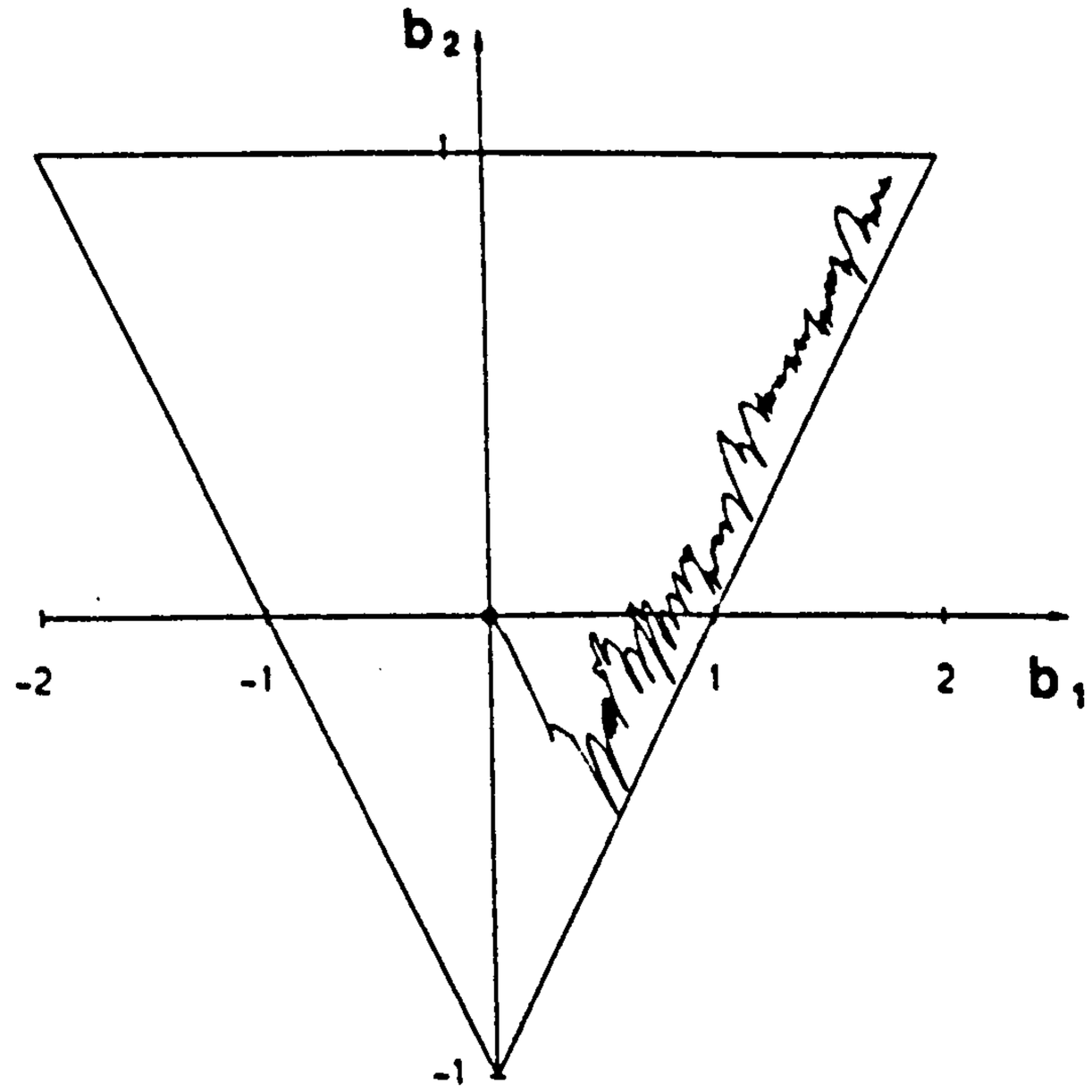
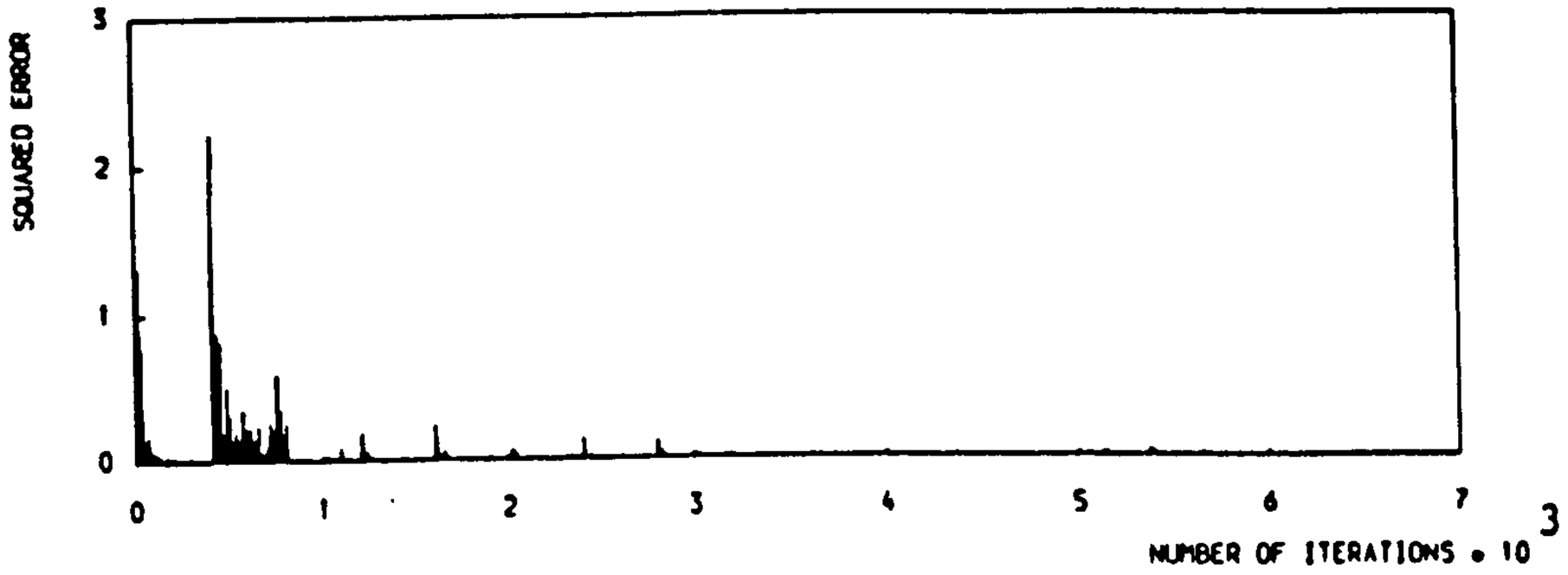


Figure 6.11 (b_1, b_2) Coefficient Track for Second-Order Example.



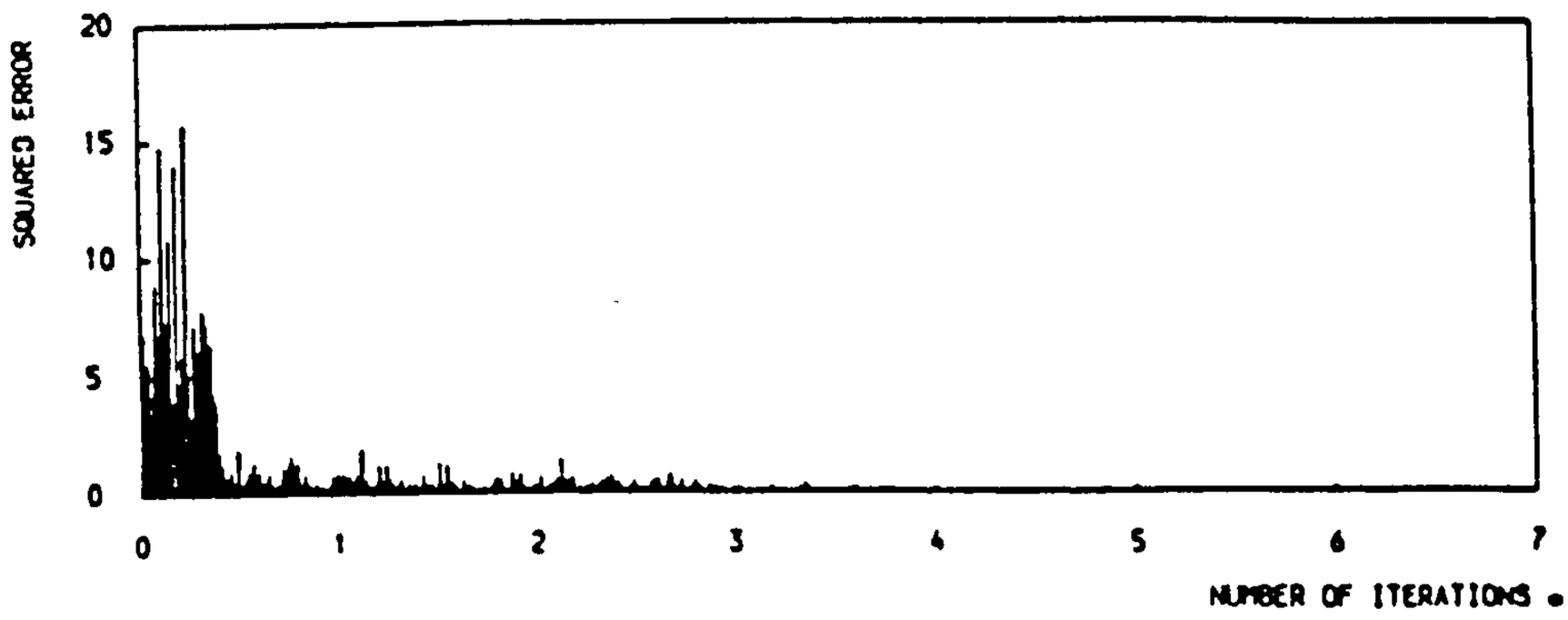
$\mu_a = 0.001 \quad \mu_b = 0.08$

$\mu_c^f = 1.0 \quad \rho = 0.9$

initial coefficients: $(a_0, a_1, a_2, a_3, a_4) = (1.0, 0.0, 0.0, 0.0, 0)$

$(b_{11}, b_{21}, b_{12}, b_{22}) = (-0.2, 0.0, 0.2, 0.0)$

a) Output-Error Convergence Plot for Master-Slave Filter.



b) Output-Error Convergence Plot for Master Filter.

Figure 6.12 Master-Slave Filter: Fourth Order Example.

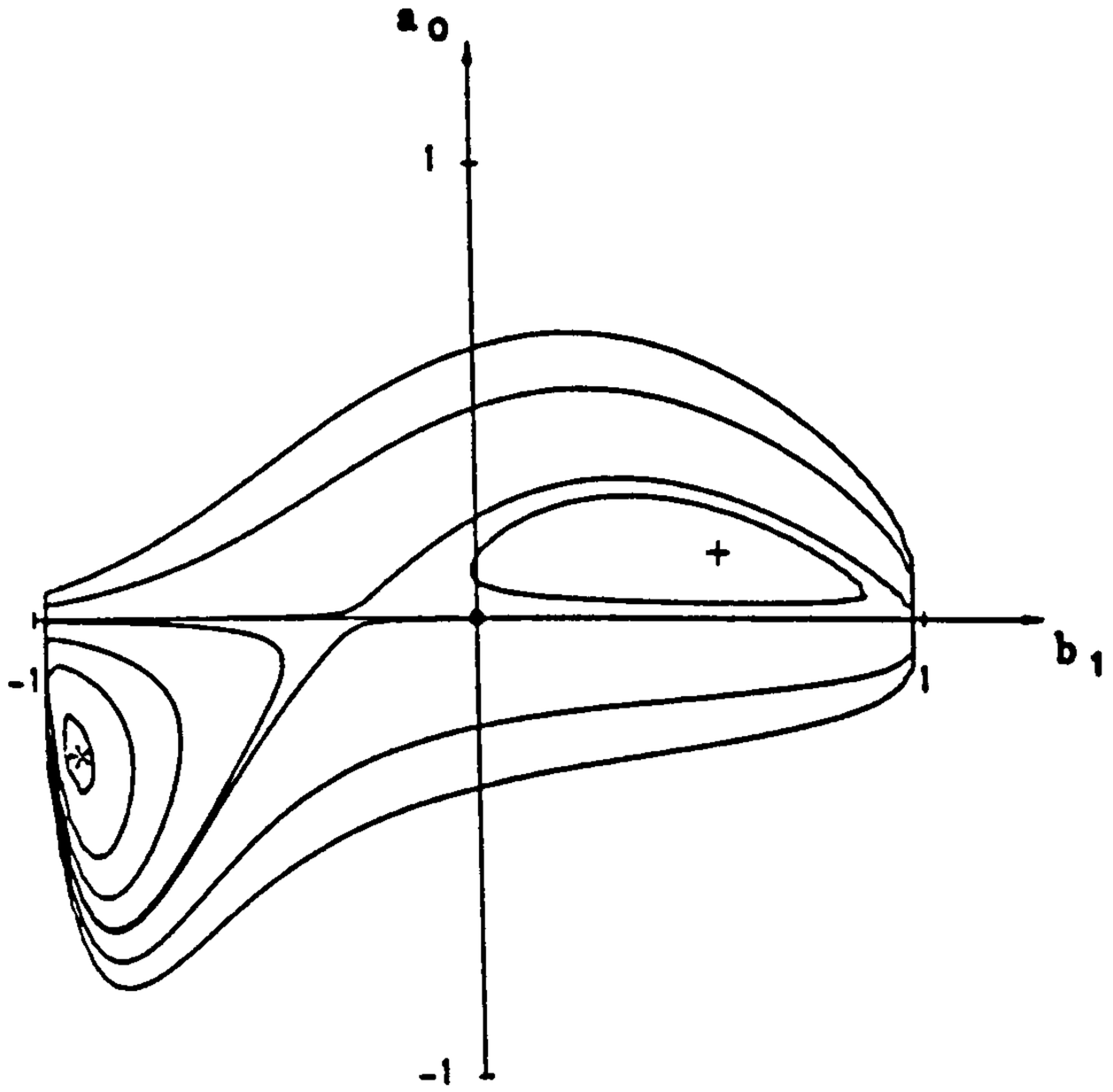


Figure 6.13 Error Surface for Insufficient Order Filter: Example 1

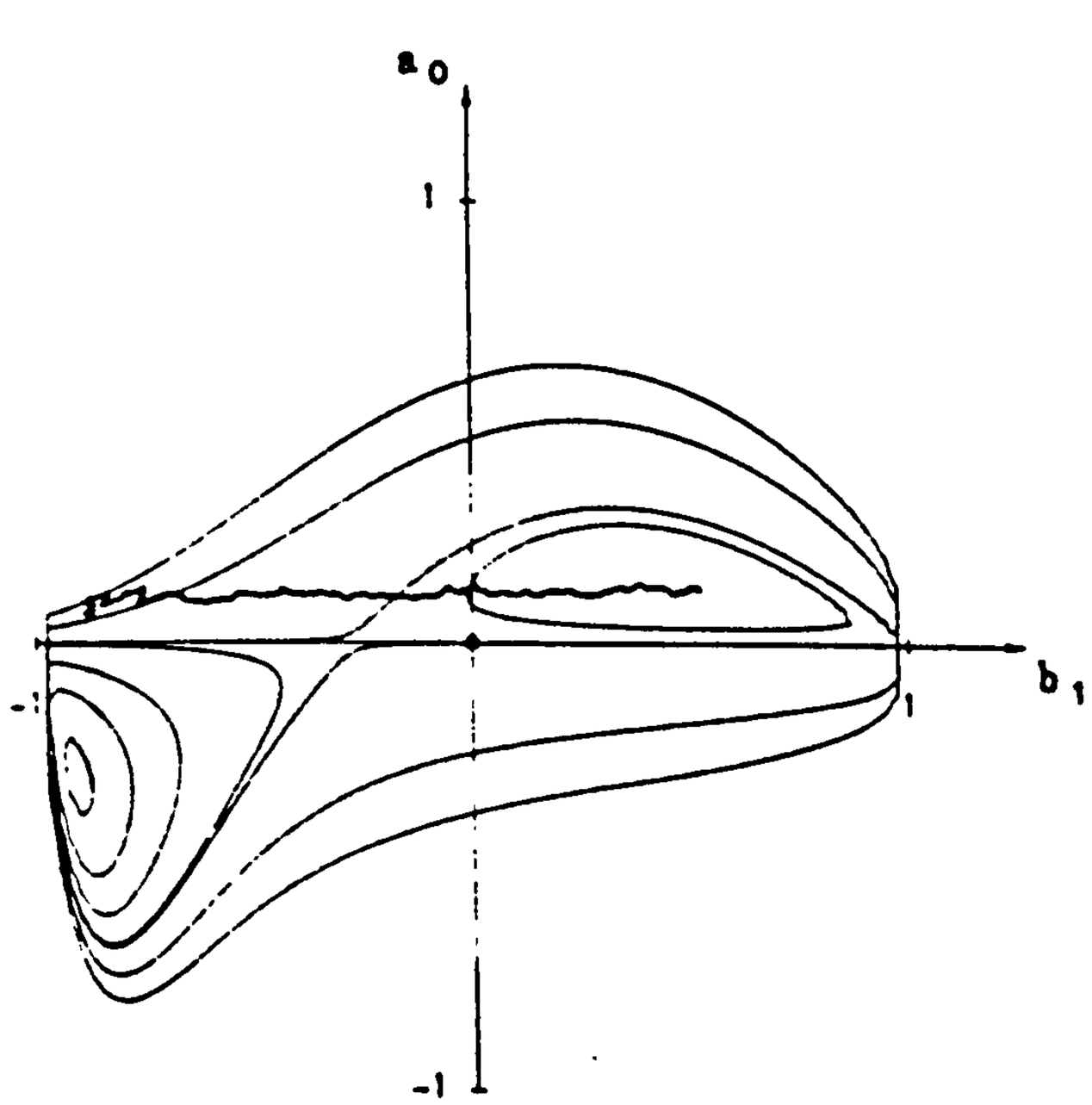


Figure 6.14 (a_0, b_1) Coefficient Track for Example 1.

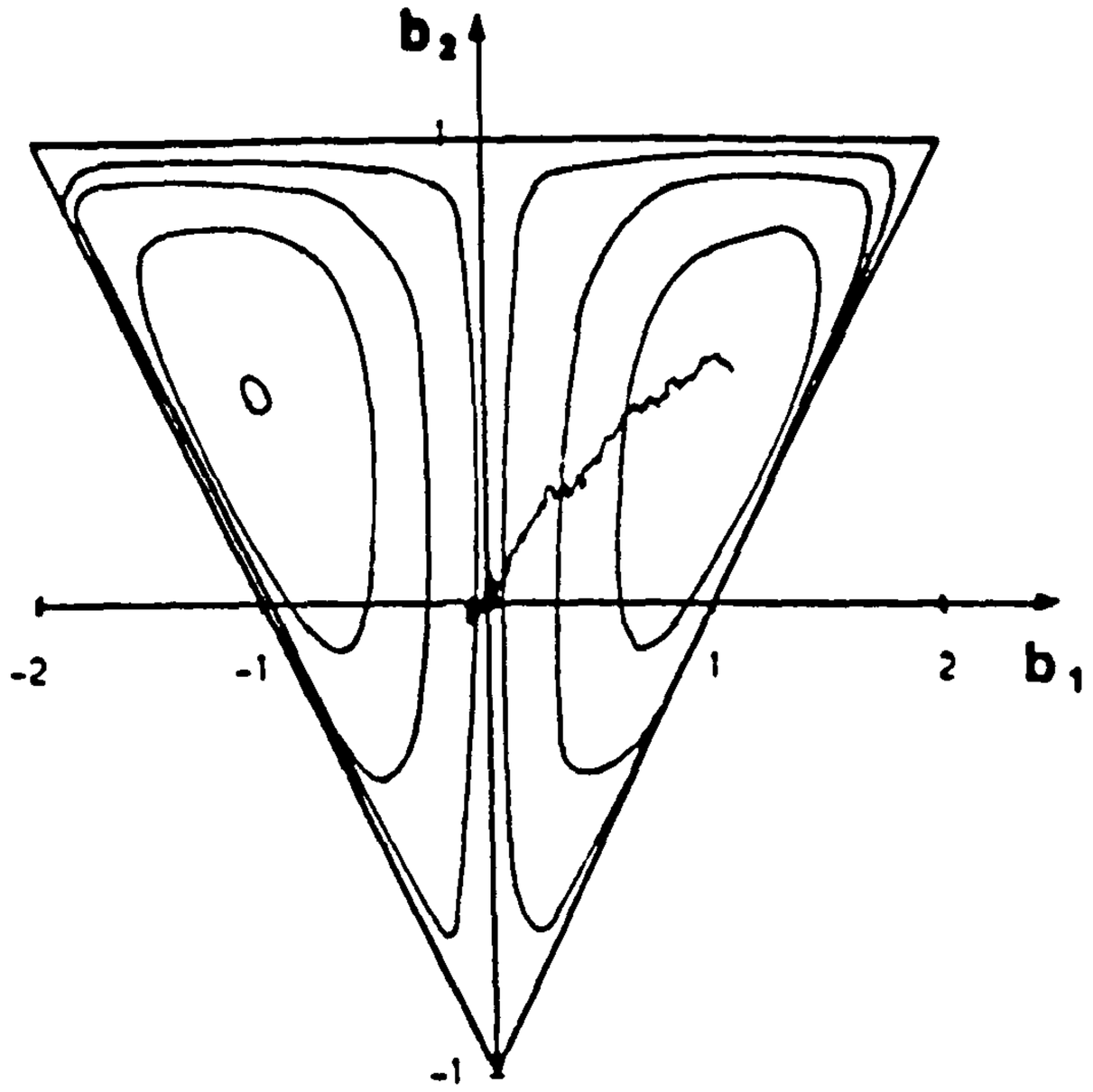


Figure 6.15 (b_1,b_2) Coefficient Track for Insufficient Order Filter, Example 2.

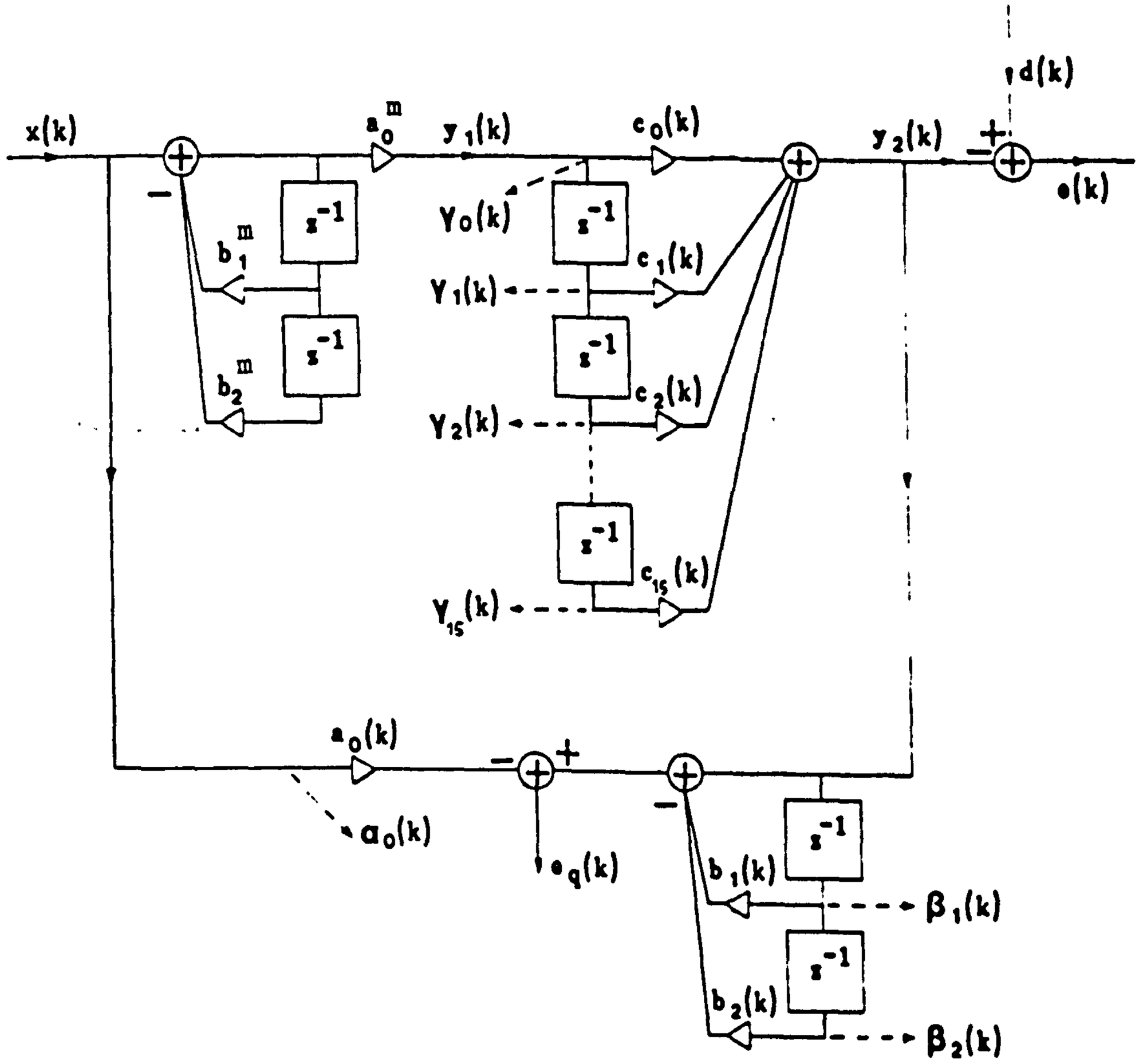


Figure 6.16 Master-Slave Filter Structure for Second-Order Example.

CHAPTER 7

7.0 CONCLUSIONS AND FUTURE DEVELOPMENTS

The work presented in this thesis has been concerned with the development of a robust algorithm for the adaptation of IIR Filters, with a view to real-time hardware implementation. It is necessary for such an algorithm to remain stable and to converge to an optimum solution with respect to the performance criterion. In addition, the speed of adaptation of the filter is significant for a number of applications, and hence should be as fast as possible.

Previously published adaptation algorithms for IIR filters have been studied, and an evaluation made of their applicability for hardware implementation. It has been shown that algorithms can be divided into three classes; gradient algorithms, hyperstable algorithms and random search algorithms, and that none of the classes offers a robust algorithm at present. However, for filter simplicity, stability and speed of adaptation, gradient algorithms using filter structures with guaranteed stability are the most applicable for hardware implementation.

Three new gradient-based Adaptive IIR Filter algorithms have been proposed using filter structures consisting of cascade and parallel arrangements of second-order filter sections. Stability of each of the filter structures is maintained by using a simple check on the filter coefficients. Using simple steepest descent adaptation algorithms, examples have shown that the proposed Modified Cascade Structure offers the best overall performance of the structures considered. However, the rates of convergence of all structures varied greatly and were highly dependent on the target transfer function characteristics. A local minimum resulting from the constraints of the cascade and parallel structures was discovered, and a technique for circumventing the problem has been proposed.

Methods for increasing the speed of convergence of gradient search adaptive IIR filers have been considered and two new algorithms (the Valley-Search and R-Cosθ algorithms) have been proposed for adaptation of the Modified Cascade Structure. The Valley-Search algorithm employs a pattern-search technique to improve convergence times for second-order error surfaces containing steep valleys. The R-Cosθ algorithm adapts the radius and angle of the filter poles independently. Although the R-Cosθ filter structure is restricted to only generating complex pole pairs and not real poles, it has been shown that the algorithm offers faster convergence times than standard RLMS adaptation algorithms.

A practical echo cancelling problem for a 144 kbit duplex data transmission system has been studied and an Adaptive IIR Filter Structure proposed for removal of the echo. A steepest descent gradient algorithm has been proposed for adaptation of the filter structure. The results presented have demonstrated advantages of using adaptive IIR filters in preference to adaptive FIR filters. In addition, a method for the avoidance of local minima has been proposed. This involves fixing the transfer function denominator coefficients at predetermined values, and exploiting the quadratic error surface of the numerator filter coefficients to allow the filter to adapt to the global minimum of the error surface. If sufficient echo cancellation is obtained without adaptation of the transfer function denominator coefficients, then optimum values of the required μ adaptation parameters can be calculated *a priori*.

A new steepest descent algorithm has been proposed for equation error minimization which simplifies a previously published algorithm. A performance assessment of the new algorithm has been made, and the filter has been incorporated into the new Master-Slave Filter Configuration. The Master-Slave filter is a novel adaptive filter

structure combining advantages of the equation error criterion and output error criterion. An algorithm for adaptation of the Master-Slave filter has been proposed and the performance of the filter has been assessed. Results have been presented to demonstrate that the algorithm gives possible adaptation to the global minimum of a multimodal performance surface. Using an arrangement of three FIR filters, the Master-Slave filter structure is stable, very simple to implement and has a fast speed of convergence.

FUTURE DEVELOPMENTS

In this thesis, a number of IIR filter adaptation algorithms have been developed and evaluated using computer simulations. The computation required for the algorithms was kept to a minimum to enable the algorithms to be implemented in real-time hardware systems. Effects caused by the implementation of both the adaptive IIR filter and the adaptation algorithm in hardware require further study; in particular the effects resulting from the finite wordlength filters of the filter feedback coefficients, and also the recursive data paths.

Results obtained using the R-Cosθ algorithm were encouraging when compared with those from other output error minimizing steepest descent algorithms. The constraints of the R-Cosθ filter structure only allow complex conjugate pole pairs to be synthesized and not split real poles. However, this constraint assists with gradient search minimization techniques on the IIR output error surface. Further investigation of this filter structure is required, possibly using adaptive μ values as in the NCRLMS algorithm.

As was discussed in Chapter 2, an alternative method of retaining stability of the adaptive filter structure, is by implementing the recursive filter section using a lattice structure rather than a cascade of second-order filter sections. Similar algorithms to those

developed in this thesis may be applied to the lattice structure. In addition, further study is required to obtain a better understanding of the lattice filter output error performance surface.

The proposed Master-Slave Adaptive Filter was assessed using a number of computer simulations and a number of avenues of further research arise from the results obtained. The number of iterations required for the Master and Slave Filter adaptation phases to obtain optimum rates of convergence needs studying. In addition, the values of the μ adaptation parameters used for the algorithm are empirical at present and the value of μ_c, in particular, could cause problems to the adaptation process. However, as the adaptive filters are all FIR filters, it may be possible to calculate optimum values.

The Slave filter is at present an FIR filter placed in cascade with the Master filter. Further investigation to find the optimum length of the slave filter is required. In addition, extra "slave" filters may be used, possibly implementing a filter in parallel with the Master Filter.

REFERENCES

1. J.W. MARK and S.S. HAYKIN,
"Adaptive Equalisation for Digital Communication", Proc. IEE, Vol. 118, No. 12, Dec 1971, pp. 1711,1720.

2. B. WIDROW et al,
"Adaptive Antenna Systems", Proc. IEEE, Vol. 55, No. 12, Dec 1967, pp. 2143,2159.

3. B. WIDROW et al,
"Adaptive Noise Cancelling: Principles and Applications", Proc. IEEE, Vol. 63, No. 12, Dec 1975, pp. 1692,1716.

4. C.R. JOHNSON, J.P. LYONS and C.HEEGARD,
"A New Adaptive Parameter Estimation Structure Applicable to ADPCM", ICASSP 83 Conf. Proc., Boston, 1983, pp. 1,4.

5. A.KIKUCHI, S.OMATO and T.SOEDA,
"Applications of Adaptive Digital Filtering to the Data Processing for the Environment System", IEEE Trans. on Acoustics, Speech and Signal Proc., Vol. ASSP-27, No. 6, Dec 1979, pp. 790,803.

6. R.A. DAVID,
"IIR Adaptive Algorithms Based on Gradient Search Techniques", Ph.D. Dissertation, Stanford Univ., Aug 1981.

7. T. CLASSEN and W. MECKLENBRAUKER,
"Overview of Adaptive Techniques in Signal Processing", Proc. of EURASIP Conf., Erlangen, Germany, 1983, pp. 747,754.

8. B. WIDROW et al,
"Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter", Proc. IEEE, Vol. 64, No. 8, Aug 1976, pp. 1151,1162.

9. J.M. McCOOL and B. WIDROW,
"Principles and Applications of Adaptive Filters: A Tutorial Review", Proc. IEE Conf., Conf. Publication 14, Aviemore, 1979, pp. 84,95.

10. B. WIDROW and J.M. McCOOL,
"A Comparison of Adaptive Algorithms Based on the Methods of Steepest Descent and Random Search", IEEE Trans. on Antennas and Propogation, Vol. AP-24, No. 5, Sept 1976, pp. 615,637.

11. S.D. STEARNS,
"Error Surfaces of Recursive Adaptive Filters", IEEE Trans. on Circ, and Sys., Vol. CAS-28, No. 6, June 1981, pp. 603,606.

12. S.D. STEARNS,
"Error Surfaces of Adaptive Recursive Filters", Sandia National Lab., SAND80-1348, Dec 1980.

13. K.J. ASTROM and T. SODERSTRAND,
"Uniqueness of the Maximum Likelihood Estimates of the parameters of an ARMA model", IEEE Trans. on Automatic Control, Vol. AC-19, No. 6, Dec 1974, pp. 769,773.

14. G.N. SARIDIS,
"Comparison of Six On-Line Identification Algorithms", Automatica, Vol. 10, 1974, pp. 69,79.

15. G.N. SARIDIS and G. STEIN,
"Stochastic Approximation Algorithms for Linear Discrete-Time System Identification", IEEE Trans. on Automatic Control, Vol. AC-13, No. 5, Oct 1968, pp. 515,523.

16. I.D. LANDAU,
"Unbiased Recursive Identification Using Model Reference Adaptive Techniques", IEEE Trans. on Automatic Control", Vol. AC-21, No. 2, Apr 1976, pp. 194,202.

17. B. FRIEDLANDER,
"System Identification Techniques for Adaptive Signal Processing", IEEE Trans. on Acoustics, Speech and Signal Proc., Vol. ASSP-30 No. 2, Apr 1982, pp. 240,246.

18. C.R. JOHNSON,
"Adaptive IIR Filtering: Current Results and Open Issues", IEEE Trans. on Information Theory, Vol. IT-30, No. 2, Mar 1984, pp. 237,250.

19. E. WALACH and B. WIDROW,
"The Least Mean Fourth (LMF) Adaptive Algorithm and its Family", IEEE Trans. on Information Theory, Vol. IT-30, No. 2, Mar 1984, pp. 275,283.

20. J.R. TREICHLER,
"Transient and Convergent Behavior of the Adaptive Line Enhancer", IEEE. Trans. on Acoustics, Speech and Signal Proc., Vol. ASSP-27, No. 1, Feb 1979, pp. 53,62.

21. J.D. GIBSON,
"Adaptive Prediction for Speech Encoding", IEEE ASSP Magazine, Jul 1984, pp. 12,26.

22. D.M. ETTER and S.D. STEARNS,
"Performance Surfaces of Adaptive Predictor Structures", Proc. Asilomar Conf. on Circ., Sys. and Comp., 1982, pp. 174,176.

23. S.A. WHITE,
"An Adaptive Recursive Digital Filter", Proc. 9th Asilomar Conf. Circ., Sys. and Comp., 1975, pp. 21,25.

24. G.R.ELLIOT, W.L. JACKLIN and S.D. STEARNS,
"The Adaptive Digital Filter", Tech. Report No. SAND76-0360, Sandia National Labs., Aug 1976.

25. S.D. STEARNS, G.R. ELLIOTT and N. AHMED,
"On Adaptive Recursive Filtering", Proc. 10th Asilomar Conf. on Circ., Sys. and Comp., 1976, pp. 5,11.

26. T.C. HSIA,
"A Simplified Adaptive Recursive Filter Design", Proc. IEEE, Vol. 69, No. 9, Sep 1981, pp. 1153,1155.

27. S. HORVATH,
"A New Adaptive Recursive LMS Filter", Digital Signal Processing, Academic Press, 1980, pp. 21,26.

28. P.L. FEINTUCH,
"An Adaptive Recursive LMS Filter", Proc. IEEE, Vol. 64, Nov 1976, pp. 1622,1624.

29. C.R. JOHNSON and M.G. LARIMORE,
"Comments on and Additions to "An Adaptive Recursive LMS Filter"", Proc. IEEE, Vol. 65, No. 9, Sep 1977, pp. 1399,1402.

30. B. WIDROW and J.M. McCOOL,
"Comments on "An Adaptive Recursive LMS Filter"", Proc. IEEE, Vol. 65, No. 9, Sep 1977, pp.1402,1404.

31. N. AHMED et al,
"Sequential Regression Considerations of Adaptive Filtering", Electronics Letters, Vol. 13, No. 15, Jul 1977, pp. 446,447.

32. D. PARIKH and N. AHMED,
"A Sequential Regression Algorithm for Recursive Filters", Electronics Letters, Vol. 14, No. 9, Apr 1978, pp. 266,268.

33. D. PARIKH and N. AHMED,
"Sequential Regression Considerations of an Adaptive Filtering Example", Proc. IEEE, Vol. 66, No. 12, Dec 1978, pp. 1660,1662.

34. D. PARIKH and N. AHMED,
"Adaptive Algorithm Considerations of Adaptive IIR Filters", Proc. 21st Midwest Symp. on Circ. and Sys., pp. 369,373.

35. D.L. SOLDAN, N. AHMED and S.D. STEARNS,
"On Using the Sequential Regression (SER) algorithm for Long-Term Signal Processing", Proc. of ICASSP 80, 1980, pp. 1018,1021.

36. D.D. PARIKH,
"The Time Domain IIR Adaptive Filtering Algorithms", Proc. of 24th Midwest Symp. on Circ. and Sys., Albuequerque, U.S.A, Jun 1981, pp. 33,37.

37. M.G. LARIMORE, J.R. TREICHLER and C.R. JOHNSON,
"SHARF: An Algorithm for Adapting IIR Digital Filters", IEEE Trans. on Acoustics, Speech and Signal Proc., Vol. ASSP-28, No. 4, Aug 1980, pp. 428,440.

38. C.R. JOHNSON et al,
"SHARF Convergence Properties", IEEE Trans. on Acoustics, Speech and Signal Proc., Vol. ASSP-29, No. 3, Jun 1981, pp. 659,669.

39. M.L. LARIMORE,
"Hyperstability and Adaptive Filtering", IEEE Trans. on Acoustics, Speech and Signal Proc., Vol. ASSP-29, No. 2, April 1981, pp. 319,320.

40. J.R. TREICHLER, M.G. LARIMORE and C.R. JOHNSON,
"Simple Adaptive IIR Filtering", Proc. of ICASSP 78, 1978, pp. 118,122.

41. C.R. JOHNSON,
"A Convergence Proof for a Hyperstable Recursive Filter", IEEE Trans. on Information Theory, Vol. IT-25, No. 6, Nov 1979, pp. 745,749.

42. D. PARIKH, S.C. SINHA and N. AHMED,
"On a Modification of the SHARF Algorithm", Proc. 22nd Midwest Symp. on Circ. and Sys., Philadelphia, Jun 1979, pp. 362,365.

43. D. PARIKH,
"Convergence Study of the Modified HARF Algorithm", Proc. 14th Asilomar Conf. on Circ., Sys. and Comp., 1980, pp. 355,360.

44. C.R. JOHNSON and T. TAYLOR,
"CHARF Convergence Studies", Proc. of Asilomar Conf. on Circ., Sys. and Comp., 1979, pp. 403,407.

45. D.M.ETTER, M.J. HICKS and K.H. CHO,
"Recursive Adaptive Filter Design using an Adaptive Genetic Algorithm", Proc. of ICASSP 82, 1982, pp. 635,638.

46. D.M. ETTER and D.C. DAYTON,
"Performance Characteristics of a Genetic Algorithm in Adaptive IIR Filter Design", Proc. of EUROSIP Conf., 1983, pp. 53,56.

47. D. PARIKH, N. AHMED and S.D.STEARNS,
"An Adaptive Lattice Algorithm for Recursive Filters",
IEEE Trans. on Acoustics Speech and Signal Proc., Vol. ASSP-28, No. 1, Feb 1980, pp. 110,111.

48. I.L. AYALA,
"On a New Adaptive Lattice Algorithm for Recursive Filters",
IEEE Trans. on ASSP, Vol. ASSP-30, No. 2, Apr 1982, pp. 316,319.

49. J.R. CASAR-CORREDERA et al,
"ARMA Adaptive Lattices with Error Filtering", Proc. EURASIP Conf. 1983, pp. 759,762.

50. R.A. DAVID,
"A Modified Cascade Structure for IIR Adaptive Algorithms",
Proc. 15th Asilomar Conf. Circ., Sys. and Comp., 1981, pp. 175,179.

51. M.A. SODERSTRAND,
"Cost and Performance Comparisons of Several Implementations of Adaptive Recursive Filters", Proc. of 13th Asilomar Conf. on Circ., Sys. and Comp., 1980, pp. 416,420.

52. M.A. SODERSTRAND and M.C. VIGIL,
"An Adaptive Filter which adapts Structure as well as Weights", Int. J. Electronics, Vol. 55, No. 4, 1983, pp. 533,549.

53. R.D. GITLIN and J.S. THOMPSON,
"A New Structure for Adaptive Digital Echo Cancellation", Proc. National Telecom. Conf., Dallas, Tx., 1976, pp. 8.2-1,8.2-7.

54. P.C. CHING and C.C. GOODYEAR,
"Adaptive Cascade Filter for Speech Analysis", IEE Proc., Vol. 130, Pt. E, No. 1, Jan 1983, pp. 11,18.

55. R.A. DAVID and S.D.STEARNS,
"Adaptive IIR Algorithms based on Gradient Search", Proc. 24th Midwest Symp. on Circ. and Sys., Jun 1981, pp. 22,27.

56. L.B.JACKSON and S.L.WOOD,
"Linear Prediction in Cascade Form", IEEE Trans. on Acoustics, Speech and Sig. Proc., Vol. ASSP-26, No. 6, Dec 1978, pp. 518,528.

57. P.C. CHING and C.C. GOODYEAR,
"L.M.S. Algorithm for Sequentially Adapting All-Zero Digital Filters in Modified Cascade Form", Electronics Letters, Vol. 16, No. 7, Mar 1980, pp. 270,271.

58. S. HENTSCHKE,
"Investigations and Logic designs of Echo Cancelling Methods for Duplex Transmission with Respect to VLSI Realization", Frequenz, Vol. 36, No. 11, Nov 1982, pp. 302,309. (German)

59. S. HENTSCHKE,
"Echokompensation unter verwendung Rekursiver Filter", (German), Source Unknown.

60. P.E. MANTEY,
"Convergent Automatic-Synthesis Procedures for Sampled-Data Networks with Feedback", Tech. Report No. 6773-1, Stanford Uni. California, Oct. 1964.

61. R.A. DAVID,
"A Cascade Structure for Equation Error Minimization", Proc. of Asilomar Conf. on Circ. Sys. and Comp., 1982, pp. 182,186.

62. R.A. DAVID and S.D. STEARNS,
"A Comparison of IIR Adaptive Algorithms", Proc. of 15th Asilomar Conf. on Circ., Sys and Comp., 1981, pp. 180,184.

63. C.R. JOHNSON Jr. and A.L. HAMM,
"Debiasing Equation Error Frequency Detection of a noisy Sinusoid without noise variance prespecification", IEEE Trans. on Acoustics, Speech and Signal Proc., Vol. ASSP-27, Dec 1979, pp. 654,656.

64. A.V. OPPENHEIM and R.W. SCHAFER,
"Digital Signal Processing", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, U.S.A., 1975.

65. P.M. HUGHES,
"The Representation, Generation and Analysis of Equivalent Digital Filter Structures", Ph.D. Thesis, University of Liverpool, 1982.

66. R.A. DAVID et al,
"IIR Algorithms for Adaptive Line Enhancement", Proc. of ICASSP 83, Boston, Mass., U.S.A, 1983, pp. 17,20.

67. C.N. TATE and C.C. GOODYEAR,
"Note on the convergence of linear predictive filters, adapted using the LMS algorithm", Proc. IEE, Vol. 130, Pt. G, No. 2, Apr 1983, pp. 61,64.

68. D. PARIKH and N. AHMED,
"On an Adaptive Algorithm for IIR Filters", Proc. IEEE, Vol. 66, No. 5, May 1978, pp. 585,588.

69. V.M. POPOV,
"Hyperstability of Control Systems", Springer-Verlag, Berlin, Germany, 1973.

70. J.HOLLAND,

"Adaptation in Natural and Artificial Systems" The University of Michigan Press, U.S.A., 1975.

ADDENDIX A

APPENDIX A

Publication based upon the research presented in this thesis.

M.C. HALL and P.M.HUGHES,

"A Simplified Adaptive IIR Cascade Structure", Proc. of 4th International Conf. on Digital Signal Processing of Signals in Communications, Loughborough, April 1985, pp. 33,38.

A SIMPLIFIED ADAPTIVE IIR CASCADE STRUCTURE

M. C. Hall* and P. M. Hughest

SUMMARY

A new gradient-based adaptive IIR cascade structure is proposed which offers a significant computational saving over previously published algorithms. Examples are given demonstrating certain advantages the cascade structure has over a direct form structure. A local minimum problem relating to the IIR cascade error surface is discovered and a solution to it offered.

1. Introduction

The potential of adaptive digital filters in applications where a priori filter design is either impractical or impossible has been recognised for some time. In a number of these applications the adaptive filter is required to have a highly selective frequency response with a correspondingly long impulse response time. In such cases, the order of the finite impulse response (FIR) adaptive filter needed can be prohibitively high. An alternative approach is to use an infinite impulse response (IIR) adaptive filter. This allows a very much lower order filter to be implemented and hence offers a reduction in computational effort and cost. There are, however, two particular problems associated with the adaptation of IIR filters which have limited their use.

As a result of noise in the adaptation algorithm or short term trends in the error signal, the transfer function poles may migrate outside the z-plane unit circle during the adaptation. This results in an unstable filter and consequent failure of the algorithm. Hence, it is necessary to incorporate some form of stability checking in the adaptation algorithm.

A further difficulty arises from the shape of the IIR performance surface used to drive the adaptation. The mean squared difference between a desired response and the adaptive filter output for all possible sets of filter coefficients, forms a surface in the coefficient plane which is non-uniform and in certain cases may even be multimodal [1]. Both these factors are undesirable and hinder the adaptation.

A number of algorithms for the adaptation of IIR filters have appeared in the literature [e.g. 5,6]. However, the majority of these do not inherently guarantee filter stability. Hence, it is necessary to incorporate some form of stability checking after each coefficient update, and ignore any steps which would result in an unstable filter. The ease with which the stability of a filter can be ascertained, however, varies from structure to structure. For the commonly used nth order direct form IIR structure (i.e. direct form II structure in [2]), for example, it is necessary to factorize the denominator of the filter transfer function or apply some other stability criterion to test for stability. Clearly, this is impractical in most real time applications, and hence stability of this structure cannot be guaranteed. The problem may be circumvented to some extent by restricting the allowable change in the filter coefficients to be very small. This reduces the possibility that a short term error will drive the filter unstable, but also considerably lengthens the time taken by the filter to adapt to its target.

An alternative approach is to realize the adaptive filter as a cascade of second order recursive sections. Stability can be maintained simply by ensuring that the recursive coefficients in each section, b_1 and b_2, are such that $|b_2| < 1$ and $|b_1| < b_2 + 1$.

A recursive least mean squares algorithm for adapting a cascade of biquadratic filter sections has been proposed by David in [3]. This was modified in [4], where the algorithm was re-developed for a filter consisting of a single nth order all-zero section followed by a cascade of second order all-pole sections. Despite the loss of the modularity of the structure, the latter has the advantage that the performance surface with respect to the coefficients of the all-zero section is quadratic [1]. In this paper a modification to the structure given in [4] is proposed, which greatly simplifies the implementation of the adaptive filter whilst maintaining the performance.

---

*University of Liverpool

†British Telecom Research Labs., Martlesham, Ipswich. Formerly University of Liverpool.

2. Derivation of new Modified Cascade Algorithm

The modified cascade structure is shown in Figure 1 where:

$$\hat{B}_j(z) = 1 + b_{1j}z^{-1} + b_{2j}z^{-2} \quad j = 1,2,\ldots,n \qquad (1)$$

$$\hat{A}(z) = a_o + a_1z^{-1} + a_2z^{-2} + \ldots + a_nz^{-n} \qquad (2)$$

and the error signal at time kT is given by:

$$e(k) = d(k) - y(k) \qquad (3)$$

y(k) and d(k) are the actual and required outputs respectively, of the adaptive filter at time kT. The filter coefficients are adapted so as to minimize $e^2(k)$, at which time the adaptive filter output y(k) will be a best least squares estimate of the desired signal d(k). For the steepest descent gradient search method, the coefficients are updated using:

$$b_{ij}(k+1) = b_{ij}(k) + \mu_b \cdot \frac{-\partial\ \overline{e^2(k)}}{\partial\ b_{ij}(k)} \qquad (4)$$

$$i = 1,2$$
$$j = 1,2,..,n$$

$$a_i(k+1) = a_i(k) + \mu_a \cdot \frac{-\partial\ \overline{e^2(k)}}{\partial\ a_i(k)} \quad i = 1,2..n \qquad (5)$$

where μ_a and μ_b are convergence factors used to control the rate of adaptation. Combining equations (3), (4) and (5) yields:

$$b_{ij}(k+1) = b_{ij}(k) + \mu_b \cdot e(k) \cdot \beta_{ij}(k) \qquad (6)$$

$$i = 1,2$$
$$j = 1,2,\ldots,n$$

$$a_i(k+1) = a_i(k) + \mu_a \cdot e(k) \cdot \alpha_i(k) \qquad (7)$$

$$i = 1,2,\ldots,n$$

where:

$$\beta_{ij}(k) \triangleq \frac{\partial y(k)}{\partial b_{ij}(k)} \qquad (8)$$

$$\alpha_i(k) \triangleq \frac{\partial y(k)}{\partial a_i(k)} \qquad (9)$$

and the instantaneous squared error $e^2(k)$ is taken as a local estimate of $\overline{e^2(k)}$ [6].

As the filter is time varying and recursive, y(k) at a particular time reflects the past variations in the coefficients. By allowing only small coefficients updates, the filter transfer function remains approximately constant over a number of iterations, and hence y(k) can be assumed to be the output from a fixed filter. Further, it can be assumed that each section in the cascade is independent of the other sections. That is, small coefficient changes in one section will not affect the inputs to the other sections.

From Figure 1 it can be seen that the z-transform of the adaptive filter output, Y(z), is given by:

$$Y(z) = X(z)\ \frac{\hat{A}(z)}{\hat{B}_1(z)\ .\ \hat{B}_2(z)\ldots.\hat{B}_n(z)} \qquad (10)$$

By taking the partial derivatives of Y(z) with respect to the adaptive coefficients, equations (8) and (9) give:

$$\beta_{ij}(z) = \frac{-Y(z).z^{-i}}{\hat{B}_j(z)} \qquad (11)$$

$$i = 1,2$$
$$j = 1,2,\ldots,n$$

$$\alpha_i(z) = \frac{X(z)\ .\ z^{-i}}{\hat{B}_1(z).\ \hat{B}_2(z)\ldots.\hat{B}_n(z)} \qquad (12)$$

$$i = 1,2\ldots..n$$

Hence, the $\alpha_i(k)$ and $\beta_{ij}(k)$ terms can be generated by a series of time varying IIR filters.

The number of filters required may be greatly reduced using a modification proposed in [5]. As $\hat{B}_j(z)$ is approximately constant over a short interval, $\hat{B}_j(z)$ at time t = kT can be approximated by $\hat{B}_j(z)$ at time t = (k - i)T in equations (11) and (12). This allows the β_{ij} terms to be calculated using j second order all-pole filters, rather than the 2_j filters previously required. By placing the $\hat{A}(z)$ zero section at the end of the cascade, the α_i terms are generated at the outputs of the delays within the filter, and hence are obtained without further processing 'cost'.

The new adaptive filter structure is shown in Figure 2. The coefficients are updated using equations (6) and (7). A comparison with the David structure shown in Figure 3 shows the two filters to be equivalent, but with the structure proposed in this paper offering a significant computational saving. The differences in the structures are a result of the order in which the simplifying assumptions are made in the analysis, and the positioning of the all-zero section at the end of the cascade.

3. Filter Performance

In order to assess the performance of the new cascade implementation, a series of system modelling tests have been performed. In these tests, a 4th-order adaptive filter, $\hat{A}(z)$, is required to model a known 4th-order plant (Figure 4). Both the adaptive filter and the fixed plant are excited by the same broadband white noise input. Results are presented for the new cascade algorithm, and for comparison, a direct form implementation adapted using the White RLMS algorithm [6]. As the mean square error surface with respect to the non-recursive coefficients can be shown to be parabolic for both structures,

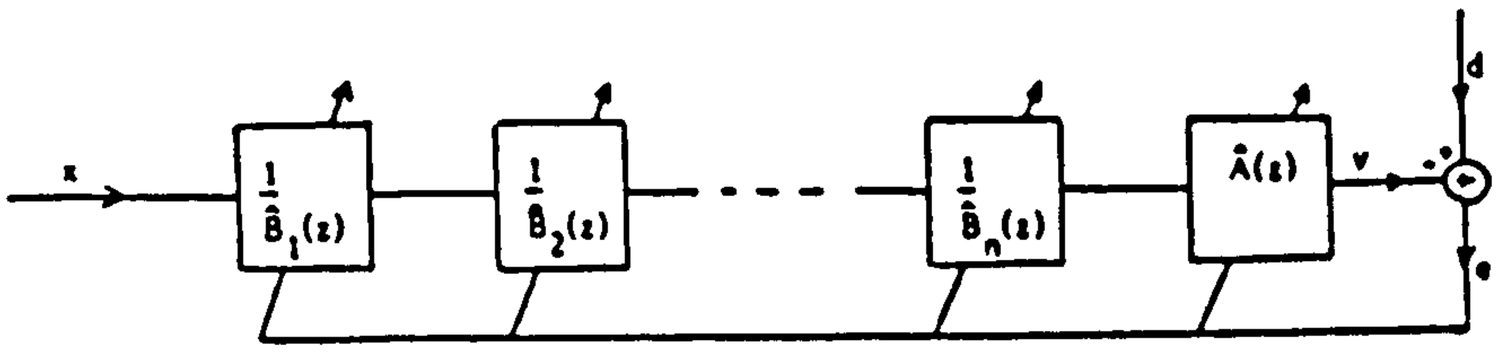


Figure 1. Adaptive Modified Cascade Structure

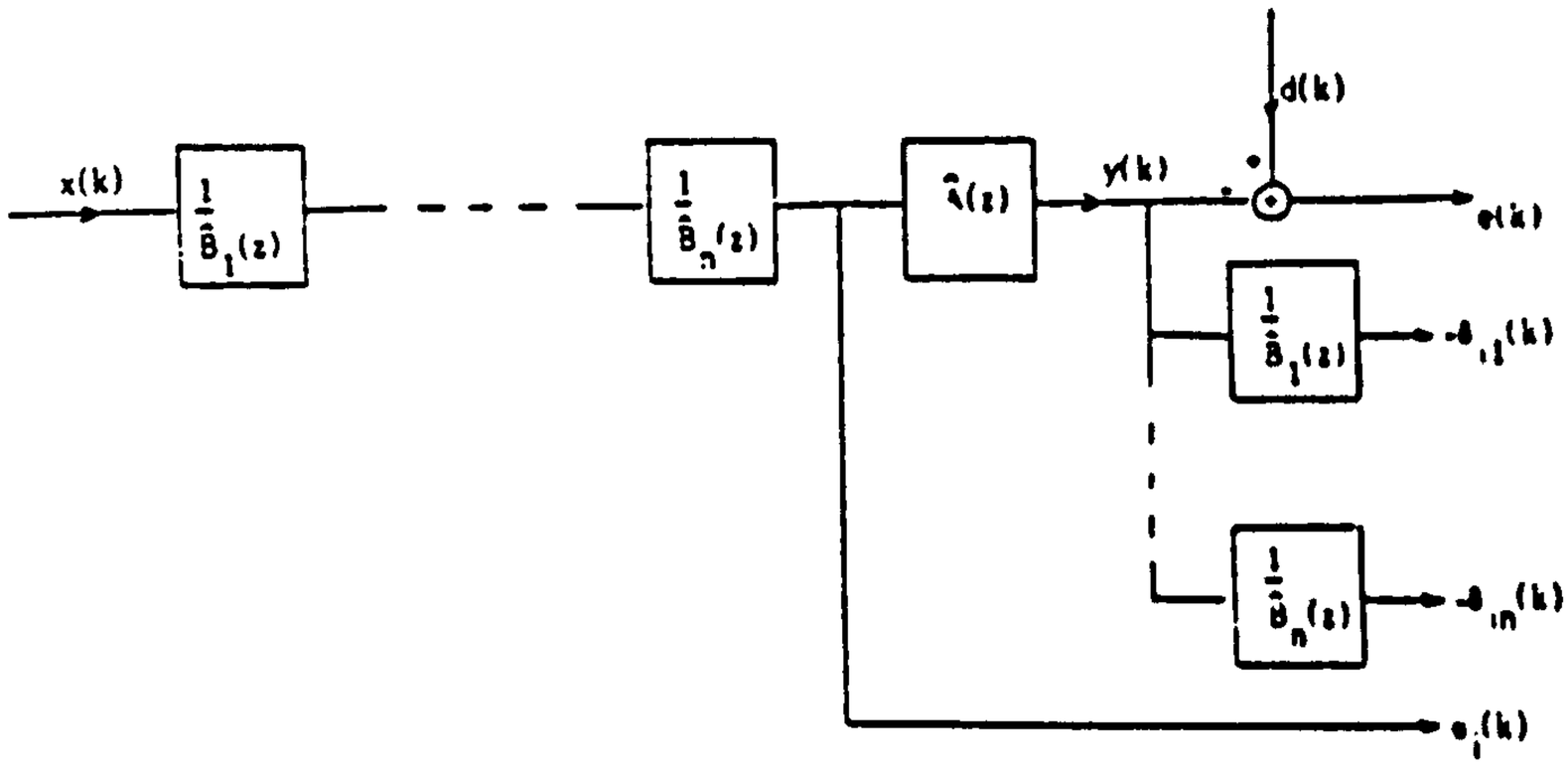


Figure 2. New Cascade Structure

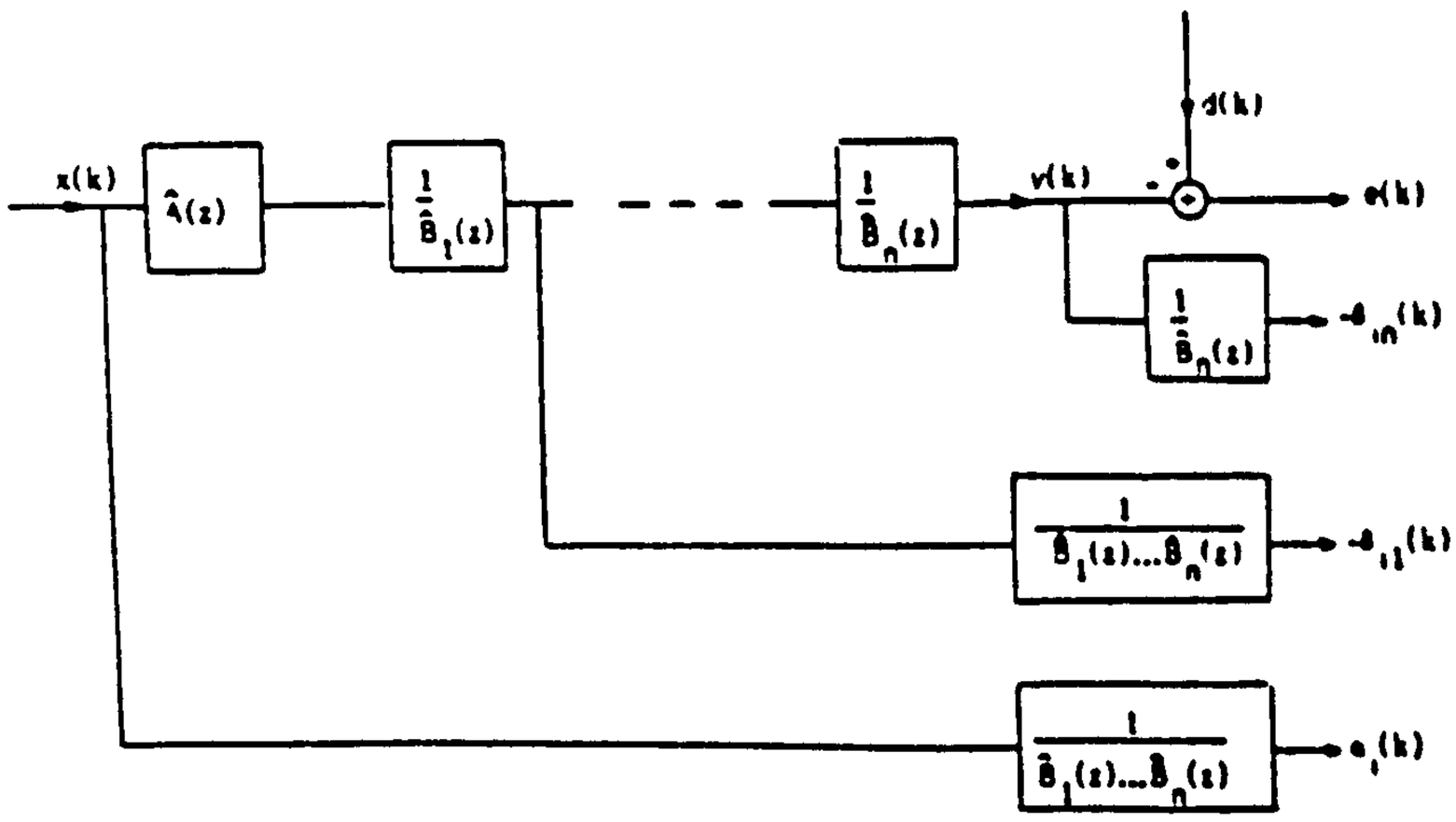


Figure 3. David Structure

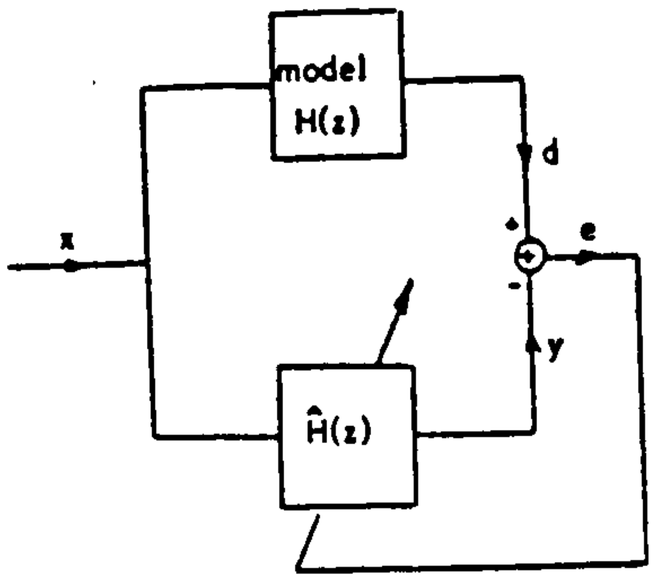
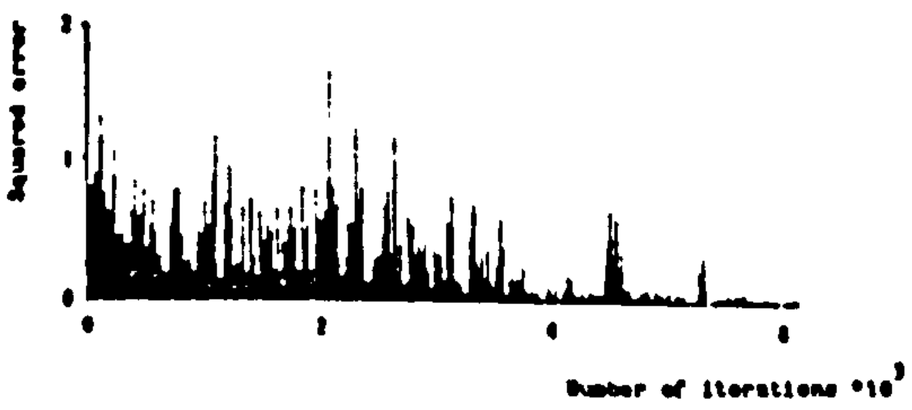
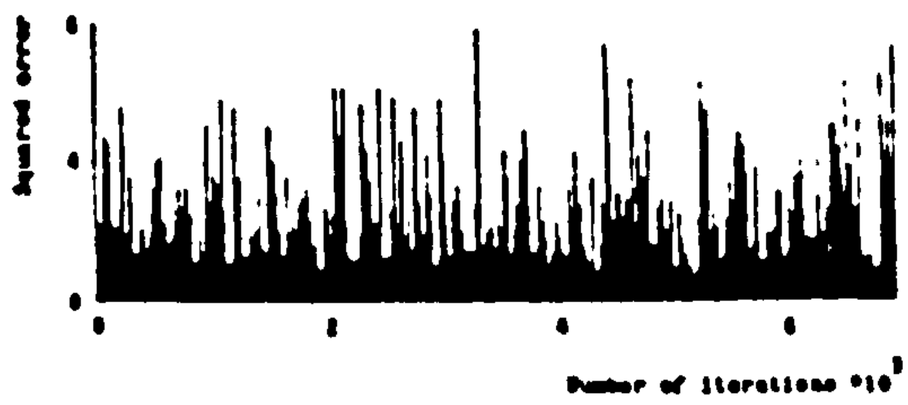


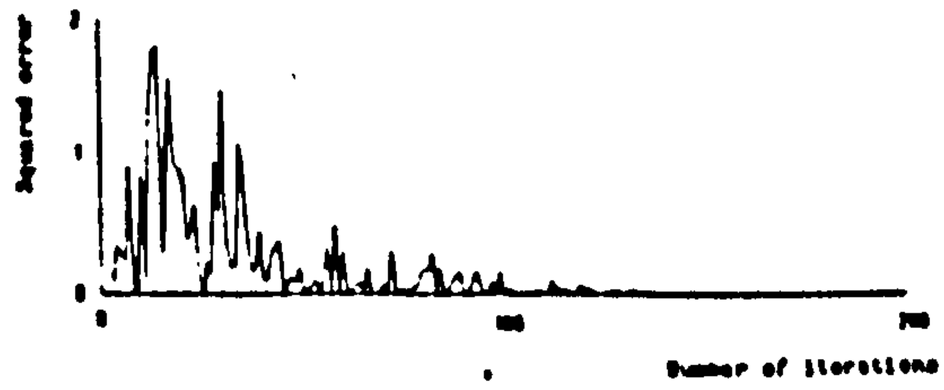
Figure 4. System Identification Configuration



a) Cascade Structure

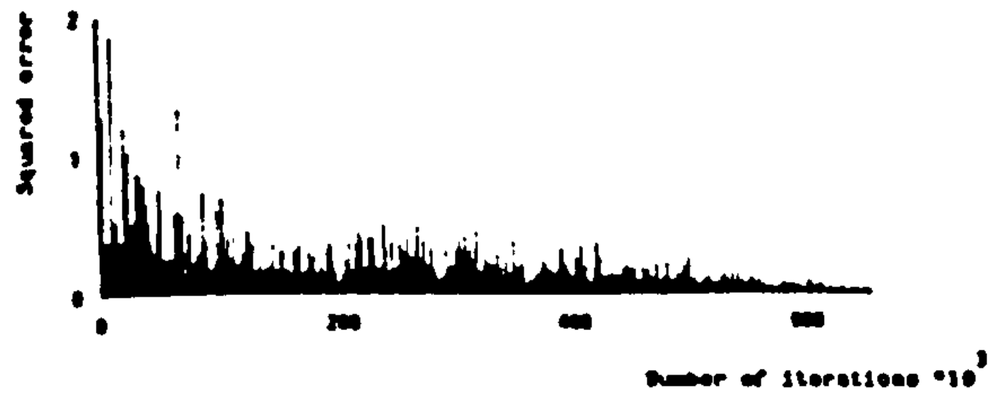


a) Cascade Structure



b) Direct Form Structure

Figure 6. Convergence Plots for Example 2



b) Direct Form Structure

Figure 5. Convergence Plots for Example 1

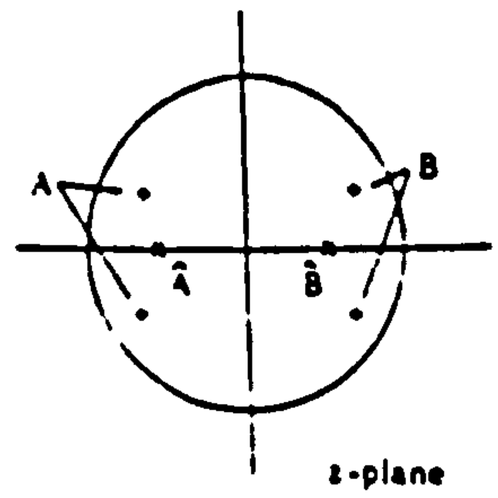


Figure 7. Local Minimum Pole Positions

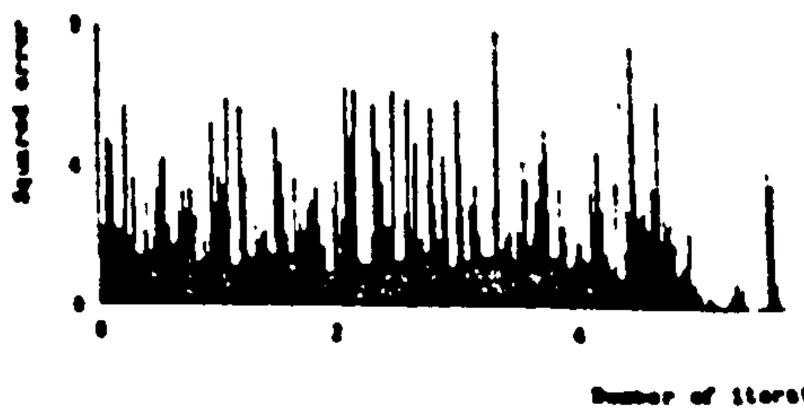


Figure 8. Convergence Plot for Cascade Structure with Modification.

then provided that the poles of $\hat{H}(z)$ can adapt successfully to their optimum positions, it may be assumed that the transfer function zeros will adapt. Hence in the following examples, the fixed plant and the adaptive filter used are all-pole filters. This implies no loss of generality.

In each individual case the value of μ_b was adjusted to that which gave the fastest convergence. The convergence plots presented are therefore noisy.

3.1 Example 1

The model transfer function implemented was:

$$H(z) = \frac{0.05}{(1 + 1.2z^{-1} + 0.8z^{-2})(1 + 1.5z^{-1} + 0.7z^{-2})} \quad (13)$$

The poles of H(z) are closely spaced in the unit circle with radii of 0.90 and 0.85, and corresponding arguments 132° and 153°. Figures 5a and 5b show convergence plots for the cascade and direct form filters respectively.

The high sensitivity of the direct form coefficients to closely spaced poles [2] coupled with possible instability, necessitated a very small value of μ_b to produce convergence. In the cascade case, the less sensitive coefficients and guaranteed stability result in much faster convergence.

3.2 Example 2

A transfer function consisting of widely spaced poles was implemented, viz:

$$H(z) = \frac{1}{(1 + 1.5z^{-1} + 0.8z^{-2})(1 - 1.5z^{-1} + 0.7z^{-2})} \quad (14)$$

The poles have radii of 0.90 and 0.85, with respective arguments of 147° and 26°. Convergence plots for the two structures are given in Figures 6a and 6b. In this example the direct form structure adapts very quickly whilst the cascade is unable to match the model and hence gives a large final mean square error.

Inspection of the cascade coefficients reveals both sections 'lock up' at $b_1 = -0.07$, $b_2 = -0.21$ due to a local minimum in the cascade error surface. This is illustrated in Figure 7. The coefficient values correspond to two pairs of split real poles, each section contributing one pole at $\hat{A}$, the other at $\hat{B}$. The adaptive filter tries to adapt the poles at $\hat{A}$ to match the plant poles at A and similarly for those at $\hat{B}$. However, the poles at $\hat{A}$ cannot become a complex pair in the cascade filter and hence the adaptation is halted.

The problem can be overcome by testing for the above condition and replacing the split real pole pairs by a complex pair at each point. Figure 8 shows a convergence plot for the cascade structure with this extra modification.

It is seen that although the cascade converges in 6k iterations the direct form requires only 150 iterations.

4. Conclusion

By implementing the recursive part of an adaptive IIR filter as a cascade of second order sections, the stability of the filter can be guaranteed by means of a simple stability check on each section. In this paper a means of adapting a cascade structure has been presented which requires substantially less computation than previously published algorithms. The performance of this algorithm has been examined in the context of system modelling.

It is shown that when adapting to a plant which has closely spaced poles, the cascade implementation requires considerably fewer iterations than the direct form structure for convergence. When the plant consists of widely spaced poles, it is found that the poles implemented by each of the sections in the cascade are driven into a local minimum in the performance surface. A method for overcoming this difficulty has been proposed. It should be noted that the existence of the constrained local minimum is not necessarily a major disadvantage, as the local minimum will lie towards, and often close to, the global minimum.

In the examples given, the values for the convergence factor μ_b were chosen so as to give the fastest convergence. The fact that the cascade structure is constrained to be stable, allows the adaptation of μ_b and significant lowering of convergence times. Thus, although the direct form structure converged faster for the widely-spaced poles plant (example 2), adapting μ_b for the cascade structure would reduce the difference considerably. Suitable strategies for the adaptation are the subject of further research.

5. Acknowledgements

One of the authors (M.C.H.), wishes to thank S.E.R.C. and British Telecom Research Laboratories, Martlesham for their financial support.

References

1) S.D. Stearns, "Error Surfaces of Recursive Adaptive Filters", IEEE Trans on Circuits and Systems, CAS-28, No.6, June 1981. pp. 603-607.

2) A.V. Oppenheim, and R.W. Schafer, "Digital Signal Processing", Prentice-Hall, Inc., 1975.

3) R.A. David, "IIR Adaptive Algorithms based on Gradient Search Techniques", Ph.D Dissertation, Stanford University, August 1981.

4) R.A. David, "A Modified Cascade Structure for IIR Adaptive Algorithms", Proc. 15th Asilomar Conf. Circuits, Systems and Computers, November 1981, pp. 175-179.

5) T.C. Hsia, "A Simplified Adaptive Recursive Filter Design", IEEE Proc, Vol. 69, No. 9, September 1981, pp. 1153-1155.

6) S.A. White, "An Adaptive Recursive Digital Filter", Proc. 9th Asilomar Conf. Circuits, Systems, and Computers, November 1975, pp. 21-25.

Adaptive IIR Filter Algorithms for Real-Time Applications

MARTIN C. HALL

The characteristics of a digital filter are determined by a set of multiplier values (the filter coefficients). By changing the coefficient values under program control, the filter characteristics can be optimized with respect to a chosen performance criterion. Adaptive filters are used in applications where fixed filter design is impractical or impossible, such as when signal properties are unknown or vary with time. Adaptive IIR filters contain both poles and zeros in their transfer function, and hence are more powerful than adaptive non-recursive (FIR) filters. However, the algorithms required for IIR filter adaptation are more complex than those for FIR filter adaptation. In addition, IIR filter algorithms have several disadvantages associated with them which have restricted their use in the past. With the increasing need for low-cost, dedicated microprocessor-based systems, the development of simple, robust algorithms for IIR filter adaptation is of great importance.

In this thesis, previously published adaptation algorithms for IIR filters are examined with a view to real-time implementation. It is shown that the algorithms developed to date are not sufficiently robust, and that many are too complex for microprocessor implementation. The algorithm assessment suggests that simple gradient-based algorithms for adaptation of filter structures with guaranteed stability, are the most applicable for hardware implementation.

Three new gradient adaptive IIR filter algorithms are proposed for filter structures consisting of parallel and cascade arrangements of second-order filter sections. An assessment of the algorithms is made using a series of system modelling tests which show advantages and disadvantages of the differing filter structures; in particular a local minimum problem arising from constraints on the filter coefficients is highlighted. It is shown that a modified cascade structure offers the best overall performance of the structures considered. More complex gradient algorithms applicable to this filter structure are studied, and two new algorithms are proposed, the Valley-Search and R-Cosθ Algorithms. Results are presented to demonstrate the improved rates of convergence obtained by these algorithms.

The modified cascade structure is applied to a practical echo cancelling problem which demonstrates advantages of using IIR filters in preference to FIR filters. In addition, a method for the avoidance of local minima in the IIR filter output error performance surface is presented.

An examination of the equation error performance criterion is presented and a new algorithm for minimization of the equation error is proposed. By combining advantages of output error and equation error minimization, a novel gradient adaptive IIR filter structure is proposed. The Master-Slave Algorithm uses adaptive FIR filters for the adaptation of both the transfer function numerator and denominator coefficients, and offers faster convergence than conventional output-error minimizing adaptive filters. Implementation of the algorithm requires very little additional computation, and stability of the filter structure is maintained easily. In addition, the Master-Slave Algorithm offers possible adaptation to the global minimum of a multi-modal output error performance surface.

The Master-Slave Filter Configuration proposed in Chapter 6 of the thesis is the subject of a UK patent application. The application, number 8629872, was filed on 15th December 1986.