SPLIT ALGORITHMS FOR LMS ADAPTIVE SYSTEMS

By

Ho, King Choi

A thesis presented to the Chinese University of Hong Kong
in partial fulfillment of the degree of Doctor of Philosophy

Department of Electronic Engineering
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong

May, 1991

ACKNOWLEDGMENT

The author is greatly indebted to his academic supervisor, Dr. P. C. Ching, for his guidance and inspiring advice during the preparation of this thesis. The author is grateful to Prof. Y. T. Chan for his valuable comments and discussions.

The author would also like to thank the Croucher Foundation for the grant of a studentship to support the project for this thesis.

ABSTRACT

Adaptive digital filtering using least mean square (LMS) algorithm has found applications in a wide variety of areas such as beamforming, channel equalization, noise cancellation and many others because of its capability in tracking nonstationary signals and changing environments. Despite its simplicity in hardware implementation, transversal LMS adaptive filter has an inherent deficiency that its convergence rate might be very poor if the eigenvalue spread of the autocorrelation matrix of the input sequence is large. Different forms of realization in filter structure can, however, result in different eigenvalue spreads and hence achieving better convergence characteristics. In this thesis, techniques of improving adaptation rate by splitting a transversal adaptive filter into both parallel and serial form will be examined. The filter structure together with the corresponding adaptation rules are formulated and the dynamic convergence behavior is being analysed vigorously. Theoretical derivations are supported by extensive simulations.

We shall demonstrate that the configuration of an adaptive filter can be implemented in either a parallel or a serial form so that substantial improvement in system performance can be obtained for many different applications. When an adaptive filter is constructed as a parallel connection of two transversal linear phase filters in which one is symmetric and the other antisymmetric, the eigenvalues of the input process are separated into two sets, one for each linear phase filter. This will eventually lead to smaller eigenvalue spreads and therefore an overall faster convergence speed. In the context of linear prediction, backward prediction can be embedded easily in the adaptation process to reduce gradient noise so that the rate

of convergence can be further enhanced.

If an adaptive linear predictor is represented as a serial connection of two adaptive subunits in transversal form, the eigenvalue spreads governing their convergence properties will be smaller than that of a single ladder structure and will decrease as adaptation proceeds. Consequently, the dynamic convergence behavior can be significantly improved. The serial split idea is explored further to develop a new configuration for adaptive time delay estimation. Theoretical analysis shows that a two-fold convergence speed-up can be achieved when compared with the traditional model.

In regard to computational complexity of the split LMS adaptive algorithms, roughly $(M+1)/2$ extra additions and subtractions are needed in the parallel split structure while $(M-1)/2$ extra multiplications and additions are required in the serial split configuration. This overhead is only a moderate increase in computations when compared with $2(M+1)$ multiplications and additions that are generally needed in the traditional LMS algorithm, where M is the filter order.

In addition, we shall describe how a new type of predictor can be realized by combining the parallel split and serial split predictor. This structure is used for speech analysis as a typical illustration. Experimental results show that its convergence characteristics are comparable to those of the gradient lattice filter but it requires much less computation.

CONTENTS

FIGURE CAPTIONS

TABLE CAPTIONS

1 INTRODUCTION

Adaptive signal processing has its roots in adaptive control and in the mathematics of iterative processes, where first attempts were made to design systems that adapt to their environments. It has undergone a remarkable increase of interest and attention in recent years. This growth has been promoted by developments in microelectronics and VLSI circuit design that increase tremendously the amount of computing capability in processing of digital signals. In this chapter, a brief introduction to adaptive filter and adaptive system will be provided. In addition, some applications will be described to illustrate their practical uses.

1.1 ADAPTIVE FILTER AND ADAPTIVE SYSTEM

Many problems encountered in signal processing and communications nowadays involve removing noise and distortion due to physical processes that may be time varying or possibly unknown. These types of processes represent most of the problems in transmitting, receiving and extracting information from an underlying data sequence.

It is well known that the Wiener filter [1] can be used to extract useful information from noisy data by minimizing the mean-square value of an error formed by the difference between some desired response and the actual filter output. However, the design of such filter requires *a priori* information about the statistical characteristics of the data to be processed. The filter can achieve optimal performance only when the statistical characteristics of the input data match the a priori knowledge on which the design of the filter is based. When this knowledge is not known completely or the statistics of the data is time varying, it may not be

possible to design the Wiener filter or the design may no longer be optimum. In such circumstances, what we need is a device that can learn the input statistics from time to time and perform optimal filtering. Adaptive filter is a device that can accomplish these purposes.

An adaptive filter is basically a self designing filter in which it relies its operation characteristics on an adaptation rule, allowing the filter to perform satisfactorily in an environment where complete knowledge of the relevant signal statistics is not available. It starts from some predetermined set of initial conditions, representing complete ignorance about the circumstance. In a stationary environment, that is the statistics of the input data is not changed, it will converge to a Wiener filter in the mean-square sense. Under a nonstationary environment, however, the algorithm offers a tracking capability to keep up with time variations in the statistics of the input data to provide optimum performance of the filter, provided that the variations are slow enough to be followed.

An adaptive filter has essentially two distinct features. First, it has a finite number of internal adjustable parameters which can be used to control its functions over a useful range. Different parameter sets correspond to different filter structures or realizations which can affect the adaptation performance. Second, it has an updating algorithm enabling the filter transfer function to be changed in a useful manner in accordance with its external environment. As a result, the updating algorithm changes its characteristics by varying its internal parameters for a particular realization to achieve optimum accomplishment.

Broadly speaking, an adaptive system can be defined as a system which is provided with a means of continuously monitoring its own performance in relation to a given figure of merit or optimal condition and a means of modifying its own

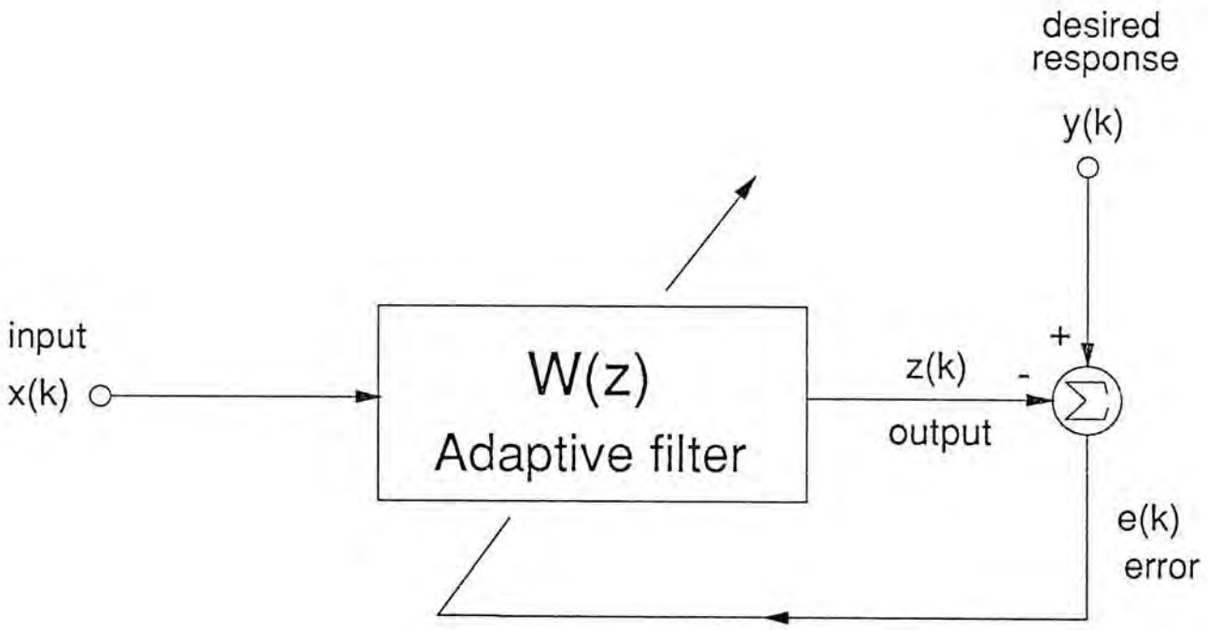


Fig. 1.1 An adaptive system

system parameters so as to approach this optimum [2]. In the context of signal processing, incorporating adaptive filters into a system forms an adaptive system. An adaptive system may be very complex in actual case. Figure 1.1 shows a simple adaptive system for illustration purpose. Referring to the diagram, $W(z)$ is a linear adaptive filter whose input is $x(k)$. The sequence $y(k)$ is the desired response representing the desired output of the adaptive filter. At each time instant k, the current value of the filter coefficients are used to achieve the filtering operation. The computed output error $e(k)$, formed by the difference between the desired response $y(k)$ and the filter output $z(k)$, is then utilized by the adaptive filter to change its internal parameters in the direction of their optimum values. As processing of $x(k)$ and $y(k)$ proceeds, the filter gradually learns the statistics of these signals and converges to its desired value and thus providing an optimal signal processing system.

1.2 APPLICATIONS OF ADAPTIVE FILTER

Adaptive filtering has been extensively used in many fields of applications, including geophysical signal processing, control, telecommunications, biomedical signal processing, the elimination of radar clutter, and sonar processing [3]-[5]. In the following sub-sections, some typical applications of adaptive filter will be described to demonstrate its practical usefulness.

1.2.1 SYSTEM IDENTIFICATION

Adaptive system identification has found a wide range of applications in digital filter design [6], adaptive control systems [7] and coherence estimation [8]. Suppose we have an unknown linear system called plant, with a set of discrete-time noisy observations in response to a known stationary excitation as the system input. The

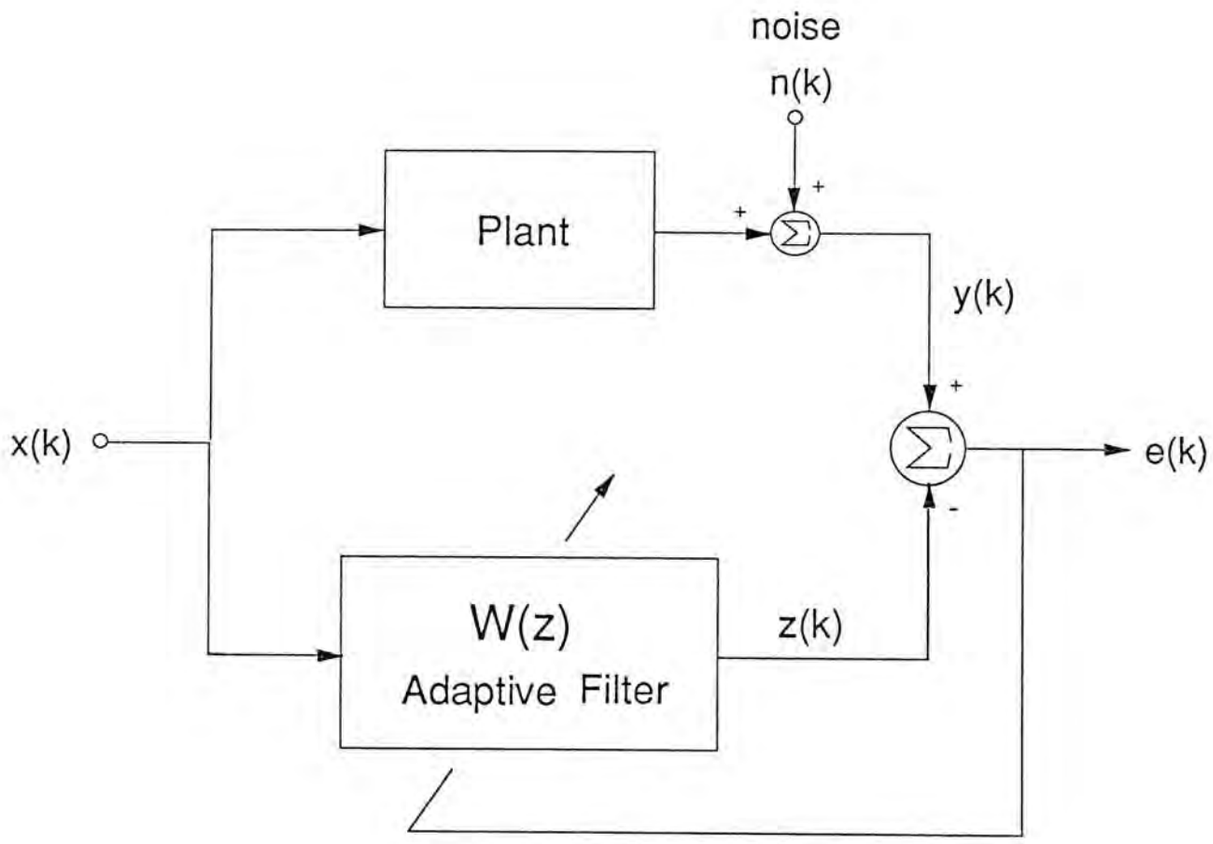


Fig. 1.2 Adaptive configuration for system identification

configuration for identifying the unknown plant is depicted in Figure 1.2. The adaptive filter $W(z)$ represents the transfer function of the plant model. The input $x(k)$ is fed into both the unknown plant and the plant model. The difference of the plant output corrupted by noise, $y(k)$, and the plant model output, $z(k)$, forms the error $e(k)$. At the beginning the output error is nonzero, implying that the plant model deviates from the unknown system. Because of the self tuning capability of the adaptive filter, $W(z)$ will adjust itself to minimize a certain criterion of the output error to produce a response that is as close as possible to the plant output. If the excitation $x(k)$ is robust in frequency content, then $W(z)$ will adapt to become a good representative of the unknown system.

When the unknown system is time varying, the output of the plant becomes nonstationary. In this case, $W(z)$ will allow itself to alter its parameters to track continuously the statistical variations of the plant.

1.2.2 NOISE CANCELLATION

Another popular application of adaptive filtering is to extract a signal which is corrupted by an interference or uncorrelated noise. This is achieved by an adaptive noise canceller as shown in Fig. 1.3. It has two inputs, one is called primary input containing signal $s(k)$ plus additive noise $n_1(k)$ while the other is called reference input $n_2(k)$ consisting a noise alone, where $n_2(k)$ is correlated to the primary input noise but uncorrelated with the primary input signal. The adaptive noise canceller tries to make use of the reference input to eliminate the noise in the primary input by utilizing an adaptive filter $W(z)$ to exploit the unknown correlation between $n_1(k)$ and $n_2(k)$.

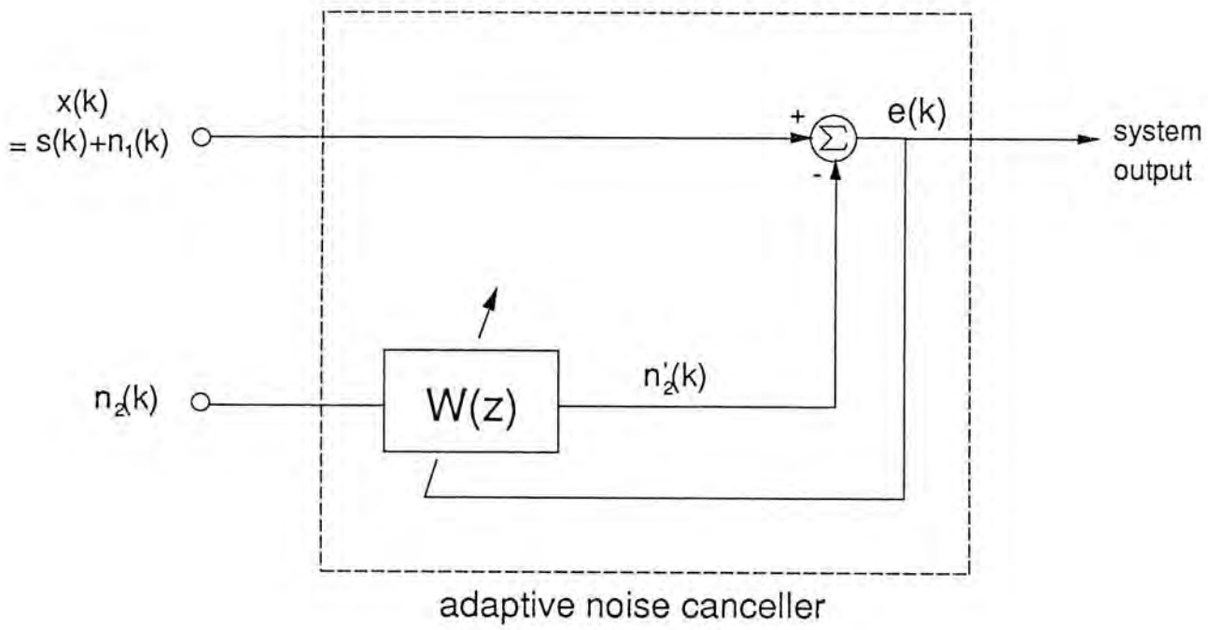


Fig. 1.3 An adaptive noise canceller

The reference input first passes through the adaptive filter $W(z)$ to form the output $n_2'(k)$. The difference of the primary input and the filter output gives an error which is used to adjust $W(z)$. As only the noise components in the two channels are correlated, $W(z)$ will adapt in such a way to generate an output which is as close a replica as possible to the primary input noise. As a result, in equilibrium state the output error will be an estimate of the primary input signal.

Applications of adaptive noise cancelling [9]-[10] include the cancellation of various forms of periodic interference in electrocardiographs, the cancelling of periodic interference in speech signals, the cancelling of broadband interference in the sidelobes of an antenna array, and the elimination of tape hum or turntable rumble during the playback of recorded broadband signals.

1.2.3 ECHO CANCELLATION

An extensive application of adaptive filtering is also found in telecommunications [11]-[15]. In telephone connections that involve the use of both four-wire and two-wire transmissions, an echo is generated at the hybrid that connects a four-wire to a two-wire transmission. When the telephone call is made over a long distance, for example via satellite, an echo represents an impairment that can be as annoying subjectively as the more obvious impairments of low volume and noise. Figure 1.4 shows a satellite circuit model with no echo protection. The hybrids at both ends of the circuit convert the two-wire transmissions used on customer loops and metallic trunks to the four-wire transmission needed for carrier circuits. Ideally, when person A on the left speaks, his speech should follow the upper transmission path to the hybrid on the right and from there be directed to the two-wire circuit. In practice, however, not all the speech energy is directed to

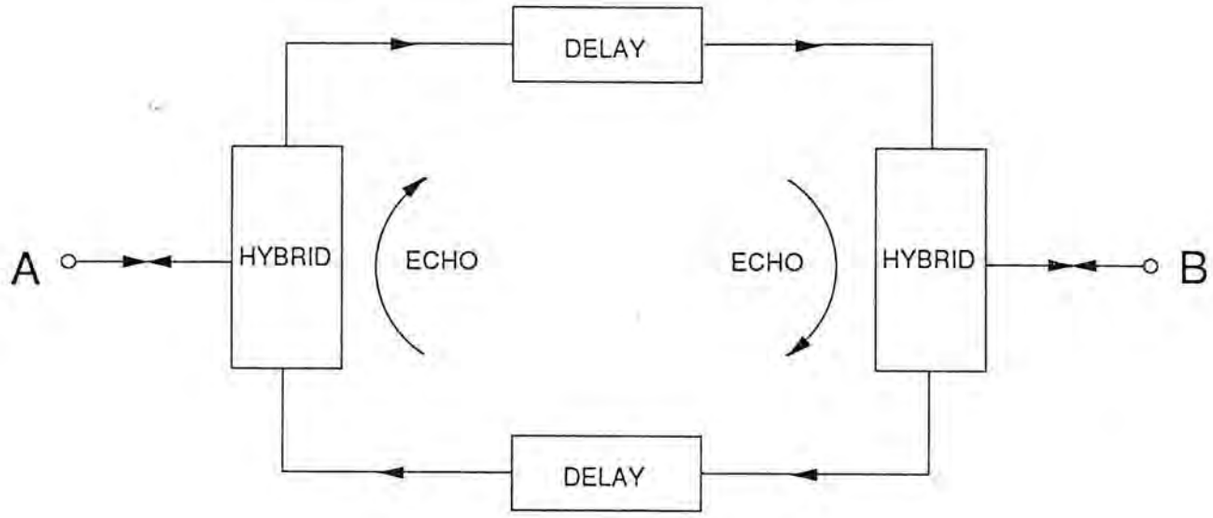


Fig. 1.4 A typical dialed connection for a long distance call

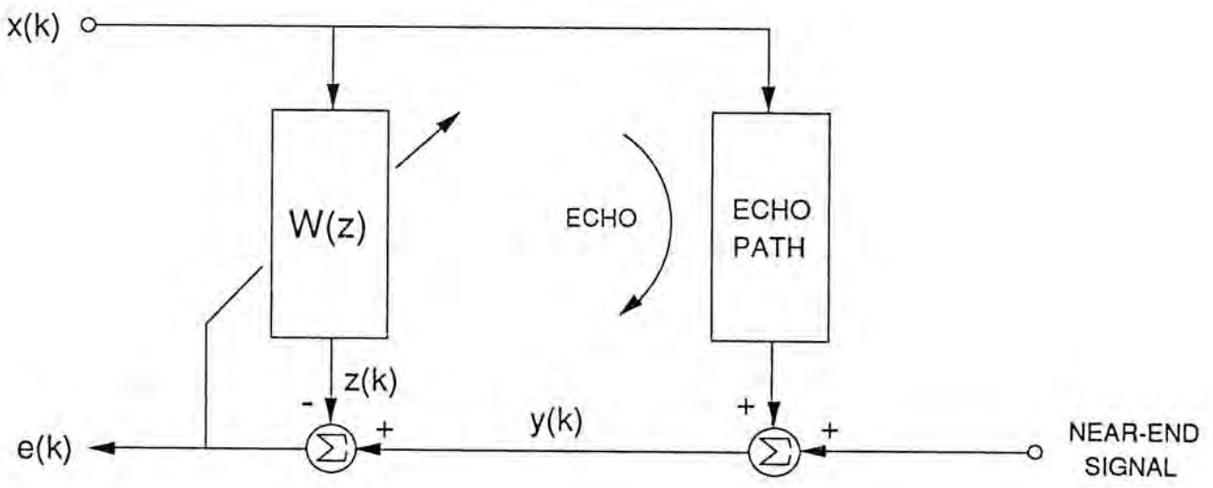


Fig. 1.5 Echo cancellation in telephone network

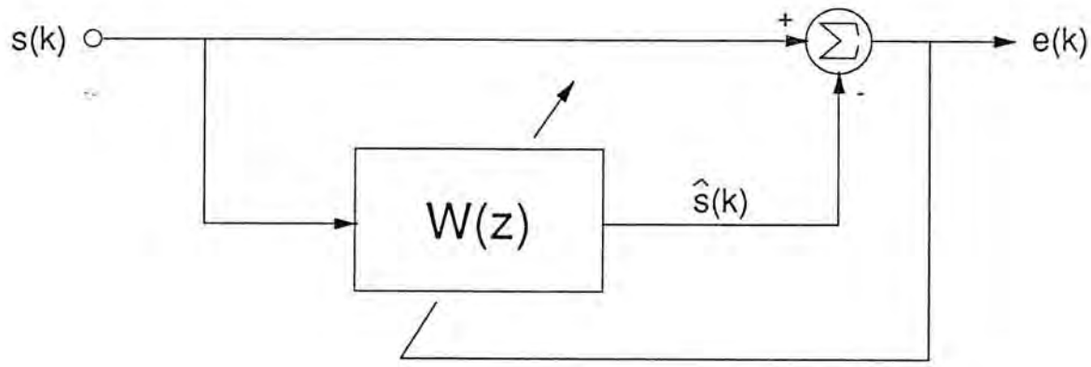
this two-wire circuit. Because of hybrid leakage, some of the energy is returned along the lower four-wire path which will be heard by the person on the left as an echo.

To overcome this problem, echo cancellers are installed in the network in pairs, one in each end. The operation of an adaptive echo canceller is illustrated in Figure 1.5. At a given end of the long-distance line, the incoming signal is fed into both the hybrid and the adaptive filter. The difference between the adaptive filter output and the hybrid output gives the output error, which is used to adjust the filter parameters. Because of hybrid leakage, the outgoing component of the hybrid is correlated with the incoming signal. The adaptive filter will model the transfer function of the leakage path of the hybrid as adaptation proceeds. In equilibrium, the adaptive filter will remove the leakage component as much as possible and leaving a clear (free of leakage) signal containing only the speech energy from the near end speaker as the output error.

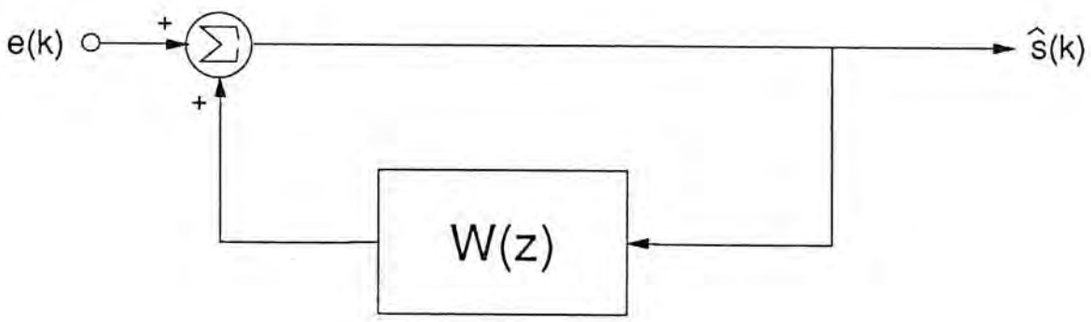
1.2.4 SPEECH PROCESSING

A major application of adaptive filtering is in the digital encoding of speech [16]. One approach to speech encoding is the analysis-synthesis method with a system block diagram shown in Figure 1.6. In this method, the speech production process is assumed to be governed by a simple model. The available speech signal $s(k)$ is used to derive the model parameters which are then quantized for transmission. To recover the speech signal in the receiving end, the speech is synthesized by emulating the speech production model from the received parameters.

One distinctive nature of speech signal is that it can be modeled by an auto-regressive (AR) process. That means, a current speech sample can be generated by a linear combination of the previous speech samples plus an innovation term.



(a)



(b)

Fig. 1.6 (a) Speech analysis model. (b) Speech synthesis model.

The innovation can either be a periodic impulse for voiced sound or a random noise for unvoiced sound. What speech analysis does is to determine the linear combination coefficients or the AR parameters and the innovation. Notice that linear combination of past samples can be represented by a filtering operation, the parameters of the model can be identified by minimizing an error function with output error given by $s(k)$ and the output of the filter $W(z)$. Afterwards, the AR parameters can be extracted from $W(z)$ and the innovation is just the output error. As speech signal is only quasi-stationary, that is it can only be considered as stationary within a short period of time, it is impossible to use a fixed filter in the model. Hence it is beneficial and indeed necessary to make $W(z)$ adaptive such that it can continuously monitor the input signal to determine the model parameters accurately. The adaptive filter $W(z)$ is usually referred to as an adaptive predictor because it predicts in an adaptive manner the present input by past input values.

Although we have only considered four applications of adaptive filtering, it is by no mean exhaustive, in fact, many other applications can be found in the literature [3]-[5].

The performance of an adaptive filter relies mainly on its adaptation rule to identify the optimum filter parameters. There are two well known optimization procedures for adaptive filtering. They are the least-mean-square (LMS) algorithm and the recursive least-square (RLS) algorithm. The details of the two algorithms will be described in the next chapter. It is understood that the RLS algorithm has a faster convergence speed than the LMS algorithm, however, the computation involved is much greater. More precisely, the LMS algorithm requires order M+1 operations while the RLS algorithm demands order $(M+1)^2$ operations, where M is the adaptive filter order. In most applications of adaptive filtering such as noise

cancellation and echo cancellation, the filter order M could be very large which makes real time implementation of the RLS algorithm very difficult if not impossible. Therefore, the LMS algorithm is usually preferred for most practical uses.

Although the implementation of an LMS adaptive filter is simple, the algorithm itself has an inherent deficiency that the adaptation speed or the learning rate depends very much on the eigenvalue spread (the ratio of the largest to the smallest eigenvalue) of the correlation matrix of the input signal. If the eigenvalue spread is large, the system will take a fairly long time to reach its optimal behavior. An open question for the LMS adaptive system is how the learning rate can be enhanced effectively without a considerable increase in system complexity.

It is widely accepted that a digital filter can be realized in different forms resulting in different system complexity and suffering from a various degree of finite wordlength effects. Whilst for adaptive filtering, it is also possible to configure an adaptive filter in many different structures. Each individual realization format will have its own distinct set of eigenvalues and hence the convergence behavior for the LMS adaptation process might be different. It seems that some simple filter structures can be devised so that the eigenvalue spread associated with the parameter adaptation can be greatly reduced while still demanding not a large increase in computations. In this thesis, the possible enhancement of convergence speed by representing an adaptive filter in two simple but novel structures, namely the parallel and the serial form, will be investigated.

It will be demonstrated in detail in the following chapters that by splitting an adaptive filter in parallel form or serial form, it will give promising result in improving the convergence performance because the eigenvalue spread for the adaptation process can be decreased substantially while the overhead in computation is only very modest.

1.3 CHAPTER SUMMARY

A brief introduction of adaptive filter and adaptive system has been given. Conceptually, an adaptive filter is simply an ordinary filter but with varying filtering characteristics to keep up with some unknown or changing environments. With the advent of VLSI technology, the use of adaptive filter has increased tremendously in the last few years, and this trend is expected to continue in the years ahead. Some of its potential applications in control, signal extraction, telecommunication and speech processing have been discussed. In the next chapter, we shall look into further details of an adaptive filter by investigating some of its widely used structures and the accompanying adaptation algorithms.

REFERENCES

[1] N. Wiener, *Extrapolation, Interpretation, and Smoothing of Stationary Time Series*. NY: MIT Press and Wiley, 1949.

[2] V. W. Eveleigh, *Adaptive Control and Optimization Techniques*. NY: McGraw-Hill, 1967.

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

[4] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[5] M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, Algorithms, and Applications*. Boston, MA: Kluwer, 1984.

[6] B. Widrow, P. F. Titchener, and R. P. Gooch, "Adaptive design of digital filters," in *Proc. ICASSP-81*, Feb. 1981, pp. 43-142.

[7] G. F. Fanklin and J. D. Powell, *Digital Control of Dynamic Systems*. Reading, MA: Addison-Wesley, 1980.

[8] D. H. Youn, N. Ahmed, and G. C. Carter, "Magnitude-squared coherence function estimation: an adaptive approach," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-31, pp. 137-142, Feb. 1983.

[9] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong and R. C. Goodlin, "Adaptive noise cancelling: principle and applications,", *Proc. IEEE*, vol. 63, Dec. 1975.

[10] E. H. Satorius, J. D. Smith and P. M. Peeves, "Adaptive noise cancelling of sinusoidal interference using lattice structure," in *Proc. IEEE ICASSP-79*, pp. 929-932.

[11] N. Dwmytko and K. English, "Echo cancellation on time-variant circuits, " *Proc. IEEE*, vol. 65, pp. 444-453, Mar. 1977.

[12] K. Ochiai, T. Araseki and T. Ogihara, "Echo canceller with two echo path models," *IEEE Trans. Commun.*, vol. COM-25, pp. 589-595, June 1977.

[13] M. M. Sondhi and D. A. Berkley, "Silencing echoes on the telephone network," *Proc. IEEE*, vol. 68, pp. 948-963, Aug. 1980.

[14] D. G. Messerschmitt, "Echo cancellation in speech and data transmission," *IEEE J. Select. Areas Commun.*, vol. SAC-2, pp. 283-297, Mar. 1984.

[15] P. C. W. Yip and D. M. Etter, "An adaptive multiple echo canceller for slowly time-varying echo paths," *IEEE Trans. Commun.*, vol. 38, pp. 1693-1698, Oct. 1990.

[16] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals.* Englewood Cliffs, NJ: Prentice-Hall, 1978.

2 ADAPTIVE FILTER STRUCTURES AND ALGORITHMS

The performance of an adaptive filter is governed by its structure and adaptation algorithm. The structure of a filter is defined by a set of parameters that characterize the filter. Different realizations of an adaptive filter corresponds to adjusting different parameter sets. On the other hand, adaptation algorithm is a training rule used to alter the parameters of an underlying filter so as to achieve certain optimum filtering task. With a particular type of algorithm, modifying different parameter sets will affect the convergence speed for their respective realizations and hence some structures may be more attractive than the others. Of course, given a specific filter structure, utilizing different types of algorithms will also affect its adaptation capability. Generally speaking, a rapid convergence algorithm requires a large amount of computations. The many possible combinations of filter structures and adaptation rules lead to a bewildering variety of adaptive filters with different performance and complexity. In this chapter, some of the commonly used adaptive filter structures and adaptation algorithms will be reviewed.

2.1 FILTER STRUCTURES FOR ADAPTIVE FILTERING

There are two basic types of adaptive filters as in non-adaptive filters, namely, adaptive finite impulse response (FIR) filter and adaptive infinite impulse response (IIR) filter. Non-adaptive IIR filter is widely used in digital signal processing because it has computational improvement over the FIR filter and its stability is always guaranteed when properly designed. However, adaptive IIR filter is of limited

applications in present moment. This is due to first of all, stability for adaptive IIR filter may not be guaranteed explicitly as a part of the adaptation algorithm's design. That means, instability of the filter because of poles straying outside the stable region may occur during adaptation. Second, adaptation of an IIR filter is a highly nonlinear process and convergence to local minimum is highly possible. Therefore, global convergence for optimal performance is not assured. Third, the theory of adaptive IIR filters is incomplete because the analysis involves highly nonlinear systems. Adaptive IIR filter is currently a popular research topic in adaptive signal processing [1]-[6]. On the contrary, adaptive FIR filter does not have the stability problem because it is unconditionally stable when certain conditions about its adaptation algorithm are fulfilled. In addition, it is relatively simple to design and construct, and there exist well understood algorithms whose performance are well documented [7]-[9]. Thus, much of the reported literature on adaptive filters has been based on FIR filter approach. In this thesis, we shall limit our scope to adaptive FIR filter only.

There are many filter structures which can be equivalently implemented an FIR filter. Specifically, two particular filter structures are almost universally employed. They are transversal and lattice structures.

Figure 2.1 shows the structure for transversal FIR filter which is a direct realization of its transfer function. The filter output $z(k)$ is formed by a linear combination of the current and the past M input samples, where M is the filter order. The filter coefficients $w_0, w_1, \ldots, w_M$ are the parameters defining a transversal form FIR filter. This structure is perhaps the simplest implementation of an FIR filter.

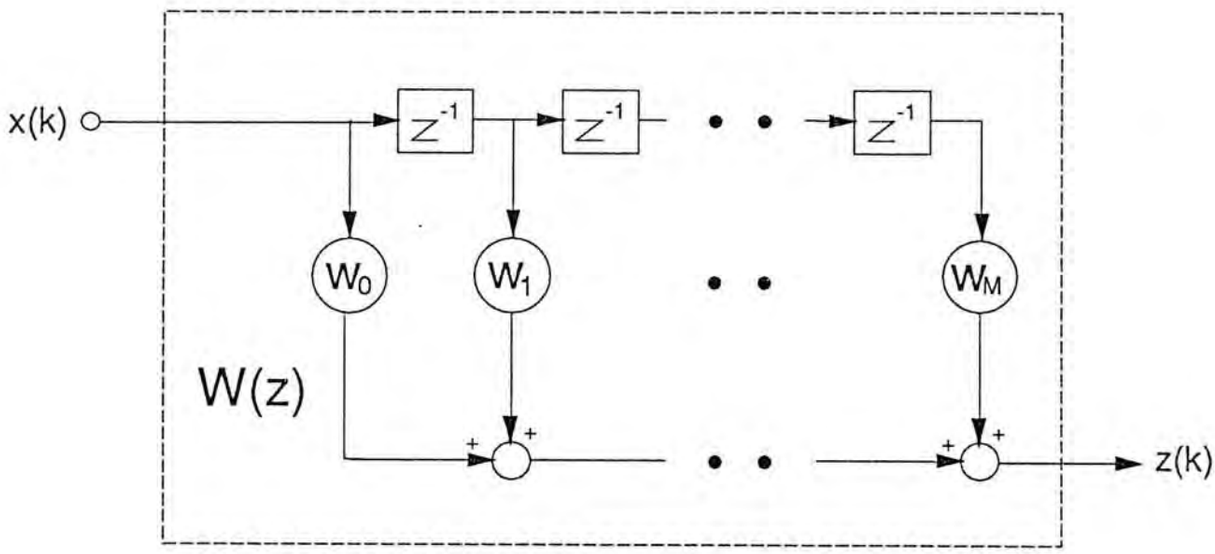


Fig. 2.1 Transversal filter structure

The lattice realization of an FIR filter is illustrated in Figure 2.2 [10]-[11]. This structure transforms the input signal into forward and backward residual samples, with delays added into the backward channel. These signals are multiplied by the partial correlation (PARCOR) coefficients, k_n, to form the filtering output. The parameters of a lattice filter are the PARCOR coefficients. Although lattice filter seems to be more complicated than transversal filter, it is found to be particularly useful in linear predictive coders for speech processing [12] because of some intriguing properties of the PARCOR coefficients.

There are many other structures that could potentially represent an FIR filter. For example, one can factorize the transfer function of an FIR filter and then implement it in cascade of second order sections [13]. Figure 2.3 shows such realization. As there are many realizations, there exists problems of what criterion should be used in choosing a structure for adaptive filtering. In the non-adaptive filter case, the primary criterion for selecting a filter structure for a digital implementation is finite word-length effects, such as the effect of round-off errors internal to the filter. This is also an important consideration for an adaptive filter, but we are more concerned with the following two factors. First, trade-off between adaptation speed and computations for an adaptive filter implementation. It has been demonstrated [14] that the convergence rate of a lattice filter is faster than that of a transversal filter, nevertheless more computations is involved. Second, there should be a simple and analytically tractable relationship between the transfer function of the filter and its parameters. In the case of a transversal filter, this relationship is simple because it is linear. Whereas for most other structures such as the cascade the relationship is nonlinear. The lattice structure also has a nonlinear relationship, but fortunately the relationship can be recursively represented. The complicated nonlinear relation between parameter and transfer function for most

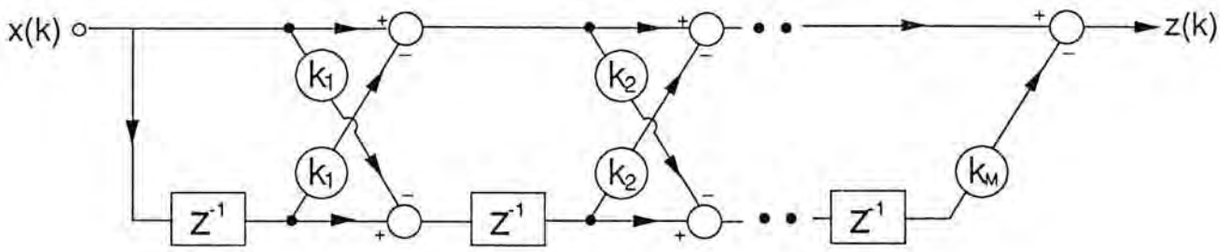


Fig. 2.2 Lattice filter structure

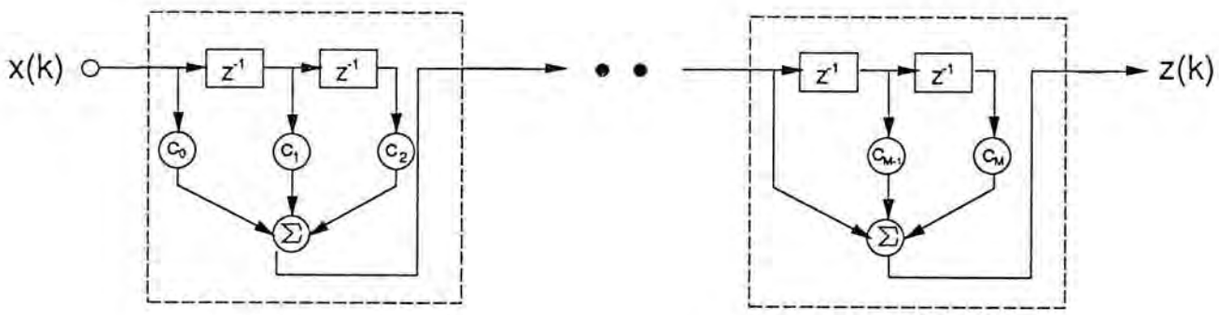


Fig. 2.3 Cascade filter structure

other structures makes their corresponding adaptation algorithms difficult analyze and assess their performance. In the adaptation algorithm provided in the following section, we shall consider the transversal structure only.

2.2 ADAPTATION ALGORITHMS

Consider a simple filtering problem as shown in Figure 2.4. $W(z)$ is an adaptive FIR filter whose input is $x(k)$. The difference between the desired response $y(k)$ and the filter output forms the output error $e(k)$,

$$e(k) = y(k) - \sum_{i=0}^{M} w_i x(k-i) = y(k) - w^t x(k) \tag{2.1}$$

where

$$w = [\, w_o \quad w_1 \quad \ldots \quad w_M \,]^t \tag{2.2}$$

is the M+1 filter parameter vector and

$$x(k) = [\, x(k) \quad x(k-1) \quad \ldots \quad x(k-M) \,]^t \tag{2.3}$$

is the M+1 input vector with the superscript t denoting transpose operation. The task is to find iteratively a set of filter parameters such that the filter output is as close to the desired response $y(k)$ as possible.

In evaluating the performance of an adaptation algorithm, three factors should be considered, namely, (a) convergence speed, (b) stability and numerical accuracy and (c) computational complexity. Convergence speed of an adaptation algorithm is an important factor because it determines the maximum rate of change of the input nonstationarities that can be successfully tracked by an adaptive filter. Stability refers to the possibility of reaching the desired solution and numerical accuracy

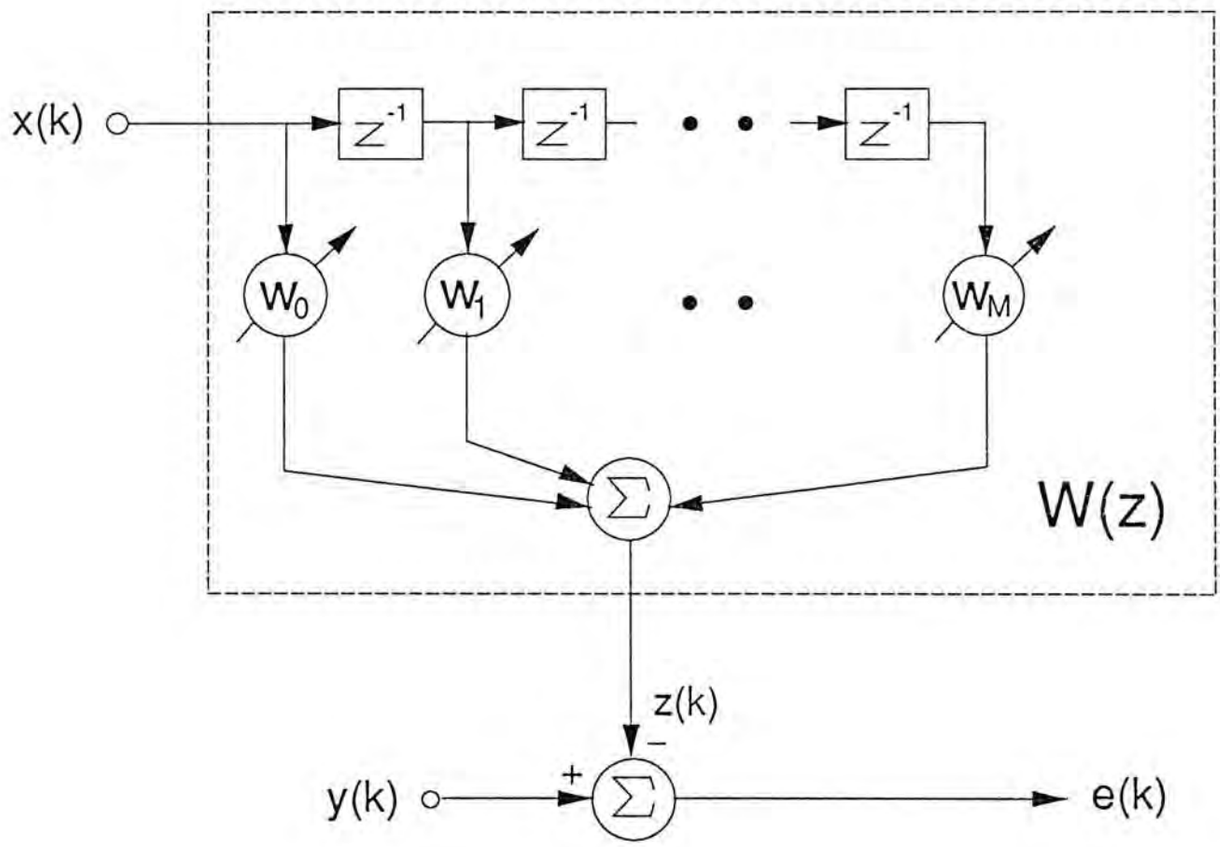


Fig. 2.4 A typical Wiener filtering configuration

relates to how closely the algorithm can track the optimal Wiener solution. Finally, computational complexity concerns with the number of operations required to update filter parameters from one time instant to the next.

There are two well known adaptation algorithms for adaptive FIR filter. They are the recursive least-square (RLS) algorithm and the Widrow-Hoff least-mean-square (LMS) algorithm. The RLS algorithm has been used extensively for system identification in control engineering and time-series analysis. In spite of its potentially superior performance, its use in signal processing applications has been relatively limited, due to its high computational requirements. However, in recent years there has been renewed interest in RLS algorithm, especially in its computationally efficient versions [15]-[18] due to its fast convergence. On the other hand, the LMS algorithm is computationally simple and easy to implement. Therefore, it has gained considerable popularity for many years.

2.2.1 THE LMS ADAPTATION ALGORITHM

The LMS algorithm was originally developed by Widrow and Hoff in 1960 [19]-[20]. Since then, it has been widely studied and found numerous applications in many areas [21]-[27]. In the LMS adaptation algorithm, the objective function ξ to be minimized is the mean-square error (MSE). From (2.1), it is given by

$$\begin{aligned}\xi &= E[e^2(k)] = E[(y(k) - \boldsymbol{w}^t \boldsymbol{x}(k))^2] \\ &= E[y^2(k)] - 2\boldsymbol{g}^t \boldsymbol{w} + \boldsymbol{w}^t \boldsymbol{R}\, \boldsymbol{w}\end{aligned} \tag{2.4}$$

where $\boldsymbol{R} = E[\boldsymbol{x}(k)\boldsymbol{x}^t(k)]$ is the input autocorrelation matrix and $\boldsymbol{g} = E[y(k)\boldsymbol{x}(k)]$ is the cross-correlation vector between the input vector $\boldsymbol{x}(k)$ and the desired response $y(k)$. Note that ξ is a quadratic function with respect to the elements of the vector $\boldsymbol{w}$, and hence there will be a single unique minimum. As a simple case for $M = 1$,

the plot of the mean-square error versus filter weights is depicted in Figure 2.5. It can be observed that the error criterion, or performance surface, is a paraboloid. The minimum point of the surface is the least possible MSE, ξ^o, and the corresponding weights are the optimal solution. The optimal weight vector w^* can be obtained by setting the derivative of ξ shown in (2.4) with respect to w to zero, which gives

$$w^* = R^{-1} g \tag{2.5}$$

At this optimal point, the minimum MSE is equal to

$$\xi^o = E[y^2(k)] - g^t w^* \tag{2.6}$$

Using (2.5) and (2.6), ξ can be expressed as

$$\xi = \xi^o + (w - w^*)^t R (w - w^*) \tag{2.7}$$

The form (2.7) of the MSE is compact and has an intuitive interpretation. At any instance during adaptation, $(w - w^*)$ is different from zero and ξ will be greater than ξ^o. After convergence, $(w - w^*)$ becomes zero and the steady state MSE will be ξ^o. Equation (2.7) is found especially useful in the analysis of the LMS algorithm.

In searching iteratively the minimum point of the performance surface, the filter weights are modified from one time instant to the next according to

$$w(k+1) = w(k) + \Delta w(k) \tag{2.8}$$

where $\Delta w(k)$ is a correction term that must be chosen properly in order to ensure the weight vector $w(k)$ will eventually converge to the optimal value w^* as more and more data become available. That is,

$$w(k) = w^* \quad , \qquad \text{as } k \to \infty \tag{2.9}$$

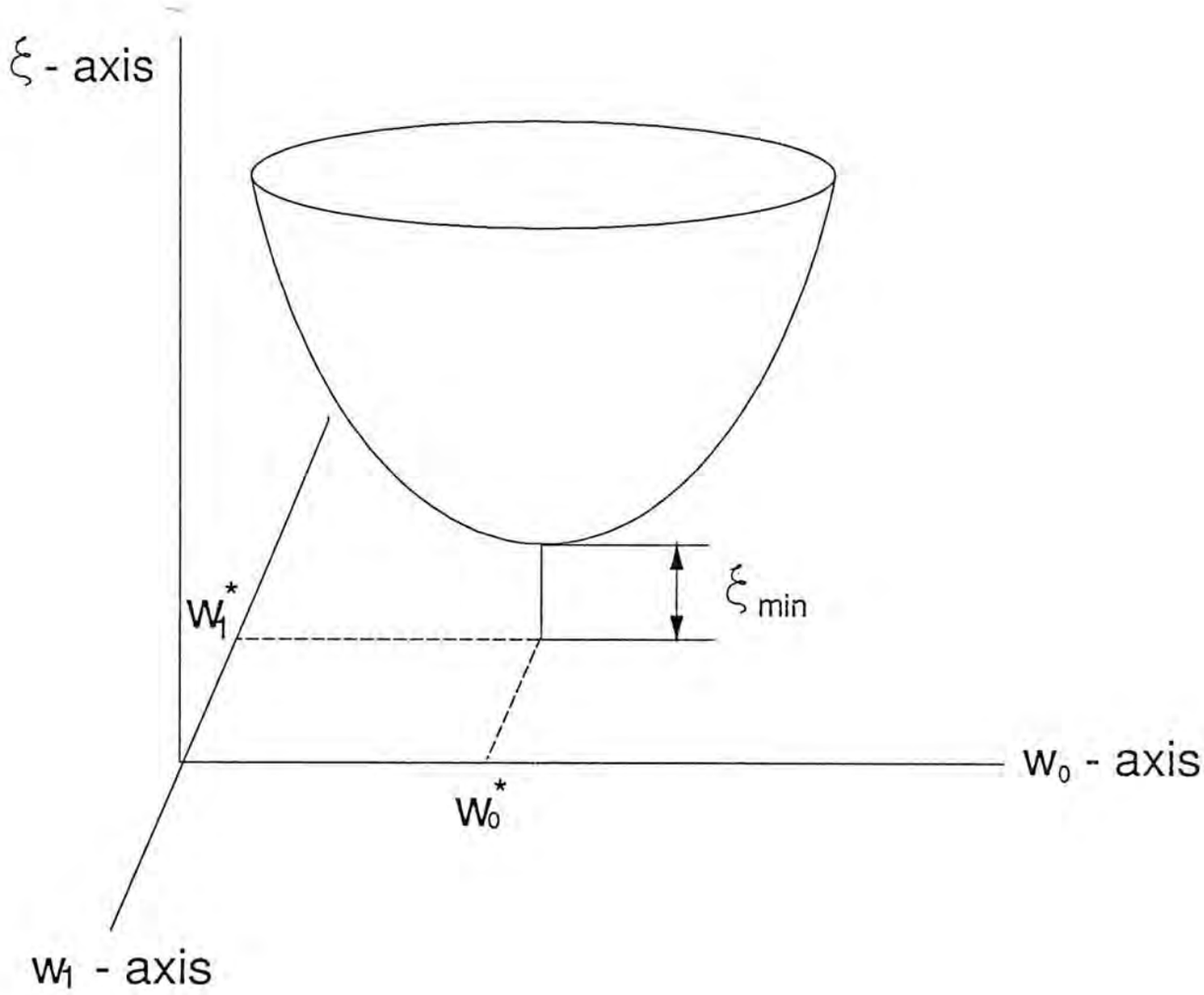


Fig. 2.5 MSE surface with two weights

A simple way to reach the minimum is to make use of the quadratic nature of the performance surface by the method of steepest descent. According to this optimization technique, the correction term $\Delta w(k)$ is proportional to the negative of the gradient at $w(k)$,

$$\Delta w(k) = -\mu \frac{\partial E[\, e^2(k)]}{\partial w(k)} = -2\,\mu \{\, \boldsymbol{R}\, w(k) - \boldsymbol{g}\,\} \tag{2.10}$$

where μ is a small positive constant called step size. With sufficiently small $\Delta w(k)$ shown in (2.10), Taylor's series expansion of the error criterion in the $(k+1)$th iteration yields

$$\begin{aligned} \xi(\, w(k+1)\,) &\approx \xi(\, w(k)\,) + \Delta w(k) \frac{\partial \xi(k)}{\partial w(k)} \\ &= \xi(\, w(k)\,) - \mu \left\{ \frac{\partial \xi(k)}{\partial w(k)} \right\}^2 \quad \leq \quad \xi(\, w(k)\,) \end{aligned} \tag{2.11}$$

Thus steepest-descent method ensures the change in filter parameters moves the error function closer to its minimum than before. At the minimum point the gradient is zero which implies (2.9).

The iterative procedure described above requires a priori knowledge of the correlations $\boldsymbol{R}$ and $\boldsymbol{g}$ which are not known in practice. The LMS algorithm is an approximation of the steepest descent method by replacing the unknown gradient with the instantaneous gradient by ignoring the expectation operation, that is

$$\Delta w(k) = -\mu \frac{\partial e^2(k)}{\partial w(k)} = -2\,\mu\, e(k)\, \boldsymbol{x}(k) \tag{2.12}$$

so that the weight adaptation equation becomes

$$w(k+1) = w(k) + 2\,\mu\, e(k)\, \boldsymbol{x}(k) \tag{2.13}$$

The iterative equation described by (2.13) is called the LMS adaptation algorithm. It is also named the stochastic gradient algorithm because of using approximate gradient instead of the true gradient. The LMS algorithm has several attractive features : it is simple and easy to implement, it does not requires any a priori information and it allows itself to real-time processing. With the step size μ chosen to be power of two, it can be observed from (2.1) and (2.13) that the algorithm requires a total of $2(M+1)$ multiplications and additions for each iteration to calculate $e(k)$ and update parameters.

2.2.1.1 CONVERGENCE ANALYSIS

Although the LMS algorithm has been extensively studied [7]-[9], [28]-[29], the iterative scheme does not permit itself to analysis for all class of data and almost all results to date have been developed on the assumption that the input vectors are statistically independent. Nevertheless, it has been shown experimentally [30] that for a sufficiently small step size μ, the results based on the independence assumption is closely agreed with the experimental results. In the analysis provided, it is also assumed that the independence assumption holds.

An important consideration of an iterative scheme is the speed with which the algorithm converges to the optimum solution. Taking statistical expectations of (2.13), we have

$$E[\,w(k+1)] = E[\,w(k)] + 2\mu g - 2\mu E[\,x(k)\,x^{t}(k)\,w(k)] \tag{2.14}$$

Assuming $x(k)$ and $w(k)$ to be independent, then (2.14) can be simplified to

$$E[\,w(k+1)] = E[\,w(k)] - 2\mu R\,\{\,w(k) - w^{*}\} \tag{2.15}$$

Introducing the parameter error vector

$$\tilde{w}(k) = w(k) - w^* \quad , \tag{2.16}$$

we have

$$E[\tilde{w}(k+1)] = (\boldsymbol{I} - 2\mu \boldsymbol{R})\, E[\tilde{w}(k)] \tag{2.17}$$

where $\boldsymbol{I}$ is an identity matrix of size M+1. To see more precisely the adaptation behavior of the weight error vector $\tilde{w}(k)$, equation (2.17) can be decoupled into M+1 independent equations. Being a real and symmetric matrix, the input correlation matrix $\boldsymbol{R}$ can be factorized into the form

$$\boldsymbol{R} = \boldsymbol{U} \Lambda \boldsymbol{U}^t \tag{2.18}$$

where $\boldsymbol{U}$ is an orthonormal matrix whose columns are the eigenvectors of $\boldsymbol{R}$ and Λ is a diagonal matrix with diagonal elements equal to the eigenvalues of $\boldsymbol{R}$. Notice that an autocorrelation matrix is always positive semidefinite [7], the eigenvalues of $\boldsymbol{R}$ are all real and non-negative. Define the transformed weight misalignment vector as

$$\tilde{w}'(k) = \boldsymbol{U}^t\, \tilde{w}(k) \tag{2.19}$$

Premultiplying both sides of (2.17) by $\boldsymbol{U}^t$ yields

$$E[\tilde{w}'(k+1)] = (\boldsymbol{I} - 2\mu \Lambda)\, E[\tilde{w}'(k)] = (\boldsymbol{I} - 2\mu \Lambda)^{k+1}\, \tilde{w}'(0) \tag{2.20}$$

where $\tilde{w}'(0)$ is the transformed initial weight error vector. As k increases, the expected weight vector $w(k)$ will reach the optimal solution, provided that the right side converges to zero. That means, convergence is guaranteed if

$$0 < \mu < \frac{1}{\lambda_{max}} \tag{2.21}$$

and λ_{max} is the largest eigenvalue of $\boldsymbol{R}$. Since λ_{max} is usually unknown in advance, μ can be chosen to satisfy the more restrictive bound

$$0<\mu<\frac{1}{tr(\boldsymbol{R})} \tag{2.22}$$

in which tr is the trace operation. As long as (2.21) is satisfied, $E[\,w(k)]$ will converge exponentially to w^* according to M+1 natural modes with the time constant of the ith natural mode given by

$$\tau_i=\frac{1}{2\,\mu\,\lambda_i}\quad,\qquad\qquad i=1,2,\ldots,M+1 \tag{2.23}$$

The smaller the time constant, the faster will be the adaptation speed.

Corresponding to the adjustment in filter weights, the mean-square error will be decreased in each iteration. From (2.7) and (2.18), the error function in the kth iteration can be expressed as

$$\begin{aligned}\xi(k)&=\xi^o+E[\,\tilde{w}'^t(k)\,\Lambda\,\tilde{w}'(k)]\\&=\xi^o+tr(\,\Lambda\,E[\,\tilde{w}'(k)\,\tilde{w}'^t(k)]\,)\end{aligned} \tag{2.24}$$

Ignoring the weight vector fluctuation during adaptation, $E[\,\tilde{w}'(k)\,\tilde{w}'^t(k)]$ can be approximated by $E[\,\tilde{w}'(k)]\,E[\,\tilde{w}'^t(k)]$. When (2.20) is substituted into (2.24), we obtain

$$\xi(k)=\xi^o+\sum_{i=1}^{M+1}\lambda_i\,(\,1-2\,\mu\,\lambda_i\,)^{2k}\,\tilde{w}_i'(0) \tag{2.25}$$

and the entity $\tilde{w}_i'(0)$ designates the ith element of the vector $\tilde{w}'(0)$. The plot of $\xi(k)$ versus the number of iteration k is called a learning curve.

When the input signal $x(k)$ is a random sequence of power σ_x^2, λ_i will be σ_x^2 and (2.25) becomes

$$\xi(k)=\xi^o+\sigma_x^2\,(\,1-2\,\mu\,\sigma_x^2\,)^{2k}\sum_{i=1}^{M+1}\tilde{w}_i'(0) \tag{2.26}$$

indicating that $\xi(k)$ converges exponentially to its optimal value. Figure 2.6 shows a typical learning curve of identifying an unknown system given by

$$W^*(z) = 0.5 + z^{-1} + 1.5\,z^{-2} + 2\,z^{-3} + 2.5\,z^{-4} \tag{2.27}$$

in this respect. The input was Gaussian distributed with unity power and a Gaussian distributed random noise of power 0.001 was added to the unknown system output. The step size was set to 0.0008. It can be seen that the MSE converges exponentially to its minimum value as predicted by (2.26). When the input is correlated, it can be observed from (2.25) that the curve consists of a sum of exponentials and each of which corresponds to a natural mode of the algorithm. The exponential decay for the ith natural mode has a time constant given by

$$\tau_{\xi,i} = \frac{1}{4\mu\lambda_i} \quad , \qquad i = 1, 2, \ldots, M+1 \tag{2.28}$$

The slowest converging mode of $\xi(k)$ corresponds to the largest value of $|1 - 2\mu\lambda_i|$. Accordingly, the longest time constant involved in (2.25) is equal to

$$\tau_{\xi,\max} = \frac{1}{4\mu\lambda_{\min}} \tag{2.29}$$

where $\lambda_{\min}$ is the smallest eigenvalue of $\boldsymbol{R}$. Using the bound on μ shown in (2.21), it can be deduced that

$$\tau_{\xi,\max} > \frac{1}{4}\frac{\lambda_{\max}}{\lambda_{\min}} \tag{2.30}$$

This leads to the insight that the adaptation speed of the LMS algorithm is determined by the eigenvalue spread, $\chi(\boldsymbol{R})$, of the input correlation matrix $\boldsymbol{R}$, where

$$\chi(\boldsymbol{R}) = \frac{\lambda_{\max}}{\lambda_{\min}} \tag{2.31}$$

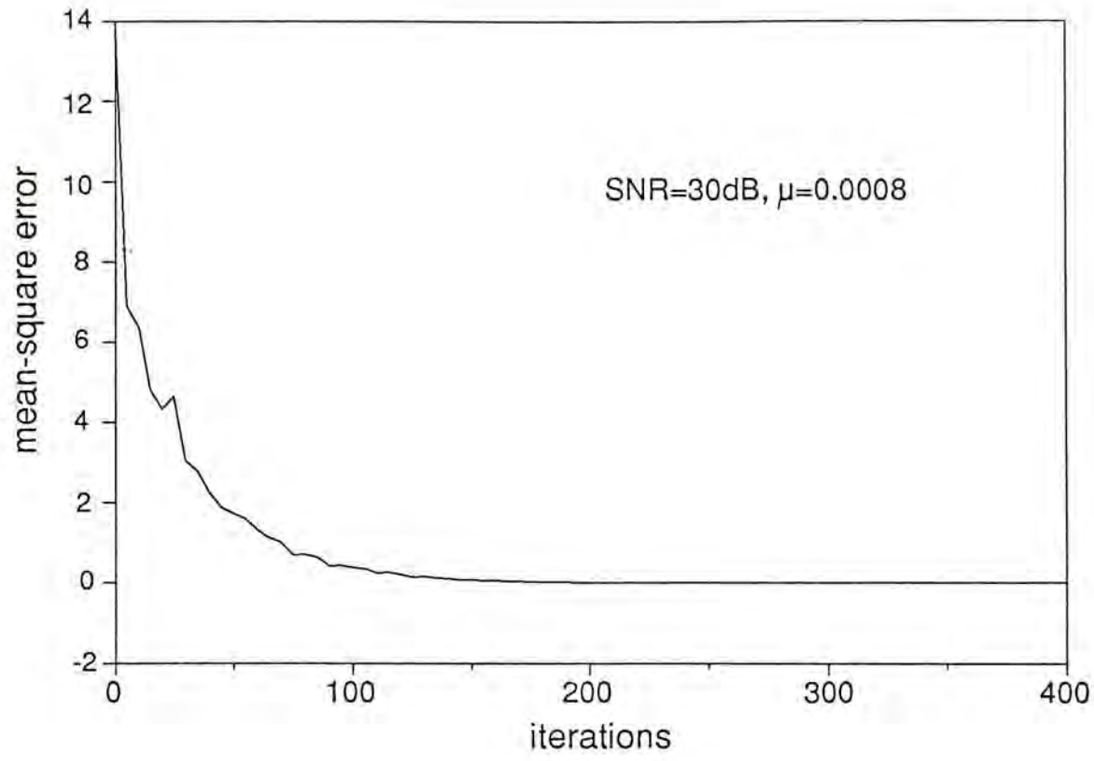


Fig. 2.6 Learning curve with random input

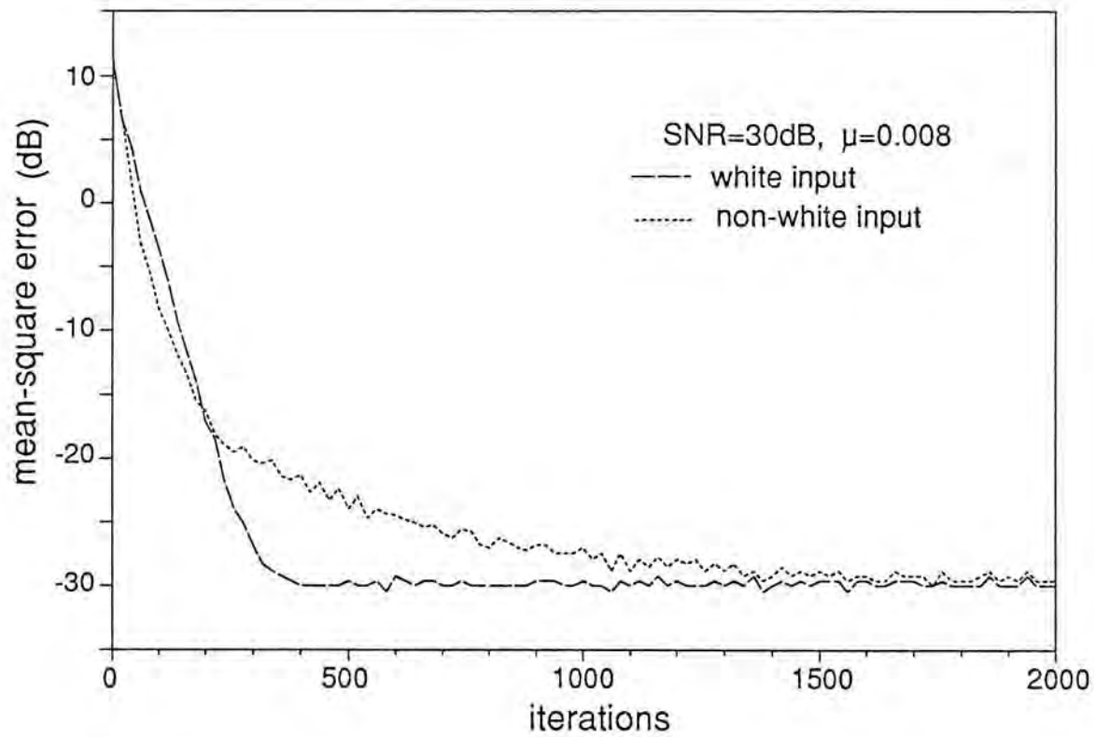


Fig. 2.7 Comparison of convergence speed with different eigenvalue spreads

The greater the spread, the longer the algorithm will take to converge to the optimum solution. That means the adaptation rate of an adaptive system will be slowed down with a correlated input signal. Alternatively, a white input signal can produce a faster adaptation speed.

Figure 2.7 illustrates the effect of eigenvalue spread on the convergence properties of the LMS algorithm. The unknown system given in (2.27) was to be identified with two distinct inputs, one is Gaussian distributed random signal while the other is a correlated signal generated from

$$\begin{aligned} x(k) &= 0.8914\,x(k-1) - 0.09688\,x(k-2) + 0.4345\,x(k-3) \\ &\quad - 0.5184\,x(k-4) + \beta(k) \end{aligned} \tag{2.32}$$

where $\beta(k)$ is a Gaussian distributed zero mean random sequence. The eigenvalue spread of the random input is unity and that of the correlated input was found to be 33.1. The input signal power was fixed to unity and ξ^o was selected to be 0.001. The step size was chosen to be 0.0008. An inspection of the plot reveals that the performance of an LMS adaptive system deteriorates when the input is not random. It has been confirmed by extensive simulation results that the larger the eigenvalue spread, the slower the adaptation speed.

2.2.1.2 STEADY STATE PERFORMANCE

Since approximate gradients are used in the LMS algorithm, the weight vector $w(k)$ never really reaches the theoretical limiting value w^*. Instead, it stabilizes and continuously fluctuates about this value at steady state. To give a quantitative measure of the fluctuation, we start by rewriting the LMS algorithm as

$$\begin{aligned} w(k+1) &= w(k) - \mu E[-2\,e(k)\,x(k)] + \mu\eta(k) \\ &= w(k) - 2\,\mu R\,\tilde{w}(k) + \mu\eta(k) \end{aligned} \tag{2.33}$$

where $\eta(k)$ is the zero mean gradient noise vector having a steady state covariance matrix given by

$$cov\{\eta(k)\} = E[\eta(k)\eta^t(k)] = 4E[e^2(k)x(k)x^t(k)]$$
$$\approx 4E[e^2(k)]E[x(k)x^t(k)] = 4\xi^o R \tag{2.34}$$

Subtracting both sides of (2.33) by w^* and premultiplying both sides by U^t, we have

$$\tilde{w}'(k+1) = (I - 2\mu\Lambda)\tilde{w}'(k) + \mu U^t \eta(k) \tag{2.35}$$

If we assume $x(k)$ is uncorrelated over time, $\eta(k)$ will be uncorrelated with $w(k)$ and hence $U^t\eta(k)$ and $\tilde{w}'(k)$ are uncorrelated. Postmultiplying both sides of (2.35) by their transposes and taking expectation gives

$$cov\{\tilde{w}'(k+1)\} = (I - 2\mu\Lambda)cov\{\tilde{w}(k))\}(I - 2\mu\Lambda)$$
$$+\mu^2 U^t cov\{\eta(k)\}U \tag{2.36}$$

In equilibrium, $cov\{w'(k+1)\}$ is identical to $cov\{\tilde{w}'(k)\}$. Using (2.34), (2.36) becomes

$$(I - \mu\Lambda)cov\{\tilde{w}'(k)\} = \mu\xi^o I \tag{2.37}$$

When μ is small, we have

$$cov\{\tilde{w}'(k)\} = \mu\xi^o I \tag{2.38}$$

The covariance of the weight vector $w(k)$ can now be expressed as

$$cov\{w(k)\} = U\,cov\{\tilde{w}'(k)\}U^t = \mu\xi^o I \tag{2.39}$$

The entity misadjustment, which is defined as the ratio of the excess MSE to the minimum MSE, is a measure of how closely the adaptive process tracks the true Wiener solution. From (2.24) and (2.38), the excess MSE is given by

$$\text{excess MSE} = tr(\Lambda E[\tilde{w}'(k)\,\tilde{w}'^{t}(k)]) = \mu\,\xi^{o}\,tr(\boldsymbol{R}) \tag{2.40}$$

and the misadjustment M is equal to

$$M = \mu\,tr(\boldsymbol{R}) \tag{2.41}$$

Obviously, a small μ can give small weight vector fluctuation and hence little misadjustment but the convergence speed will be slowed down as indicated in (2.28). This is a basic trade-off of the LMS algorithm between accuracy and convergence speed.

2.2.2 THE RLS ADAPTATION ALGORITHM

Referring to the simple adaptive filtering problem shown in Figure 2.4, the output error of estimating $y(i)$ from $x(i)$ with the filter coefficients equal to $w(k)$ is given by

$$e_k(i) = y(i) - \boldsymbol{x}^{t}(i)\,w(k) \tag{2.42}$$

The RLS algorithm finds the filter coefficients at iteration k such that the cumulative squared error measure

$$\xi(k) = \sum_{i=0}^{k} \gamma^{k-i}\,e_k^2(i) \tag{2.43}$$

is minimized, where $x(k)$ is assumed to be zero when $k < 0$. The parameter γ with $0 < \gamma \leq 1$ is a data-weighing factor used to emphasize recent data more heavily in the least-square computation. It is usually set to unity in stationary environment while a smaller value is usually employed under nonstationary environment. Let the vectors

$$\boldsymbol{e}(k) = [\,e_k(0) \quad e_k(1) \quad \ldots \quad e_k(k)\,]^{t} \quad , \tag{2.44}$$

$$y(k) = [\, y(0) \quad y(1) \quad \ldots \quad y(k) \,]^t \tag{2.45}$$

and the matrix

$$X(k) = [\, x(0) \quad x(1) \quad \ldots \quad x(k) \,] \quad , \tag{2.46}$$

then the error vector $e(k)$ can be written as

$$e(k) = y(k) - X^t(k)\, w(k) \tag{2.47}$$

and the error criterion shown in (2.43) becomes

$$\xi(k) = e^t(k)\, \Gamma(k)\, e(k) \tag{2.48}$$

where the matrix $\Gamma(k)$ is given by

$$\Gamma(k) = diag\, \{\, \gamma^k \quad \gamma^{k-1} \quad \ldots \quad 1 \,\} \tag{2.49}$$

Denote

$$R(k) = X(k)\, \Gamma(k)\, X^t(k) = \sum_{i=0}^{k} \gamma^{k-i}\, x(i)\, x^t(i) \tag{2.50}$$

and

$$g(k) = X(k)\, \Gamma(k)\, y(k) = \sum_{i=0}^{k} \gamma^{k-i}\, y(i)\, x(i) \quad , \tag{2.51}$$

equation (2.48) can be expressed as

$$\xi(k) = y^t(k)\, \Gamma(k)\, y(k) - 2\, w^t(k)\, g(k) + w^t(k)\, R(k)\, w(k) \tag{2.52}$$

The optimum parameter vector that minimizes $\xi(k)$ is then given by

$$w(k) = R^{-1}(k)\, g(k) \tag{2.53}$$

This least-square solution is computationally intensive, as matrix inversion is involved for each adaptation. The RLS algorithm avoids this inversion by allowing the coefficient vector to be updated from its previous value when a new data point is usable.

Suppose that

$$w(k-1)=R^{-1}(k-1)\,g(k-1) \tag{2.54}$$

is available and a new data point $y(k)$ and $x(k)$ is entered into the system. We want to determine $w(k)$ from $w(k-1)$. The autocorrelation matrix $R(k)$ and the correlation vector $g(k)$ can be expressed as

$$R(k)=\gamma R(k-1)+x(k)\,x^{t}(k) \tag{2.55}$$

and $$g(k)=\gamma g(k-1)+y(k)\,x(k) \tag{2.56}$$

Using the matrix inversion lemma [31], we have

$$R^{-1}(k)=\frac{1}{\gamma}\left\{R^{-1}(k-1)-\frac{R^{-1}(k-1)\,x(k)\,x^{t}(k)\,R^{-1}(k-1)}{\gamma+\alpha(k)}\right\} \tag{2.57}$$

with the scalar $\alpha(k)$ equal to

$$\alpha(k)=x^{t}(k)\,R^{-1}(k-1)\,x(k) \tag{2.58}$$

Denote $P(k)=R^{-1}(k)$ and

$$\kappa(k)=\frac{P(k-1)\,x(k)}{\gamma+\alpha(k)} \quad , \tag{2.59}$$

equation (2.57) can be rewritten as

$$P(k)=\gamma^{-1}\{P(k-1)-\kappa(k)\,x^{t}(k)\,P(k-1)\} \tag{2.60}$$

When we substitute (2.56) and (2.60) into (2.53), the weight vector that minimize $\xi(k)$ is equal to

$$\begin{aligned} w(k) &= P(k)\,g(k) \\ &= P(k-1)\,g(k-1)-\kappa(k)\,x^{t}(k)\,P(k-1)\,g(k-1) \\ &\quad +\gamma^{-1}\,y(k)\{P(k-1)\,x(k)-\kappa(k)\,\alpha(k)\} \\ &= w(k-1)+\kappa(k)\{y(k)-x^{t}(k)\,w(k-1)\} \end{aligned} \tag{2.61}$$

Initialization :

$w(0)=0$, $x(-1)=0$

Operation :

For k=1 to final do

(1) Acquire $y(k)$, $x(k)$

(2) Compute error :

$$e_k(k)=y(k)-\boldsymbol{x}^t(k)\,w(k-1)$$

(3) Calculate the gain vector $\kappa(k)$:

$$\alpha(k)=\boldsymbol{x}^t(k)\,\boldsymbol{P}(k-1)\,\boldsymbol{x}(k)$$

$$\kappa(k)=\frac{\boldsymbol{P}(k-1)\,\boldsymbol{x}(k)}{\gamma+\alpha(k)}$$

(4) Update weight vector :

$$w(k)=w(k-1)+\kappa(k)\,e_k(k)$$

(5) Update matrix for next iteration :

$$\boldsymbol{P}(k)=\gamma^{-1}\{\boldsymbol{P}(k-1)-\kappa(k)\,\boldsymbol{x}^t(k)\,\boldsymbol{P}(k-1)\}$$

Table 2.1 The RLS adaptation algorithm

Equation (2.61) is the desired result for the update recursion of the parameter vector $w(k)$. The complete set of equations necessary for the RLS algorithm are listed in Table 2.1 [32]. The essence of the RLS algorithm is to compute the gain vector $\kappa(k)$ used in updating $w(k-1)$ to $w(k)$ in a least-square fashion.

When comparing the RLS and the LMS algorithm, it is found that both methods incorporate the latest scalar error for adjustment but the vector portion of their correction terms is different. The LMS algorithm simply uses the input vector whereas the RLS method utilizes the gain vector $\kappa(k)$ for adaptation.

It is noted that the RLS algorithm has a faster convergence speed than the LMS algorithm as it computes the optimal vector for each iteration. However, the RLS algorithm requires $O((M+1)^2)$ operations per iteration. Although lately, computationally efficient versions of the RLS algorithm [15]-[18] are derived by exploiting the shifting property of the sample autocorrelation matrix, which can reduce the complexity to $7(M+1)$ to $11(M+1)$ operations. Nevertheless, they are still greater than that of the LMS algorithm at least by a factor of 3.5.

2.3 CHAPTER SUMMARY

An adaptive filter is composed of two basic elements, filter structure and adaptation algorithm. Filter structure of an adaptive filter refers to different realizations leading to adjustment of distinct parameter sets. Some adaptive filter structures are more promising than others by having faster adaptation rate but an increase in system complexity is necessary.

Associated with each filter structure, there is an adaptation algorithm which is used to alter the filter parameters in acquiring certain filtering goals. There are two well known adaptation algorithms in the context of adaptive signal processing. They are the RLS and the LMS adaptation algorithm. The RLS algorithm has limited

utilization due to its high computational complexity. On the other hand the LMS algorithm, which is a variant of the method of steepest descent, is relatively simple and easy to implement. Therefore it is widely used in diverse fields of applications nowadays. In the LMS algorithm, fast convergence rate will lead to a large steady state parameter fluctuation and a compromise between adaptation speed and steady state performance is essential. A deficiency of the LMS algorithm is that its adaptation speed is dependent on the signal statistics. When the autocorrelation matrix of an input signal has a large eigenvalue spread, an exceedingly long time is required for the system to converge. In subsequent chapters, new structures for adaptive filters will be developed so as to remove this drawback for parameters adaptation using LMS algorithm.

REFERENCES

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

[2] K. Teng and C. E. Rohrs, "The use of large adaptation gains to remove the SPR condition from recursive adaptive algorithms," in *Proc. IEEE ICASSP-87*, Dallas, Texas, Apr. 1987, pp. 129-132.

[3] Y. H. Tam, P. C. Ching and Y. T. Chan, "Adaptive recursive filter in cascade form," IEE Proc. Part F, vol. 134, pp. 245-252, June 1987.

[4] M. Nayeri and W. K. Jenkins, "Analysis of alternate realizations of adaptive IIR filters," in *Proc. IEEE ISCAS-88*, Finland, June 1988, pp. 2157-21601.

[5] H. Fan, "Application of Benveniste's convergence results in a study of adaptive IIR filtering algorithms," *IEEE Trans. Inform. Theory*, vol. 34, pp. 692-709, July 1988.

[6] J. J. Shynk, "Adaptive IIR filtering using parallel form realizations," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 519-533, Apr. 1989.

[7] S. Haykin, *Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[8] M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, Algorithms, and Applications*. Boston, MA: Kluwer Academic Publishers, 1984.

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

[10] L. J. Griffiths, "A continuously adaptive filter implemented as a lattice structure," in *Proc. IEEE ICASSP-77*, Hartford CT, pp. 683-686, 1977.

[11] B. Friedlander, "Lattice filters for adaptive processing," *Proc. IEEE*, vol. 70, pp. 829-867, Aug. 1982.

[12] J. Markoul, "A class of all zero lattice digital filters: properties and applications," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, pp. 304-314, Aug. 1978.

[13] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[14] E. H. Satorius and S. T. Alexander, "Channel equalization using adaptive lattice algorithms," *IEEE Trans. Commun.*, vol. COM-27, pp. 899-905, June 1979.

[15] D. D. Falconer and L. Ljung, "Application of fast Kalman estimation to adaptive equalization," *IEEE Trans. Commun.*, vol. COM-26, pp. 1439-1446, Oct. 1978.

[16] L. Ljung, M. Morf and D. Falconer, "Fast calculations of gain matrices for recursive estimation schemes," *Int. J. Control*, vol. 27, pp. 1-19, 1978.

[17] G. C. Carayannis, D. Manolakis and N. Kalouptsidis, "A fast sequential algorithm for least-squares filtering and prediction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1394-1402, Dec. 1983.

[18] J. Cioffi and T. Kailath, "Fast, recursive least-squares, transversal filters for adaptive processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 304-337, Apr. 1984.

[19] B. Widrow and M. E. Hoff, "Adaptive switching circuits," *IRE Wescon Conv. Rec.*, Part 4, pp. 96-104, 1960.

[20] B. Widrow, "Adaptive filters," in *Aspects of Network and System Theory*, N. de Claris and R. E. Kalman, eds., Holt, Rinehart, and Winston, New York, 1971.

[21] B. Widrow, *et al.*, "Adaptive antenna systems," *Proc. IEEE*, vol. 55, pp. 2143-2159, Dec. 1967.

[22] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong and R. C. Goodlin, "Adaptive noise cancelling: principles and applications," *Proc. IEEE*, vol. 63, pp. 1692-1716, Dec. 1975.

[23] B. Widrow, J. M. McCool, M. G. Larimore, C. R. Johnson, "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, pp. 1151-1162, Aug. 1976.

[24] R. A. Monzingo and T. W. Miller, *Introduction to Adaptive Array*. NY: Wiley-Interscience, 1980.

[25] J. Treichler, "Transient and convergent behavior of the ALE," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 53-63, Feb. 1979.

[26] B. Widrow and E. Walach, "On the statistical efficiency of the LMS algorithm with nonstationary inputs," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 211-221, March, 1984.

[27] S. T. Alexander and S. A. Rajala, "Image compression results using the LMS adaptive algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 712-715, June 1985.

[28] N. J. Bershad and L. Z. Qu, "LMS adaptation with correlated data - a scalar example," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 695-700, Aug. 1984.

[29] A. Feuer and E. Weinstein, "Convergence analysis of LMS adaptive filters with uncorrelated gaussian data," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 222-230, Feb. 1985.

[30] R. D. Gitlin and S. B. Weinstein, "On the required tap-weight precision for digitally implemented, adaptive, mean-squared equalizers," *Bell Syst. Tech. J.*, vol. 58, pp. 301-321, Feb. 1979.

[31] G. L. Bierman, *Factorization Methods for Discrete Sequential Estimation.* NY: Academic Press, 1977.

[32] S. T. Alexander, *Adaptive Signal Processing: Theory and Applications.* NY: Springer-Verlag, 1986.

3 PARALLEL SPLIT ADAPTIVE SYSTEM

In chapter 2, we find that the performance of an LMS adaptive filter may be affected by its implementation structure as well as the associated adaptation algorithm. Although the transversal ladder form is the simplest realization scheme, it suffers from having a fairly slow convergence rate. The lattice filter, on the other hand, has a much faster rate of adaptation but at the expense of a higher computational load [1]. For most practical applications, we need to design an adaptive filter that can adapt much faster than a transversal filter in order to cope with the nonstationary signals and environments. In addition, we also wish to keep the computational complexity as low as possible so as to make hardware implementation feasible. In this chapter, we propose a novel adaptive structure by splitting an adaptive transversal filter into two subunits connected in parallel. Each subunit comprises an adaptive filter of its own and has either a symmetric or an antisymmetric property. The performance of this new adaptive system is analyzed both theoretically and by computation simulations. It can be shown that the proposed adaptive filter can perform significantly better than a simple transversal filter and there is only a modest increase in computations.

3.1 PARALLEL FORM ADAPTIVE FILTER

Let the transfer function of a simple transversal adaptive filter $W(z)$ be

$$W(z)=\sum_{i=0}^{M} w_i\, z^{-i} \tag{3.1}$$

where M is the filter order and w_i, $0 \le i \le M$, denotes the filter coefficients. $W(z)$ can be decomposed into the sum of two filters $P(z)$ and $Q(z)$. Express explicitly, we have

$$W(z) = \frac{P(z) + Q(z)}{2} \tag{3.2}$$

where $P(z)$ and $Q(z)$ are of the form

$$P(z) = \sum_{i=0}^{M_p} p_i z^{-i} \tag{3.3a}$$

and

$$Q(z) = \sum_{i=0}^{M_q} q_i z^{-i} \tag{3.3b}$$

respectively. Here, M_p and M_q represent the order and p_i and q_i are the coefficients of the transfer function $P(z)$ and $Q(z)$. To ensure validity of (3.2), the filter orders M_p and M_q must be chosen to satisfy the following condition,

$$L = \max(M_p, M_q) \ge M \tag{3.4}$$

Padding zeros into $P(z)$ or $Q(z)$, they can be rewritten as

$$P(z) = \sum_{i=0}^{L} p_i z^{-i} \tag{3.5a}$$

and

$$Q(z) = \sum_{i=0}^{L} q_i z^{-i} \tag{3.5b}$$

where $p_i = 0$ for $i = M_p + 1, \ldots, L$ when $M_p < L$; or $q_i = 0$ for $i = M_q + 1, \ldots, L$ when $M_q < L$. Comparing the coefficients of z^{-i} on both sides of (3.2), the original filter parameters, w_i, can be related to p_i and q_i by

$$\begin{cases} w_i = \frac{1}{2}(p_i + q_i), & i = 0, 1, \ldots, M \\ p_i = -q_i, & i = M+1, M+2, \ldots, L \end{cases} \tag{3.6}$$

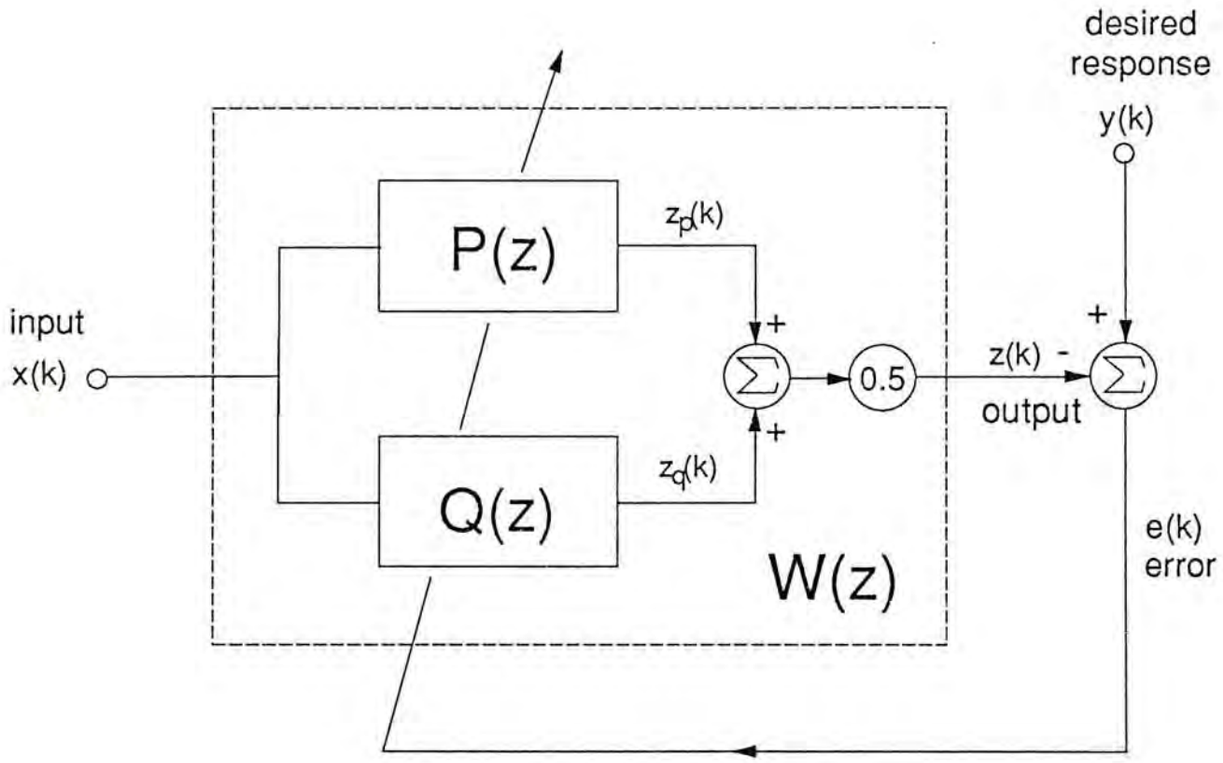


Fig. 3.1 Parallel split adaptive filter

Figure 3.1 depicts the schematic block diagram of an adaptive system with the filter $W(z)$ being split into two paths that are connected in parallel. The input signal $x(k)$ is passed through the two adaptive subunits to generate outputs $z_p(k)$ and $z_q(k)$, where

$$z_p(k) = \sum_{i=0}^{M} p_i x(k-i) \tag{3.7a}$$

and

$$z_q(k) = \sum_{i=0}^{M} q_i x(k-i) \tag{3.7b}$$

The output error can be obtained by

$$e(k) = y(k) - \frac{1}{2}\{ z_p(k) + z_q(k) \} \tag{3.8}$$

and the adaptive filter parameters p_i and q_i can be computed iteratively by minimizing the output mean-square error (MSE).

There are many ways of decomposing a filter into a sum of two units. In the context of adaptive signal processing, the major consideration in splitting $W(z)$ is based on the following two issues:

1. to keep the complexity due to decomposition to minimum
2. to maintain the same number of adapting parameters

One separation method that can fulfil the above requirement is to choose $P(z)$ and $Q(z)$ as transversal linear phase filters of order M. In this case, we set

$$p_i = -p_{M-i} \tag{3.9a}$$

and

$$q_i = q_{M-i} \tag{3.9b}$$

so that $P(z)$ becomes an antisymmetric linear phase filter and $Q(z)$ a symmetric linear phase filter. With the linear phase property of $P(z)$ and $Q(z)$, equation (3.6) can now be expressed as

$$\begin{cases} w_i = \dfrac{p_i + q_i}{2} \\ w_{M-i} = \dfrac{-p_i + q_i}{2} \end{cases}, \qquad i = 0, 1, \ldots, \left\lfloor \frac{M}{2} \right\rfloor \tag{3.10}$$

Alternatively, subtracting and adding w_i and w_{M-i} in turn will give

$$\begin{cases} p_i = w_i - w_{M-i} \\ q_i = w_i + w_{M-i} \end{cases}, \qquad i = 0, 1, \ldots, \left\lfloor \frac{M}{2} \right\rfloor \tag{3.11}$$

where the symbol $\lfloor \cdot \rfloor$ represents the integral part of $\cdot$. Notice that when M is even, we have $p_{\frac{M}{2}} = 0$ as shown in (3.9*a*).

In the following sections, we shall apply the above split-path adaptive filter model with the two subunits constrained to linear phase filters to the problem of joint process estimation and linear prediction. In either cases, it will be proved analytically that the split-path adaptive model can achieve a considerable improvement in convergence rate whilst the increase in computation is only very little.

3.2 JOINT PROCESS ESTIMATION WITH A SPLIT-PATH ADAPTIVE FILTER [2]

Adaptive LMS filter has been successfully applied in diverse fields of applications including plant modeling, noise cancelling and channel equilization [3]-[5]. In these applications of joint process estimation, the underlying signal processing problems can be formulated as the identification of an unknown system

which can be represented by a polynomial transfer function given by

$$H(z) = \sum_{i=0}^{M} h_i z^{-i} \tag{3.12}$$

where h_i are the system weights to be determined and M is the order. The unknown system is usually referred to as the plant and a typical configuration to perform system identification is shown in Figure 3.2. The input signal $x(k)$ is fed into the plant and its output is corrupted by an independent zero-mean random noise $n(k)$ of power σ_n^2 to form the measurement output $y(k)$. Our aim is to determine the unknown plant in an adaptive manner such that the expected output MSE $E[e^2(k)]$ is minimized, and $e(k)$ is the difference between $y(k)$ and the output of the plant model $W(z)$.

Let the plant model be given by (3.1), where w_i denotes the weights of the plant model. The plant model shares the same input with the unknown plant to form the output $z(k)$. Define

$$\boldsymbol{w} = [\, w_0 \quad w_1 \quad \ldots \quad w_M \,]^t \tag{3.13}$$

be the weight vector of $W(z)$,

$$\boldsymbol{h} = [\, h_0 \quad h_1 \quad \ldots \quad h_M \,]^t \tag{3.14}$$

be the parameter vector of the unknown plant $H(z)$ and

$$\boldsymbol{x}(k) = [\, x(k) \quad x(k-1) \quad \ldots \quad x(k-M) \,]^t \tag{3.15}$$

be the vector of the current and previous M input samples. The output error $e(k)$ can now be expressed as

$$e(k) = y(k) - \boldsymbol{x}^t(k)\,\boldsymbol{w} = n(k) - (\boldsymbol{w} - \boldsymbol{h})^t\,\boldsymbol{x}(k) \tag{3.16}$$

with the superscript t representing the transpose operation. Squaring (3.16) and taking statistical expectation, the error criterion to be minimized is given by

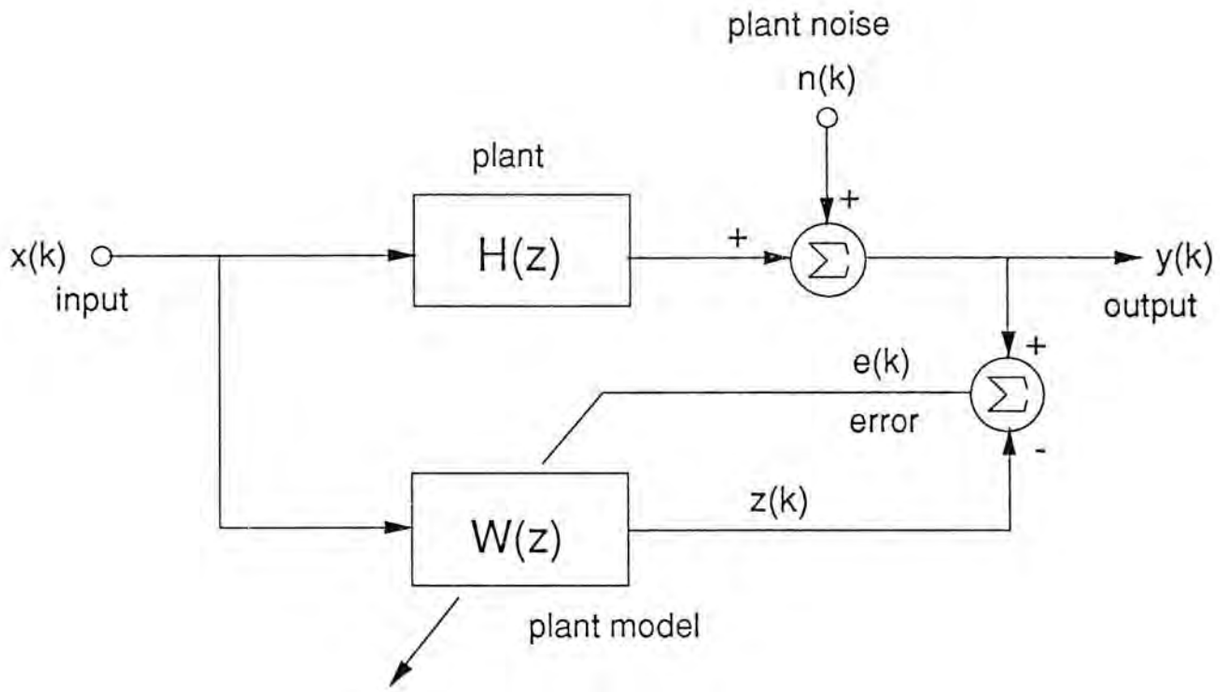


Fig. 3.2 Conventional adaptive configuration for system modeling

$$\xi(k) = E[\, e^2(k)] = \sigma_n^2 + (\, w - h)^t \, R \, (\, w - h) \tag{3.17}$$

where $R = E[\, x(k)\, x^t(k)]$ is the autocorrelation matrix of the input vector $x(k)$. To determine the optimal parameter vector w^* that minimizes ξ, we equate the derivative of ξ with respect to w to zero, which yields

$$\frac{\partial \xi}{\partial w} = 2 R \, (\, w^* - h) = 0 \tag{3.18}$$

As long as R is a positive definite matrix of full rank, the inverse of R exists. The solution to (3.18) is

$$w^* = h \tag{3.19}$$

showing the plant model will exactly identify the unknown plant as equilibrium is reached.

According to Widrow's LMS algorithm, the weight updating equations for w is given by

$$w(k+1) = w(k) - \mu_w \frac{\partial e^2(k)}{\partial w(k)} = w(k) + 2\, \mu_w \, e(k)\, x(k) \tag{3.20}$$

where μ_w is the step size that determines the algorithm rate of convergence and stability. From the analytical results of the LMS adaptation algorithm illustrated in chapter 2, the stability range for μ_w is

$$0 < \mu_w < \frac{1}{\lambda_{\max}} \tag{3.21}$$

where $\lambda_{\max}$ is the maximum eigenvalue of the input correlation matrix R. Let the eigenvalues of R be λ_i, then the relaxation time constants of the MSE are

$$\tau_{\xi,i} = \frac{1}{4\, \mu_w \, \lambda_i} \quad , \qquad i = 1, \ldots, M+1 \tag{3.22}$$

When steady state is reached, the variation of the weight vector $w(k)$ due to gradient noise is given by

$$cov\{w(k)\} = E[\{w(k)-w^*\}\{w(k)-w^*\}^t] = \mu_w \xi^o I_{M+1} \tag{3.23}$$

and the final MSE is equal to

$$\xi_{ss} = \xi^o + \mu_w \xi^o tr(R) \tag{3.24}$$

where $\xi^o = \sigma_n^2$ is the minimum possible MSE, I_{M+1} is an identity matrix of size $M+1$ and $tr(R)$ denotes the trace of the matrix R.

In most practical applications, the input $x(k)$ is not random. That means the eigenvalues of R are different. The convergence speed of $w(k)$ is then dependent on the eigenvalue spread (the ratio of the maximum to minimum eigenvalue) of the correlation matrix R. The larger the spread, the longer the algorithm will take to reach equilibrium. An intuitive way to improve adaptation speed is to reduce the eigenvalue spread of R. As will be shown later, when the plant model is split into two linear phase filters connected in parallel, it can come up with two eigenvalue spreads, one for each linear phase filter. They are unequal and both of them are usually smaller than the eigenvalue spread of R. Thus with appropriate step sizes for the two linear phase filters, the performance of the underlying adaptive system can be improved.

3.2.1 THE NEW ADAPTIVE SYSTEM IDENTIFICATION CONFIGURATION

The new filter structure for system identification using a split-path plant model is depicted in Figure 3.3. The plant model is now represented by two linear phase filters connected in parallel as shown in (3.2), in which $P(z)$ is an antisymmetric

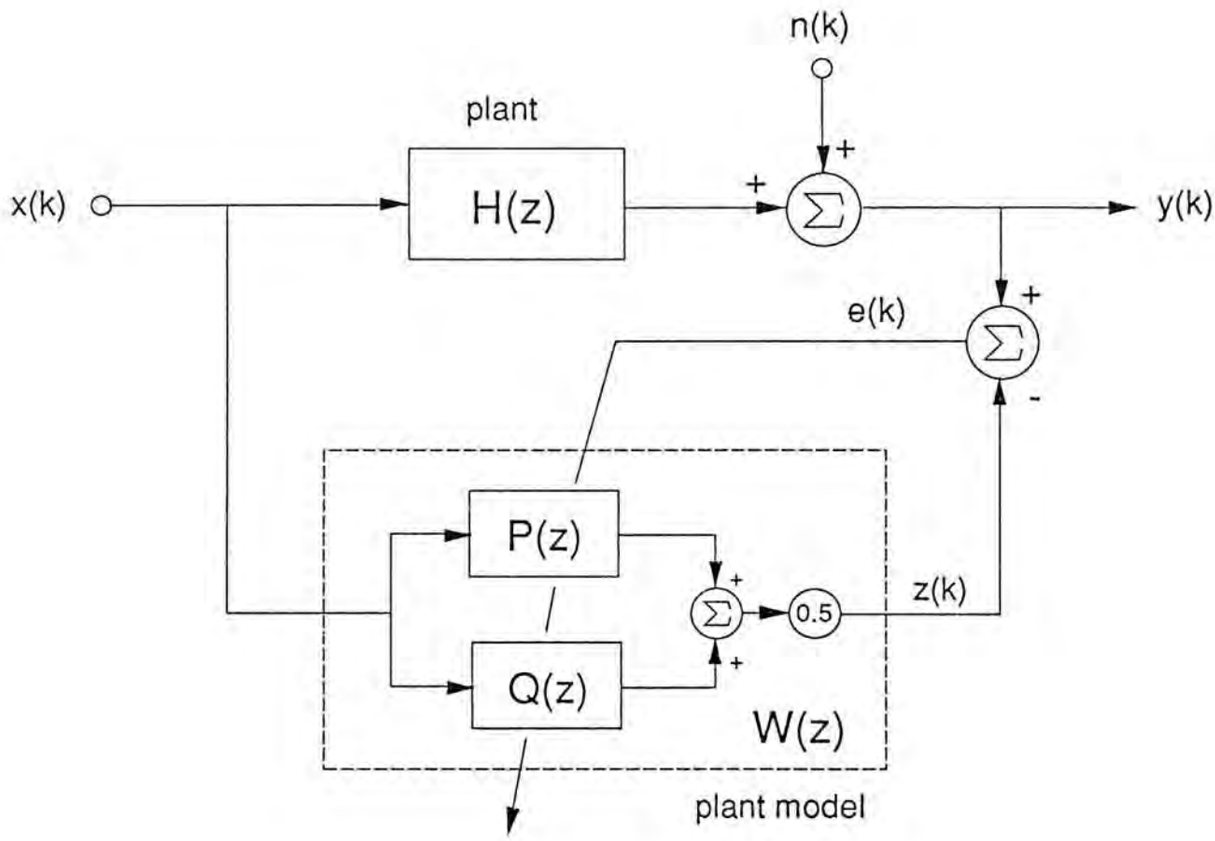


Fig. 3.3 The new split-path adaptive filter for system modeling

linear phase filter and $Q(z)$ is a symmetric linear phase filter. They are being updated simultaneously to optimize the system performance by minimizing the output MSE. For the sake of simplicity, we shall first consider the case for M being an odd number. Let $M = 2N-1$, $P(z)$ and $Q(z)$ can be expressed as

$$P(z) = \sum_{i=0}^{N-1} p_i \left(z^{-i} - z^{-2N+1+i} \right) \tag{3.25a}$$

and

$$Q(z) = \sum_{i=0}^{N-1} q_i \left(z^{-i} + z^{-2N+1+i} \right) \tag{3.25b}$$

Denote the respective parameter vectors of $P(z)$ and $Q(z)$ by

$$\boldsymbol{p} = [\, p_0 \quad p_1 \quad \cdots \quad p_{N-1} \,]^t \tag{3.26a}$$

and

$$\boldsymbol{q} = [\, q_0 \quad q_1 \quad \cdots \quad q_{N-1} \,]^t \tag{3.26b}$$

In matrix notation, it can be derived from (3.11) that the filter coefficients of $W(z)$, $P(z)$ and $Q(z)$ are related by

$$\begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{q} \end{bmatrix} = \Theta \, \boldsymbol{w} = \begin{bmatrix} \boldsymbol{I}_N & -\boldsymbol{J}_N \\ \boldsymbol{I}_N & \boldsymbol{J}_N \end{bmatrix} \boldsymbol{w} \tag{3.27}$$

where $\boldsymbol{J}_N$ is an off diagonal matrix of size N given by

$$\boldsymbol{J}_N = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \cdot & \cdot & \cdots & \cdot & \cdot \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix} = \boldsymbol{J}_N^{-1} \tag{3.28}$$

It is easy to verify that Θ is an orthogonal matrix, in particular,

$$\Theta^{-1} = \frac{1}{2} \Theta^t \tag{3.29}$$

Therefore, we can obtain $\boldsymbol{w}$ from $\boldsymbol{p}$ and $\boldsymbol{q}$ by

$$w = \Theta^{-1}\begin{bmatrix} p \\ q \end{bmatrix} = \frac{1}{2}\begin{bmatrix} I_N & I_N \\ -J_N & J_N \end{bmatrix}\begin{bmatrix} p \\ q \end{bmatrix} \tag{3.30}$$

which is equivalent to (3.10). Let $v_p(k)$ and $v_q(k)$ be two Nx1 column vectors and define $v(k)$ as

$$v(k) = [\, v_p^t(k) \quad v_q^t(k)]^t = \Theta\, x(k) \tag{3.31}$$

Using (3.29), (3.30) and (3.31), the output error shown in (3.16) can be expressed in terms of p and q by

$$e(k) = y(k) - \frac{1}{2} v^t(k)\begin{bmatrix} p \\ q \end{bmatrix} = y(k) - \frac{1}{2}\{\, v_p^t(k)\, p + v_q^t(k)\, q\,\} \tag{3.32}$$

As seen from (3.32), the MSE performance surface is quadratic with respect to the elements of p and q. There exists a single global minimum point $[\, p^* \quad q^*]^t$ which relates to w^* by (3.27).

Pre-multiplying both sides of (3.20) by Θ and substituting (3.27) and (3.31), we have

$$\begin{bmatrix} p(k+1) \\ q(k+1) \end{bmatrix} = \begin{bmatrix} p(k) \\ q(k) \end{bmatrix} + 2\,\mu_w\, e(k)\begin{bmatrix} v_p(k) \\ v_q(k) \end{bmatrix} \tag{3.33}$$

When different step sizes for $p(k)$ and $q(k)$ are employed, their update equations become

$$p(k+1) = p(k) + 2\,\mu_p\, e(k)\, v_p(k) \tag{3.34a}$$

and

$$q(k+1) = q(k) + 2\,\mu_q\, e(k)\, v_q(k) \tag{3.34b}$$

3.2.2 ANALYSIS OF THE SPLIT-PATH SYSTEM MODELING STRUCTURE

Let $r(t)=E[x(k)x(k-t)]$ be the correlation function of $x(k)$. The input correlation matrix $\boldsymbol{R}$ can be partitioned into the form

$$\boldsymbol{R}=\begin{bmatrix} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \boldsymbol{R}_{12}^{t} & \boldsymbol{R}_{11} \end{bmatrix} \tag{3.35}$$

where

$$\boldsymbol{R}_{11}=\begin{bmatrix} r(0) & r(1) & \dots & r(N-1) \\ r(1) & r(0) & \dots & r(N-2) \\ \cdot & \cdot & \dots & \cdot \\ r(N-1) & r(N-2) & \dots & r(0) \end{bmatrix} \tag{3.36}$$

and

$$\boldsymbol{R}_{12}=\begin{bmatrix} r(N) & r(N+1) & \dots & r(2N-1) \\ r(N-1) & r(N) & \dots & r(2N-2) \\ \cdot & \cdot & \dots & \cdot \\ r(1) & r(2) & \dots & r(N) \end{bmatrix} \tag{3.37}$$

Using the Hermitian Toeplitz property of $\boldsymbol{R}$, it can be verified that

$$\boldsymbol{R}_{11}=\boldsymbol{J}_N\boldsymbol{R}_{11}\boldsymbol{J}_N \tag{3.38}$$

and

$$\boldsymbol{R}_{12}\boldsymbol{J}_N=\boldsymbol{J}_N\boldsymbol{R}_{12}^{t} \tag{3.39}$$

Hence from (3.31),

$$\begin{aligned} E[\,v(k)\,v^{t}(k)] &= \begin{bmatrix} E[\,v_p(k)\,v_p^{t}(k)] & E[\,v_p(k)\,v_q^{t}(k)] \\ E[\,v_q(k)\,v_p^{t}(k)] & E[\,v_q(k)\,v_q^{t}(k)] \end{bmatrix} \\ &= \Theta\boldsymbol{R}\,\Theta^{t}=\begin{bmatrix} \boldsymbol{I}_N & -\boldsymbol{J}_N \\ \boldsymbol{I}_N & \boldsymbol{J}_N \end{bmatrix}\begin{bmatrix} \boldsymbol{R}_{11}-\boldsymbol{R}_{12}\boldsymbol{J}_N & \boldsymbol{R}_{11}+\boldsymbol{R}_{12}\boldsymbol{J}_N \\ \boldsymbol{R}_{12}^{t}-\boldsymbol{R}_{11}\boldsymbol{J}_N & \boldsymbol{R}_{12}^{t}+\boldsymbol{R}_{11}\boldsymbol{J}_N \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{R}_p & \boldsymbol{O}_N \\ \boldsymbol{O}_N & \boldsymbol{R}_q \end{bmatrix}=2\begin{bmatrix} \boldsymbol{R}_{11}-\boldsymbol{R}_{12}\boldsymbol{J}_N & \boldsymbol{O}_N \\ \boldsymbol{O}_N & \boldsymbol{R}_{11}+\boldsymbol{R}_{12}\boldsymbol{J}_N \end{bmatrix} \end{aligned} \tag{3.40}$$

where $\boldsymbol{R}_p = E[\, \boldsymbol{v}_p(k)\, \boldsymbol{v}_p^t(k)]$, $\boldsymbol{R}_q = E[\, \boldsymbol{v}_q(k)\, \boldsymbol{v}_q^t(k)]$ and $\boldsymbol{O}_N$ denotes a zero matrix of size N. Therefore, $\boldsymbol{v}_p(k)$ and $\boldsymbol{v}_q(k)$ are uncorrelated. Let the respective parameter error vectors be

$$\tilde{\boldsymbol{p}}(k) = \boldsymbol{p}(k) - \boldsymbol{p}^* \tag{3.41a}$$

and

$$\tilde{\boldsymbol{q}}(k) = \boldsymbol{q}(k) - \boldsymbol{q}^* \tag{3.41b}$$

Then the output error $e(k)$ can be expressed as

$$e(k) = n(k) - \frac{1}{2}\boldsymbol{v}^t(k)\begin{bmatrix}\tilde{\boldsymbol{p}}(k)\\ \tilde{\boldsymbol{q}}(k)\end{bmatrix} = n(k) - \frac{1}{2}\{\, \boldsymbol{v}_p^t(k)\,\tilde{\boldsymbol{p}}(k) + \boldsymbol{v}_q^t(k)\,\tilde{\boldsymbol{q}}(k)\} \tag{3.42}$$

If μ_p and μ_q are chosen sufficiently small, $v(k)$ will be independent of $\tilde{\boldsymbol{p}}(k)$ and $\tilde{\boldsymbol{q}}(k)$. Now, by putting (3.42) into (3.34a) and taking expectation, we obtain

$$\begin{aligned} E[\,\tilde{\boldsymbol{p}}(k+1)] &= E[\,\tilde{\boldsymbol{p}}(k)] - \mu_p \{\, E[\, \boldsymbol{v}_p(k)\, \boldsymbol{v}_p^t(k)]\, E[\,\tilde{\boldsymbol{p}}(k)] \\ &\quad + E[\, \boldsymbol{v}_p(k)\, \boldsymbol{v}_q^t(k)]\, E[\,\tilde{\boldsymbol{q}}(k)]\,\} \end{aligned} \tag{3.43}$$

Since $\boldsymbol{v}_p(k)$ and $\boldsymbol{v}_q(k)$ are orthogonal to each other, (3.43) becomes

$$E[\,\tilde{\boldsymbol{p}}(k+1)] = (\boldsymbol{I}_N - \mu_p \boldsymbol{R}_p)\, E[\,\tilde{\boldsymbol{p}}(k)] \tag{3.44a}$$

Similarly, it can be derived that

$$E[\,\tilde{\boldsymbol{q}}(k+1)] = (\boldsymbol{I}_N - \mu_q \boldsymbol{R}_q)\, E[\,\tilde{\boldsymbol{q}}(k)] \tag{3.44b}$$

It is easy to verify from (3.44) that the stability range of μ_p and μ_q are, respectively, given by

$$0 < \mu_p < \frac{2}{\lambda_{p,\max}} \tag{3.45a}$$

and

$$0 < \mu_q < \frac{2}{\lambda_{q,\max}} \tag{3.45b}$$

where $\lambda_{p,\max}$ and $\lambda_{q,\max}$ are the maximum eigenvalues of $\boldsymbol{R}_p$ and $\boldsymbol{R}_q$. As indicated in (3.29) and (3.40), we have

$$\begin{bmatrix} \boldsymbol{R}_p & \boldsymbol{O}_N \\ \boldsymbol{O}_N & \boldsymbol{R}_q \end{bmatrix} = 2\,\Theta \boldsymbol{R}\,\Theta^{-1} \tag{3.46}$$

Notice that the autocorrelation matrix $\boldsymbol{R}$ can be decomposed into

$$\boldsymbol{R} = \boldsymbol{U}^{-1}\,\Lambda\,\boldsymbol{U} \tag{3.47}$$

where $\boldsymbol{U}$ is an orthonormal matrix containing the eigenvectors of $\boldsymbol{R}$ as its columns and Λ is a diagonal matrix with its diagonal elements equal to the eigenvalues of $\boldsymbol{R}$. Subsequently, equation (3.46) can be rewritten as

$$\begin{bmatrix} \boldsymbol{R}_p & \boldsymbol{O}_N \\ \boldsymbol{O}_N & \boldsymbol{R}_q \end{bmatrix} = 2\,(\boldsymbol{U}\,\Theta^{-1})^{-1}\,\Lambda\,(\boldsymbol{U}\,\Theta^{-1}) \tag{3.48}$$

It is possible to partition the diagonal matrix Λ into

$$\Lambda = \begin{bmatrix} \Lambda_1 & \boldsymbol{O}_N \\ \boldsymbol{O}_N & \Lambda_2 \end{bmatrix} \tag{3.49}$$

so that Λ_1 and Λ_2 are matrices of the same size N. Denoting $\lambda_{p,i}$ and $\lambda_{q,i}$, $i = 1, 2, \ldots, N$, be the respective eigenvalues of $\boldsymbol{R}_p$ and $\boldsymbol{R}_q$, the characteristic expression of the matrix shown in (3.48) is

$$\begin{aligned} &\begin{bmatrix} \lambda_p \boldsymbol{I}_N & \boldsymbol{O}_N \\ \boldsymbol{O}_N & \lambda_q \boldsymbol{I}_N \end{bmatrix} - \begin{bmatrix} \boldsymbol{R}_p & \boldsymbol{O}_N \\ \boldsymbol{O}_N & \boldsymbol{R}_q \end{bmatrix} \\ &= 2\,(\boldsymbol{U}\,\Theta^{-1})^{-1} \begin{bmatrix} \frac{1}{2}\lambda_p \boldsymbol{I}_N - \Lambda_1 & \boldsymbol{O}_N \\ \boldsymbol{O}_N & \frac{1}{2}\lambda_q \boldsymbol{I}_N - \Lambda_2 \end{bmatrix} (\boldsymbol{U}\,\Theta^{-1}) \end{aligned} \tag{3.50}$$

Thus, the eigenvalues of $\boldsymbol{R}_p$ and $\boldsymbol{R}_q$ are obtained by solving

$$det\left\{\begin{bmatrix} \lambda_p I_N & O_N \\ O_N & \lambda_q I_N \end{bmatrix} - \begin{bmatrix} R_p & O_N \\ O_N & R_q \end{bmatrix}\right\} = 0 \tag{3.51}$$

with the symbol $det\{\cdot\}$ denoting the determinant of the matrix $\cdot$. However,

$$det\left\{\begin{bmatrix} \lambda_p I_N & O_N \\ O_N & \lambda_q I_N \end{bmatrix} - \begin{bmatrix} R_p & O_N \\ O_N & R_q \end{bmatrix}\right\} = 2\, det\left\{\begin{bmatrix} \frac{1}{2}\lambda_p I_N - \Lambda_1 & O_N \\ O_N & \frac{1}{2}\lambda_q I_N - \Lambda_2 \end{bmatrix}\right\}$$

$$= 2 \prod_{i=1}^{N}\left(\frac{1}{2}\lambda_p - \lambda_i\right)\prod_{i=1}^{N}\left(\frac{1}{2}\lambda_q - \lambda_{N+i}\right) \tag{3.52}$$

Hence the eigenvalues of $\boldsymbol{R}_p$ and those of $\boldsymbol{R}_q$ form a partition of the eigenvalues of $\boldsymbol{R}$ with a scaling factor of 2. That is,

$$\lambda_{p,i} = 2\lambda_i \quad , \qquad i = 1, 2, \ldots, N \tag{3.53a}$$

and

$$\lambda_{q,i} = 2\lambda_{N+i} \quad , \qquad i = 1, 2, \ldots, M+1-N \tag{3.53b}$$

Without loss of generality, let us assume that the largest eigenvalue of $\boldsymbol{R}$ is λ_{2N}. $\lambda_{q,\max} = 2\lambda_{2N}$ is then greater than $\lambda_{p,\max}$. According to (3.45), the stability range for μ_q is equal to that of μ_w in the conventional system but μ_p, now, can enjoy a larger stability range. Using (3.44) and (3.53), it can be derived that the relaxation time constant for the parameter vectors p and q are respectively given by

$$\tau_{p,i} = \frac{1}{2\mu_p\lambda_i} \quad , \qquad i = 1, 2, \ldots, N \tag{3.54a}$$

and

$$\tau_{q,i} = \frac{1}{2\mu_q\lambda_{N+i}} \quad , \qquad i = 1, 2, \ldots, M+1-N \tag{3.54b}$$

From (3.42) and using the uncorrelated property of $v_p(k)$ and $v_q(k)$, the output MSE at iteration k can be expressed as

$$\xi(k) = \xi^o + \frac{1}{4} E[\,\tilde{p}^t(k)\, \boldsymbol{R}_p\, \tilde{p}(k)\,] + \frac{1}{4} E[\,\tilde{q}^t(k)\, \boldsymbol{R}_q\, \tilde{q}(k)\,] \tag{3.55}$$

If we ignore weight vector fluctuation during the learning period, (3.55) can be approximated by

$$\xi(k) = \xi^o + \frac{1}{4} E[\,\tilde{p}(k)\,]^t\, \boldsymbol{R}_p\, E[\,\tilde{p}(k)\,] + \frac{1}{4} E[\,\tilde{q}(k)\,]^t\, \boldsymbol{R}_q\, E[\,\tilde{q}(k)\,] \tag{3.56}$$

When (3.44) and (3.54) are taken into consideration, the time constant for the ith mode of the MSE is found to be

$$\begin{cases} \tau_{\xi,i} = \dfrac{1}{4\,\mu_p\,\lambda_i} \quad , & i = 1, \ldots, N \\ \tau_{\xi,i} = \dfrac{1}{4\,\mu_q\,\lambda_i} \quad , & i = N+1, \ldots, M+1 \end{cases} \tag{3.57}$$

To complete the analysis, we now evaluate the steady state performance of the split-path adaptive model. The parameter update equation (3.34) can be expressed as [3]

$$\begin{aligned} \tilde{p}(k+1) &= \tilde{p}(k) - \mu_p\, E[\,2\, e(k)\, v_p(k)\,|\,p(k)\,] + \mu_p\, \eta_p(k) \\ &= (\,\boldsymbol{I}_N - \mu_p\, \boldsymbol{R}_p\,)\, \tilde{p}(k) + \mu_p\, \eta_p(k) \end{aligned} \tag{3.58a}$$

and

$$\begin{aligned} \tilde{q}(k+1) &= \tilde{q}(k) - \mu_q\, E[\,2\, e(k)\, v_q(k)\,|\,q(k)\,] + \mu_q\, \eta_q(k) \\ &= (\,\boldsymbol{I}_N - \mu_q\, \boldsymbol{R}_q\,)\, \tilde{q}(k) + \mu_q\, \eta_q(k) \end{aligned} \tag{3.58b}$$

where $\eta_p(k)$ and $\eta_q(k)$ are the zero mean gradient noise vectors associated with the adjustment of $\boldsymbol{p}(k)$ and $\boldsymbol{q}(k)$. The steady state correlation matrices of the two gradient noise vectors are given by

$$cov\{\,\eta_p(k)\,\} = 4\, E[\,e^2(k)\, \boldsymbol{v}_p(k)\, \boldsymbol{v}_p^t(k)\,] = 4\, \xi^o\, \boldsymbol{R}_p \tag{3.59a}$$

and

$$cov\{\,\eta_q(k)\,\} = 4\, E[\,e^2(k)\, \boldsymbol{v}_q(k)\, \boldsymbol{v}_q^t(k)\,] = 4\, \xi^o\, \boldsymbol{R}_q \tag{3.59b}$$

Post-multiplying both sides of (3.58*a*) by their transpose and assuming $v_p(k)$ is uncorrelated over time, we have

$$cov\{\tilde{p}(k+1)\} = (I_N - \mu_p R_p)\, cov\{\tilde{p}(k)\}\, (I_N - \mu_p R_p) + \mu_p^2\, cov\{\eta_p(k)\} \tag{3.60}$$

On account to the fact that $cov\{\tilde{p}(k+1)\} = cov\{\tilde{p}(k)\}$ when equilibrium is attained, (3.60) can be simplified to

$$R_p\, cov\{\tilde{p}(k)\} + cov\{\tilde{p}(k)\}\, R_p = 4\mu_p\, \xi^o\, R_p \tag{3.61}$$

Hence

$$cov\{p(k)\} = 2\mu_p\, \xi^o\, I_N \tag{3.62a}$$

Similarly, the weight vector variation of $q(k)$ is found to be

$$cov\{q(k)\} = 2\mu_q\, \xi^o\, I_N \tag{3.62b}$$

Post-multiplying $\tilde{p}(k+1)$ by the transpose of $\tilde{q}(k+1)$ and then taking expectation, we obtain

$$E[\tilde{p}(k+1)\,\tilde{q}^t(k+1)] = (I_N - \mu_p R_p)\, E[\tilde{p}(k)\,\tilde{q}^t(k)]\, (I_N - \mu_q R_q) + E[\eta_p(k)\,\eta_q^t(k)] \tag{3.63}$$

Remember that $v_p(k)$ and $v_q(k)$ are orthogonal, we have

$$E[\eta_p(k)\,\eta_q^t(k)] = 4\,E[e^2(k)]\,E[v_p(k)\,v_q^t(k)] = 0 \tag{3.64}$$

Therefore (3.63) becomes

$$\mu_p R_p\, E[\tilde{p}(k)\,\tilde{q}^t(k)] + \mu_q\, E[\tilde{p}(k)\,\tilde{q}^t(k)]\, R_q = 0 \tag{3.65}$$

Notice that the step sizes μ_p and μ_q are both greater than zero and the matrices R_p and R_q are positive definite, (3.65) holds if and only if

$$E[\tilde{p}(k)\tilde{q}^{t}(k)]=0 \tag{3.66}$$

which means that the steady state parameter error vectors $\tilde{p}(k)$ and $\tilde{q}(k)$ are independent to each other.

Finally, from (3.55), the steady state MSE in the split-path adaptive system can be computed easily and is given by

$$\begin{aligned}\xi(k) &= \xi^{o}+\frac{1}{4}tr(\boldsymbol{R}_{p}\,cov\{p(k)\})+\frac{1}{4}tr(\boldsymbol{R}_{q}\,cov\{q(k)\}) \\ &= \xi^{o}\left\{1+\frac{\mu_{p}}{2}tr(\boldsymbol{R}_{p})+\frac{\mu_{q}}{2}tr(\boldsymbol{R}_{q})\right\}\end{aligned} \tag{3.67}$$

3.2.3 COMPARISON WITH THE NON-SPLIT CONFIGURATION

From the above discussion, it is noted that splitting $W(z)$ into two linear phase filters connected in parallel has the effect of partitioning the eigenvalues of the input correlation matrix $\boldsymbol{R}$ into two groups and there are two eigenvalue spreads instead of one. Denote the eigenvalue spread of $\boldsymbol{R}$, $\boldsymbol{R}_{p}$ and $\boldsymbol{R}_{q}$ be $\chi(\boldsymbol{R}_{p})$, $\chi(\boldsymbol{R}_{p})$ and $\chi(\boldsymbol{R}_{q})$ respectively. Again, let $\lambda_{\max}=\lambda_{2N}$ such that $\lambda_{q,\max}>\lambda_{p,\max}$. In accordance with (3.45), the step size μ_{q} is set to μ_{w} and μ_{p} is chosen to be greater than μ_{w}. Depending on which groups the minimum eigenvalue $\lambda_{\min}$ will fall into, there are two possible cases :

(1) $2\lambda_{\min}$ is an eigenvalue of $\boldsymbol{R}_{p}$. Then $\chi(\boldsymbol{R}_{p})$ and $\chi(\boldsymbol{R}_{q})$ are both less than $\chi(\boldsymbol{R})$. The vector $q(k)$ will converge faster than $w(k)$ in the conventional tapped delay line plant model because the largest time constant of $q(k)$ is smaller than that of $w(k)$. In addition, the vector $p(k)$ also has a faster adaptation speed because it has a smaller time constant with μ_{p} larger than μ_{w}.

(2) $2\lambda_{min}$ is an eigenvalue of R_q. Then $\chi(R_q)$ is equal to $\chi(R)$ but $\chi(R_p)$ is smaller. The convergence rate of $q(k)$ and $w(k)$ is expected to be identical. On the other hand, $p(k)$ is capable of achieving a rapid convergence because the largest time constant of $p(k)$ is smaller than that of $w(k)$ even with $\mu_p = \mu_w$. In this case the dynamic convergence speed is improved, before the smallest eigenvalue of R_q dominates the convergence rate.

Consequently, we can conclude that the split-path plant model has a superior performance because of its smaller eigenvalue spreads [2].

In certain applications, the steady state performance of an adaptive system should be well under control. For instance, the variance of the plant model parameters has to be kept below some specified values. This implies that the step sizes μ_p and μ_q cannot be chosen freely based upon (3.45) because a larger μ_p will give rise to a greater variation in $p(k)$. In this situation, we make use of the mean-square estimation error or mean-square difference (MSD) of the plant model parameters which is defined as

$$\text{MSD}(w(k)) = E[\tilde{w}^t(k)\,\tilde{w}(k)] = E\left[\sum_{i=0}^{M}(w_i(k) - w_i^*)^2\right] \tag{3.68}$$

to be a performance index to contrast the convergence behavior of two different systems. The MSD has identical time constant with the MSE. Using (3.29) and (3.30), the MSD of the plant model parameters w in the split-path system can be related to the MSD of the parameter vectors p and q by

$$\begin{aligned}\text{MSD}(w(k)) &= E[\tilde{w}^t(k)\,\tilde{w}(k)] = E\left[\,[\tilde{p}^t(k) \quad \tilde{q}^t(k)]\,\Theta^{-t}\,\Theta^{-1}\begin{bmatrix}\tilde{p}(k)\\ \tilde{q}(k)\end{bmatrix}\right] \\ &= \frac{1}{2}\{E[\tilde{p}^t(k)\,\tilde{p}(k)] + E[\tilde{q}^t(k)\,\tilde{q}(k)]\} \\ &= \frac{1}{2}\{\text{MSD}(p(k)) + \text{MSD}(q(k))\}\end{aligned} \tag{3.69}$$

When equilibrium is attained, on using (3.62), the steady state MSD of the plant model parameters in the split-path system is equal to

$$\begin{aligned} \mathrm{MSD}_{ss}(w(k)) &= \frac{1}{2}\{ tr(cov\{\tilde{p}(k)\tilde{p}^{t}(k)\}) + tr(cov\{\tilde{q}(k)\tilde{q}^{t}(k)\}) \} \\ &= N(\mu_p + \mu_q)\xi^{o} \end{aligned} \tag{3.70}$$

Whereas the steady state MSE for the traditional transversal configuration can be evaluated from (3.23) and is given by

$$\mathrm{MSD}_{ss}(w(k)) = 2N\mu_w\xi^{o} \tag{3.71}$$

If the same final MSD is maintained for the two structures, we have

$$\mu_p + \mu_q = 2\mu_w \tag{3.72}$$

When $\mu = \mu_q = \mu_w$, it can be seen from (3.22) and (3.57) that the adaptation speed of the two systems is essentially identical. However, with the constraint (3.72) on the step sizes, one can choose a larger step size for $P(z)$ while decrease the step size for $Q(z)$ to obtain a faster adaptation in case (1) as described above. Notice that μ_q must not be lessened to such an extent that the largest time constant of $Q(z)$ is greater than that of $P(z)$. In case (2), μ_q can be chosen to be larger than μ_p for a better performance as long as (3.45*b*) is fulfilled. In both cases, the split-path model can achieve a better performance. In all the simulation experiments we have performed, we found that λ_{max} and λ_{min} always belong to different groups. This reveals that it is highly unlikely that both λ_{max} and λ_{min} come from the same correlation matrix. Thus a larger step size can be selected for the linear phase filter with a smaller set of eigenvalues to acquire a better performance. Now, the question remained is how to select μ_p and μ_q such that the best performance can be assured.

For a matrix K, $tr(K)$ is equal to the sum of eigenvalues of K. If K is an autocorrelation matrix, then it is positive semidefinite and its eigenvalues would be all non-negative. It is, therefore, reasonable to assume that a small value of $tr(K)$ itself would indicate that K has small eigenvalues. Hence, an intuitive way of choosing the step size μ for an adaptive process with autocorrelation matrix K is to equate it to $C/tr(K)$, where C is a constant less than or equal to unity. The convergence of the adaptive process is still guaranteed because $C/tr(K) < 1/\lambda_{max}$, where λ_{max} is the largest eigenvalue of K, is again satisfied. In this way, a larger μ is essentially used if the autocorrelation matrix has small eigenvalues and thereby increasing the convergence speed. The selection scheme for the step sizes for the split-path model can be summarized as follow,

$$\mu_p = \frac{C}{tr(R_p)} \tag{3.73a}$$

and

$$\mu_q = \frac{C}{tr(R_q)} \tag{3.73b}$$

As condition (3.72) must be satisfied in order to maintain an identical steady state performance of the two systems, solving (3.72) and (3.73) yields

$$C = \frac{tr(R_p)\,tr(R_q)}{tr(R_p)+tr(R_q)} 2\mu_w \tag{3.74}$$

Hence

$$\mu_p = \frac{tr(R_q)}{tr(R_p)+tr(R_q)} 2\mu_w \tag{3.75a}$$

and

$$\mu_q = \frac{tr(R_p)}{tr(R_p)+tr(R_q)} 2\mu_w \tag{3.75b}$$

Equation (3.75) provides a simple way of selecting the step sizes in the new adaptive configuration to achieve a better performance. But the calculation of $tr(\boldsymbol{R}_p)$ and $tr(\boldsymbol{R}_q)$ requires a lot of computations since evaluation of the autocorrelation function of the input sequence is required. To obtain a compromise between adaptation performance and computational load, we shall determine whether $tr(\boldsymbol{R}_p)$ or $tr(\boldsymbol{R}_q)$ is greater than the other. A larger step size can be assigned to that particular filter whose corresponding autocorrelation matrix has a smaller trace. In most applications, deciding which one of $tr(\boldsymbol{R}_p)$ or $tr(\boldsymbol{R}_q)$ is larger can be made with a little a priori knowledge about the input signal statistics. For instance, if the input signal $x(k)$ is relatively rich in low frequency content like speech signal, it can be verified that $tr(\boldsymbol{R}_p)$ is always less than $tr(\boldsymbol{R}_q)$ and a larger μ_p can always be selected to obtain a faster adaptation.

Figure 3.4 shows a practical implementation of the split-path system modeling configuration. It is observed that no extra hardware is actually required. In regard to computational complexity, as indicated in (3.31) and (3.34), only N extra addition and subtraction operations are needed to calculate the tap input vectors $\boldsymbol{v}_p(k)$ and $\boldsymbol{v}_q(k)$.

3.2.4 SOME NOTES ON EVEN FILTER ORDER CASE

When $M = 2N$ is even, the transfer function of $P(z)$ and $Q(z)$ is chosen to be

$$P(z) = \sum_{i=0}^{N-1} p_i \left(z^{-i} - z^{-2N+i} \right) \tag{3.76a}$$

and

$$Q(z) = \sum_{i=0}^{N-1} q_i \left(z^{-i} + z^{-2N+i} \right) + \sqrt{2}\, q_N z^{-N} \tag{3.76b}$$

The vectors $\boldsymbol{q}(k)$ and $\boldsymbol{v}_q(k)$ now has N+1 components. The matrix Θ for conversion of filter parameters is

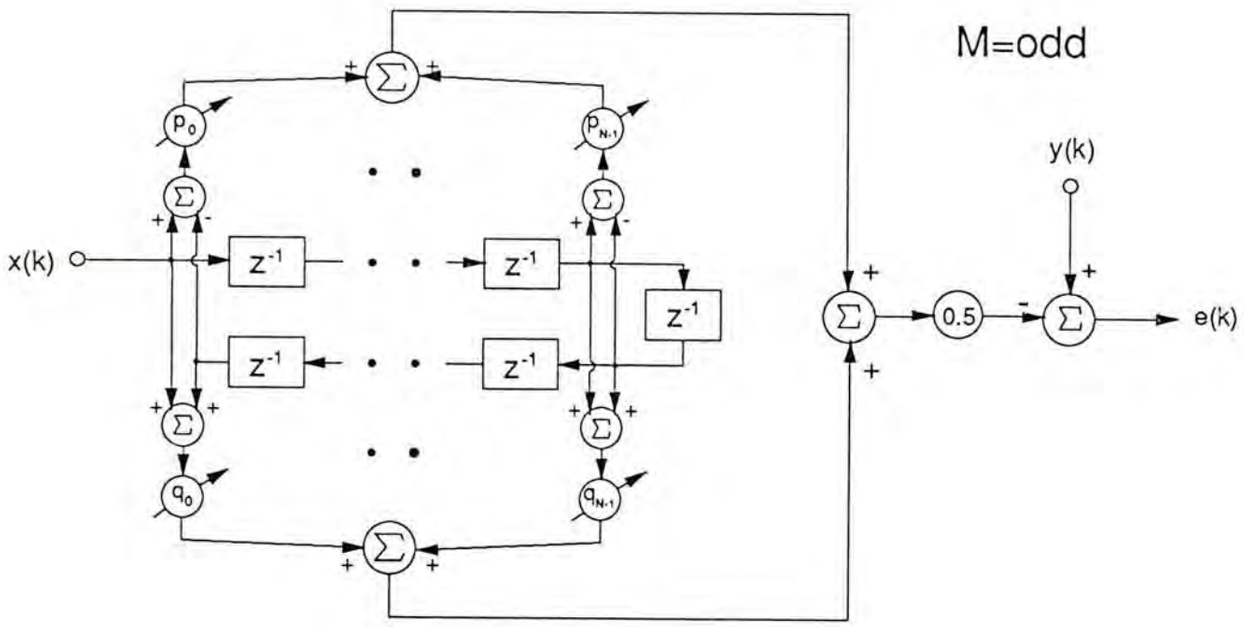


Fig. 3.4 Hardware implementation of a split-path adaptive model for system modeling

$$\Theta = \begin{bmatrix} I_N & 0_N & -J_N \\ I_N & 0_N & J_N \\ 0_N^t & \sqrt{2} & 0_N^t \end{bmatrix} \tag{3.77}$$

with 0_N representing a Nx1 column vector of zero elements. It can be verify that the orthogonal property (3.29) can be satisfied for Θ given by (3.77). Partition the autocorrelation matrix into the form

$$R = \begin{bmatrix} R_{11} & R_{13} & R_{12} \\ R_{13}^t & r(0) & R_{13}^t J_N \\ R_{12}^t & J_N R_{13} & R_{11} \end{bmatrix} \tag{3.78}$$

where

$$R_{11} = \begin{bmatrix} r(0) & r(1) & \dots & r(N-1) \\ r(1) & r(0) & \dots & r(N-2) \\ \cdot & \cdot & \cdot & \cdot \\ r(N-1) & r(N-2) & \dots & r(0) \end{bmatrix} \tag{3.79}$$

$$R_{12} = \begin{bmatrix} r(N+1) & r(N+2) & \dots & r(2N) \\ r(N) & r(N+1) & \dots & r(2N-1) \\ \cdot & \cdot & \cdot & \cdot \\ r(2) & r(3) & \dots & r(N+1) \end{bmatrix} \tag{3.80}$$

and $$R_{13} = [\, r(N) \quad r(N-1) \quad \cdot \quad \cdot \quad \cdot \quad r(1) \,]^t \tag{3.81}$$

It is easy to show that the expressions (3.38) and (3.39) are still valid. The autocorrelation matrix of $v(k)$ is found to be

$$E[\, v(k)\, v^t(k)] = \begin{bmatrix} E[\, v_p(k)\, v_p^t(k)] & E[\, v_p(k)\, v_q^t(k)] \\ E[\, v_q(k)\, v_p^t(k)] & E[\, v_q(k)\, v_q^t(k)] \end{bmatrix} = \Theta R \Theta^t$$

$$= \begin{bmatrix} R_p & 0_N\, 0_{N+1}^t \\ 0_{N+1}\, 0_N^t & R_q \end{bmatrix} = 2 \begin{bmatrix} R_{11} - R_{12} J_N & O_N & 0_N \\ O_N & R_{11} + R_{12} J_N & \sqrt{2} R_{13} \\ 0_N^t & \sqrt{2} R_{13}^t & r(0) \end{bmatrix} \tag{3.82}$$

showing that $v_p(k)$ and $v_q(k)$ are uncorrelated to each other. It can be proved that the eigenvalues of $\boldsymbol{R}_p$ and $\boldsymbol{R}_q$ again form a partition of the eigenvalues of $\boldsymbol{R}$ with a multiplication factor of 2. Consequently, the results derived in previous section for the case when M is odd can still be applied when M is even. When the steady state MSD of the plant model parameters of the split and non-split system is fixed, μ_p and μ_q can be chosen according to

$$N\mu_p + (N+1)\mu_q = (2N+1)\mu_w \tag{3.83}$$

Figure 3.5 shows the implementation of the split-path adaptive system for even M. In can be observed that one additional multiplication in this case is necessary.

3.2.5 SIMULATION RESULTS

Extensive simulations have been taken to study the behavior of the new adaptive system. The signal $x(k)$ was generated by an AR(4) process given by

$$\begin{aligned} x(k) = {} & 0.8915\,x(k-1) - 0.09684\,x(k-2) + 0.4345\,x(k-3) \\ & - 0.5184\,x(k-4) + \beta(k) \end{aligned} \tag{3.84}$$

where $\beta(k)$ was a Gaussian distributed random signal of unity variance. The pole locations of the signal $x(k)$ are $0.9\angle\pm 20°$ and $0.8\angle\pm 120°$. Numerical evaluations of the AR process reveals that the eigenvalues of $\boldsymbol{R}$ are $\{18.23, 6.46, 0.81, 0.77, 0.70, 0.22\}$ and that of $\boldsymbol{R}_p$ and $\boldsymbol{R}_q$ are $\{12.92, 1.54, 0.44\}$ and $\{36.5, 1.62, 1.4\}$. The eigenvalue spreads of $\boldsymbol{R}$, $\boldsymbol{R}_p$ and $\boldsymbol{R}_q$ are 81.7 , 29.3 and 23.75 respectively. The system to be identified was

$$H(z) = 0.1 + 0.2z^{-1} + 0.62z^{-2} - 0.22z^{-3} + 0.5z^{-4} + 0.4z^{-5} \tag{3.85}$$

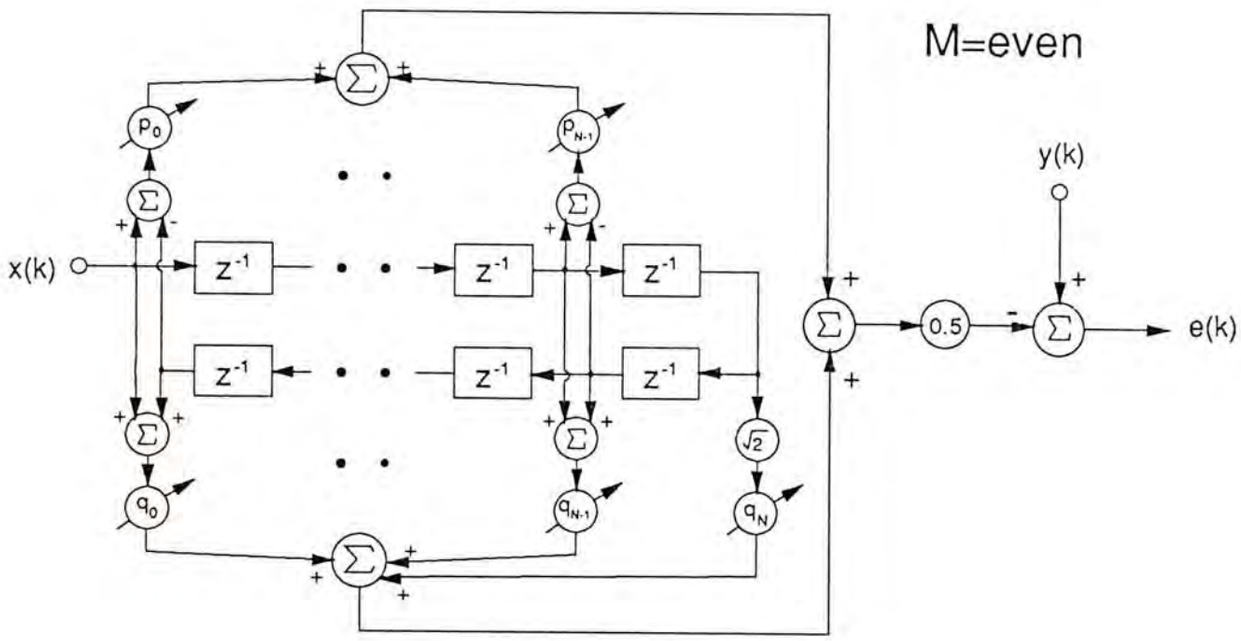


Fig. 3.5 Hardware implementation of a split-path plant model for system identification

The corrupting noise $n(k)$ was also Gaussian distributed but with a variance of 0.1. The control parameter μ_w was set to 0.002 and the results obtained were the average of 100 independent runs.

Figure 3.6 shows the MSD for the two split filters $P(z)$ and $Q(z)$ when their step sizes are both set to μ_w. It is seen that the MSD for $Q(z)$ converges much faster. This is validated by (3.54) since $Q(z)$ has a larger set of eigenvalues compared to $P(z)$. The resulting MSD for the plant model parameters which is obtained from (3.69) is depicted in Figure 3.7. The one for the non-split plant model is also provided for comparison. It is recognized that the trajectories of the MSD for both systems are essentially identical as expected because the two structures essentially have the same time constant when $\mu_p = \mu_q = \mu_w$.

We now investigate the effect on convergence speed by selecting different μ_p and μ_q. Figure 3.8 demonstrates the adaptation behavior of $P(z)$ and $Q(z)$ when μ_p and μ_q are selected according to (3.75). It can be observed that the two curves have similar slope during adaptation, which implies that their adaptation speed are roughly identical. Because of the decrease of μ_q and increase of μ_p, the steady state MSD for $q(k)$ is less than that for $p(k)$. The corresponding MSD for the plant model parameters in the split-path structure and the non-split model are compared in Figure 3.9. Although the steady state MSD for $p(k)$ and $q(k)$ are different, the MSD for the plant model parameters in the split model is approximately the same as that in the non-split system, which thus verifies (3.72). Due to the synchronization of convergence between the parameters $p(k)$ and $q(k)$, it is clear that the split configuration outperforms the non-split model. In fact, the new system achieved a mean-square parameter error of -28dB at about 1900 iterations whereas in the old system, 3800 iterations were required instead.

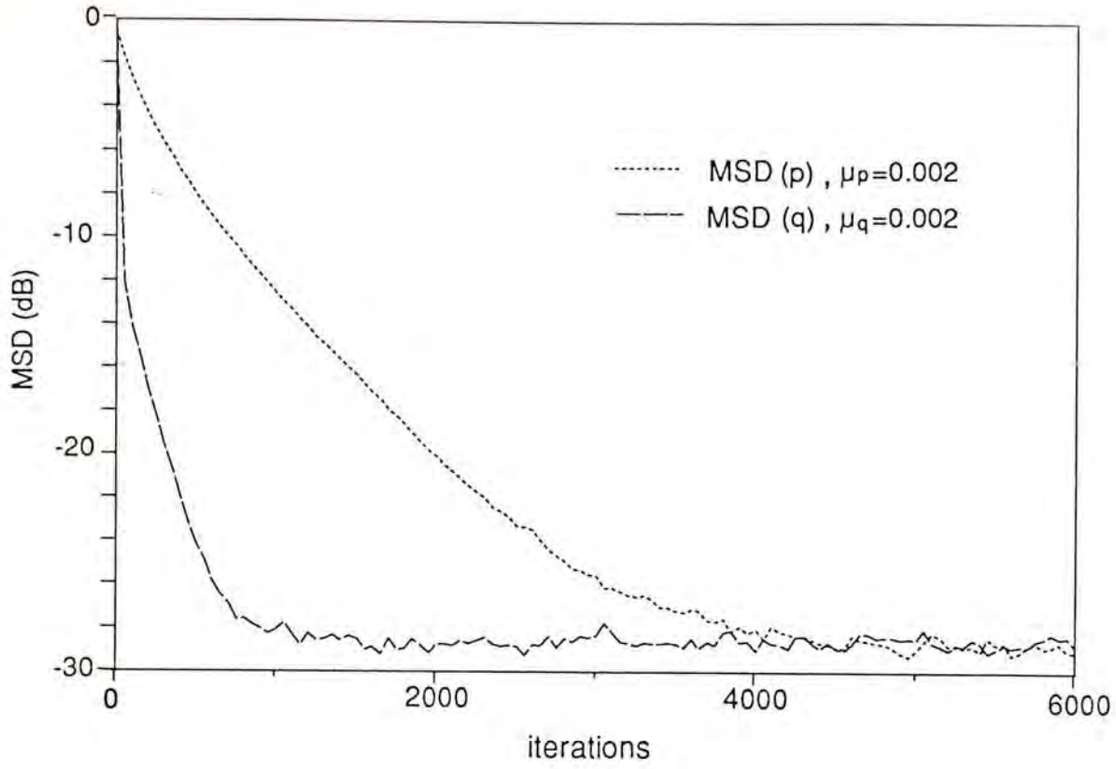


Fig. 3.6 Comparison of MSD for the two split filters with $\mu_p = \mu_q$

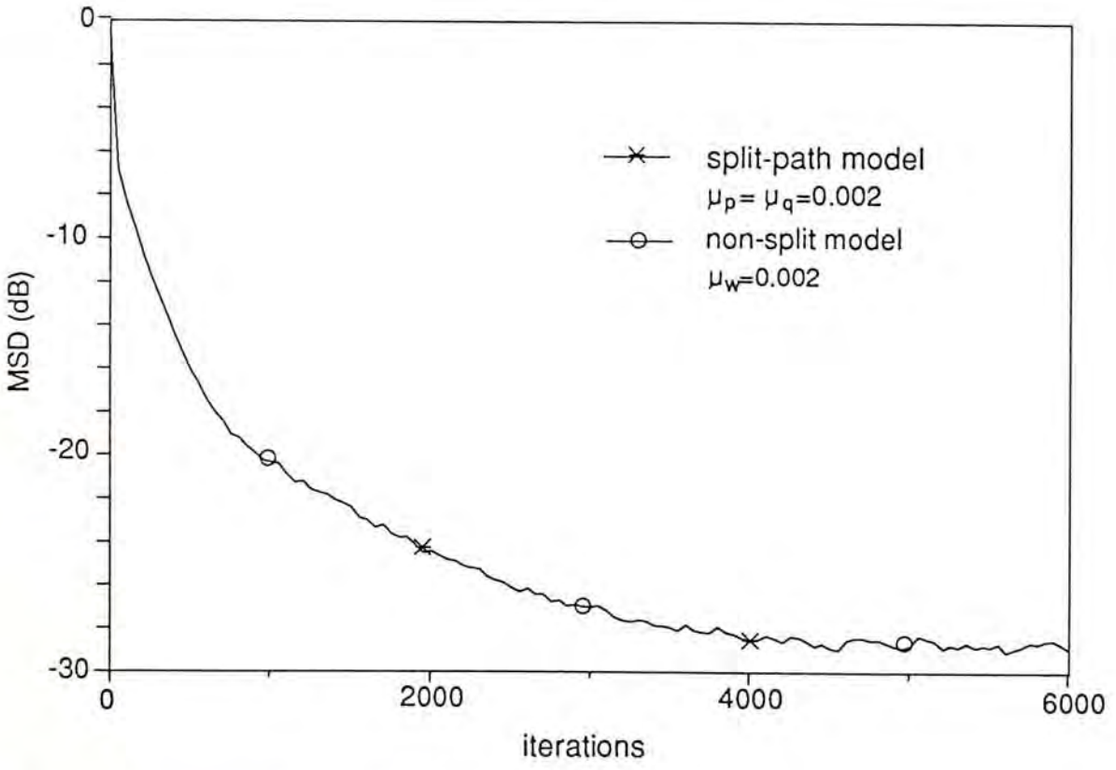


Fig. 3.7 Comparison of MSD for the plant model parameters with $\mu_p = \mu_q$

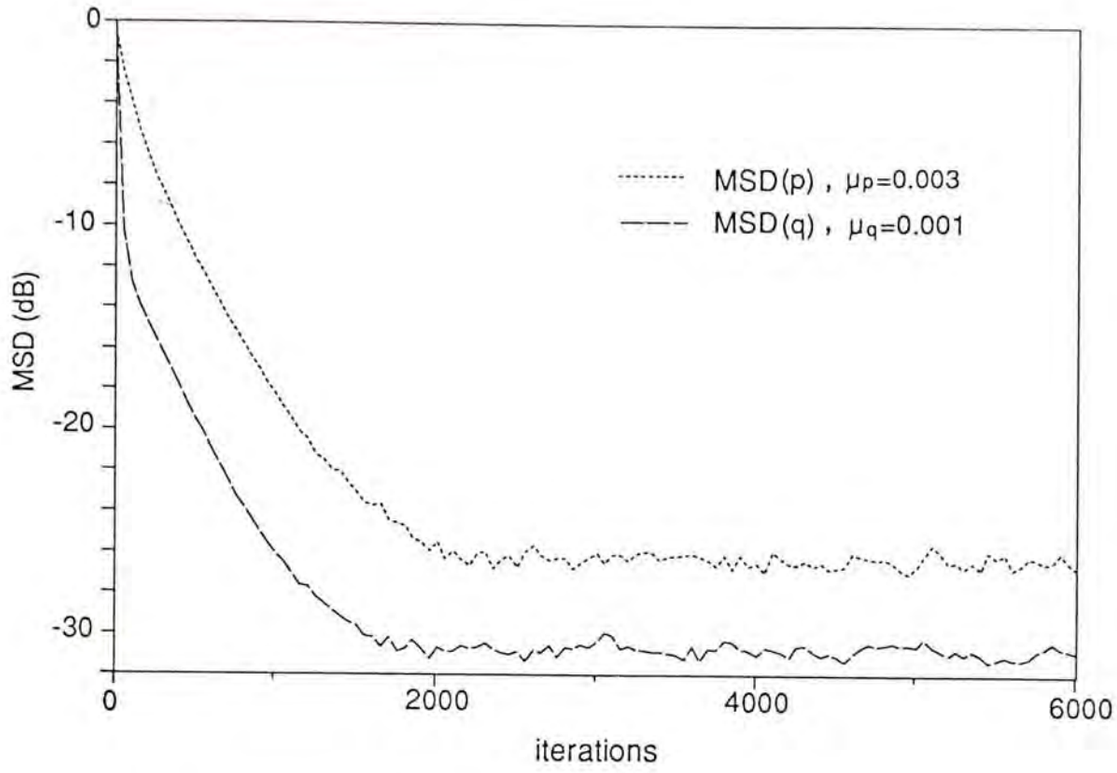


Fig. 3.8 Comparison of MSD for the two split filters with $\mu_p \neq \mu_q$

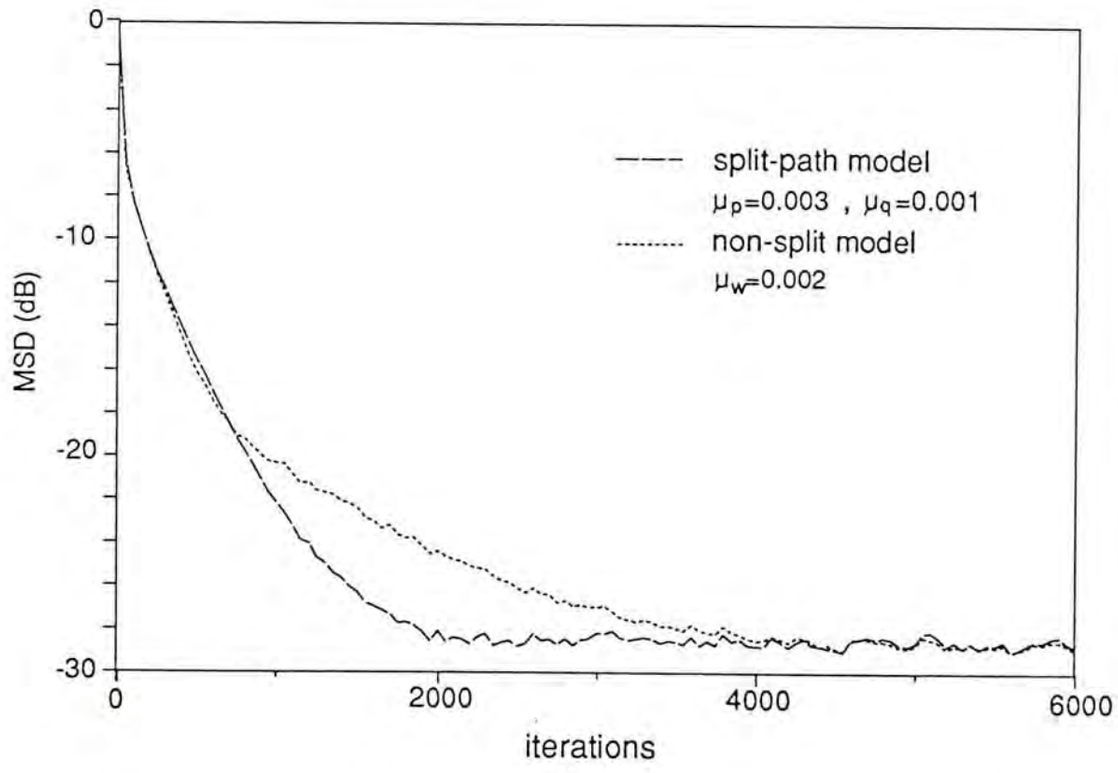


Fig. 3.9 Comparison of MSD for plant model parameters with $\mu_p \neq \mu_q$

The parallel split model applied to joint process estimation which is formulated as system modeling with non-white input is investigated. It is found that the effect of the split operation is to partition the eigenvalues of the input correlation matrix into two sets, which in turn gives rise to a reduction in eigenvalue spread. When suitable step sizes are chosen for the two split-paths, a faster adaptation speed can be attained.

3.3 AUTOREGRESSIVE MODELING WITH A SPLIT-PATH ADAPTIVE FILTER [6]

The autoregressive (AR) model has been extensively studied in a variety of fields such as statistics, econometrics, geophysics, and engineering. Recently, it has received much interest in the areas of spectral estimation [7]-[11] and speech processing [12]-[14]. A typical Mth order AR process can be expressed as

$$x(k)=\sum_{i=1}^{M}-g_i\,x(k-i)+\beta(k) \tag{3.86}$$

where $\beta(k)$ is a zero mean random process of variance σ_β^2, $x(k)$ is the generated signal driven by $\beta(k)$ and g_i are the system variables usually referred as the AR parameters. The problem is to determine the model parameters from the output sequence $x(k)$.

Many efficient methods have been developed for estimating the AR parameters. Most of them are based on a whitening filter approach with a conventional system configuration as shown in Figure 3.10. The sequence $x(k)$ passes through an all-zero filter, $W(z)$, which has the effect of flattening the spectrum of the output signal. The filter coefficient w_0 is fixed to unity while the other coefficients, w_i^*, $i=1,\ldots,M$, are obtained by solving the following highly structured Yule-Walker equation [12], which are identical to the AR parameters,

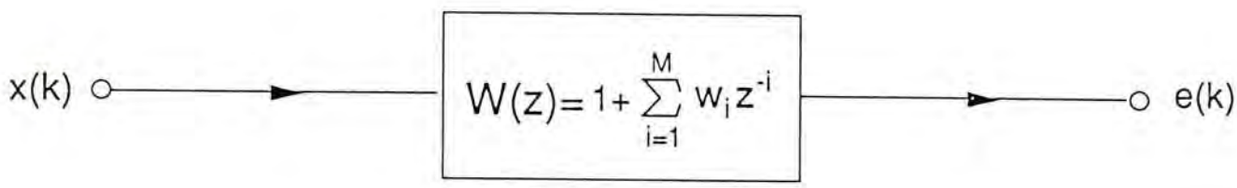


Fig. 3.10 A whitening filter model

$$\sum_{i=1}^{M} w_i^* r(j-i) = -r(j) \quad , \qquad j = 1, \ldots, M \tag{3.87}$$

where $r(j-i) = E[x(k-i)x(k-j)]$ is the autocorrelation function of the sequence $x(k)$. It is convenient to express (3.87) into matrix form. Let

$$w = [\, w_1 \quad w_2 \quad \ldots \quad w_M \,]^t \tag{3.88}$$

be the parameter vector of $W(z)$,

$$x_M(k-1) = [\, x(k-1) \quad x(k-2) \quad \ldots \quad x(k-M) \,]^t \tag{3.89}$$

be a vector of M pass input samples and

$$R_M = E[\, x_M(k)\, x_M^t(k) \,] \tag{3.90}$$

be the autocorrelation matrix of size M. Equation (3.87) can now be expressed as

$$R_M\, w^* = -g \tag{3.91}$$

where w^* is the optimal weight vector and

$$g = E[\, x(k)\, x_M(k-1) \,] = [\, r(1) \quad r(2) \quad \ldots \quad r(M) \,]^t \tag{3.92}$$

is the correlation vector between $x(k)$ and the vector $x_M(k-1)$. The optimal solution of the weight vector w^* can be evaluated from

$$w^* = -R_M^{-1}\, g \tag{3.93}$$

Solving (3.93) directly involves massive computations as matrix inversion is required. However, since R_M is a Toeplitz matrix, there are many efficient algorithms available that can solve (3.91) recursively without matrix manipulation such as Levinson's algorithm [15] and Durbin's algorithm [16].

Adaptive filtering is another technique which is widely used nowaday to extract AR coefficients. In this approach, the whitening filter $W(z)$ is made adaptive by

sequentially adjusting its weights to minimize a predefined cost function. Because the coefficients are determined iteratively based on the input and output signal of the system, one major advantage of this method is that any nonstationary characteristics of the input signal can be easily coped with.

Widrow's LMS algorithm is, again, commonly used to minimize the mean-square value of the filter output so as to flatten its frequency spectrum by means of a "noisy steepest descent" procedure. In this case, the weight vector $w(k)$ of the adaptive filter is determined, on a sample by sample basis, according to

$$e(k) = x(k) + w^t(k-1)\, x_M(k-1) \tag{3.94a}$$

$$w(k) = w(k-1) - 2\,\mu_w\, e(k)\, x_M(k-1) \tag{3.94b}$$

where μ_w is a predefined step size that controls the rate of convergence and stability of the adaptive process. It has been shown that when μ_w is chosen properly, the adaptive process will converge to the Wiener solution given by (3.93). The convergence behavior of this adaptive system can be characterized by the time constant of $\xi = E[\, e^2(k)]$, which is given by

$$\tau_{\xi,i} = \frac{1}{4\,\mu_w\,\lambda_i} \quad , \qquad i = 1, \ldots, M \tag{3.95}$$

where λ_i are the eigenvalues of $\boldsymbol{R}_M$. In steady state, the variation of the weight vector and the final MSE, ξ_{ss}, are respectively given by

$$cov\{\, w(k)\, w^t(k)\,\} = E[\,\{\, w(k) - w^*\,\}\,\{\, w(k) - w^*\,\}^t\,] = \mu_w\, \xi^o\, \boldsymbol{I}_M \tag{3.96}$$

and

$$\xi_{ss} = \xi^o\,\{\, 1 + \mu_w\, tr(\boldsymbol{R}_M)\,\} \tag{3.97}$$

where $\xi^o = \sigma_\beta^2$ is the minimum possible MSE.

In this section, application of the split-path adaptive technique in AR modeling will be investigated. It is found that when $W(z)$ is being represented by the split-path model, the output of the two linear phase filters are independent to each other. By adapting the two subunits separately using their respective outputs as error functions, the convergence behavior can be significantly improved at least by a factor of two. This is not only due to the decrease in eigenvalue spread of the input correlation matrix as in the case for system identification, but also due to the incorporation of backward prediction in the adaptation algorithm that help to reduce the gradient noise.

3.3.1 THE SPLIT-PATH ADAPTIVE FILTER FOR AR MODELING

The schematic block diagram of a split-path adaptive filter for AR modeling is shown in Figure 3.11. The original whitening filter $W(z)$ is now divided into two filters $P(z)$ and $Q(z)$, and they are related by (3.2). The input signal $x(k)$ is being fed to both filters and the whitening output $e(k)$ is given by

$$e(k)=\frac{e_p(k)+e_q(k)}{2} \tag{3.98}$$

where $e_p(k)$ and $e_q(k)$ are the outputs of $P(z)$ and $Q(z)$ respectively. The two filters are both of FIR type of order $(M+1)$ and have linear phase characteristics. As before, we put $P(z)$ to have antisymmetric property whilst $Q(z)$ have symmetric property. Let us first consider the case $M=2N$ for ease of exposition. The transfer function of the two subunits are of the usual form

$$P(z)=\sum_{i=0}^{N} p_i\,(z^{-i}-z^{-2N-1+i}) \tag{3.99a}$$

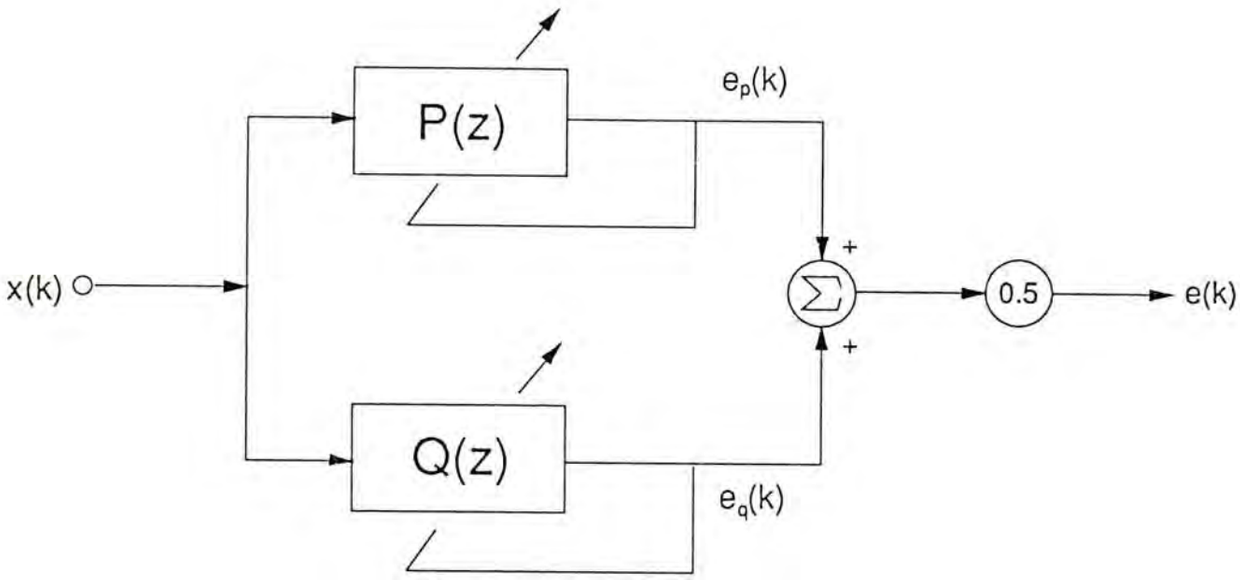


Fig. 3.11 A split-path adaptive whitening filter model

and $$Q(z) = \sum_{i=0}^{N} q_i \left(z^{-i} + z^{-2N-1+i} \right) \tag{3.99b}$$

In our study, both p_0 and q_0 are set to unity without loss of generality. The realization form of the split-path structure is shown in Figure 3.12. Define the vectors

$$p = [\, p_1 \quad \cdots \quad p_N \,]^t \tag{3.100a}$$

and $$q = [\, q_1 \quad \cdots \quad q_N \,]^t \tag{3.100b}$$

as the parameter vectors for $P(z)$ and $Q(z)$ respectively. Putting (3.99) into (3.2), we find that the filter coefficients between the new and old configuration are related by

$$\begin{bmatrix} 1 \\ w \\ 0 \end{bmatrix} = \frac{1}{2} \left\{ \begin{bmatrix} 1 \\ p \\ -J_N p \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ q \\ J_N q \\ 1 \end{bmatrix} \right\} \tag{3.101}$$

or $$\begin{bmatrix} p \\ q \end{bmatrix} = \Theta w = \begin{bmatrix} I_N & -J_N \\ I_N & J_N \end{bmatrix} w \tag{3.102}$$

In matrix notation, $e_p(k)$ and $e_q(k)$ can be expressed as

$$e_p(k) = [\, 1 \quad p^t \,] \, [\, I_{N+1} \quad -J_{N+1} \,] \, x_{M+2}(k) \tag{3.103a}$$

and $$e_q(k) = [\, 1 \quad q^t \,] \, [\, I_{N+1} \quad J_{N+1} \,] \, x_{M+2}(k) \tag{3.103b}$$

The filter parameters p and q are obtained by minimizing the total MSE, $E[\, e^2(k) \,]$ as usual. However, $e_p(k)$ and $e_q(k)$ are totally uncorrelated, since

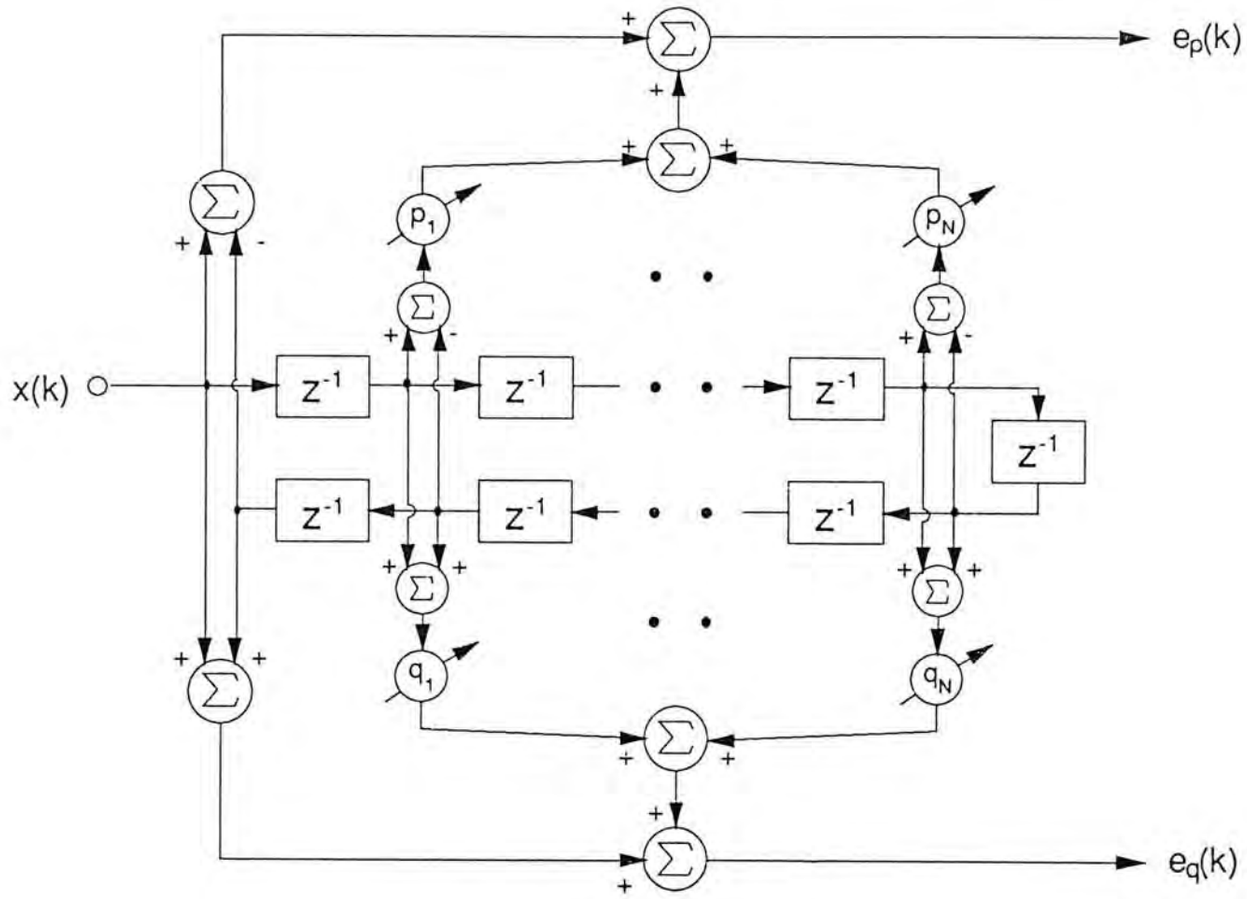


Fig. 3.12 Hardware implementation of the split-path whitening filter for even M

$$E[\,e_p(k)\,e_q(k)] = [\,1 \quad p^t\,]\,[\,I_{N+1} \quad -J_{N+1}\,]\,R_{M+2}\begin{bmatrix} I_{N+1} \\ J_{N+1} \end{bmatrix}\begin{bmatrix} 1 \\ q \end{bmatrix}$$

$$= [\,1 \quad p^t\,]\,[\,I_{N+1} \quad -J_{N+1}\,]\begin{bmatrix} O_{N+1} & J_{N+1} \\ J_{N+1} & O_{N+1} \end{bmatrix}$$

$$R_{M+2}\begin{bmatrix} O_{N+1} & J_{N+1} \\ J_{N+1} & O_{N+1} \end{bmatrix}\begin{bmatrix} I_{N+1} \\ J_{N+1} \end{bmatrix}\begin{bmatrix} 1 \\ q \end{bmatrix}$$

$$= [\,1 \quad p^t\,]\,[\,-I_{N+1} \quad J_{N+1}\,]\,R_{M+2}\begin{bmatrix} I_{N+1} \\ J_{N+1} \end{bmatrix}\begin{bmatrix} 1 \\ q \end{bmatrix}$$

$$= -E[\,e_p(k)\,e_q(k)] = 0 \tag{3.104}$$

Hence, from (3.98), the error criterion becomes

$$\xi(k) = E[\,e^2(k)] = \frac{1}{4}E[\,e_p^2(k)] + \frac{1}{4}E[\,e_q^2(k)] \tag{3.105}$$

In other words, minimizing the expectation of $e^2(k)$ is just the same as minimizing the expectation of $e_p^2(k)$ and $e_q^2(k)$ separately. As a result, the original adaptive system can be decomposed into two decoupled subsystems. As $E[\,e_p^2(k)]$ is a quadratic function with respect to p_i while $E[\,e_q^2(k)]$ is also a quadratic function with respect to q_i, there exists a unique global minimum which in turn will provide the optimum weight vectors p^* and q^*. From (3.102), we also have

$$\begin{bmatrix} p^* \\ q^* \end{bmatrix} = \begin{bmatrix} I_N & -J_N \\ I_N & J_N \end{bmatrix} w^* \tag{3.106}$$

Using the LMS adaptive algorithm, the parameter updating equations for the split-path system can be formulated as follow,

$$e_p(k) = x(k) - x(k-M-1) + p^t(k-1)\,v_p(k-1) \tag{3.107a}$$

$$p(k) = p(k-1) - 2\,\mu_p\,e_p(k)\,v_p(k-1) \tag{3.107b}$$

and $$e_q(k) = x(k) + x(k-M-1) + q^t(k-1)\, v_q(k-1) \tag{3.107c}$$

$$q(k) = q(k-1) - 2\,\mu_q\, e_q(k)\, v_q(k-1) \tag{3.107d}$$

where $$v_p(k-1) = [\, I_N \quad -J_N \,]\, x_M(k-1) \tag{3.108a}$$

and $$v_q(k-1) = [\, I_N \quad J_N \,]\, x_M(k-1) \tag{3.108b}$$

3.3.2 ANALYSIS OF THE SPLIT-PATH AR MODELING STRUCTURE

When the signal $x(k)$ is generated by an AR process of order M as in (3.86) with the AR parameters equal to w^*, we obtain

$$x(k) = \beta(k) - w^{*t}\, x_M(k-1) \tag{3.109a}$$

Along with this forward model, we can equivalently consider a backward model,

$$x(k-M-1) = \beta'(k-M-1) - w^{*t}\, J_M\, x_M(k-1) \tag{3.109b}$$

where $\beta'(k-M-1)$ is another zero-mean white noise sequence of variance σ_β^2. Note that $\beta(k)$ and $\beta'(k-M-1)$ are uncorrelated. By subtracting and adding (3.109*a*) and (3.109*b*) and then using (3.106) and (3.108), we have

$$\begin{aligned} x(k) - x(k-M-1) &= \beta(k) - \beta'(k-M-1) - w^{*t}\,(I_M - J_M)\, x_M(k-1) \\ &= \beta(k) - \beta'(k-M-1) - w^{*t} \begin{bmatrix} I_N & -J_N \\ -J_N & I_N \end{bmatrix} x_M(k-1) \\ &= \beta(k) - \beta'(k-M-1) - w^{*t} \begin{bmatrix} I_N \\ -J_N \end{bmatrix} v_p(k-1) \\ &= \beta(k) - \beta'(k-M-1) - p^{*t}\, v_p(k-1) \end{aligned} \tag{3.110a}$$

and

$$x(k)+x(k-M-1) = \beta(k)+\beta'(k-M-1)-w^{*t}(I_M+J_M)\,x_M(k-1)$$

$$= \beta(k)+\beta'(k-M-1)-w^{*t}\begin{bmatrix} I_N & J_N \\ J_N & I_N \end{bmatrix} x_M(k-1)$$

$$= \beta(k)+\beta'(k-M-1)-w^{*t}\begin{bmatrix} I_N \\ J_N \end{bmatrix} v_q(k-1)$$

$$= \beta(k)+\beta'(k-M-1)-q^{*t}\,v_q(k-1) \qquad (3.110b)$$

Substituting (3.110*a*) into (3.107*a*) and (2.110*b*) into (3.107*c*), $e_p(k)$ and $e_q(k)$ can be expressed as

$$e_p(k)=\tilde{p}^t(k-1)\,v_p(k-1)+\beta(k)-\beta'(k-M-1) \qquad (3.111a)$$

and

$$e_q(k)=\tilde{q}^t(k-1)\,v_q(k-1)+\beta(k)+\beta'(k-M-1) \qquad (3.111b)$$

where

$$\tilde{p}(k-1)=p(k-1)-p^* \qquad (3.112a)$$

and

$$\tilde{q}(k-1)=q(k-1)-q^* \qquad (3.112b)$$

are the weight vector errors. When (3.111) is put into (3.107*b*) and (3.107*d*), we obtain

$$\tilde{p}(k) = \{I_N-2\mu_p\,v_p(k-1)\,v_p^t(k-1)\}\,\tilde{p}(k-1)$$
$$-2\mu_p\,v_p(k-1)\{\beta(k)-\beta'(k-M-1)\} \qquad (3.113a)$$

and

$$\tilde{q}(k) = \{I_N-2\mu_q\,v_q(k-1)\,v_q^t(k-1)\}\,\tilde{q}(k-1)$$
$$-2\mu_q\,v_q(k-1)\{\beta(k)+\beta'(k-M-1)\} \qquad (3.113b)$$

Making use of the independence assumption that $\tilde{p}(k-1)$ and $v_p(k-1)$ are independent on one hand and $q(k-1)$ and $v_q(k-1)$ on the other hand, and noting that $x_M(k-1)$, $\beta(k)$ and $\beta'(k-M-1)$ are also independent to each other, taking statistical average of (3.113) yields

$$E[\tilde{p}(k)] = (I_N - 2\mu_p R_p)\, E[\tilde{p}(k-1)] \tag{3.114a}$$

and $$E[\tilde{q}(k)] = (I_N - 2\mu_q R_q)\, E[\tilde{q}(k-1)] \tag{3.114b}$$

where $$R_p = E[v_p(k-1)\, v_p^t(k-1)] \tag{3.115a}$$

and $$R_q = E[v_q(k-1)\, v_q^t(k-1)] \tag{3.115b}$$

Equations (3.114*a*) and (3.114*b*) give the adaptation trajectories for the parameter error vectors which also govern the convergence characteristics of the filter coefficients. It can be seen from (3.114) that the stability ranges for μ_p and μ_q are given by

$$0 < \mu_p < \frac{1}{\lambda_{p,\max}} \tag{3.116a}$$

and $$0 < \mu_q < \frac{1}{\lambda_{q,\max}} \tag{3.116b}$$

where $\lambda_{p,\max}$ and $\lambda_{q,\max}$ are, respectively, the maximum eigenvalues of R_p and R_q. Recognize that the symmetric Toeplitz matrix R_M is identical to the matrix R in the previous case for system identification, it can be obtained from (3.46) that

$$\begin{bmatrix} I_N & -J_N \\ I_N & J_N \end{bmatrix} R_M \begin{bmatrix} I_N & I_N \\ -J_N & J_N \end{bmatrix} = \begin{bmatrix} R_p & O_N \\ O_N & R_q \end{bmatrix} = 2\,\Theta\, R_M\, \Theta^{-1} \tag{3.117}$$

Hence the eigenvalues of R_p and that of R_q will form a partition of the set of eigenvalues of R_M with a scaling factor of 2. Consequently, if $\lambda_{q,\max}$ is greater than $\lambda_{p,\max}$, then $\lambda_{q,\max}$ will be equal to 2 times the maximum eigenvalue of R_M. In this case, the stability range for μ_q is half of that of μ_w in the conventional system while the stability range for μ_p will be larger than that of μ_q. According to (3.114), the time constant for the ith mode of the parameter vectors $p(k)$ and $q(k)$ are of the form

$$\tau_{p,i} = \frac{1}{4\,\mu_p\,\lambda_i} \quad , \qquad i = 1, \ldots, N \tag{3.118a}$$

and

$$\tau_{q,i} = \frac{1}{4\,\mu_q\,\lambda_{N+i}} \quad , \qquad i = 1, \ldots, M-N \tag{3.118b}$$

From (3.111), the mean-square output of $P(z)$ and $Q(z)$ are given by

$$\xi_p(k) = E[\,e_p^2(k)\,] = 2\,\xi^o + E[\,\tilde{p}^t(k-1)\,\boldsymbol{R}_p\,\tilde{p}(k-1)\,] \tag{3.119a}$$

and

$$\xi_q(k) = E[\,e_q^2(k)\,] = 2\,\xi^o + E[\,\tilde{q}^t(k-1)\,\boldsymbol{R}_q\,\tilde{q}(k-1)\,] \tag{3.119b}$$

When the gradient noise in the parameter vectors during transient state is ignored, we obtain

$$\xi_p(k) = 2\,\xi^o + E[\,\tilde{p}^t(k-1)\,]\;\boldsymbol{R}_p\,E[\,\tilde{p}(k-1)\,] \tag{3.120a}$$

and

$$\xi_q(k) = 2\,\xi^o + E[\,\tilde{q}^t(k-1)\,]\;\boldsymbol{R}_q\,E[\,\tilde{q}(k-1)\,] \tag{3.120b}$$

Taking (3.114) and (3.118) into consideration, the time constant for the ith mode of ξ_p and ξ_q are

$$\tau_{\xi p,i} = \frac{1}{8\,\mu_p\,\lambda_i} \quad , \qquad i = 1, \ldots, N \tag{3.121a}$$

and

$$\tau_{\xi q,i} = \frac{1}{8\,\mu_q\,\lambda_{N+i}} \quad , \qquad i = 1, \ldots, M-N \tag{3.121b}$$

respectively. It is intriguing to see that partitioning of eigenvalues will generally make the eigenvalue spreads of the two split filters not the same and both spreads are usually smaller than that of $W(z)$. If μ_p and μ_q are assigned to the same value, the filter with a larger set of eigenvalues will converge faster because of having smaller time constants. In order to average out the convergence speed of the two split-paths to obtain a better performance, a larger step size can be assigned to the filter with a smaller set of eigenvalues as in the case for joint process estimation.

Next, we shall study the steady state behavior of the new adaptive system for AR modeling. From (3.119), the steady state MSE of the two filter outputs, $\xi_{p,ss}$ and $\xi_{q,ss}$ are given by

$$\xi_{p,ss} = 2\xi^o + tr(\boldsymbol{R}_p\, cov\{\, p(k-1)\,\}) \tag{3.122a}$$

and

$$\xi_{q,ss} = 2\xi^o + tr(\boldsymbol{R}_q\, cov\{\, q(k-1)\,\}) \tag{3.122b}$$

With the independence assumption and sufficiently small step sizes, it can be derived from (3.113) that

$$cov\{\, p(k)\,\} = cov\{\, p(k-1)\,\} - 2\mu_p \boldsymbol{R}_p\, cov\{\, p(k-1)\,\} - 2\mu_p\, cov\{\, p(k-1)\,\}\, \boldsymbol{R}_p + 8\mu_p^2 \xi^o \boldsymbol{R}_p \tag{3.123a}$$

and

$$cov\{\, q(k)\,\} = cov\{\, q(k-1)\,\} - 2\mu_q \boldsymbol{R}_q\, cov\{\, q(k-1)\,\} - 2\mu_q\, cov\{\, q(k-1)\,\}\, \boldsymbol{R}_q + 8\mu_q^2 \xi^o \boldsymbol{R}_q \tag{3.123b}$$

In steady state, (3.123) gives

$$cov\{\, p(k-1)\,\} = 2\xi^o \mu_p \boldsymbol{I}_N \tag{3.124a}$$

and

$$cov\{\, q(k-1)\,\} = 2\xi^o \mu_q \boldsymbol{I}_N \tag{3.124b}$$

Therefore

$$\xi_{p,ss} = 2\xi^o \{\, 1 + \mu_p\, tr(\boldsymbol{R}_p)\,\} \tag{3.125a}$$

and

$$\xi_{q,ss} = 2\xi^o \{\, 1 + \mu_q\, tr(\boldsymbol{R}_q)\,\} \tag{3.125b}$$

Notice that $\xi_{p,ss}$ and $\xi_{q,ss}$ are, in general, different. Putting (3.125) into (3.105), the total output final MSE is given by

$$\xi_{ss} = \xi^o \left\{ 1 + \frac{\mu_p}{2} tr(\boldsymbol{R}_p) + \frac{\mu_q}{2} tr(\boldsymbol{R}_q) \right\} \tag{3.126}$$

Finally, multiplying (3.113*a*) with the transpose of (3.113*b*) and noting that $v_p(k-1)$ and $v_q(k-1)$ are orthogonal, we have

$$E[\tilde{p}(k)\,\tilde{q}^t(k)] = E[\tilde{p}(k-1)\,\tilde{q}^t(k-1)] - 2\mu_p R_p E[\tilde{p}(k-1)\,\tilde{q}^t(k-1)] - 2\mu_q E[\tilde{p}(k-1)\,\tilde{q}^t(k-1)]\,R_q \tag{3.127}$$

when the step sizes are keeping small. Thus

$$E[\tilde{p}(k)\,\tilde{q}^t(k)] = 0 \tag{3.128}$$

that is the steady state variation of the parameters $p(k)$ and $q(k)$ are orthogonal to each other.

3.3.3 COMPARISON WITH TRADITIONAL AR MODELING SYSTEM

In this section, the excess MSE is used as a performance index to contrast the adaptation characteristics of the split-path and non-split AR modeling configuration. From (3.40), we have

$$R_p + R_q = 4R_{11} \tag{3.129}$$

Taking the trace on both sides of (3.129) gives

$$tr(R_p + R_q) = 4N\,r(0) = 2\,tr(R_M) \tag{3.130}$$

We observe from (3.95), (3.97), (3.121) and (3.126) that for $\mu_p = \mu_q = \mu_w$, the convergence time constants of the split-path system are reduced by a factor of 2, while the excess MSE for both systems remains identical. In fact, one can additionally vary μ_p and μ_q separately to further improve the adaptation speed by keeping the MSE unchanged. By equating (3.97) and (3.126) to maintain the same excess MSE for both systems, μ_p and μ_q can be chosen to satisfy the following

constraint

$$\mu_p \, tr(R_p) + \mu_q \, tr(R_q) = 2\,\mu_w \, tr(R_M) \tag{3.131}$$

The details on how to select the step sizes to achieve yet a better performance will be described in subsection 3.3.4.

As far as computational complexity is concerned, the conventional adaptive system requires $2M$ multiplications and $2M$ additions whilst the split-path system requires $N+1$ extra additions and subtractions as indicated in (3.107) and (3.108). The extra computation involved is not significant and in any case will not exceed 25% when M is large. In regard of component requirement for hardware implementation, only an inexpensive shift register is needed.

3.3.4 SELECTION OF STEP SIZES

In this section, the selection of step sizes for the split-path filters $P(z)$ and $Q(z)$ in the application of AR modeling will be considered. By properly choosing the control parameters, μ_p and μ_q, the overall adaptation speed of the system can be further improved. Two methods are hereby suggested. The first method assigns different step sizes for the two filter paths while the other uses unequal step sizes for each of the filter tap weights.

A. Method 1

It has been noted that the convergence rate might be increased by giving a larger step size to an adaptive filter with a smaller set of eigenvalues. In addition, equation (3.125) shows that a more evenly distributed excess MSE of the two split paths can be obtained when μ_p and μ_q are chosen properly according to the

autocorrelation of the input signal $x(k)$. The next task is to devise a mechanism to choose μ_p and μ_q such that the best possible adaptation performance can be achieved.

From equation (3.73), we observe that the adaptation speed of the two split-path filters can be somewhat synchronized if μ_p and μ_q are set accordingly. Furthermore, substituting (3.73) into (3.125), we found that the excess MSE for the two split-paths are identical. Therefore, (3.73) can also be used in selecting step sizes for the adaptive filters in the AR modeling system. Solving (3.73) and (3.131) yields

$$\mu_p = \frac{tr(\boldsymbol{R}_M)}{tr(\boldsymbol{R}_p)}\mu_w \tag{3.132a}$$

and

$$\mu_q = \frac{tr(\boldsymbol{R}_M)}{tr(\boldsymbol{R}_q)}\mu_w \tag{3.132b}$$

Notice that (3.132) is different from (3.75) because different performance measure is used, but in either cases a larger step size can always be assigned if the corresponding filter path has a smaller trace.

B. Method 2

Unlike the tapped delay line whitening filter, the tap inputs of the split-path structure have different signal powers. In fact, from (3.36), (3.37) and (3.40), $\boldsymbol{R}_p$ and $\boldsymbol{R}_q$ are given by

$$\boldsymbol{R}_p = 2\begin{bmatrix} r(0)-r(2N-1) & r(1)-r(2N-2) & \dots & r(N-1)-r(N) \\ r(1)-r(2N-2) & r(0)-r(2N-3) & \dots & r(N-2)-r(N-1) \\ \vdots & \vdots & \dots & \vdots \\ r(N-1)-r(N) & r(N-2)-r(N-1) & \dots & r(0)-r(1) \end{bmatrix} \tag{3.133a}$$

and

$$R_q = 2\begin{bmatrix} r(0)+r(2N-1) & r(1)+r(2N-2) & \dots & r(N-1)+r(N) \\ r(1)+r(2N-2) & r(0)+r(2N-3) & \dots & r(N-2)+r(N-1) \\ \cdot & \cdot & \dots & \cdot \\ r(N-1)+r(N) & r(N-2)+r(N-1) & \dots & r(0)+r(1) \end{bmatrix} \qquad (3.133b)$$

Hence the power of the ith tap of $P(z)$ is

$$\begin{aligned} \sigma_{p,i}^2 &= E[\, v_{p,i}^2(k)] \\ &= 2\,\{\, r(0) - r(2N+1-2i)\} \quad , \qquad i = 1, 2, \dots, N \end{aligned} \qquad (3.134a)$$

and that of $Q(z)$ is

$$\begin{aligned} \sigma_{q,i}^2 &= E[\, v_{q,i}^2(k)] \\ &= 2\,\{\, r(0) + r(2N+1-2i)\} \quad , \qquad i = 1, 2, \dots, N \end{aligned} \qquad (3.134b)$$

where $v_{p,i}(k)$ and $v_{q,i}(k)$ are the ith element of the vectors $\boldsymbol{v}_p(k)$ and $\boldsymbol{v}_q(k)$ respectively. It is likely that convergence behavior of the adaptive system could be improved by utilizing different step sizes, which are normalized to the filter taps' input power [17], for each filter weight. Thus, the weight updating equations (3.107b) and (3.107d) could now be modified to

$$\boldsymbol{p}(k) = \boldsymbol{p}(k-1) - 2\,\mu_p\, e_p(k)\, \Lambda_p^{-2}\, \boldsymbol{v}_p(k-1) \qquad (3.135a)$$

and $$\boldsymbol{q}(k) = \boldsymbol{q}(k-1) - 2\,\mu_q\, e_q(k)\, \Lambda_q^{-2}\, \boldsymbol{v}_q(k-1) \qquad (3.135b)$$

where $$\Lambda_p^2 = diag\{\, \sigma_{p,1}^2, \sigma_{p,2}^2, \dots, \sigma_{p,N}^2\} \qquad (3.136a)$$

and $$\Lambda_q^2 = diag\{\, \sigma_{q,1}^2, \sigma_{q,2}^2, \dots, \sigma_{q,N}^2\} \qquad (3.136b)$$

The power estimates $\sigma_{p,i}^2$ and $\sigma_{q,i}^2$ are found by taking an exponentially weighted average of the past samples,

$$\sigma_{p,i}^2(k) = \alpha\, \sigma_{p,i}^2(k-1) + (1-\alpha)\,\{\, v_{p,i}^2(k-1)\} \quad , \qquad 0 < \alpha < 1 \qquad (3.137a)$$

and $$\sigma_{q,i}^2(k) = \alpha\, \sigma_{q,i}^2(k-1) + (1-\alpha)\,\{\, v_{q,i}^2(k-1)\} \quad , \qquad 0 < \alpha < 1 \qquad (3.137b)$$

Initial power estimates $\sigma_{p,i}^2(0)$ and $\sigma_{q,i}^2(0)$ are usually set to smaller values so as to speed up the rate of convergence right at the beginning. The modified algorithm shown in (3.135) should have the update mechanism disabled for the filter tap if its power estimate falls below an acceptable level.

Assume α is large such that Λ_p^{-2} and Λ_q^{-2} can be considered to be fixed after the initial transient, it can be proved easily that the steady state weight vector variation for $P(z)$ and $Q(z)$ are given by

$$cov\{\tilde{p}(k-1)\} = 2\mu_p \xi^o \Lambda_p^{-2} \tag{3.138a}$$

and

$$cov\{\tilde{q}(k-1)\} = 2\mu_q \xi^o \Lambda_q^{-2} \tag{3.138b}$$

The output MSE of the modified algorithm then becomes

$$\xi_{ss} = \xi^o \left\{ 1 + \frac{\mu_p}{2} N + \frac{\mu_q}{2} N \right\} \tag{3.139}$$

If the excess MSE in the split-path and in the non-split system are kept at the same value, the step sizes in the split-path structure should be chosen according to

$$\mu_p + \mu_q = \frac{2\mu_w}{N} tr(\boldsymbol{R}_M) \tag{3.140}$$

Although computational burden is increased in this method, a faster convergence rate is expected since more information about the signal statistics is exploited in the adaptation. Notice that method 2 is not suitable for performance enhancement when MSD is used as the performance index because the filter tap that has a smaller power will produce a very large parameter variation as indicated by (3.138).

3.3.5 SOME NOTES ON ODD FILTER ORDER CASE

When the filter order is odd, say $M = 2N + 1$, similar results can also be derived although the notations might be a little bit more complicated. The transfer function of $P(z)$ and $Q(z)$ are now given by

$$P(z) = \sum_{i=0}^{N} p_i \left(z^{-i} - z^{-2N-2+i} \right) \tag{3.141a}$$

and

$$Q(z) = \sum_{i=0}^{N} p_i \left(z^{-i} - z^{-2N-2+i} \right) + \sqrt{2}\, q_{N+1}\, z^{-N-1} \tag{3.141b}$$

and the realization scheme is shown in Figure 3.13. The vector $\boldsymbol{q}$ now has N+1 elements and $\boldsymbol{p}$ and $\boldsymbol{q}$ are related to w by

$$\begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{q} \end{bmatrix} = \Theta\, w = \begin{bmatrix} \boldsymbol{I}_N & 0_N & -\boldsymbol{J}_N \\ \boldsymbol{I}_N & 0_N & \boldsymbol{J}_N \\ 0_N^t & \sqrt{2} & 0_N^t \end{bmatrix} w \tag{3.142}$$

The adaptation algorithm is the same as before with $\boldsymbol{v}_p(k-1)$ and $\boldsymbol{v}_q(k-1)$ become

$$\boldsymbol{v}_p(k-1) = [\, \boldsymbol{I}_N \quad 0_N \quad -\boldsymbol{J}_N \,]\, \boldsymbol{x}_M(k-1) \tag{3.143a}$$

and

$$\boldsymbol{v}_q(k-1) = \begin{bmatrix} \boldsymbol{I}_N & 0_N & \boldsymbol{J}_N \\ 0_N^t & \sqrt{2} & 0_N^t \end{bmatrix} \boldsymbol{x}_M(k-1) \tag{3.143b}$$

respectively. The selection criterion for the step sizes in the split-path system described previously can still be applied in this case.

3.3.6 SIMULATION RESULTS

Extensive computer simulations have been carried out in order to evaluate the convergence behavior and to assess the overall adaptive performance of the split-path system. In these experiments, the AR process being used is given by (3.84). Because

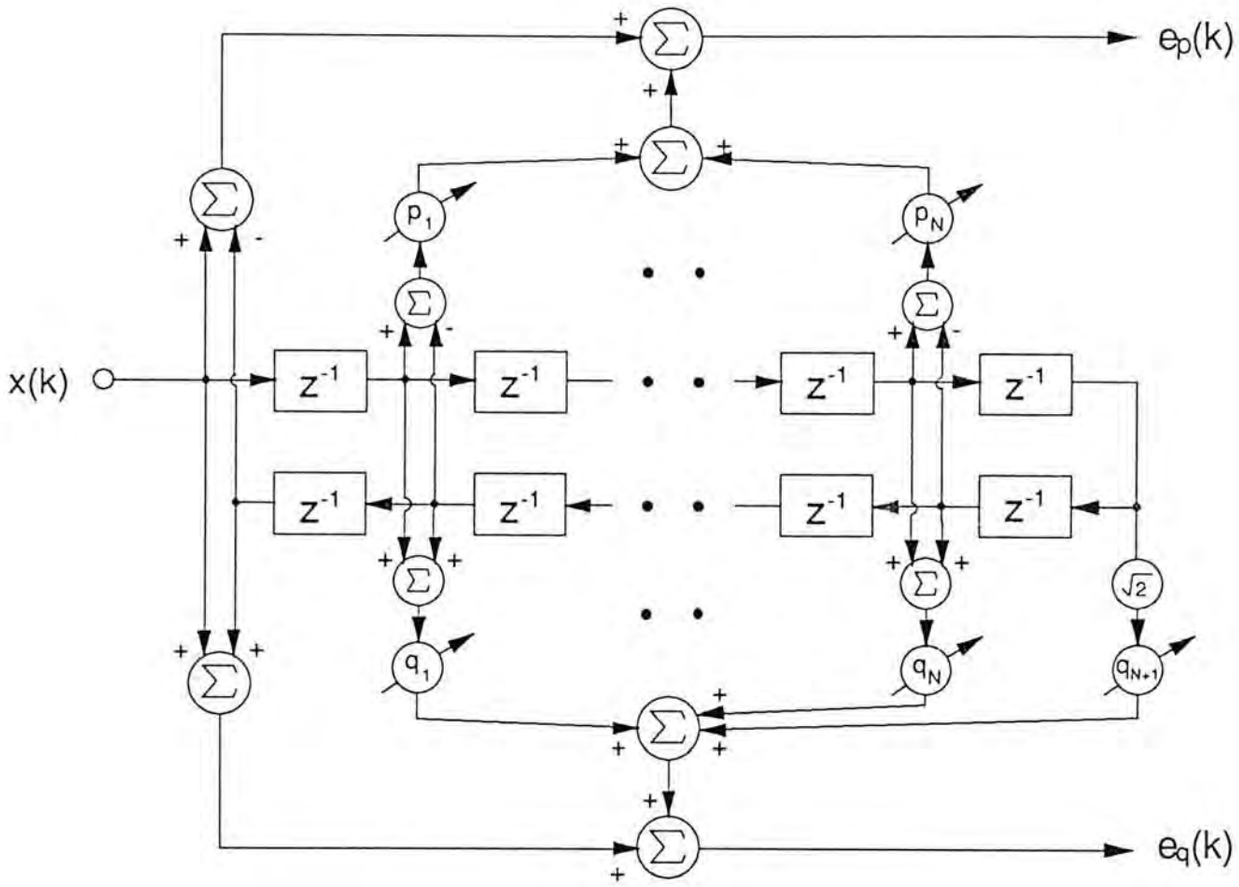


Fig. 3.13 Hardware implementation of the split-path whitening filter for odd M

of the incorporation of the backward prediction knowledge to the adaptive algorithm, the split-path system is able to reduce the time constant for convergence by a factor of 2 when $\mu_p = \mu_q = \mu_w$. In addition, due to the decrease in eigenvalue spreads, further improvement can be achieved by choosing different suitable values for the step sizes. The simulation results obtained were the average of 1000 independent runs to eliminate temporal fluctuation.

Figure 3.14 compares the learning curves in which the MSE was plotted against the number of iterations for the split-path and the conventional system. The total MSE in the split-path case was obtained from (3.105). The step sizes for the two methods were assigned the same values such that both systems would have identical excess MSE. As shown in the diagram, the split-path model indeed converged at a faster rate as predicted and the MSE dropped to a value of 0.125 at about 700 iterations. Whereas for the conventional system, 1400 iterations were required.

The convergence characteristic of the split-path structure was also investigated when using different values for μ_p and μ_q. The learning curve is depicted in Figure 3.15 in which the two step sizes were chosen in accordance with (3.132) to achieve the best performance. A similar curve for the case with $\mu_p = \mu_q$ is also given for ease of comparison. It is seen that using different step sizes can improve the dynamic convergence behavior significantly. In fact, the MSE has dropped to 0.125 at about 300 iterations which is roughly half of the time as required in the former test.

Figure 3.16 compares the performance of the two step size selection methods as suggested in subsection 3.3.4. The step sizes for the two split-path systems were chosen such that identical steady state excess MSE were obtained. In method 1, μ_p and μ_q were selected according to (3.132). Whilst, in method 2, they were adjusted to follow equation (3.140) to produce the best possible performance. The

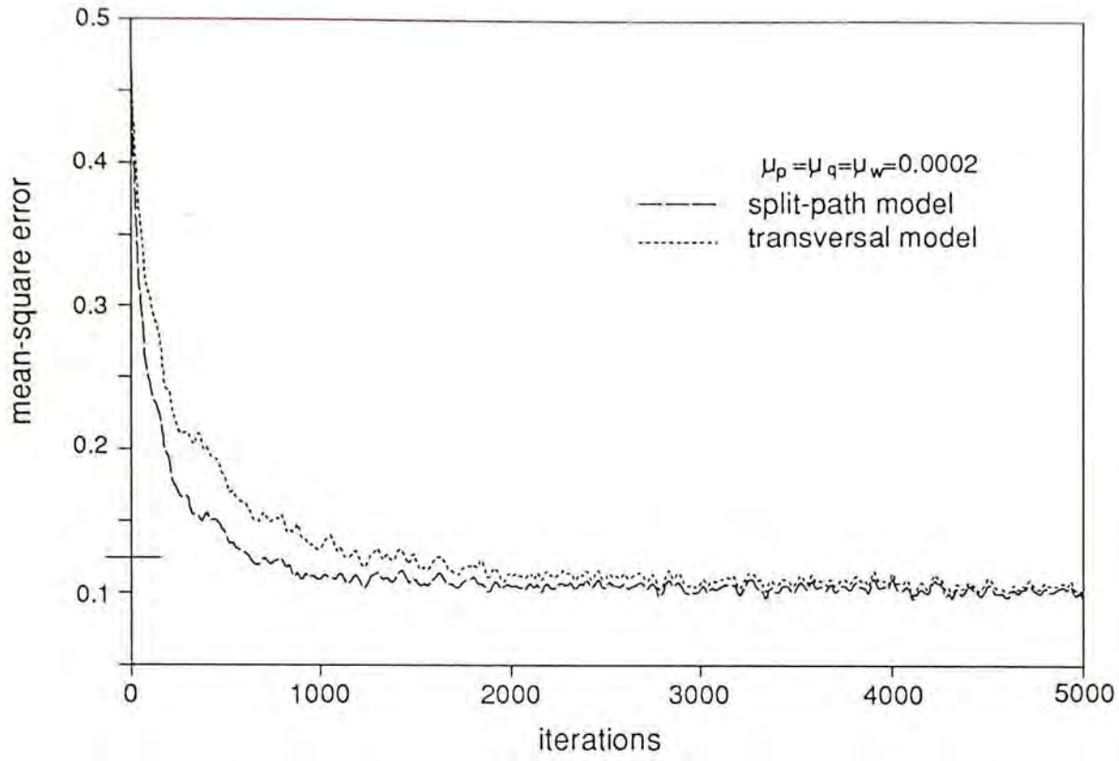


Fig. 3.14 Comparison of adaptation characteristics

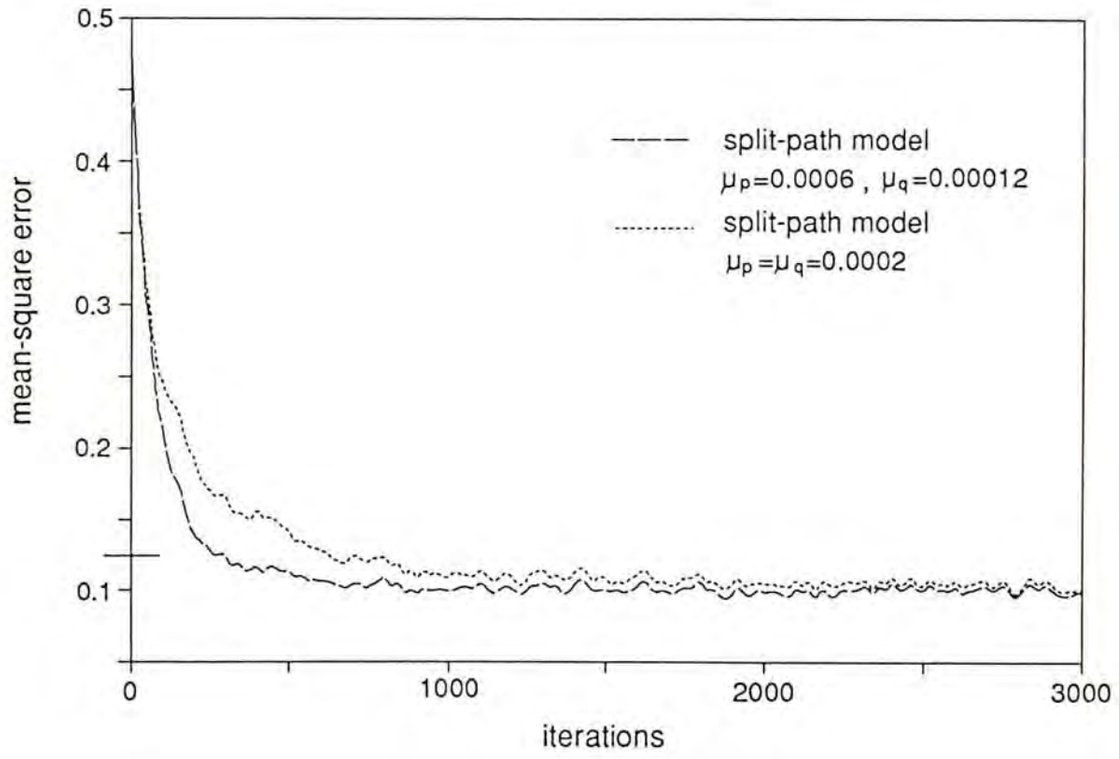


Fig. 3.15 Comparison of learning curves in the split-path model when using different μ

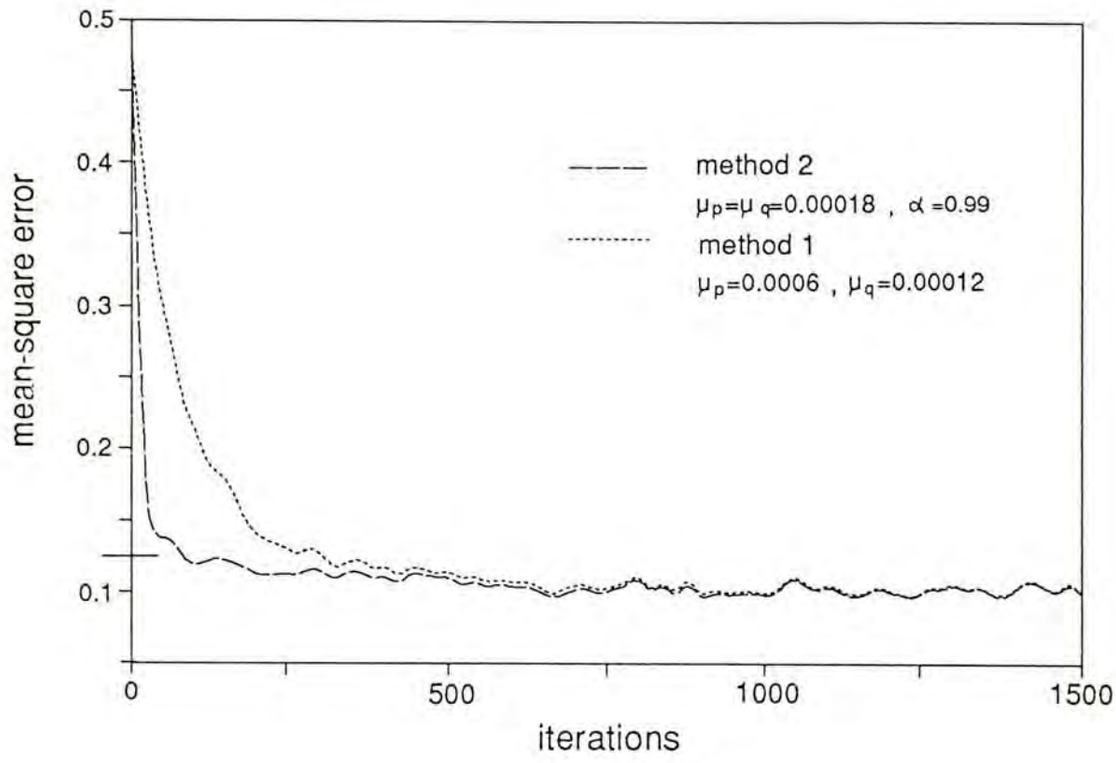


Fig. 3.16 Comparison of learning curves with step sizes being selected according to method 1 and 2 for the split-path AR modeling system

variable α was set to 0.99 and the initial power estimates were assigned to 0.02. It can be observed from Figure 3.16 that method 2 will achieve a better dynamic convergence behavior. Indeed, in this case only 90 iterations were required to have the MSE reduced to a value of 0.125 whereas for method 1, almost 300 iterations were needed. However, computations that were involved in method 2 was much higher.

3.3.7 APPLICATION TO NOISE CANCELLATION [18]-[19]

In this subsection, the application of the proposed filter structure to a typical problem of noise cancellation is examined. Simulation results are provided to demonstrate the superiority of the split-path structure over the traditional configuration.

Noise cancelling is a kind of optimal filtering that has found applications in many areas [3], [4], [20]-[26]. Assume that we have a noisy measurements $x(k)$ which is given by

$$x(k) = s(k) + n_1(k) \tag{3.144}$$

where $s(k)$ is the signal and $n_1(k)$ is the corrupting noise. The purpose is to extract the signal $s(k)$ from the noisy observations $x(k)$.

Figure 3.17 illustrates the concept of noise cancellation. It makes use of an auxiliary input $n_2(k)$ derived from one or more sensors located at points in the noise field where the signal is weak or undetectable. The input is then passed through an optimal filter to get as close a replica as possible of the noise in the primary input $x(k)$. The filter output is then subtracted from the primary signal and, as a result, the additive noise is attenuated or eliminated and the desired signal will be the output of the process.

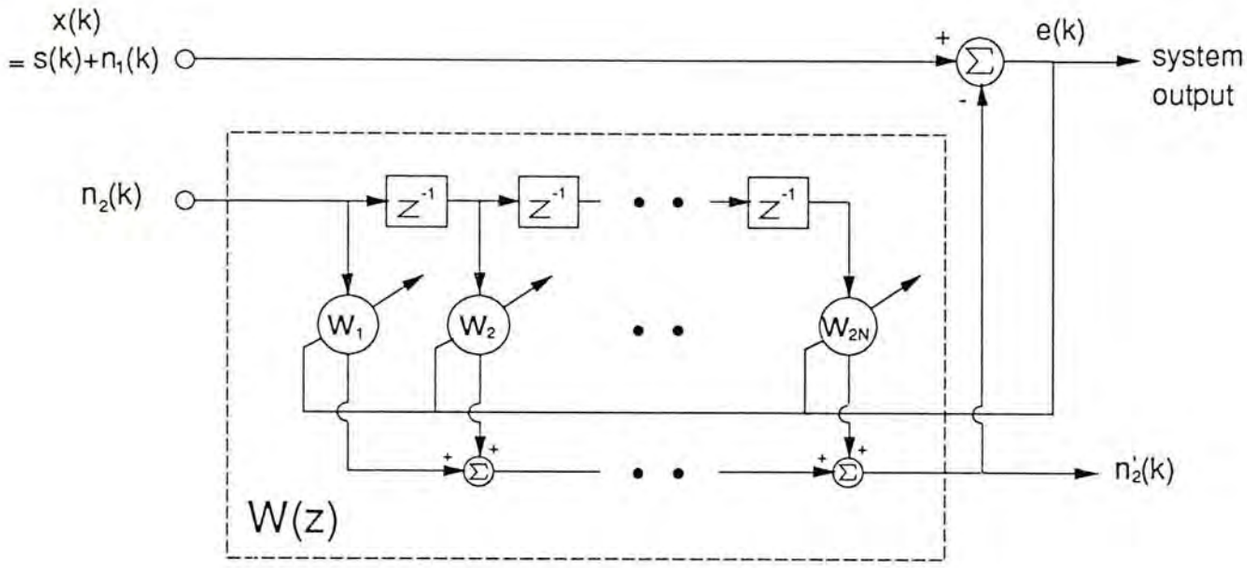


Fig. 3.17 An adaptive noise canceller

Optimal design of such a filter must be based on the knowledge of the signal statistics. However, in practical situation, the operating environment is always changing and this information is usually unavailable. One possible way to alleviate this difficulty is to make the canceller adaptive. The filter $W(z)$ of the canceller is now replaced by an adaptive filter with its coefficients adjusted iteratively according to the LMS adaptation algorithm to minimize the squared output error. Because of the simplicity and ease of implementation, the LMS adaptive noise canceller has been widely used in echo cancellation [20]-[21] and speech enhancement [23]-[24].

The operation principle of the adaptive noise canceller can be explained as follows. Let the filter output of $W(z)$ be $n_2'(k)$. Assuming the signal $s(k)$, the noise $n_1(k)$ and $n_2(k)$ are statistically stationary and have zero means. Moreover, it is assumed that $s(k)$ is uncorrelated with $n_1(k)$ and $n_2(k)$, and that $n_2(k)$ is correlated with $n_1(k)$. The output of the canceller is given by

$$e(k)=s(k)+n_1(k)-n_2'(k) \tag{3.145}$$

Squaring both sides of (3.145) and taking expectation yields

$$\begin{aligned} E[e^2(k)] &= E[s^2(k)]+E[(n_1(k)-n_2'(k))^2]+2E[s(k)(n_1(k)-n_2'(k))] \\ &= E[s^2(k)]+E[(n_1(k)-n_2'(k))^2] \end{aligned} \tag{3.146}$$

Notice that the signal power $E[s^2(k)]$ is unaffected when the filter is operated to minimize the output MSE $E[e^2(k)]$. Accordingly, the minimum output MSE is given by

$$\min\{E[e^2(k)]\}=E[s^2(k)]+\min\{E[(n_1(k)-n_2'(k))^2]\} \tag{3.147}$$

Thus the MSE will attain its minimum value when $E[(n_1(k)-n_2'(k))]=0$ which implies at steady state,

$$n_1(k) = n_2'(k) \tag{3.148}$$

Substituting (3.148) into (3.145), the steady state output error is equal to

$$e(k) = s(k) \tag{3.149}$$

which shows that the adaptive canceller can perfectly retrieve the signal $s(k)$.

When the signal to be enhanced is a sum of sinusoids while the corrupting noise is completely random, the auxiliary input can be replaced by a unit delayed version of the primary input, that is, $n_2(k) = x(k-1)$. This situation is of particular interest to many applications and the resulting canceller is usually referred as an adaptive line enhancer (ALE) [25]-[26]. The performance characteristics of a typical ALE can be found in [25]. In this case, the signal component in both channels are correlated as opposed to the random noise. When the output squared error is being minimized, the correlated component between the two input would be cancelled and the filter output then becomes the enhanced signal.

Assuming that the signal $s(k)$ is of the form

$$s(k) = \sum_{i=1}^{L} A_i \sin(\omega_i k + \phi_i) \tag{3.150}$$

where L is the number of sinusoids present whose amplitudes, frequencies and phases are respectively given by A_i, ω_i and ϕ_i. The signal $s(k)$ can roughly be expressed as a linear combination of its previous samples [25],

$$s(k) = \sum_{j=1}^{2N} h_j s(k-j) \tag{3.151}$$

where h_j is known as the matched filter response which is the sum of sampled sinusoids with frequencies at ω_i. Notice that if $n_1(k)$ is a broad band random noise, (3.151) can be expressed as

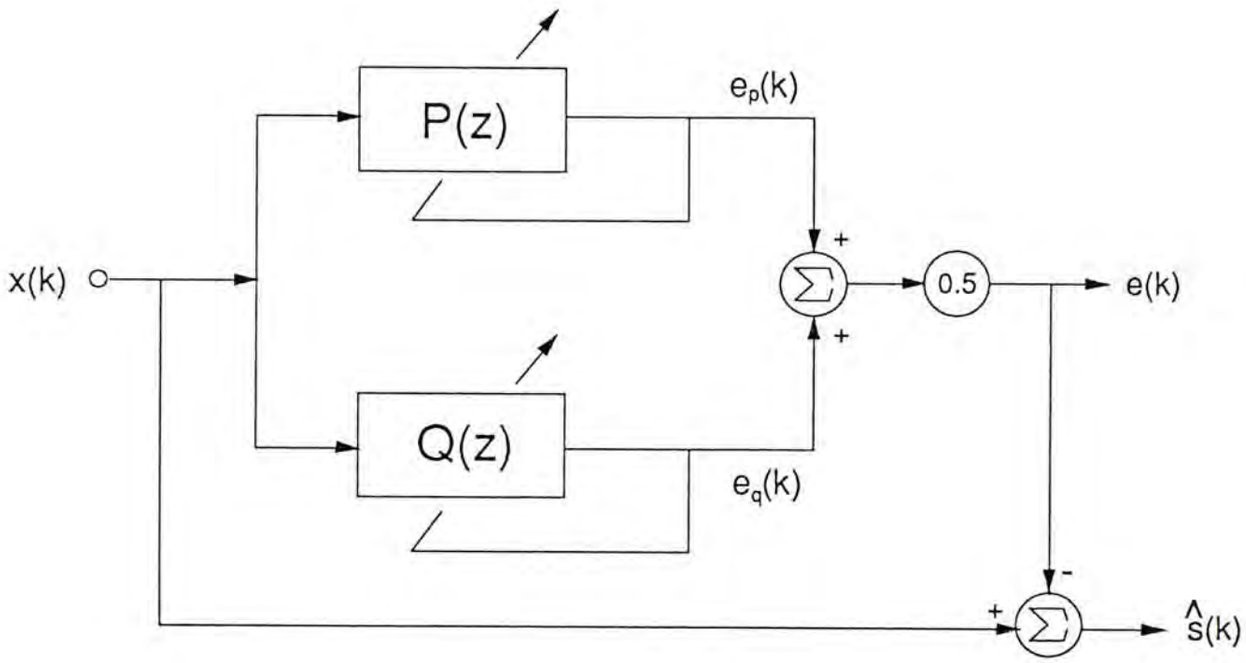


Fig. 3.18 A split-path ALE

$$s(k)=K\sum_{i=1}^{2N} h_i\{s(k-i)+n_1(k-i)\}=\sum_{i=1}^{2N} -g_i\,x(k-i) \tag{3.152}$$

where K is a positive constant less than unity being dependent on the signal to noise power. Now, (3.144) can be rewritten to

$$x(k)=\sum_{i=1}^{2N} -g_i\,x(k-i)+n_1(k) \tag{3.153}$$

(3.153) shows that the noisy observations $x(k)$ can be modelled by an AR($2N$) process. When $x(k)$ is passed through an optimal whitening filter whose impulse responses are g_i, the filter output $n_2'(k)$ will be $n_1(k)$. Our signal estimate, $\hat{s}(k)$, can then be retrieved by subtracting the filter output from the noisy input, that is,

$$\hat{s}(k)=x(k)-n_2'(k) \tag{3.154}$$

A drawback for this particular application is its slow convergence. Since $x(k)$ can be considered as an AR process, Ching and Ho [18] have shown that the split-path adaptive filter for AR modeling can be applied to improve the adaptation speed. The schematic block diagram of a split-path adaptive line enhancer is depicted in Figure 3.18. The transfer function of the two split-path filters $P(z)$ and $Q(z)$ are again given by (3.99). The filter weights are adjusted iteratively in accordance with the adaptation rule governed by (3.107). The total output error is obtained by (3.98) and finally, the signal estimate is determined from (3.154).

The split-path noise canceller was simulated on a micro-computer. Figure 3.19 shows the output of the new and the old system when the input signal $x(k)$ was a composite of three sinusoids and an uncorrelated Gaussian random noise, that is

$$\begin{aligned} x(k) &= 0.8\cos(0.015\pi k)+1.0\cos(0.0124\pi k) \\ &\quad +1.2\cos(0.0048\pi k)+n_1(k) \end{aligned} \tag{3.155}$$

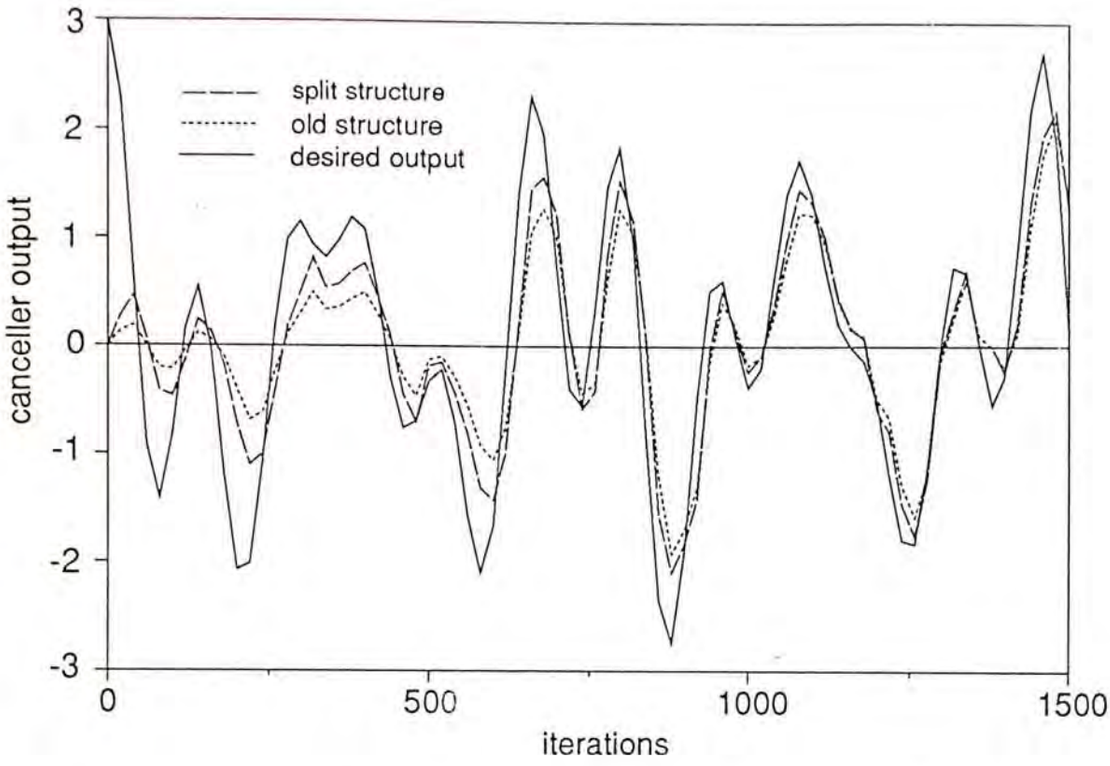


Fig. 3.19 The outputs of the split and the traditional ALE

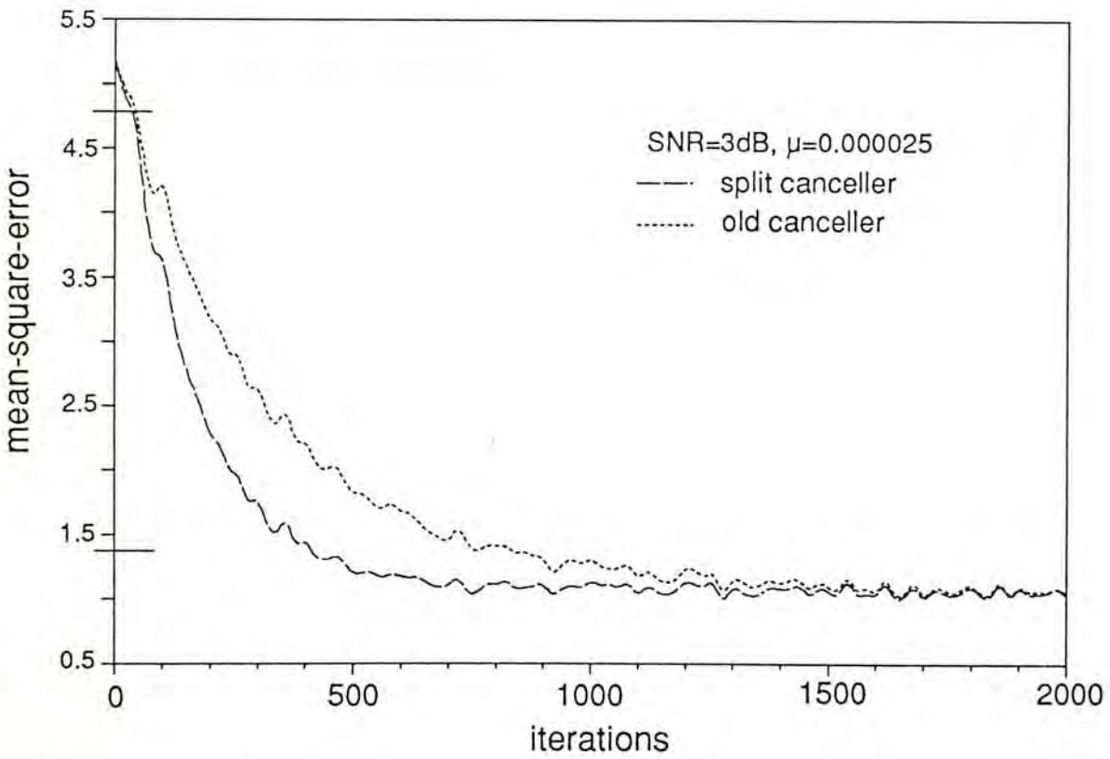


Fig. 3.20 Comparison of convergence for the two types of cancellers

The noise power was set to unity and the order of the split and that of the non-split canceller were both chosen to be $N = 16$. The step sizes for the two systems were all fixed to 0.000025. The desired output was also provided for ease of comparison. It is clear from Figure 3.19 that the new canceller can reach the desired output with a shorter transition than its counterpart.

To compare the convergence characteristics in more detail, a signal which is given by

$$x(k) = 2\cos(0.125\pi k) + n_1(k) \tag{3.156}$$

was used as the input and the learning characteristics for the two systems are plotted in Figure 3.20. The noise power, the value N and the step sizes were all assigned to the same values as before. Let the adaptation time be the number of iterations required for the output MSE to fall between 90% and 10% levels drawn between the input power level and the asymptotic equilibrium level. It can be easily seen that the adaptation time for the new canceller is about half of that obtained by the conventional structure and the improvement in convergence speed is almost doubled, which agreed with our theoretical developments.

At high SNR, one can yet enhance the adaptation rate by choosing different step sizes for the two filter paths or each of the filter taps. However, when the SNR is low, the autocorrelation matrix of the input sequence tends to a diagonal matrix. In such circumstances, choosing different step sizes cannot achieve further improvement.

3.4 CHAPTER SUMMARY

A new adaptive model formed by splitting an adaptive filter to two linear phase filters, one antisymmetric and the other symmetric, connected in parallel is investigated. This model was applied to system identification and AR modeling.

In the system identification case, the two linear phase filters are adapted by Widrow's LMS algorithm to minimize the total output MSE. It is found that the essence of splitting is to partition the eigenvalues of the input correlation matrix into two sets, giving rise to two eigenvalue spreads that are usually smaller as compared with the non-split model. When appropriate step sizes for the two filters are chosen in accordance with their eigenvalue spreads, the system performance in terms of convergence speed can be enhanced.

When the split-path model is used for AR modeling, unlike system identification, there are two distinct features in this application. First, the backward prediction knowledge can be incorporated naturally into the two linear phase filters so that the gradient noise associated with the adaptation process can be halved. Second, the output errors of the two linear phase filters are orthogonal to each other and independent adaptation for the two filters is feasible. As a result, when the two linear phase filters are adapted separately with each other using the LMS algorithm to minimize their respective outputs, a two-fold increase in adaptation speed can be achieved and, also, a parallel processing architecture for ease of hardware realization is possible. Due to the decrease in eigenvalue spreads, extra gain in convergence speed can be obtained if unequal step sizes are selected for the two split-path filters. In addition, experimental results have confirmed that choosing different step sizes for individual filter taps will further enhance the adaptation performance of the split-path system.

The split-path adaptive system for AR modeling can also be applied to tackle the typical noise cancellation problem in extracting sinusoidal signals corrupted by random noise. Simulation results verify the superiority of the split-path structure over the conventional structure by having a significant improvement in adaptation speed.

An important point needed to emphasise on the split-path system is its simplicity. In contrast to the transversal model, additional hardware required is roughly M addition operations which only accounts for a modest increase in computational burden. In linear prediction case, only one more shift register is needed.

REFERENCES

[1] E. H. Satorius and S. T. Alexander, "Channel equalization using adaptive lattice algorithms," *IEEE Trans. Commun.*, vol. COM-27, pp. 899-905, June 1979.

[2] K. C. Ho and P. C. Ching, "System identification using a pair of linear phase filter," in *Proc. IEEE ISCAS-91*, Singapore, June 1991, pp. 552-555.

[3] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong and R. C. Goodlin, "Adaptive noise cancelling: principles and applications," *Proc. IEEE*, vol. 63, pp. 1692-1716, Dec. 1975.

[4] G. S. Furno and W. J. Tompkins, "A learning filter for removing noise interference," *IEEE Trans. Biomed. Eng.*, vol. BME-30, pp. 234-235, Apr. 1983.

[5] S. U. H. Qureshi, "Adaptive equalization," *Proc. IEEE,* vol. 73, pp. 1349-1387, Sept. 1985.

[6] K. C. Ho and P. C. Ching, "Performance analysis of a split-path LMS adaptive filter for AR modeling," to appear in *IEEE Trans. Signal Processing*.

[7] S. M. Kay, "The effects of noise on the autoregressive spectral estimator," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 478-485, Oct. 1979.

[8] N. Beamish and M. B. Priestly, "A study of autoregressive and window spectral estimation," *Appl. Statist.*, vol. 30, pp. 41-58, 1981.

[9] T. S. Lee, "Large sample identification and spectral estimation of noisy multivariate autoregressive processes," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 76-82, Feb. 1983.

[10] D. F. Ginras and E. Masry, "Autoregressive spectral estimation in additive noise," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 490-501, Apr. 1988.

[11] S. Li and B. W. Dickinson, "Performance contours of autoregressive estimates," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 608-610, Apr. 1988.

[12] J. Makoul, "Linear prediction: A tutorial review," *Proc. IEEE*, vol. 63, pp. 561-580, 1975.

[13] R. Andre-Obrecht, "Automatic segmentation of continuous speech signals based on autoregressive statistical model," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 29-40, Jan. 1988.

[14] K. K. Paliwal, "Estimation of noise variance from noisy autoregressive signal and its application in speech enhancement," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 292-294, Feb. 1988.

[15] E. A. Robinson, *Statistical communication and Detection.* NY: Hafner, 1967.

[16] J. Durbin, "The fitting of time-series models," *Rev. Inst. Int. Statist.*, vol. 28, pp. 233-243, 1960.

[17] S. S. Narayan, A. M. Perterson and M. J. Narasimha, "Transform domain LMS algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 609-615, June 1983.

[18] P. C. Ching and K. C. Ho, "A split structure for adaptive line enhancer," *Int. J. Electronics*, vol. 70, pp. 565-571, Mar. 1991.

[19] K. C. Ho and P. C. Ching, "Noise cancellation using a split-path adaptive filter," in *Proc. Int. Conf. Circuits Syst.*, Shenzhen, China, June 1991, pp. 149-152.

[20] M. M. Sondhi and D. Mitra, "New results on the performance of a well-known class of adaptive filters," *Proc. IEEE*, vol. 64, pp. 1583-1597, Nov. 1976.

[21] S. Yamamoto and S. Kitayama, "An adaptive echo canceller and variable step gain method," *Trans. IECE Japan*, vol. E65, pp. 1-8, Jan. 1982.

[22] W. B. Mikhael *et al.*, "Optimum adaptive algorithms with applications to noise cancellation," *IEEE Trans. Circuits Syst.*, vol. CAS-31, pp. 312-315, Mar. 1984.

[23] W. A. Harrison, J. S. Lim and E. Singer, "A new application of adaptive noise cancellation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 21-27, Feb. 1986.

[24] E. Singer, "Adaptive noise reduction in aircraft communication systems," in *Proc. IEEE ICASSP-87*, Texas, Apr. 1987, pp. 169-172.

[25] J. R. Zeidler, E. H. Satorius, D. M. Chabries and H. T. Wexler, "Adaptive enhancement of multiple sinusoids in uncorrelated noise," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, pp. 240-254, June 1978.

[26] J. R. Treichler, "Transient and convergent behavior of the adaptive line enhancer," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 53-62, Feb. 1979.

4 SERIAL SPLIT ADAPTIVE SYSTEM

It has been shown in chapter 3 that an adaptive filter $W(z)$ in transversal form can be represented by two linear phase filters connected in parallel. Besides parallel split, $W(z)$ can also be implemented by connecting two adaptive subunits in series. In this chapter, we shall study the effect incurred on adaptation characteristics and behavior by serial splitting $W(z)$, particularly in the application of adaptive time delay estimation.

4.1 SERIAL FORM ADAPTIVE FILTER

Let us denote the filter order of $W(z)$ by $M=2N$, where N is assumed to be an even number. Here, we only consider splitting a transversal form adaptive filter $W(z)$ into two components, $A(z)$ and $C(z)$, of the same order connected in cascade. When N is odd, $W(z)$ is decomposed in a way that $A(z)$ is of order $N+1$ and $C(z)$ is of order $N-1$ to avoid the possibility of having complex filter coefficients due to complex zero pairs. $W(z)$ can now be expressed as

$$W(z)=\sum_{i=0}^{M} w_i z^{-i}=A(z)C(z)=\left(\sum_{i=0}^{N} a_i z^{-i}\right)\left(\sum_{j=0}^{N} c_j z^{-j}\right) \tag{4.1}$$

where w_i, a_i and c_j are the filter coefficients of $W(z)$, $A(z)$ and $C(z)$ respectively. The coefficient c_0 is fixed to unity in order to maintain the same number of adapting parameters in these two models. By comparing the coefficients of z^{-i} on both sides of (4.1), the filter weights between the split and non-split structure are related by

$$w_i = \begin{cases} \sum_{j=0}^{i} a_j c_{i-j} , & 0 \le i \le N \\ \sum_{j=i-N}^{N} a_j c_{i-j} , & N+1 \le i \le 2N \end{cases} \tag{4.2}$$

The schematic block diagram of an adaptive system corresponds to serial splitting of an adaptive filter is shown in Figure 4.1. The input signal $x(k)$ passes through the two filters $A(z)$ and $C(z)$ to form the output $z(k)$. The error signal obtained by subtracting $z(k)$ from the desired response $y(k)$ is used to adjust the two filters $A(z)$ and $C(z)$. Let

$$\boldsymbol{a} = [\, a_1 \quad a_2 \quad \ldots \quad a_N \,]^t \tag{4.3}$$

$$\boldsymbol{c} = [\, c_1 \quad c_2 \quad \ldots \quad c_N \,]^t \tag{4.4}$$

$$\boldsymbol{x}(k-1) = [\, x(k-1) \quad x(k-2) \quad \ldots \quad x(k-N) \,]^t \tag{4.5}$$

and

$$\boldsymbol{X}(k-1) = [\, \boldsymbol{x}(k-2) \quad \boldsymbol{x}(k-3) \quad \ldots \quad \boldsymbol{x}(k-N-1) \,]^t = \boldsymbol{X}^t(k-1) \tag{4.6}$$

Then the output error can be expressed as

$$\begin{aligned} e(k) &= y(k) - [\, 1 \quad \boldsymbol{c}^t \,] \begin{bmatrix} x(k) & \boldsymbol{x}^t(k-1) \\ \boldsymbol{x}(k-1) & \boldsymbol{X}(k-1) \end{bmatrix} \begin{bmatrix} a_0 \\ \boldsymbol{a} \end{bmatrix} \\ &= y(k) - x(k)\, a_0 - \boldsymbol{x}^t(k-1)\, \boldsymbol{a} - \boldsymbol{c}^t\, \boldsymbol{x}(k-1)\, a_0 - \boldsymbol{c}^t\, \boldsymbol{X}(k-1)\, \boldsymbol{a} \end{aligned} \tag{4.7}$$

where the superscript t denotes the transpose operation. Notice that the mean-square error (MSE), $\xi = E[\, e^2(k) \,]$, to be minimized is a quadratic function with respect to the parameters a_0, $\boldsymbol{a}$ and $\boldsymbol{c}$, and hence a unique global minimum exists for $a_0 \neq 1$. The optimum solution a_i^* and c_i^* are related to the desired weights w_i^* of $W(z)$ by (4.2). After taking partial derivatives of $e(k)$ with respect to a_0, $\boldsymbol{a}$ and $\boldsymbol{c}$, the LMS adaptation formulas for the parameters are

$$\boldsymbol{c}(k+1) = \boldsymbol{c}(k) + 2\mu\, e(k) \begin{bmatrix} \boldsymbol{x}(k-1) & \boldsymbol{X}(k-1) \end{bmatrix} \begin{bmatrix} a_0 \\ \boldsymbol{a}(k) \end{bmatrix} \tag{4.8}$$

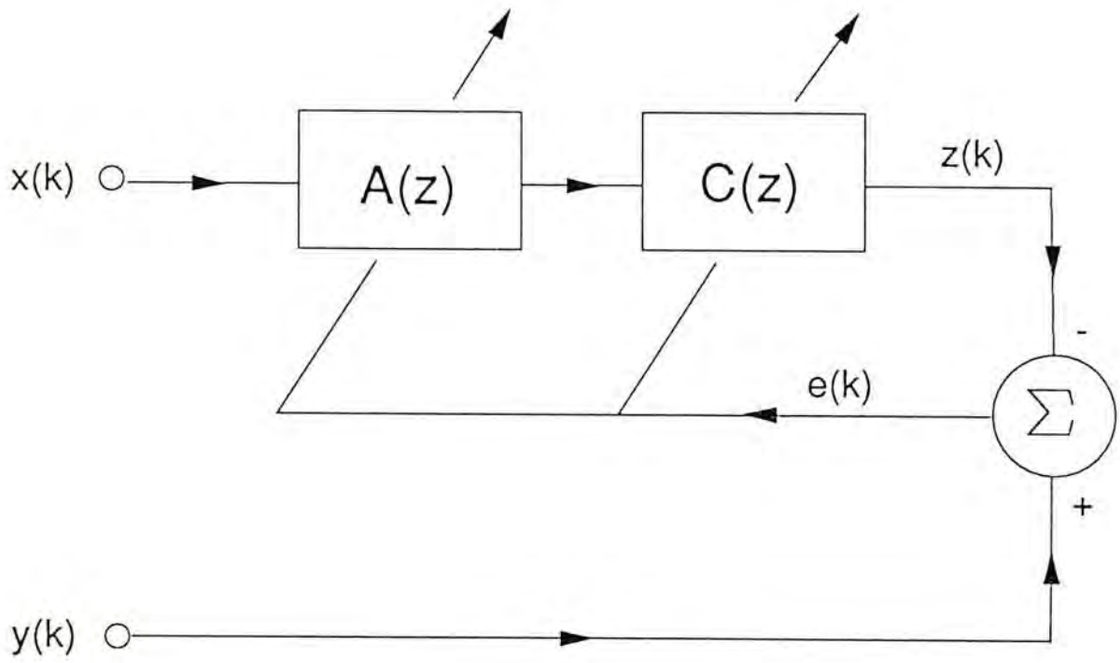


Fig. 4.1 A serial split adaptive filter

and
$$\begin{bmatrix} a_0(k+1) \\ \boldsymbol{a}(k+1) \end{bmatrix} = \begin{bmatrix} a_0(k) \\ \boldsymbol{a}(k) \end{bmatrix} + 2\mu\, e(k) \begin{bmatrix} x(k) & \boldsymbol{x}^t(k-1) \\ \boldsymbol{x}(k-1) & \boldsymbol{X}(k-1) \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{c}(k) \end{bmatrix} \tag{4.9}$$

Comparing (4.8) and (4.9) with the updating equations for transversal adaptive filter, an additional N multiplications and N additions is required to calculate the gradient to adjust $c_1(k)$. This is because the gradient of $c_{i-1}(k)$ is equal to that of $c_i(k-1)$ and the same is true for the gradient of $a_{i-1}(k)$ and $a_i(k-1)$, when the step size μ is small. Moreover, the gradient to adjust $a_0(k)$ is readily available by storing intermediate values during the calculation of the error $e(k)$. When (4.7) is put into (4.8), we have

$$\begin{aligned} \boldsymbol{c}(k+1) = (\boldsymbol{I} - 2\mu \{ a_0(k)\boldsymbol{x}(k-1) + \boldsymbol{X}(k-1)\boldsymbol{a}(k) \} \{ a_0(k)\boldsymbol{x}^t(k-1) + \boldsymbol{a}^t(k)\boldsymbol{X}(k-1) \})\boldsymbol{c}(k) \\ + 2\mu \{ a_0(k)\boldsymbol{x}(k-1) + \boldsymbol{X}(k-1)\boldsymbol{a}(k) \} \{ y(k) - a_0(k)x(k) - \boldsymbol{a}^t(k)\boldsymbol{x}(k-1) \} \end{aligned} \tag{4.10}$$

where $\boldsymbol{I}$ is an identity matrix of size N. Taking expectation and following the analysis of the LMS adaptation algorithm, the relaxation time constant for $E[\boldsymbol{c}(k)]$ is given by

$$\tau_{c,i} = \frac{1}{2\mu\lambda_{c,i}} \quad , \qquad i = 1, 2, \ldots, N \tag{4.11}$$

where $\lambda_{c,i}$ are the eigenvalues of the $N \times N$ matrix which is of the form

$$\boldsymbol{R}_c = E\left[[\boldsymbol{x}(k-1) \quad \boldsymbol{X}(k-1)] \begin{bmatrix} a_0(k) \\ \boldsymbol{a}(k) \end{bmatrix} [a_0(k) \quad \boldsymbol{a}^t(k)] \begin{bmatrix} \boldsymbol{x}^t(k-1) \\ \boldsymbol{X}(k-1) \end{bmatrix} \right] \tag{4.12}$$

From (4.4)-(4.6), we observe that the ith element of the vector $a_0\boldsymbol{x}(k-1) + \boldsymbol{X}(k-1)\boldsymbol{a}$ is given by $a_0 x(k-i) + \boldsymbol{x}^t(k-i-1)\boldsymbol{a}$, which is the output of $A(z)$ with input $x(k-i)$. Therefore $a_0\boldsymbol{x}(k-1) + \boldsymbol{X}(k-1)\boldsymbol{a}$ is a vector of N output samples of $A(z)$. That means, $\boldsymbol{R}_c$ is the autocorrelation matrix of the input sequence filtered by $A(z)$.

It can also be derived in a similar manner that the relaxation time constant for $a_0(k)$ and $E[\,a(k)]$ are

$$\tau_{a,i}=\frac{1}{2\,\mu\,\lambda_{a,i}} \quad , \qquad i=1\,,2\,,\ldots\,,N+1 \tag{4.13}$$

where $\lambda_{a,i}$ are the eigenvalues of the $(N+1)\,x\,(N+1)$ matrix $\boldsymbol{R}_a$. It can be verified that $\boldsymbol{R}_a$ is the autocorrelation matrix of the input sequence filtered by $C(z)$ and can be expressed as

$$\boldsymbol{R}_a=E\left[\begin{bmatrix} x(k) & \boldsymbol{x}^t(k-1) \\ \boldsymbol{x}(k-1) & \boldsymbol{X}(k-1) \end{bmatrix}\begin{bmatrix} 1 \\ \boldsymbol{c}(k) \end{bmatrix}[\,1 \quad \boldsymbol{c}^t(k)]\begin{bmatrix} x(k) & \boldsymbol{x}^t(k-1) \\ \boldsymbol{x}(k-1) & \boldsymbol{X}(k-1) \end{bmatrix}\right] \tag{4.14}$$

To investigate how $A(z)$ and $C(z)$ affect the eigenvalue spread of the input sequence and hence the adaptation speed of the two subunits, we first derive the relationship between the eigenvalue spread and the power spectral density of an input process $x(k)$. Let $\boldsymbol{u}_i$ be the eigenvector associated with the eigenvalue λ_i of the autocorrelation matrix $\boldsymbol{R}$ generated from $x(k)$ and the size of $\boldsymbol{R}$ be M+1. Then by definition of eigenvalue, it is given by

$$\lambda_i=\frac{\boldsymbol{u}_i^t\,\boldsymbol{R}\,\boldsymbol{u}_i}{\boldsymbol{u}_i^t\,\boldsymbol{u}_i}=\frac{\sum_{m=0}^{M}\sum_{l=0}^{M} r_x(m-l)\,u_{i,m}\,u_{i,l}}{\sum_{m=0}^{M} u_{i,m}^2} \quad , \qquad i=1\,,2\,,\ldots\,,M+1 \tag{4.15}$$

where $r_x(m-l)=E[\,x(k-m)\,x(k-l)]$ is the autocorrelation function of $x(k)$ and $u_{i,m}$ denotes the (m+1)th element of the vector $\boldsymbol{u}_i$. Making use of the fact that the autocorrelation function $r_x(m-l)$ and the power spectral density $S_x(\omega)$ of $x(k)$ forms a Fourier Transform pair, we have

$$r_x(m-l)=\frac{1}{2\,\pi}\int_{-\pi}^{\pi} S_x(\omega)\,e^{j\,\omega(m-l)}\,d\omega \tag{4.16}$$

Therefore, the entity $\boldsymbol{u}_i^t\,\boldsymbol{R}\,\boldsymbol{u}_i$ can be expressed as

$$\boldsymbol{u}_i^t \boldsymbol{R}\, \boldsymbol{u}_i = \frac{1}{2\pi}\int_{-\pi}^{\pi} S_x(\omega) \sum_{m=0}^{M} u_{i,m}\, e^{j\omega m} \sum_{l=0}^{M} u_{i,l}\, e^{-j\omega l}\, d\omega$$

$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} S_x(\omega)\, |S_{ui}(\omega)|^2\, d\omega \qquad (4.17)$$

where
$$S_{ui}(\omega) = \sum_{l=0}^{M} u_{i,l}\, e^{-j\omega l} \qquad (4.18)$$

is the Fourier Transform of the sequence $u_{i,l}$, $l = 0, 1, \ldots, M$. Using the inverse Fourier Transform of $u_{i,l}$, it can be shown that

$$\boldsymbol{u}_i^t \boldsymbol{u}_i = \frac{1}{2\pi}\int_{-\pi}^{\pi} |S_{ui}(\omega)|^2\, d\omega \qquad (7.19)$$

As a result, we can define the eigenvalue λ_i in terms of the power spectral density $S_x(\omega)$, that is

$$\lambda_i = \frac{\int_{-\pi}^{\pi} S_x(\omega)\, |S_{ui}(\omega)|^2\, d\omega}{\int_{-\pi}^{\pi} |S_{ui}(\omega)|^2\, d\omega} \qquad (4.20)$$

Let $S_{x,\min}$ and $S_{x,\max}$ be the minimum and maximum values of $S_x(\omega)$, then

$$S_{x,\min}\int_{-\pi}^{\pi} |S_{ui}(\omega)|^2\, d\omega \le \int_{-\pi}^{\pi} S_x(\omega)\, |S_{ui}(\omega)|^2\, d\omega \qquad (4.21a)$$

and
$$\int_{-\pi}^{\pi} S_x(\omega)\, |S_{ui}(\omega)|^2\, d\omega \le S_{x,\max}\int_{-\pi}^{\pi} |S_{ui}(\omega)|^2\, d\omega \qquad (4.21b)$$

From which it can be deduced that λ_i is limited by

$$S_{x,\min} \le \lambda_i \le S_{x,\max} \quad , \qquad i = 1, 2, \ldots, M+1 \qquad (4.22)$$

Consequently, the eigenvalue spread $\chi(\boldsymbol{R})$ of the input autocorrelation matrix $\boldsymbol{R}$ is bounded by

$$\chi(\boldsymbol{R}) = \frac{\lambda_{\max}}{\lambda_{\min}} \leq \frac{S_{x,\max}}{S_{x,\min}} \tag{4.23}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and minimum eigenvalue of $\boldsymbol{R}$. Equation (4.23) shows that the flatter the power spectrum of the input process, the smaller will be the eigenvalue spread.

Let the discrete Fourier transform of the impulse responses of $A(z)$ and $C(z)$ be $A(\omega)$ and $C(\omega)$ respectively. Recognize that the power spectral density of an output sequence obtained from passing $x(k)$ through $C(z)$ is given by $|C(\omega)|^2 S_x(\omega)$, the eigenvalue spread of $\boldsymbol{R}_a$, $\chi(\boldsymbol{R}_a)$, is bounded by

$$\chi(\boldsymbol{R}_a) = \frac{\lambda_{a,\max}}{\lambda_{a,\min}} = \frac{\{|C(\omega)|^2 S_x(\omega)\}_{\max}}{\{|C(\omega)|^2 S_x(\omega)\}_{\min}} \tag{4.24}$$

Similarly, the eigenvalue spread of $\boldsymbol{R}_c$, $\chi(\boldsymbol{R}_c)$, is bounded by

$$\chi(\boldsymbol{R}_c) = \frac{\lambda_{c,\max}}{\lambda_{c,\min}} = \frac{\{|A(\omega)|^2 S_x(\omega)\}_{\max}}{\{|A(\omega)|^2 S_x(\omega)\}_{\min}} \tag{4.25}$$

Suppose that the input signal $x(k)$ is white of power σ_x^2. Its power spectrum $S_x(\omega)$ is equal to a constant σ_x^2, which indicates from (4.23) that the eigenvalue spread of the input autocorrelation matrix for the transversal filter model is unity. On the other hand when $W(z)$ is split into two subunits, it can be observed from (4.24) and (4.25) that $\chi(\boldsymbol{R}_a)$ and $\chi(\boldsymbol{R}_c)$ are always greater than unity for non-zero a_i and c_i, $i = 1, 2, \ldots, N$. Since a large eigenvalue spread would lead to a slow adaptation, the convergence speed of the serial split system would therefore be degraded.

In joint process estimation with correlated input $x(k)$, it is difficult to see whether $\chi(R_a)$ or $\chi(R_c)$ is less than $\chi(R)$. This is because impulse responses of the adaptive subunits $A(z)$ and $C(z)$ depend not only on the input $x(k)$ but also on the desired response $y(k)$. For the same input, different desired signal will give rise to unequal frequency response of the two subunits and different adaptation behavior will be observed. It is expected that unless the spectrum $|C^*(\omega)|^2 S_x(\omega)$ and $|A^*(\omega)|^2 S_x(\omega)$ are both flatter than $S_x(\omega)$, splitting $W(z)$ into two components will also slow down the adaptation speed of the system. Here, $A^*(\omega)$ and $C^*(\omega)$ denote the discrete Fourier transform of the optimal parameter vectors $\boldsymbol{a}^*$ and $\boldsymbol{c}^*$.

However, if the serial split adaptive system as shown in Figure 4.1 is applied to perform linear prediction, a better performance can be obtained. In this case, the value of $y(k)$ is set to zero and the coefficient a_0 is fixed to unity. The adaptation algorithm is the same as in the case for joint process estimation except that a_0 is not adjusted and the matrix that determines the convergence speed of $\boldsymbol{a}(k)$ becomes an $N \times N$ matrix given by

$$\boldsymbol{R}_a = E\left[[x(k-1) \quad X(k-1)] \begin{bmatrix} 1 \\ c(k) \end{bmatrix} [1 \quad \boldsymbol{c}^t(k)] \begin{bmatrix} \boldsymbol{x}^t(k-1) \\ X(k-1) \end{bmatrix} \right] \tag{4.26}$$

The function of $A(z)$ and $C(z)$ are then to successively whiten the correlated input sequence $x(k)$ and therefore at steady state, the output of the prediction error filter $A(z)C(z)$ should have a uniform spectrum. This indicates that the sequence obtained by filtering $x(k)$ through either $A(z)$ or $C(z)$ will have a flatter spectrum than the input sequence $x(k)$. Hence $\chi(\boldsymbol{R}_a)$ and $\chi(\boldsymbol{R}_c)$ are both smaller than $\chi(\boldsymbol{R})$, which in turn gives rise to a faster adaptation speed. Furthermore, if $A(z)$ and $C(z)$ are set properly at the beginning such that they are close to their optimal solutions, additional enhancement in adaptation rate is anticipated.

Experiments were performed to study the convergence characteristics of the proposed serial split model and to verify the analysis as described above. In the first experiment, both the split and the non-split adaptive model were used to identify a system with transfer function

$$\begin{aligned} H(z) &= (1+2z^{-1}+3z^{-2})(0.4-0.5z^{-1}+0.6z^{-2}) \\ &= 0.4+0.3z^{-1}+0.8z^{-2}-0.3z^{-3}+1.8z^{-4} \end{aligned} \tag{4.27}$$

The input signal was white and had a power of 10 units. The desired response $y(k)$ was contaminated by a random noise with unity power. Both the input $x(k)$ and the random noise were Gaussian distributed. The step size was chosen to maintain the same misadjustment for the two configurations. Figure 4.2 compares the learning characteristics of the two adaptive systems which was obtained by 500 ensemble runs. It can be observed that the non-split model performs better than the split model. This confirms our finding because the eigenvalue spreads for $\boldsymbol{R}_a$ and $\boldsymbol{R}_c$ in the split model are larger than that of $\boldsymbol{R}$ in the non-split model.

For linear prediction, two tests with different inputs were performed and the results were the average of 500 independent trials. In the first case, the input signal was generated recursively by the formula

$$\begin{aligned} x(k) &= 2.3588x(k-1)-2.6970x(k-2)+1.6456x(k-3) \\ &\quad -0.5184x(k-4)+\beta(k) \end{aligned} \tag{4.28}$$

where $\beta(k)$ is a zero mean random sequence of unity power with Gaussian distribution. The pole pairs of $x(k)$ are $0.9\angle\pm 30°$ and $0.8\angle\pm 60°$. From (4.28), the optimal whitening filter is given by

$$\begin{aligned} W^*(z) &= 1-2.3588z^{-1}+2.6970z^{-2}-1.6456z^{-3}+0.5184z^{-4} \\ &= (1-1.5588z^{-1}+0.81z^{-2})(1-0.8z^{-1}+0.64z^{-2}) \end{aligned} \tag{4.29}$$

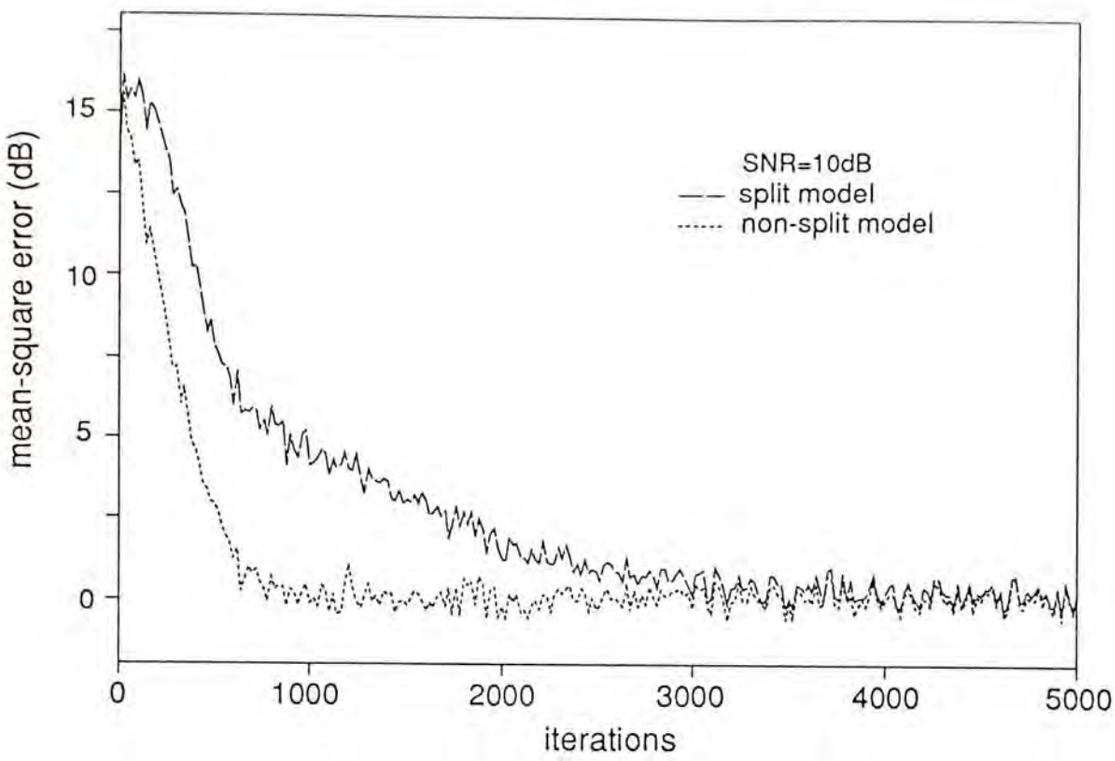


Fig. 4.2 Comparison of convergence rate with white noise input

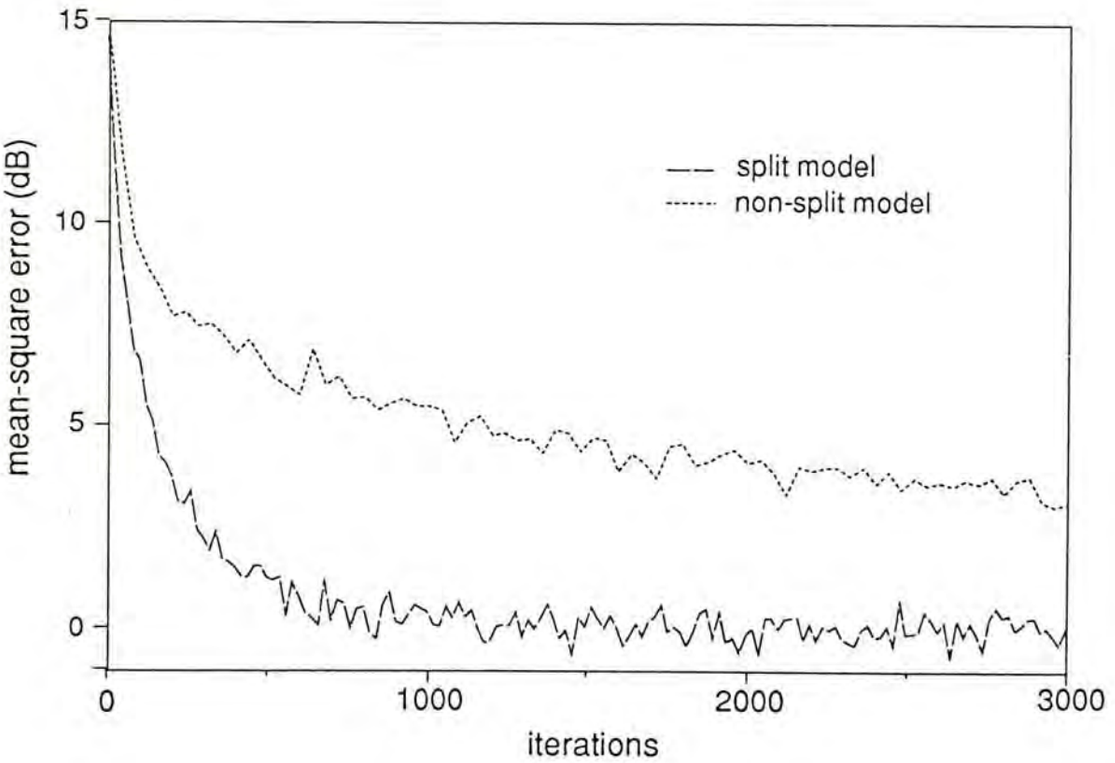


Fig. 4.3 Comparison of convergence rate with input sequence having a large eigenvalue spread

The eigenvalue spread of the input sequence was found to be 6432. On the other hand, the eigenvalue spread of R_a and R_c at steady state was found to be 13.5 and 2.91 respectively. This shows that the split system can gradually decrease the eigenvalue spreads of the inputs of the two filters $A(z)$ and $C(z)$. A substantial enhancement in adaptation speed is thus anticipated.

The learning curves for the two adaptive models are depicted in Figure 4.3. The step sizes were chosen appropriately to keep identical misadjustment in both systems. It is important to note that $A(z)$ and $C(z)$ should start with different initial values to avoid the problem of lock-up during adaptation. In our tests, the transfer function of $A(z)$ and $C(z)$ were initially set to

$$\begin{cases} A(z) = 1 - z^{-1} + z^{-2} \\ C(z) = 1 + z^{-1} + z^{-2} \end{cases} \tag{4.30}$$

respectively, which corresponds to zeros at $1\angle\pm 60°$ and $1\angle\pm 120°$. Whilst for transversal model, all the coefficients were set to zero at the beginning of adaptation. From Figure 4.3, it can be seen that the split adaptive system converges much faster than its non-split counterpart. The new system takes about 700 iterations to reach a MSE of value 1dB but the conventional model needs an exceedingly long time to converge. This verifies our theoretical arguments.

In the conventional non-split adaptive linear prediction system, it is understood that the convergence characteristics will only depend on the eigenvalues of the input correlation matrix. However, in the split model, the convergence properties will also be affected by the choice of initial values for the filter parameters. This phenomenon is illustrated in Figure 4.4 which shows two learning curves for the split adaptive system with different initial conditions. The curve with slower convergence speed has the same initial setting given by (4.30) while the other is obtained by first assigning $A(z)$ and $C(z)$ as follow,

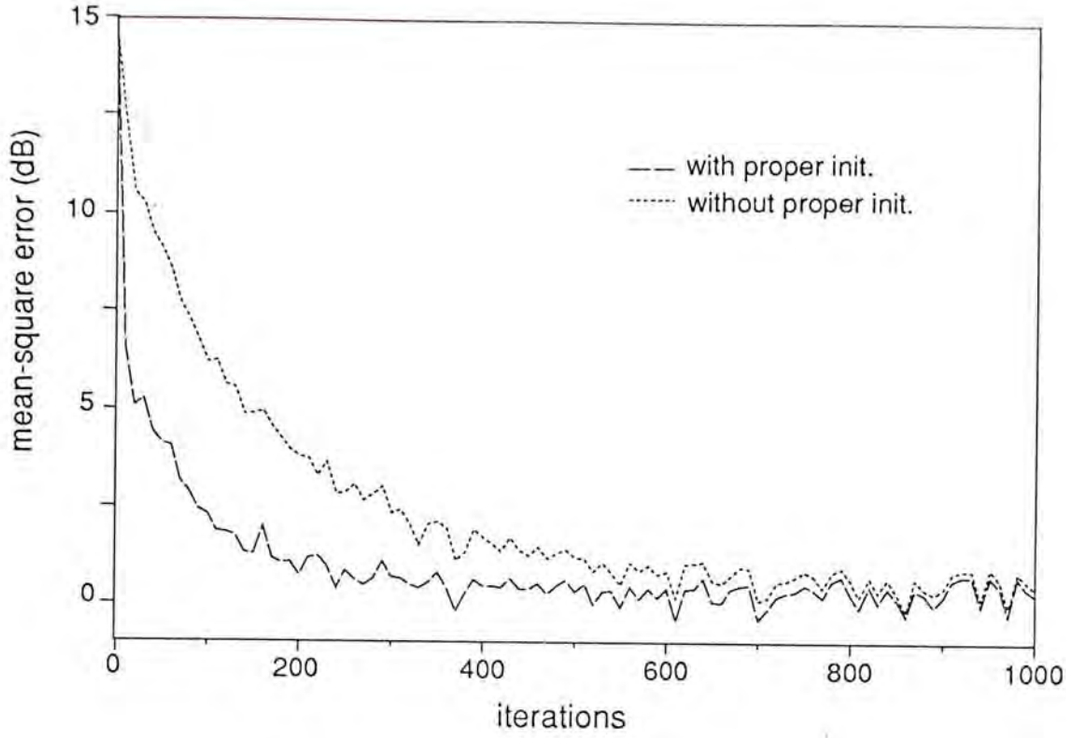


Fig. 4.4 Adaptation performance of a split system having different initial conditions

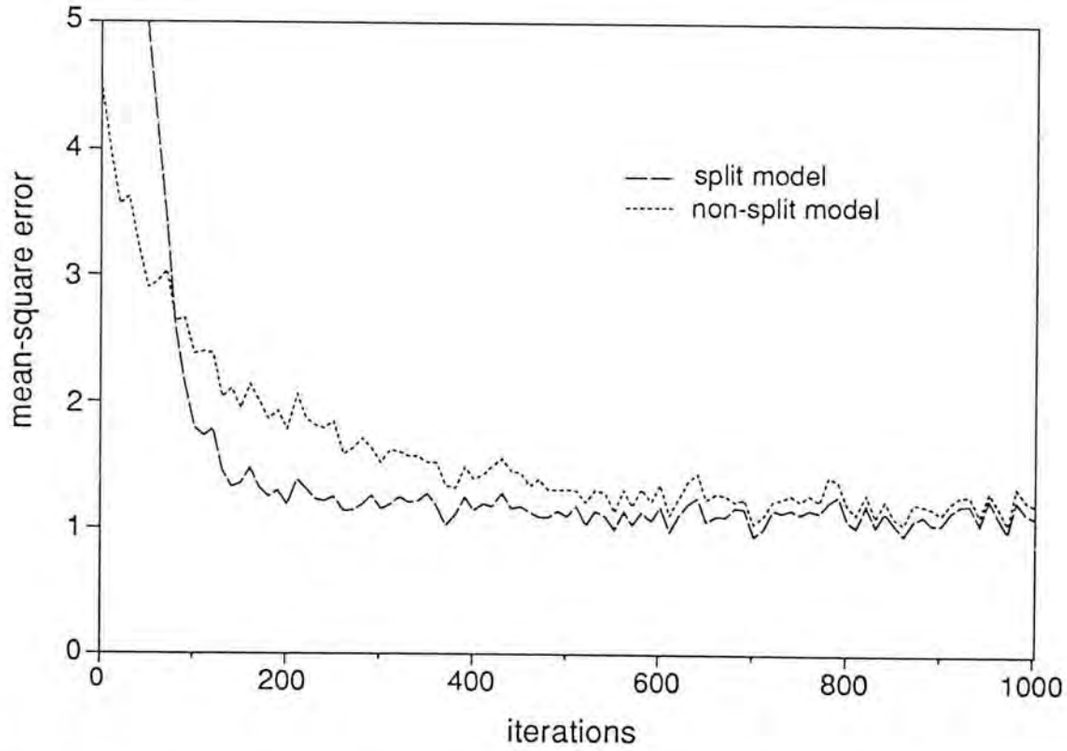


Fig. 4.5 Comparison of convergence rate with input sequence having a small eigenvalue spread

$$\begin{cases} A(z) = 1 - 1.131\, z^{-1} + 0.64\, z^{-2} \\ C(z) = 1 - 0\, z^{-1} + 0.64\, z^{-2} \end{cases}, \tag{4.31}$$

with the corresponding zeros at $0.8 \angle \pm 45°$ and $0.8 \angle \pm 90°$, which are much closer to the actual pole locations of $x(k)$. It is intriguing to note that if a priori knowledge of the optimal solution is available, proper initialization of the split filter can further speed up the adaptation rate. Therefore, serial split predictor is especially useful for tracking nonstationary signals because in such cases the optimal solution usually deviates very little from the current filter parameter values.

In the second experiment, the input was chosen to be

$$\begin{aligned} x(k) =& 0.8914\, x(k-1) - 0.09688\, x(k-2) + 0.4345\, x(k-3) \\ & - 0.5184\, x(k-4) + \beta(k) \end{aligned} \tag{4.32}$$

which had pole locations at $0.9 \angle \pm 20°$ and $0.8 \angle \pm 120°$. The corresponding optimal predictor is of the form

$$\begin{aligned} W^*(z) =& (1 - 0.8914\, z^{-1} + 0.09688\, z^{-2} - 0.4345\, z^{-3} + 0.5184\, z^{-4}) \\ =& (1 - 1.6914\, z^{-1} + 0.81\, z^{-2})(1 + 0.8\, z^{-1} + 0.64\, z^{-2}) \end{aligned} \tag{4.33}$$

The eigenvalue spread of $\boldsymbol{R}$ was equal to 33.1 and that of $\boldsymbol{R}_a$ and $\boldsymbol{R}_c$ in steady state were 29.5 and 2.88 respectively. Because of the small difference in eigenvalue spreads for the two models, the split configuration can only provide marginal improvement. The learning curves for the two configurations were shown in Figure 4.5. The step sizes were again chosen to maintain identical excess MSE in both systems and $A(z)$ and $C(z)$ were initially set to the condition as shown in (4.30). It is clear from Figure 4.5 that the split model still has a better performance although the improvement is not very remarkable.

In conclusion, serial splitting of an adaptive transversal FIR filter is, in general, not appropriate for improving adaptation speed for joint process estimation. However, for linear prediction, the split adaptive filter model can provide a significant increase in adaptation speed when the input signal is highly correlated which usually has a larger eigenvalue spread. Typical applications of this approach including speech analysis and time delay estimation which will be discussed later.

4.2 TIME DELAY ESTIMATION WITH A SERIAL SPLIT ADAPTIVE FILTER

In this section, we shall explore the possibility of applying the split idea to adaptive time delay estimation (TDE). The objective is to design a new filter structure for the time shifter in TDE that is capable of providing convergence speed-up. In the new model, the time shift filter is also represented by two adaptive subunits. However, unlike the serial split configuration, the two adaptive filters are separately implemented, one in the upper channel while the other in the lower channel. They are adapted by minimizing the output MSE to achieve the purpose of optimization. When certain restrictions is imposed between the two subunits, it is found that a two fold increase in adaptation speed can be achieved. We shall first give a brief review on adaptive TDE. The new model is then introduced and a detailed analysis and comparison of the system performance will be followed.

4.2.1 ADAPTIVE TDE

It is often necessary to determine the time delay D, which is not necessarily an integer, from two sequences

$$x(k) = s(k) + n_1(k) \tag{4.34a}$$

and $$y(k) = s(k-D) + n_2(k) \tag{4.34b}$$

where the signal $s(k)$, whose delayed version is $s(k-D)$, and $n_1(k)$ and $n_2(k)$ are random processes uncorrelated with each other. The most common application of time delay estimation is in passive sonar [1]-[2] for bearing estimation. As depicted in Figure 4.6, $s(k)$ is the signal radiating from a source. This signal is received at one sensor whose output is $x(k)$. Another sensor at a distance l apart receives $s(k-D)$, giving an output $y(k)$. The additive noises are $n_1(k)$ and $n_2(k)$ and they are assumed to have the same power. Knowing the distance between sensors and D, it is then possible to find the bearing of the source with respect to the sensors. From trigonometry, the angle θ can be computed from

$$\theta = \cos^{-1}\left(\frac{C\,D}{l}\right) \tag{4.35}$$

where C is the known signal propagation speed. When three or more sensors are present, the relative differences between the arrival times (time delays) of $s(k)$ at those sensors will give rise to three bearing lines, whose interception determines the source position [2]. Other application examples of time delay estimation are in speed measurement [3] and geophysics [4].

A basic approach to determine D from the two sensor measurements is to cross-correlate $x(k)$ and $y(k)$ and the time lag at which the cross-correlation function peaks will be an delay estimate [5]-[7]. In the passive sonar case, however, the source is generally moving so that the time delay parameter is also time-varying and adaptive techniques are needed for its estimation [8]-[11].

Assuming unity sampling interval for simplicity but without loss of generality. Moreover, let $\mathcal{F}\{\cdot\}$ and $\mathcal{F}^{-1}\{\cdot\}$ be , respectively, the Fourier transform and its inverse of $\{\cdot\}$. Then

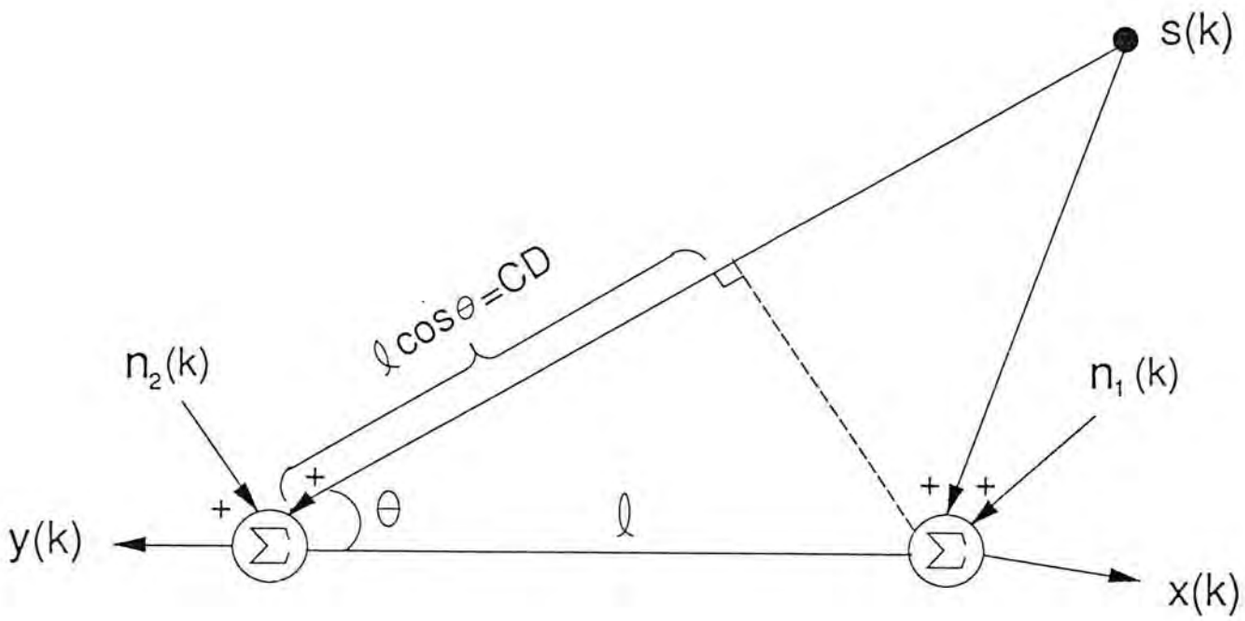


Fig. 4.6 Bearing estimation from time delay measurements

$$\mathcal{F}\{ s(k-D)\} = e^{-j\omega D}\, \mathcal{F}\{ s(k)\} \tag{4.36}$$

Using the convolution theorem, we have

$$s(k-D) = \mathcal{F}^{-1}\{ e^{-j\omega D}\} * s(k) \tag{4.37}$$

where the symbol * denotes the convolution operation. In the processing of discrete time signals, all signals are first low-pass filtered before sampling. Hence

$$\mathcal{F}^{-1}\{ e^{-j\omega D}\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-j\omega D}\, e^{j\omega k}\, d\omega = sinc(k-D) \tag{4.38}$$

where $sinc(\cdot)$ is an even function defined as

$$sinc(\cdot) = \frac{\sin(\pi \cdot)}{(\pi \cdot)} \tag{4.39}$$

From (4.37) and (4.38), we obtain

$$s(k-D) = \sum_{i=-\infty}^{\infty} sinc(i-D)\, s(k-i) \tag{4.40}$$

Equation (4.40) shows that by passing $s(k)$ through an infinite order filter whose coefficients have values $sinc(i-D)$, the output will be $s(k-D)$. In practice, the summation range is limited to some reasonable number so that an approximation of (4.40) is

$$s(k-D) \approx \sum_{i=-P}^{P} h_i\, s(k-i) = \sum_{i=-P}^{P} sinc(i-D)\, s(k-i) \tag{4.41}$$

The approximation error can be calculated for a given filter order and D [10]. In the following, it is assumed that the value P relative to D is chosen to be large enough such that the error due to truncation of filter order can be ignored. The delay estimation problem is now transformed to a parameter estimation problem of

the filter coefficients. Once the filter coefficients h_i are determined, an estimate of delay can be obtained by identifying the location at which the maximum value of the function $f(t)$ interpolated from filter coefficients occurs, where [8]

$$f(t) = \sum_{i=-P}^{P} h_i \, sinc(t-i) \tag{4.42}$$

For time-varying D, adaptive algorithms are incorporated to adjust the filter parameters. In particular, when the LMS adaptation algorithm is used, this parametric adaptive time delay estimation method is termed as LMSTDE [8]-[9], [12].

Figure 4.7 shows the LMSTDE system. The two channel inputs are sensor measurements $x(k)$ and $y(k)$. $W(z)$ is an adaptive filter whose task is to insert an appropriate time shift to $x(k)$. Let the filter parameter vector and the input vector be

$$\boldsymbol{w} = [\, w_{-P} \quad w_{-P+1} \quad \cdots \quad w_0 \quad w_1 \quad \cdots \quad w_P \,]^t \tag{4.43}$$

and

$$\boldsymbol{x}(k) = [\, x(k+P) \quad x(k+P-1) \quad \cdots \quad x(k) \quad \cdots \quad x(k-P) \,]^t \;, \tag{4.44}$$

Then the output error, $e(k)$, can be expressed as

$$e(k) = y(k) - \boldsymbol{w}^t \boldsymbol{x}(k) \tag{4.45}$$

where $(2P+1)$ is the length of the filter. The filter weights are adjusted by minimizing the MSE, $E[\, e^2(k)]$, according to Widrow's LMS algorithm as follow,

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + 2\,\mu_w \, e(k)\, \boldsymbol{x}(k) \tag{4.46}$$

where μ_w is the step size for adjusting $\boldsymbol{w}(k)$. Since $E[\, \boldsymbol{w}(k+1)] = E[\, \boldsymbol{w}(k)]$ at equilibrium, the steady state solution $\boldsymbol{w}^*$ can be obtained from (4.46) by solving $E[\, e(k)\, \boldsymbol{x}(k)] = 0$. Substituting (4.45), we find

$$\boldsymbol{w}^* = \boldsymbol{R}^{-1} \boldsymbol{g} \tag{4.47}$$

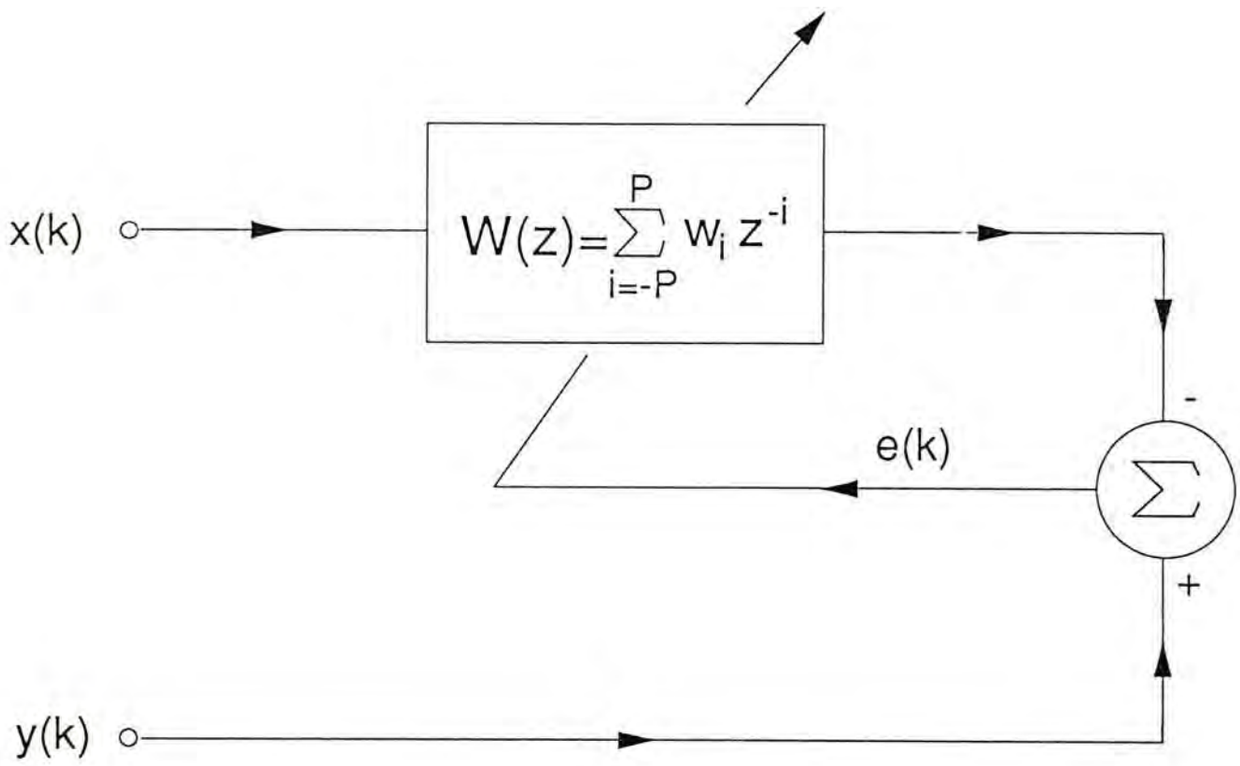


Fig. 4.7 The LMSTDE model

where $\boldsymbol{R}=E[\,x(k)\,x^t(k)]$ is the autocorrelation matrix of the input vector and $g=E[\,y(k)\,x(k)]$ is the cross-correlation vector between the input vector $x(k)$ and the desired response $y(k)$. Here, we only consider the situation where the signal to noise ratios in the two sensor measurements are high enough such that $x(k)$ and $y(k)$ are roughly equal to $s(k)$ and $s(k-D)$. Notice that $s(k)$ and $n_1(k)$ are random processes and mutually uncorrelated, $\boldsymbol{R}$ is equal to

$$\boldsymbol{R}=\sigma_x^2\,\boldsymbol{I}=(\sigma_s^2+\sigma_n^2)\,\boldsymbol{I}\approx\sigma_s^2\,\boldsymbol{I} \tag{4.48}$$

where $\boldsymbol{I}$ denotes an identity matrix of size (2P+1), σ_x^2 represents the power of $x(k)$ and σ_s^2 and σ_n^2 are, respectively, the signal and noise power. Using (4.40), the cross-correlation vector g is given by

$$g=\sigma_s^2\,[\,sinc(-N-D)\quad sinc(-N+1-D)\quad\ldots\quad sinc(N-D)]^t \tag{4.49}$$

Hence, the optimal weight vector w^* is equal to

$$\begin{aligned} w^* &= \frac{\sigma_s^2}{\sigma_s^2+\sigma_n^2}\,[\,sinc(-N-D)\quad sinc(-N+1-D)\quad\ldots\quad sinc(N-D)]^t \\ &\approx [\,sinc(-N-D)\quad sinc(-N+1-D)\quad\ldots\quad sinc(N-D)]^t \end{aligned} \tag{4.50}$$

Using the results obtained from the LMS adaptation algorithm described in chapter 2, it can be derived from (4.46) that the time constant τ_w of $E[\,w(k)]$ is equal to

$$\tau_w=\frac{1}{2\,\mu_w\,\sigma_x^2}\approx\frac{1}{2\,\mu_w\,\sigma_s^2}\quad,\qquad 0<\mu_w<\frac{1}{\sigma_x^2} \tag{4.51}$$

and the misadjustment M_w in the traditional LMSTDE system is given by

$$M_w=\mu_w\,tr(\boldsymbol{R})=\mu_w\,(2P+1)\,(\sigma_s^2+\sigma_n^2) \tag{4.52}$$

Since delay estimate, $\hat{D}$, is extracted from the filter weights by interpolation, the learning properties of $\hat{D}(k)$ will be characterized by the time constant as specified in (4.51).

The LMSTDE method is a simple but effective means for the implementation of the Roth processor [2]. It has the potential advantages that a priori knowledge about the signal statistics is not required and it is capable of tracking time-varying delay efficiently.

4.2.2 SPLIT FILTER APPROACH TO ADAPTIVE TDE

Equation (4.36) can be rewritten as

$$\mathcal{F}\{s(k-D)\} = \mathcal{F}\{s(k)\}\, e^{-j\omega(D-d)}\, e^{-j\omega d} \tag{4.53}$$

where d is an arbitrary value. From (4.53) and using the convolution theorem, we can deduce that

$$s(k-D) \approx \sum_{i=-P}^{P} \sum_{j=-P}^{P} sinc(i-D+d)\, sinc(j-d)\, s(k-i-j) \tag{4.54}$$

Thus, $s(k-D)$ can be generated by passing $s(k)$ through two FIR filters whose coefficients are, respectively, $sinc(i-D+d)$ and $sinc(j-d)$. Compared with (4.41), the single filter to provide time shift is now split into two filters connected in series. The adaptive TDE model derived from (4.54) has the same configuration shown in Figure 4.1, in which there are two filters, $A(z)$ and $C(z)$, instead of a single filter $W(z)$ in the upper channel. They are made adaptive to minimize the MSE with the output error given by

$$e(k) = y(k) - \sum_{i=-P}^{P} \sum_{j=-P}^{P} a_i\, c_j\, s(k-i-j) \tag{4.55}$$

Upon reaching steady state, it is expected that the optimal solution will be

$$\begin{cases} a_i^* = sinc(i - D + d) \\ c_i^* = sinc(i - d) \end{cases} \tag{4.56}$$

Interpolation procedure is used to determine the time shift inserted by $A(z)$ and $C(z)$. An estimate of delay will be the total time shifts provided by them.

As noted earlier when the input process is random, splitting $W(z)$ into two filters connected in cascade will result in degradation in performance, which is undesirable. Now suppose we multiply both sides of (4.53) by $e^{j\omega d}$, we have

$$\mathcal{F}\{ s(k-D)\}\, e^{j\omega d} = \mathcal{F}\{ s(k)\}\, e^{-j\omega(D-d)} \tag{4.57}$$

Again, using convolution theorem and truncating filter order yields

$$\sum_{i=-P}^{P} sinc(i+d)\, s(k-D-i) = \sum_{i=-P}^{P} sinc(i-D+d)\, s(k-i) \tag{4.58}$$

which implies that the filter $C(z)$ in Figure 4.1 can be placed in the lower channel with $1/C(z)$ performing an insertion of d advance to the desired response $y(k)$. Let $1/C(z) = B(z)$, the TDE model corresponds to (4.58) is illustrated in Figure 4.8. The error signal now becomes

$$e(k) = \sum_{i=-P}^{P} b_i\, y(k-i) - \sum_{i=-P}^{P} a_i\, x(k-i) \tag{4.59}$$

where b_i are the coefficients of $B(z)$. When the output MSE is minimized, the filter weights are expected to converge to

$$\begin{cases} a_i^* = sinc(i - D + d) \\ b_i^* = sinc(i + d) \end{cases} \tag{4.60}$$

Unlike the previous case, an estimate of delay is the difference of the time shifts provided by $A(z)$ and $B(z)$. It is obvious to see from (4.59) that the output error is linear with respect to a_i and b_i. Hence, global convergence is guaranteed.

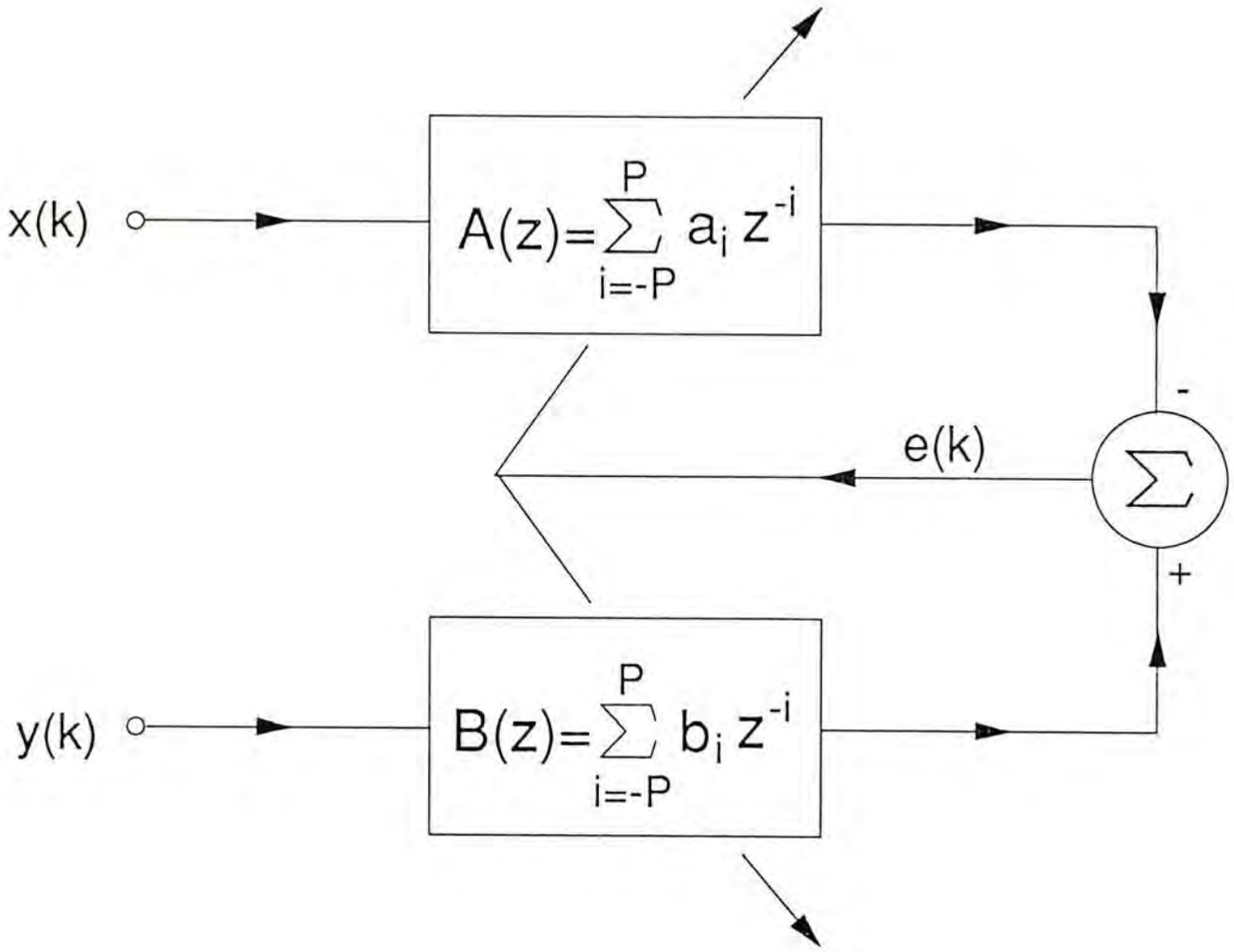


Fig. 4.8 A modified LMSTDE model

There are two drawbacks in the model depicted in Figure 4.8. First of all, we need to adapt two filters instead of one, which implies almost a double in computational load. Second, when two filters are used to represent the single time shift filter $W(z)$, the degree of freedom will be increased which provides many possible optimal solutions. This in fact can be observed from (4.60) by noting that d is a quantity that can vary. To resolve these problems, we impose the following simple restriction to the filter weights,

$$b_i = a_{-i} \quad , \qquad i = -P \, , \ldots , P \tag{4.61}$$

Now, as the weights of $B(z)$ can be easily mapped from that of $A(z)$, adaptation of one filter, say, $A(z)$ is sufficient. In addition, equation (4.61) implies that $b_i^* = a_{-i}^*$. On using (4.60), we can immediately derive that the optimal solution which are given by

$$\begin{cases} d = \dfrac{D}{2} \\ a_i^* = sinc\left(i - \dfrac{D}{2}\right) \\ b_i^* = sinc\left(i + \dfrac{D}{2}\right) \end{cases} \tag{4.62}$$

It can be seen from (4.62) that $A(z)$ and $B(z)$ provide essentially the same amount of time shift, but one is a time delay in the upper channel while the other is a time advance in the lower channel when equilibrium is reached.

The TDE model depicted in Figure 4.8 together with the restriction specified in (4.61) constitutes the basic structure for the proposed new TDE model. The advantages of this novel configuration for TDE include fast adaptation speed with ease of hardware implementation.

4.2.3 ANALYSIS OF THE NEW TDE SYSTEM

Preliminary study of the proposed system can be found in [13]-[14]. Here, we provide a more detail treatment in the analysis of the new adaptive TDE structure. Referring to Figure 4.8, the output of the upper channel can be expressed as

$$z_2(k) = \sum_{i=-P}^{P} a_i\, x(k-i) = \sum_{i=-P}^{P} a_i \{ s(k-i) + n_1(k-i) \} \tag{4.63}$$

Imposing the condition (4.61) on $B(z)$, the output of the lower channel is given by

$$\begin{aligned} z_1(k) &= \sum_{i=-P}^{P} a_{-i}\, y(k-i) = \sum_{i=-P}^{P} a_i\, y(k+i) \\ &= \sum_{i=-P}^{P} a_i \{ s(k-D+i) + n_2(k+i) \} \end{aligned} \tag{4.64}$$

Hence the output error $e(k)$ is equal to

$$\begin{aligned} e(k) &= z_1(k) - z_2(k) \\ &= \sum_{i=-P}^{P} a_i \{ s(k-D+i) - s(k-i) + n_2(k+i) - n_1(k-i) \} \end{aligned} \tag{4.65}$$

Define the following vectors as

$$\boldsymbol{a} = [\, a_{-P} \quad a_{-P+1} \quad \ldots \quad a_0 \quad a_1 \quad \ldots \quad a_P \,]^t \tag{4.66}$$

$$\begin{aligned} \boldsymbol{s}_D(k) = [\, & s(k-D-P) \quad s(k-D-P+1) \quad \ldots \\ & s(k-D) \quad \ldots \quad s(k-D+P) \,]^t \end{aligned} \tag{4.67}$$

$$\boldsymbol{s}(k) = [\, s(k+P) \quad s(k+P-1) \quad \ldots \quad s(k) \quad \ldots \quad s(k-P) \,]^t \tag{4.68}$$

$$\boldsymbol{n}_1(k) = [\, n_1(k+P) \quad n_1(k+P-1) \quad \ldots \quad n_1(k) \quad \ldots \quad n_1(k-P) \,]^t \tag{4.69}$$

$$\boldsymbol{n}_2(k) = [\, n_2(k-P) \quad n_2(k-P+1) \quad \ldots \quad n_2(k) \quad \ldots \quad n_2(k+P) \,]^t \tag{4.70}$$

$$\boldsymbol{y}(k) = \boldsymbol{s}_D(k) + \boldsymbol{n}_2(k) \tag{4.71}$$

$$\boldsymbol{x}(k) = s(k) + \boldsymbol{n}_1(k) \tag{4.72}$$

we then have

$$e^2(k) = \boldsymbol{a}^t \{ \boldsymbol{y}(k) - \boldsymbol{x}(k) \} \{ \boldsymbol{y}(k) - \boldsymbol{x}(k) \}' \boldsymbol{a} \tag{4.73}$$

When the input signal-to-noise ratio (SNR) is high, $e^2(k)$ can be approximated by

$$e^2(k) = \boldsymbol{a}^t \{ s_D(k) - s(k) \} \{ s_D(k) - s(k) \}' \boldsymbol{a} \tag{4.74}$$

Suppose $s(t)$ is a stationary bandlimited white noise process [15] of power σ_s^2, then using (4.40) we obtain

$$\begin{aligned} E[s(k)\, s(k-d)] &= \sum_{i=-\infty}^{\infty} sinc(i-d)\, E[s(k)\, s(k-i)] \\ &= \sigma_s^2\, sinc(d) = r_s(d) \end{aligned} \tag{4.75}$$

Now,

$$\begin{aligned} E[s_D(k)\, s_D^t(k)] &= E[s(k)\, s^t(k)] = \boldsymbol{R} \\ &= \begin{bmatrix} r_s(0) & 0 & \dots & 0 \\ 0 & r_s(0) & \dots & 0 \\ . & . & \dots & . \\ 0 & 0 & \dots & r_s(0) \end{bmatrix} = \sigma_s^2 \boldsymbol{I} \end{aligned} \tag{4.76}$$

and

$$\begin{aligned} E[s_D(k)\, s^t(k)] &= E[s(k)\, s_D^t(k)] = \boldsymbol{R}_D \\ &= \begin{bmatrix} r_s(D+2P) & r_s(D+2P-1) & \dots & r_s(D) \\ r_s(D+2P-1) & r_s(D+2P-2) & \dots & r_s(D-1) \\ . & . & \dots & . \\ r_s(D) & r_s(D-1) & \dots & r_s(D-2P) \end{bmatrix} \end{aligned} \tag{4.77}$$

Hence, the expectation of (4.74) can be expressed as

$$\xi = E[e^2(k)] = 2\, \boldsymbol{a}^t (\boldsymbol{R} - \boldsymbol{R}_D)\, \boldsymbol{a} \tag{4.78}$$

4.2.3.1 LEAST-MEAN-SQUARE SOLUTION

Following [16] and differentiating (4.78) with respect to a, we found that the optimal weight vector a^* must satisfy

$$(R - R_D)\, a^* = 0 \tag{4.79}$$

It is obvious that $a^* = 0$ can satisfy (4.79) but it is not a desirable solution. On the other hand, non-trivial or non-unique solutions exist if $(R - R_D)$ is singular. Therefore, it is necessary to show that $(R - R_D)$ has at least one eigenvalue equal to zero and that

$$a_{\frac{D}{2}} = \left[sinc\left(-\frac{D}{2} - P\right) \quad \ldots \quad sinc\left(-\frac{D}{2}\right) \quad sinc\left(-\frac{D}{2} + P\right) \right]^t \tag{4.80}$$

will satisfy (4.79). The filter parameter set $a_{\frac{D}{2}}$ in this case provides a time shift of $D/2$ in the upper channel and will form the desired vector in this particular application as illustrated in (4.62). Now, equation (4.79) can be rewritten as

$$R_D\, a^* = R\, a^* = \sigma_s^2\, a^* \tag{4.81}$$

which indicates that a^* is an eigenvector of R_D corresponding to an eigenvalue of σ_s^2. To prove that $a_{\frac{D}{2}}$ in (4.80) is a possible solution of (4.81), let us consider the reconstruction of $s(t)$ from its samples $s(k)$ according to

$$s(t) = \sum_{i=-\infty}^{\infty} s(i)\, sinc(t - i) = \sum_{i=-\infty}^{\infty} s(-i)\, sinc(t + i) \tag{4.82}$$

Auto-correlating $s(t)$ with its time shifted version

$$s(t - \gamma + j) = \sum_{l=-\infty}^{\infty} s(-l)\, sinc(t - \gamma + j + l) \tag{4.83}$$

yields,

$$E[\,s(t-\gamma+j)\,s(t)\,] = \sigma_s^2\, sinc(-\gamma+j)$$

$$= \sigma_s^2 \sum_{i=-\infty}^{\infty} sinc(t-\gamma+j+i)\, sinc(t+i) \tag{4.84}$$

Let $t=-\frac{D}{2}$ and $\gamma=\frac{D}{2}$, then

$$\sum_{i=-\infty}^{\infty} sinc(D-j-i)\, sinc\left(-\frac{D}{2}+i\right) = sinc\left(-\frac{D}{2}+j\right) \tag{4.85}$$

It is now seen that for $j=-P, -P+1, \ldots, P$, and with the summation of i limited from $-P$ to P, (4.85) is equivalent to (4.81), hence the proof. The error introduced by the finite summation terms decreases with increasing P.

In a similar manner, it can be shown that $\boldsymbol{R}_D$ has an eigenvalue of $-\sigma_s^2$. Let $t=-D+\gamma$ and $\gamma \neq \frac{D}{2}$. Define

$$\boldsymbol{a}_\gamma = [\,sinc(-\gamma-P) \quad \ldots \quad sinc(-\gamma) \quad \ldots \quad sinc(-\gamma+P)\,]^t \tag{4.86}$$

and $$\boldsymbol{a}_{D-\gamma} = [\,sinc(-D+\gamma-P) \quad \ldots \quad sinc(-D+\gamma) \quad \ldots \quad sinc(-D+\gamma+P)\,]^t \tag{4.87}$$

Then, from (4.84) and for $j=-P, \ldots, P$ with i limited from $-P$ to P, it follows that

$$\boldsymbol{R}_D\, \boldsymbol{a}_{D-\gamma} = \sigma_s^2\, \boldsymbol{a}_\gamma \tag{4.88a}$$

and $$\boldsymbol{R}_D\, \boldsymbol{a}_\gamma = \sigma_s^2\, \boldsymbol{a}_{D-\gamma} \tag{4.88b}$$

Hence

$$\boldsymbol{R}_D\,(\boldsymbol{a}_\gamma - \boldsymbol{a}_{D-\gamma}) = -\sigma_s^2\,(\boldsymbol{a}_\gamma - \boldsymbol{a}_{D-\gamma}) \tag{4.89}$$

and this indicates that $\boldsymbol{a}_\gamma - \boldsymbol{a}_{D-\gamma}$ is an eigenvector of $\boldsymbol{R}_D$ with eigenvalue equals $-\sigma_s^2$.

It is also noted that the vector $a_{\frac{D}{2}}$ is not the only solution for (4.81). The addition of (4.88*a*) and (4.88*b*) gives

$$\boldsymbol{R}_D(a_\gamma + a_{D-\gamma}) = \sigma_s^2(a_\gamma + a_{D-\gamma}) \tag{4.90}$$

which immediately confirms that $a_\gamma + a_{D-\gamma}$ is also a solution of (4.81). With $t = -D + \gamma$, equation (4.84) is valid for finite i if γ is small and it is required to determine the range of γ such that (4.88) is satisfied. Numerical calculations show that for

$$\begin{cases} -P + 2 + int(D) < \gamma < P - 1.5 \ , & D > 0 \\ -P + 1.5 < \gamma < P - 2 + int(D) \ , & D < 0 \end{cases} \tag{4.91}$$

the error for the largest element of a_γ and $a_{D-\gamma}$ is always less than 6%, where $int(\cdot)$ represents the integral part of $\cdot$. The larger the γ in comparison with P, the greater is the error and equation (4.88) will then become invalid.

When γ is chosen properly, we can obtain the subset of orthogonal eigenvectors for eigenvalues σ_s^2 and $-\sigma_s^2$ of $\boldsymbol{R}_D$. When $a_{\frac{D}{2}}$ is independent of $a_\gamma + a_{D-\gamma}$, we have

$$\begin{aligned} a_{\frac{D}{2}}^t(a_\gamma + a_{D-\gamma}) &= \sum_{i=-P}^{P} sinc\left(i - \frac{D}{2}\right) sinc(\gamma - i) \\ &\quad + \sum_{i=-P}^{P} sinc\left(i - \frac{D}{2}\right) sinc(D - \gamma - i) \\ &= 2\, sinc\left(\gamma - \frac{D}{2}\right) = 0 \end{aligned} \tag{4.92}$$

which implies

$$\gamma = \frac{D}{2} + i \quad , \qquad i = -L, \ldots, -1, 1, \ldots, L \tag{4.93}$$

where the value L is chosen to be $L = P - 2 - int((|D|+1)/2)$ such that (4.91) can be satisfied. In addition, when γ_1 and γ_2 are chosen according to (4.93), the condition for any two vectors $a_{\gamma 1} + a_{D-\gamma 1}$ and $a_{\gamma 2} + a_{D-\gamma 2}$ to be independent is

$$(a_{\gamma 1} + a_{D-\gamma 1})^t (a_{\gamma 2} + a_{D-\gamma 2}) = 0 \tag{4.94}$$

which implies

$$\begin{cases} \gamma 1 \neq \gamma 2 \\ \gamma 1 + \gamma 2 \neq D \end{cases} \tag{4.95}$$

From (4.93) and (4.95), we can conclude that the set of vectors

$$\left\{ (a_\gamma + a_{D-\gamma}) \mid \gamma = \frac{D}{2} + i \, , i = 0, \ldots, L \right\} \tag{4.96}$$

will form a subset of orthogonal eigenvectors of eigenvalue σ_s^2 of $\boldsymbol{R}_D$. Following the same argument, it can also be illustrated that

$$\left\{ (a_\gamma - a_{D-\gamma}) \mid \gamma = \frac{D}{2} + i \, , i = 1, \ldots, L \right\} \tag{4.97}$$

is the subset of orthogonal eigenvectors of eigenvalue $-\sigma_s^2$ of $\boldsymbol{R}_D$. Thus, $\boldsymbol{R}_D$ has at least $L+1$ eigenvalues of σ_s^2 and L eigenvalues of $-\sigma_s^2$.

It is well known that linear combination of the orthogonal eigenvectors corresponding to eigenvalue σ_s^2 is also an eigenvector of eigenvalue σ_s^2. Therefore, the solution of equation (4.81), in general form, can be expressed as

$$a^* = \alpha_0 \left(a_{\frac{D}{2}} \right) + \sum_{i=1}^{L} \alpha_i \left(a_{\frac{D}{2}+i} + a_{\frac{D}{2}-i} \right) \tag{4.98}$$

where α_i are any arbitrary constants. When $\alpha_i = sinc\left(\gamma - \frac{D}{2} + i\right) + sinc\left(\gamma - \frac{D}{2} - i\right)$ for $i = 0, \ldots, L$ with γ satisfying (4.91), using (4.85) and ignoring the negligible truncation error, we have

$$a^{*} = \sum_{i=-L}^{L} sinc\left(\gamma - \frac{D}{2} + i\right) a_{\frac{D}{2}-i} + \sum_{i=-L}^{L} sinc\left(\gamma - \frac{D}{2} + i\right) a_{\frac{D}{2}+i}$$

$$= a_{\gamma} + a_{D-\gamma} \tag{4.99}$$

which thus verifies $a_{\gamma} + a_{D-\gamma}$ is a possible solution of (4.81). Figure 4.9 demonstrates a possible realization for this special case. Simulation study has found that convergence of the filter weights to a particular solution is dependent on the initial conditions but convergence to the desired solution (4.80) is always obtained if $|D| < 1$. In addition, it will be shown later that it is fairly easy to guarantee convergence to the optimal weight vector $a_{\frac{D}{2}}$ by applying the *sinc* function constraint to the filters weights [17].

4.2.3.2 ADAPTATION ALGORITHM AND PERFORMANCE EVALUATION

A well known procedure in minimizing $E[e^2(k)]$ is to apply Widrow's LMS algorithm. Accordingly, from (3.73) the updating equation for the filter weight vector is

$$a(k+1) = a(k) - \mu_a \frac{\partial e^2(k)}{\partial a(k)}$$

$$= a(k) - 2\mu_a e(k)\{y(k) - x(k)\} \tag{4.100}$$

where μ_a represents the step size for adjusting $a(k)$ in controlling the rate of convergence and stability of the adaptive process. For simplicity, it is assumed that SNR in the two input channels is high enough such that during the learning period, (3.73) can be approximated by (3.74). Let the weight vector error be

$$\tilde{a}(k) = a(k) - a^{*} \tag{4.101}$$

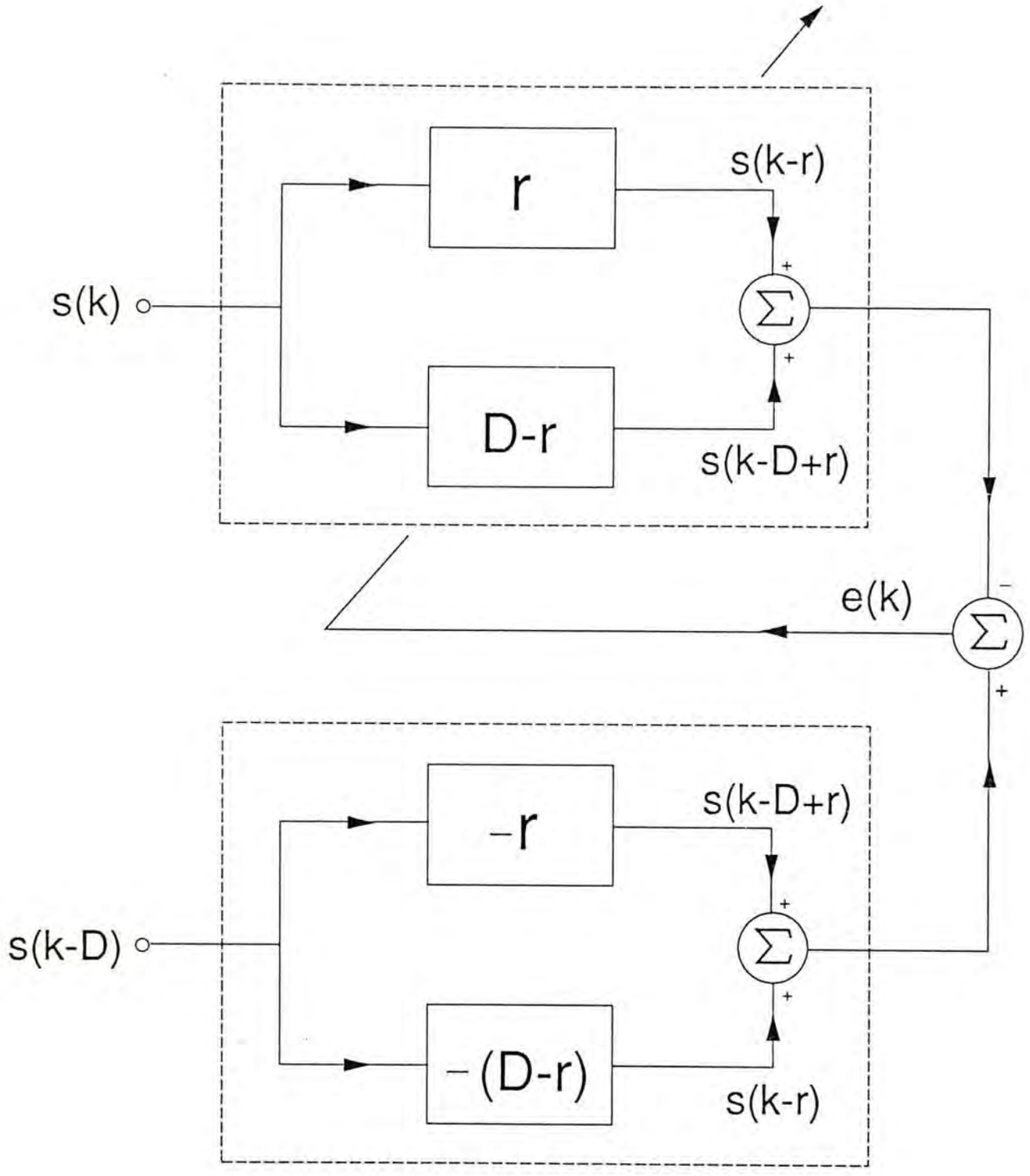


Fig. 4.9 A possible non-unique solution for the proposed TDE system

With the independence assumption that $\tilde{a}(k)$ is independent of $y(k)$ and $x(k)$, using (4.74) and then taking expectation of (4.100) gives

$$E[\tilde{a}(k+1)] = \{ \boldsymbol{I} - 4\mu_a (R - R_D) \} E[\tilde{a}(k)] \tag{4.102}$$

Since $\boldsymbol{R}_D$ is a real symmetric matrix, it can be decomposed into the form

$$\boldsymbol{R}_D = \boldsymbol{U} \Lambda_D \boldsymbol{U}^t \tag{4.103}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1 \boldsymbol{u}_2 \ldots \boldsymbol{u}_{2P+1}]$ is the orthonormal eigenvector matrix of $\boldsymbol{R}_D$. Let $\sigma_s^2 \lambda_i$, $i = 2L+2, \ldots, 2P+1$, be the remaining nondetermined eigenvalues of $\boldsymbol{R}_D$, then

$$\Lambda_D = \sigma_s^2 \, diag \left\{ \underbrace{-1, -1, \ldots, -1}_{L}, \underbrace{1, \ldots, 1}_{L+1}, \lambda_{2L+2}, \lambda_{2L+3}, \ldots, \lambda_{2P+1} \right\} \tag{4.104}$$

is a diagonal matrix that contains the eigenvalues of $\boldsymbol{R}_D$. Using the orthogonal property of $\boldsymbol{U}$ and noting from (4.89) and (4.90) that $\boldsymbol{a}_\gamma - \boldsymbol{a}_{D-\gamma}$ and $\boldsymbol{a}_\gamma + \boldsymbol{a}_{D-\gamma}$ are the respective eigenvectors of eigenvalues $-\sigma_s^2$ and σ_s^2 for the appropriate range of γ, we have

$$\boldsymbol{u}_i^t (\boldsymbol{a}_\gamma + \boldsymbol{a}_{D-\gamma}) = 0 \quad , \qquad i \in \{ j = 2L+2, \ldots, 2P+1 \mid \lambda_j \neq 1 \text{ or } -1 \} \tag{4.105a}$$

and

$$\boldsymbol{u}_i^t (\boldsymbol{a}_\gamma - \boldsymbol{a}_{D-\gamma}) = 0 \quad , \qquad i \in \{ j = 2L+2, \ldots, 2P+1 \mid \lambda_j \neq 1 \text{ or } -1 \} \tag{4.105b}$$

Hence,

$$\boldsymbol{u}_i^t \boldsymbol{a}_\gamma = 0 \quad , \qquad i \in \{ j = 2L+2, \ldots, 2P+1 \mid \lambda_j \neq 1 \text{ or } -1 \} \tag{4.106}$$

In addition, because $\boldsymbol{a}^*$ is a linear combination of eigenvectors corresponding to eigenvalue σ_s^2 of $\boldsymbol{R}_D$, it can be deduced that

$$\boldsymbol{u}_i^t \boldsymbol{a}^* = 0 \quad , \qquad i \in \{ j = 2L+2, \ldots, 2P+1 \mid \lambda_j \neq 1 \text{ or } -1 \} \tag{4.107}$$

Noting that $\boldsymbol{R}=\sigma_s^2\boldsymbol{I}$, any orthonormal matrix is an eigenvector matrix of $\boldsymbol{R}$. Thus, we have

$$\boldsymbol{R}-\boldsymbol{R}_D=\boldsymbol{U}\,\Lambda\,\boldsymbol{U}^t \tag{4.108}$$

where

$$\begin{aligned}\Lambda &=\sigma_s^2\boldsymbol{I}-\Lambda_D \\ &=\sigma_s^2\,diag\left\{\underbrace{2,\ldots,2}_{L},\underbrace{0,\ldots,0}_{L+1},1-\lambda_{2L+2},\ldots,1-\lambda_{2P+1}\right\}\end{aligned} \tag{4.109}$$

Define

$$\tilde{a}'(k)=\boldsymbol{U}^t\,\tilde{a}(k)=[\,\tilde{a}'_1(k),\ldots,\tilde{a}'_{2P+1}(k)\,]^t \tag{4.110}$$

be the transformed weight error vector. From (4.102), we have

$$E[\,\tilde{a}'(k+1)]=(\boldsymbol{I}-4\,\mu_a\,\Lambda)\,E[\,\tilde{a}'(k)] \tag{4.111a}$$

or

$$E[\,\tilde{a}'(k)]=(\boldsymbol{I}-4\,\mu_a\,\Lambda)^k\,E[\,\tilde{a}'(0)] \tag{4.111b}$$

In studying the dynamic behavior of an adaptive time delay estimator, we always have $E[\,\boldsymbol{a}(0)]=\boldsymbol{a}_{d0}$, where $d0$ is the delay estimate at time reference 0 which is much less than P. From (4.106), (4.107) and (4.110), the ith element of $E[\,\tilde{a}'(0)]$ is then equal to

$$\begin{aligned}E[\,\tilde{a}_i'(0)] &=\boldsymbol{u}_i^t\,\boldsymbol{a}_{d0}-\boldsymbol{u}_i^t\,\boldsymbol{a}^* \\ &=0\quad,\qquad i\in\{\,j=2L+2,\ldots,2P+1\,|\,\lambda_j\neq 1\text{ or }-1\,\}\end{aligned} \tag{4.112}$$

Now, it can be observed from (4.111b) that only the transformed weight errors corresponding to eigenvalue $2\sigma_s^2$ of Λ are adjusted during adaptation because those related to eigenvalue 0 of Λ are not changed and the initial transformed weight errors corresponding to other eigenvalues of Λ are all zero. Hence, the time constant for the transformed filter weights of $A(z)$ in the new system, τ_a, is given by

$$\tau_a = \frac{1}{8\,\mu_a\,\sigma_s^2} \tag{4.113}$$

Since the maximum eigenvalue of $R - R_D$ is $2\sigma_s^2$, the condition for convergence is

$$0 < \mu_a < \frac{1}{2\,\sigma_s^2} \tag{4.114}$$

We shall next investigate the steady state performance of the new system. Let the correlation matrices of $n_1(k)$ and $n_2(k)$ be $\boldsymbol{R}_{n1}$ and $\boldsymbol{R}_{n2}$ so that $\boldsymbol{R}_{n1} = \boldsymbol{R}_{n2} = \sigma_n^2 \boldsymbol{I}$. Then, in steady state, with high SNR, we can rewrite (4.100) as

$$\tilde{a}(k+1) = (\boldsymbol{I} - 2\,\mu_a \boldsymbol{R}_{xy})\,\tilde{a}(k) + \mu_a\,\eta(k) \tag{4.115}$$

where $\boldsymbol{R}_{xy} = E[\{y(k) - x(k)\}\{y(k) - x(k)\}^t] = 2(\boldsymbol{R} - \boldsymbol{R}_D) + \boldsymbol{R}_{n1} + \boldsymbol{R}_{n2}$ and $\eta(k)$ is the zero mean gradient noise vector which has a steady state covariance matrix given by [16]

$$\begin{aligned} E[\eta(k)\,\eta(k)^t] &= 4\,E[e^2(k)\{y(k) - x(k)\}\{y(k) - x(k)\}^t] \\ &= 4\,\xi_a^o\,\boldsymbol{R}_{xy} \end{aligned} \tag{4.116}$$

with ξ_a^o representing the minimum possible MSE of the new system. From (4.115), we can immediately obtain

$$\begin{aligned} cov\{a(k+1)\} = (\boldsymbol{I} - 2\,\mu_a \boldsymbol{R}_{xy})\,cov\{a(k)\}\,(\boldsymbol{I} - 2\,\mu_a \boldsymbol{R}_{xy}) \\ + 4\,\mu_a^2\,E[\eta(k)\,\eta^t(k)] \end{aligned} \tag{4.117}$$

which gives (with small μ_a)

$$cov\{a(k)\} = \mu_a\,\xi_a^o\,\boldsymbol{I} \tag{4.118}$$

where $cov\{a(k)\} = E[\tilde{a}(k)\tilde{a}^t(k)]$ is the covariance of the weight vector. Due to gradient noise, the minimum possible MSE can never be achieved. Therefore, the excess MSE (EMSE), the difference between the MSE in steady state and the minimum possible MSE, is introduced which is given by

$$\begin{aligned} \text{EMSE} &= E[a^t(k)\{y(k)-x(k)\}^t\{y(k)-x(k)\}a(k)] \\ &\quad -E[a^{*t}\{y(k)-x(k)\}^t\{y(k)-x(k)\}a^*] \end{aligned} \tag{4.119}$$

Using the independence assumption and equations (4.76), (4.77) and (4.119), we have

$$\begin{aligned} \text{EMSE} &= tr\{E[\{y(k)-x(k)\}\tilde{a}(k)\tilde{a}(k)^t\{y(k)-x(k)\}^t]\} \\ &= \mu_a \xi_a^o tr\{R_{xy}\} = \mu_a \xi_a^o 2\{(2P+1)(\sigma_s^2+\sigma_n^2) - tr\{R_D\}\} \\ &\le \mu_a \xi_a^o 2(2P+1)(\sigma_s^2+\sigma_n^2) \end{aligned} \tag{4.120}$$

Note that $0 \le tr\{R_D\} = \sigma_s^2\left\{\sum_{i=-P}^{P} sinc(D+2i)\right\} \le \sigma_s^2$ is being used in (4.120). With large P, the misadjustment of the new system, M_a, can be approximated by

$$M_a = \frac{\text{EMSE}}{\xi_a^o} = \mu_a 2(2P+1)(\sigma_s^2+\sigma_n^2) \tag{4.121}$$

4.2.4 COMPARISON WITH TRADITIONAL ADAPTIVE TDE METHOD

When the performance of two adaptive systems is to be compared, the misadjustment, a measure of the adaptive process that tracks the Wiener solution in steady state, is usually fixed with the convergence speed being contrasted. From (4.52) and (4.121), keeping the same misadjustment yields

$$\mu_a = \frac{1}{2}\mu_w \tag{4.122}$$

That means the time constant of the proposed system will be improved by a factor of 2 as indicated by (4.51) and (4.113), which in turn gives rise to a faster adaptation speed.

4.2.5 SYSTEM IMPLEMENTATION

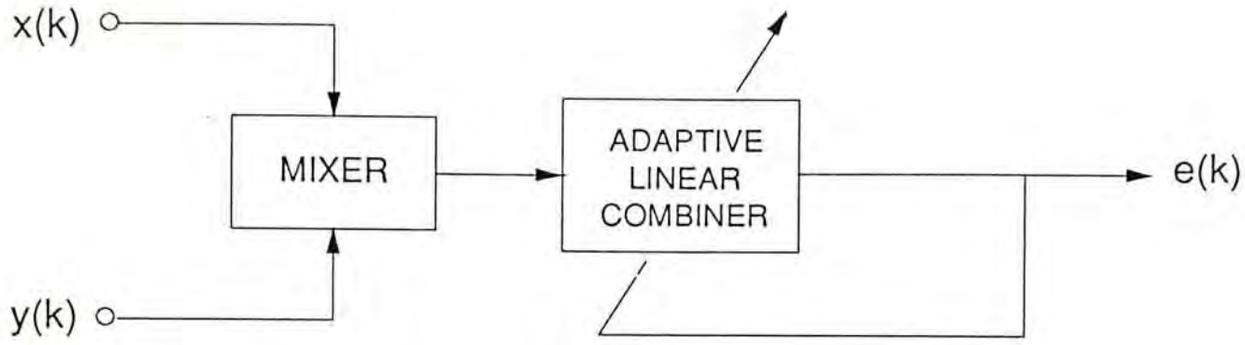
The two filters in Figure 4.8 serve as a conceptual illustration of the new configuration. The actual implementation requires, however, only a single adaptive filter. Recall from (4.65) that the output error is given by

$$e(k) = \sum_{i=-P}^{P} a_i \{ y(k+i) - x(k-i) \} \tag{4.123}$$

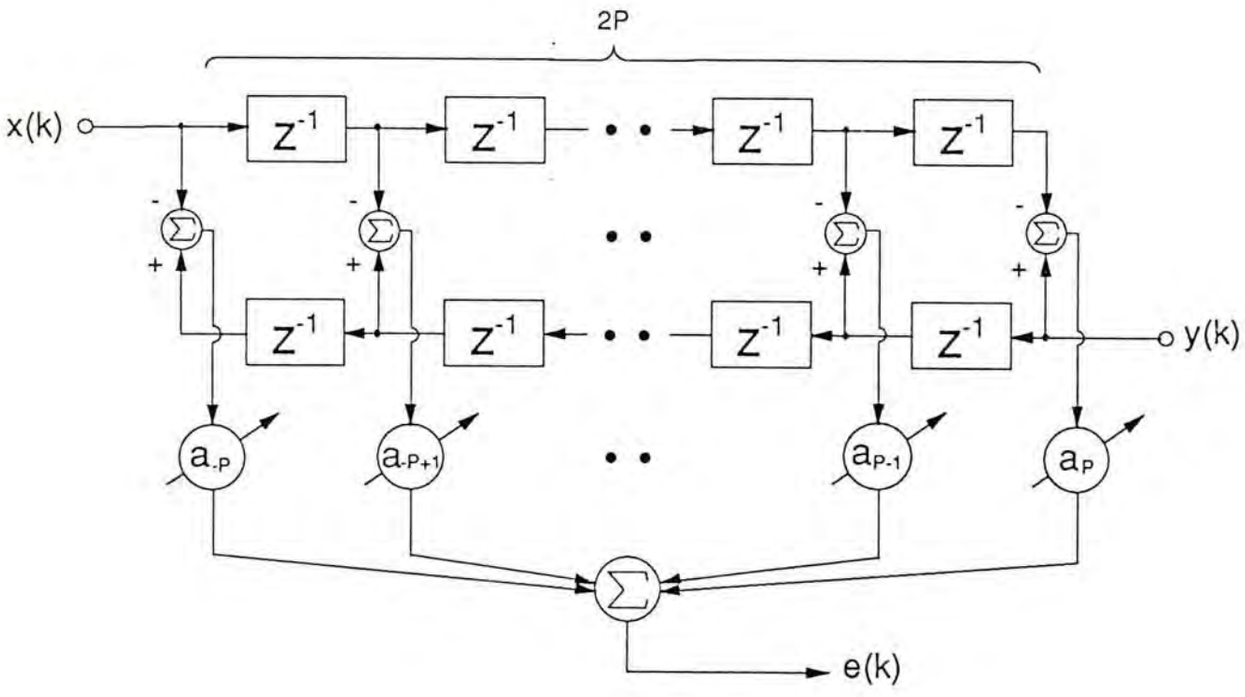
which shows that $e(k)$ can be obtained by the upper filter, if the two inputs are appropriately mixed (shifted and added). To implement equation (4.123), Figure 4.10(a) depicts the schematic block diagram while Figure 4.10(b) describes the details of the mixing and filtering. The time delay can again be retrieved from the coefficients a_i and adaptation is needed for a single filter only. Comparing with the implementation of the conventional model as shown in Figure 4.11, the new system only requires $2P+1$ extra adders and an additional P shift registers for hardware realization.

4.2.6 SIMULATION RESULTS

In this section, we shall describe the simulation tests that have been run to verify the theoretical development of the new TDE system. To check the validity that (4.80) satisfies (4.81), the left hand side of (4.81) was computed for various values of D and P and compared against the right hand side. The results obtained



(a)



(b)

Fig. 4.10 (a) Schematic block diagram of the new TDE system. (b) Hardware realization filter structure of the new model

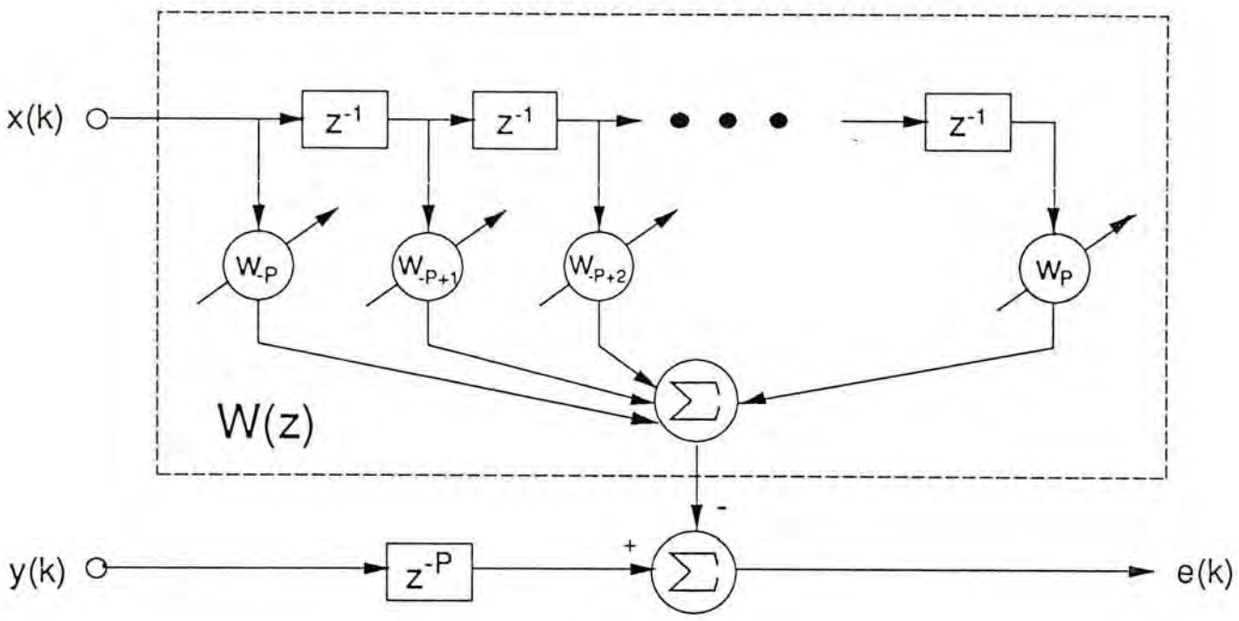


Fig. 4.11 Hardware implementation of a conventional LMSTDE model

follow pretty closely with the theory and the accuracy actually increases as P is made larger. For example, at $D=0.5$ and $P=5$, i.e., a 10th order filter, the approximation error for (4.80) for the dominant element, i.e. the middle parameter of $a_{\frac{D}{2}}$, is 2.9%. Whereas for $P=10$, this error decreases to 1.5%. Similar results are obtained for other values of D.

Different values of γ were employed and the difference between the left and right hand side of (4.88) were then calculated in order to determine the range for γ such that the equality holds with a small enough error. The error increases with increasing γ. For P=5 and 10, the largest error for the most significant element of a_γ and $a_{D-\gamma}$ with the value γ satisfying (4.91) was found to be less than 6%.

Next, the eigenvalues of $\boldsymbol{R}_D$ were computed at $P=5$ and 10 for different values of D. Table 4.1 lists the eigenvalues and they confirm the previous assertion that there are at least $L+1$ eigenvalues of σ_s^2 and L eigenvalues of $-\sigma_s^2$.

Two experiments were conducted to compare the convergence speed of the new and the conventional configurations for TDE. The signal $s(k)$ and the noises $n_1(k)$ and $n_2(k)$ were Gaussian distributed and were generated from a random number generator. The signal power was fixed to unity and the SNR for the two input channels, σ_s^2/σ_n^2, were set to 20dB. The delayed signal, $s(k-D)$, was obtained by passing $s(k)$ through a 40th order FIR filter. The adaptive filter had 21 weights, i.e., $P=10$. The LMS adaptation algorithm was used in both systems with the step sizes being chosen as $\mu_a=0.0005$ and $\mu_w=0.001$ according to (4.122). In order to ensure that both systems will give approximately the same level of misadjustment, the actual values were calculated numerically and the results were found to be $M_a=0.0211$ and $M_w=0.0208$. This shows that by satisfying the condition as stated

P	D	eigenvalues of $\boldsymbol{R}_D$ (x σ_s^2)
5	0.2	Four 1's, Three -1's, 0.9962, 0.8817, -0.9997, -0.9704
5	0.4	Four 1's, Three -1's, 0.9889, 0.5816, -0.9991, -0.9058
10	0.2	Nine 1's, Eight -1's, 0.9927, 0.8725, -0.9991, -0.9589
10	0.4	Nine 1's, Eight -1's, 0.9783, 0.5517, -0.9973, -0.8722
10	1.4	Eight 1's, Seven -1's, 0.9993, 0.9246, -0.9999, -0.9919, -0.6060, -0.0008
10	2.4	Eight 1's, Seven -1's, 0.9961, 0.6390, 0.0024, 0.0000, -0.9998, -0.9485

Table 4.1 Eigenvalues of the matrix $\boldsymbol{R}_D$

in (4.122), we shall have roughly the same misadjustment for both systems. The delay estimate was obtained from interpolation of the filter coefficients and simulation results were derived from an average of 10 independent runs.

Figure 4.12 compares the trajectories of the delay estimate in the new and conventional system with D equal to 0.3. The delay estimate was first set to null in both systems. It can be seen that the new system has a convergence speed that is approximately twice as fast. Indeed, it attained a delay estimate of value 0.28 at 1000 iterations whereas 1700 iterations were required for the old configuration.

To study the tracking ability of the new system, random step changes and sinusoidal changes in time delay were introduced to the signal $s(k-D)$ and the corresponding delay trajectories are shown in Figure 4.13 and 4.14. The trajectory obtained from the old system were also included for ease of comparison. As shown in the diagrams, both systems were able to follow these changes, however, the new model can track time-varying delays more effectively and efficiently. When the SNR is decreased to 15dB, the performance was roughly the same. In cases where the corrupting noise power is high, the analysis and performance evaluation of the proposed TDE system is still under investigation.

A very popular application of adaptive filters is in noise cancellation [16]. Referring to Figure 4.7, the lower channel, known as the primary input, contains the signal plus random noise $s(k)+n_1(k)$ whilst the upper channel contains the noise reference which is an advanced version of $n_1(k)$, $n_1(k+D)$. The noise $n_1(k)$ is independent of the signal $s(k)$ and when the adaptive filter produces the proper delay to cancel out the noise component in the lower channel, $e(k)$ will contain only $s(k)$. The configuration shown in Figure 4.8 can perform a similar cancellation operation although when steady state is reached, $e(k)=s\left(k+\frac{D}{2}\right)$ in this situation.

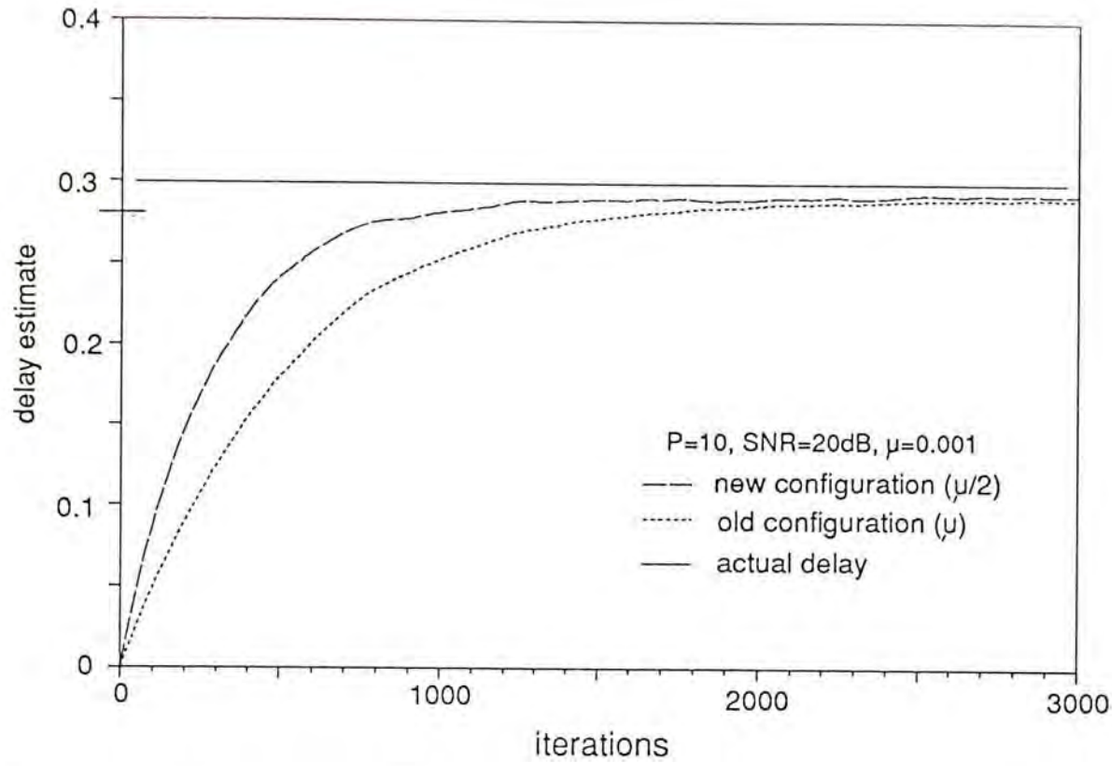


Fig. 4.12 Comparison of convergence speed for the two TDE systems

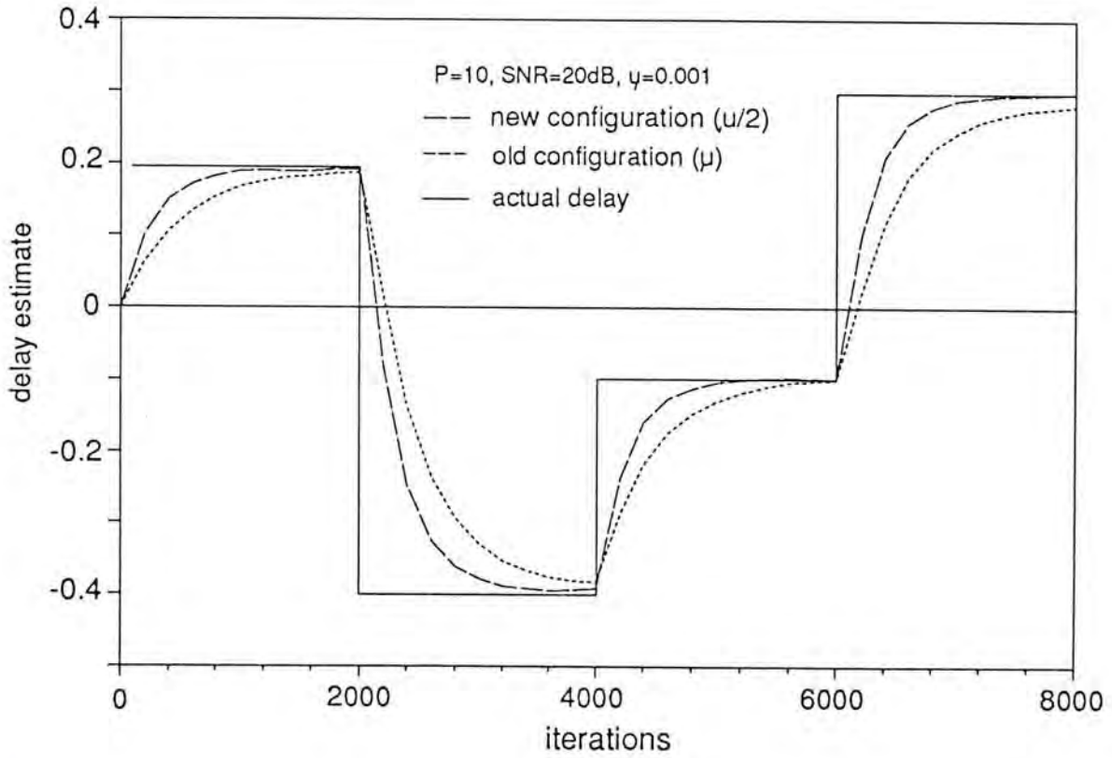


Fig. 4.13 Comparison of convergence speed with step change in delays

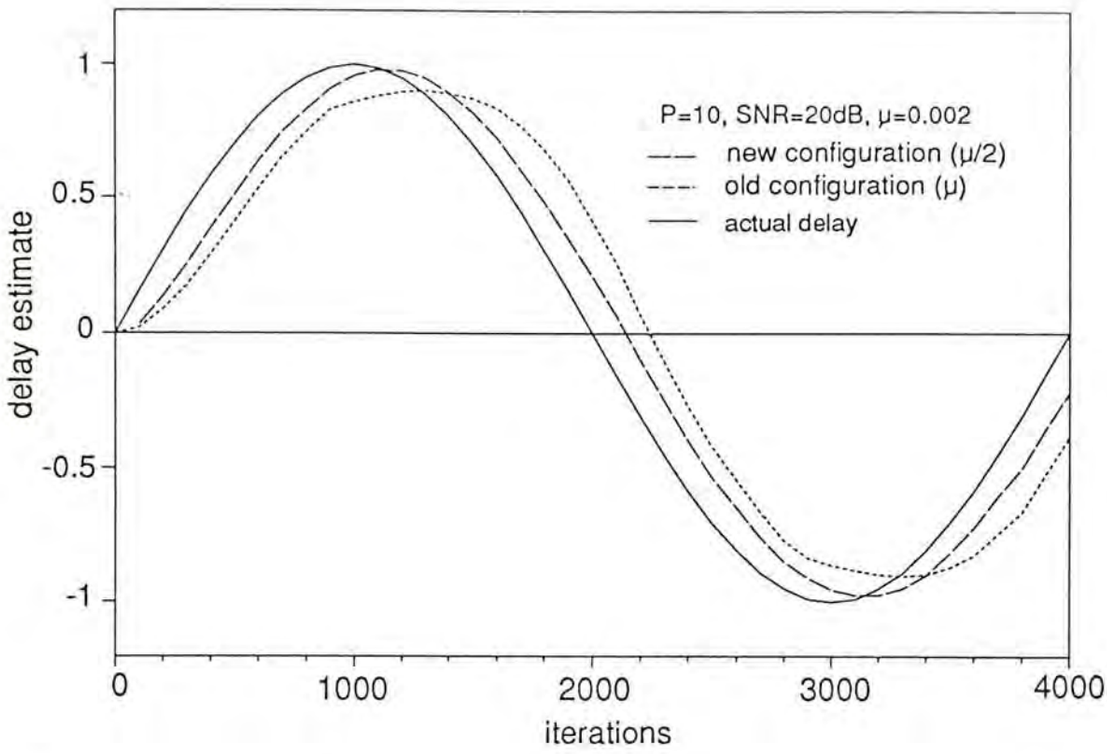


Fig. 4.14 Comparison of convergence speed with sinusoidal change in delay

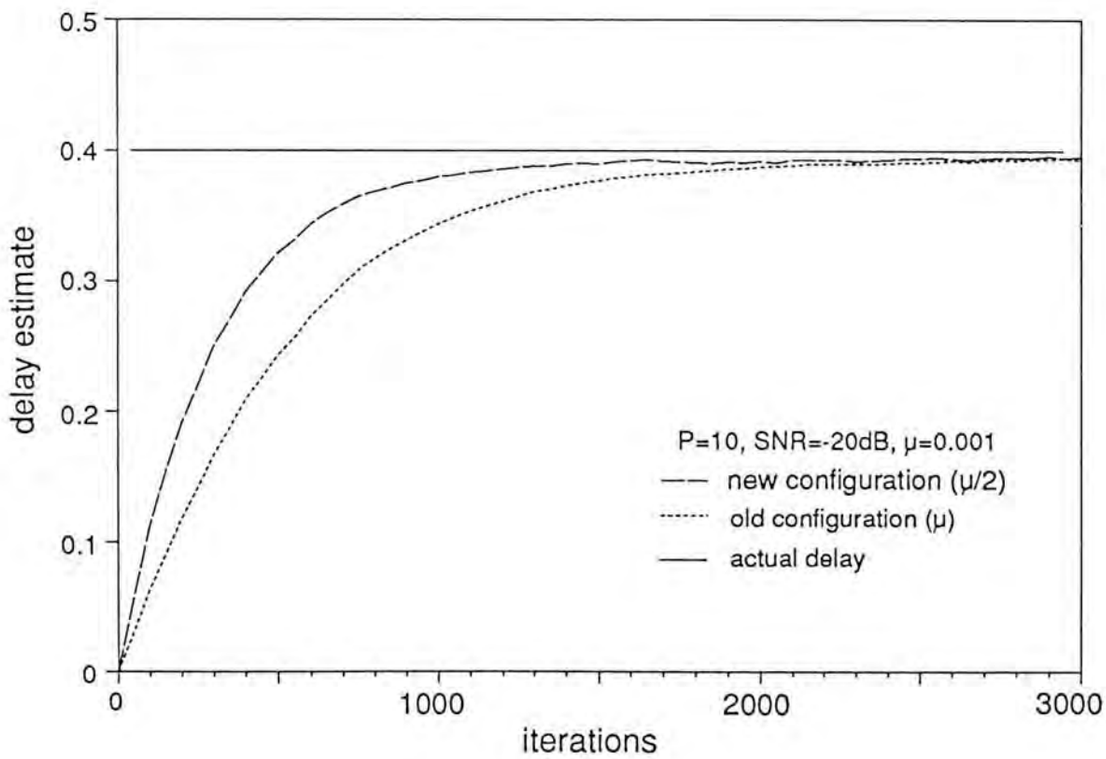


Fig. 4.15 Convergence speed up for noise cancellation application

This is of no particular concern since noise cancellation is the main objective. In any case, the time shift in $e(k)$ can be removed by passing $e(k)$ through another FIR filter chosen to have a $\frac{D}{2}$ delay.

By going through a similar procedure, it can be analysed that at low SNR and $|D|<1$, the adaptive process will converge to the desired weight vector specified in (4.80) and hence the output error will be given by $e(k)=s\left(k+\frac{D}{2}\right)$. Here, the time constant is also improved by a factor of 2, when the same misadjustment of the new and old systems are maintained. Figure 4.15 shows the noise cancellation result where $s(k)$ is an ARMA process given by

$$\begin{aligned} s(k) =& 1.1425\, s(k-1) - 0.7931\, s(k-2) + 0.5234\, s(k-3) - 0.3136\, s(k-4) \\ &+ \beta(k) - 0.2084\, \beta(k-1) + 0.36\, \beta(k-2) \end{aligned} \tag{4.124}$$

The random sequence $\beta(k)$ was Gaussian distributed and independent of the random noise $n_1(k)$. The delay D was set to 0.4 and the SNR was fixed to -20dB. It can be seen from Figure 4.15 that the adaptation speed of the new configuration is again improved by approximately 2 times.

4.2.7 CONSTRAINED ADAPTATION FOR THE NEW TDE SYSTEM [18]

Recently, Ching and Chan [17] proposed that the filter weights in traditional LMSTDE system can be constrained to *sinc* function samples such that computations for adaptation can be reduced and faster adaptation speed can be obtained. As the desired solution in the new TDE system is a *sinc* function vector, it seems that the *sinc* function constraint can also be applied to the new model to achieve yet a better performance. Incorporating this constraint in the new system has another advantage

of precluding convergence to other possible solutions. The purpose of this section is to evaluate experimentally the performance of the new system under the application of constraints.

The essence of constrained TDE [17] is to force the adaptive filter $W(z)$ as shown in Figure 4.7 to behave as a time shift device by restricting its coefficients to some *sinc* function values. That means, the filter weights w_i are given by

$$w_i = sinc(i-\Omega) \quad , \qquad i = -P, \ldots, 0, \ldots, P \tag{4.125}$$

where Ω is the time difference inserted. The error criterion can now be expressed as

$$\xi = E[\,(y(k)-x(k-\Omega))^2] \tag{4.126}$$

When steady state is reached, ξ will be minimum which implies that with high SNR,

$$\Omega^* \approx D \tag{4.127}$$

where Ω^* denotes the optimal solution of Ω.

The actual delay D can be expressed as

$$D = D_I + D_F \tag{4.128}$$

where D_I is the rounded value of D so that $|D_F| < 0.5$. As D_I can simply be deduced from the tap position of the largest filter coefficient, only $|D| < 0.5$ is needed to be considered for constrained adaptation.

The adaptive scheme of constrained TDE is to adjust the filter coefficient with the largest magnitude, say, w_m and then utilize w_m as an index to obtain the other

coefficients from a pre-stored lookup table. The table is a two dimensional matrix of size Kx$(2P+1)$ containing samples of the *sinc* functions with delay ranging from 0 to 0.5. More precisely, the elements of the table are

$$h_{ij} = sinc(j - d_i) \quad , \qquad j = -P, \ldots, 0, \ldots P \tag{4.129}$$

where

$$d_i = \frac{i}{2(K-1)} \quad , \qquad i = 0, \ldots, (K-1) \tag{4.130}$$

which gives a resolution of delay estimate equal to $1/\{2(K-1)\}$. The delay estimate is taken to be either, $m - d_i$ or $m + d_i$, depending on the signs of the two adjacent elements of w_m. Thus, the interpolation step for obtaining the time delay is no longer needed. In fact, incorporating constraints has the merits of not only simplifying the adaptive process and reducing computational load, but also achieving a faster adaptation rate.

We now consider the application of this specific constraint to the new model [18]. Suppose $|D| < 0.5$. With the constraint, the filter weights are restricted to

$$a_i = sinc(i - \Omega) \quad , \qquad i = -P, \ldots, 0, \ldots, P \tag{4.131}$$

Consequently, from (4.123), the error criterion now becomes

$$\xi = E[\{y(k+\Omega) - x(k-\Omega)\}^2] \tag{4.132}$$

In equilibrium, the optimal solution will be

$$\Omega^* \approx \frac{D}{2} \tag{4.133}$$

and thus delay estimate is taken to be

$$\hat{D} = 2\Omega^* \tag{4.134}$$

It is important to note that a look-up table containing delay ranging from 0 to 0.25 is sufficient for the new system since delay estimate is equal to twice the value obtained from the table. However, the table size in the new and the old system should be identical in order to maintain the same resolution for the delay estimate.

The detail procedures of applying constraint to the new system is outlined as follow:

(a) Set all filter coefficients to zero at the beginning.

(b) Adapt $W(z)$ for a few iterations, say, 200, using the LMS algorithm described in (4.46).

(c) Determine the maximum tap weight, w_m, of $W(z)$. Pre-shift $x(k)$ by m samples correspondingly. The initial delay inserted by $A(z)$ in the configuration shown in Figure 4.8 is set to -0.125 if $w_{m-1} > 0$ and $w_{m+1} < 0$. Otherwise, the delay is set to 0.125. Assign filter weights, which are values of a *sinc* function, to $A(z)$ according to the initial set delay with maximum at a_0. The rounded value of D is now given by m.

(d) Adaptation with constraint was applied to $A(z)$.

(e) The delay estimate at iteration k, $\hat{D}(k)$, is computed from

$$\hat{D}(k) = m + 2\Omega(k) \tag{4.135}$$

Simulation experiments were performed to evaluate the performance of the new LMSTDE system with constraint adaptation and to compare its performance with the system proposed by Ching and Chan. The simulation environments were set to be the same as the cases without constraint. The look-up table for time difference and filter weights conversion had a size of 512x21. Note that the look-up table

for the new system only needs to contain delay ranging from 0 to 0.25, the resolution of fractional delay estimate is approximately 0.001. Same table size was used in the old model to maintain equal resolution of delay estimate. Simulation results were the average of 20 independent ensembles to reduce point-to-point fluctuation.

Figure 4.16 compares the adaptive behavior of the basic and the new LMSTDE model in the absence of noise. The step sizes μ_a and μ_w were chosen to be 0.0005 and 0.001 respectively and the actual delay was set to 0.9. Let the time at which the estimate enters and stays within ±1.0% of the actual delay be the capture time. As shown in Figure 4.16, the capture time for the new and old system are roughly 1000 iterations and 1800 iterations, respectively. Thus the improvement of convergence speed in terms of capture time is about 800 iterations.

A similar test has also been run to study the performance of the two models in the presence of white noise. Figure 4.17 shows a typical result in this regard. Gaussian distributed random noise at SNR of 30dB were added to both input channels while μ_a and μ_w were set to the same values as the case without noise. The actual delay D was assigned to 2.1. It is apparent from the graph that only marginal degradation in performance was recorded and, again, the new configuration outperforms the traditional system. For both new and old LMSTDE system, deviation of the delay estimate from the desired value would be occurred if SNR was decreased. However, for a SNR as low as 10dB, both systems will still converge though with a discrepancy of around 10%.

Figure 4.18 depicts the trajectories of the two adaptive systems with nonstationary delays. In this example, the unknown time delay was given a series of step offsets and constrained adaptive algorithms were applied to estimate and track the dynamic changes of the time difference parameter. The SNR in the two input channels were both set to 30dB. It is seen that both systems responded to

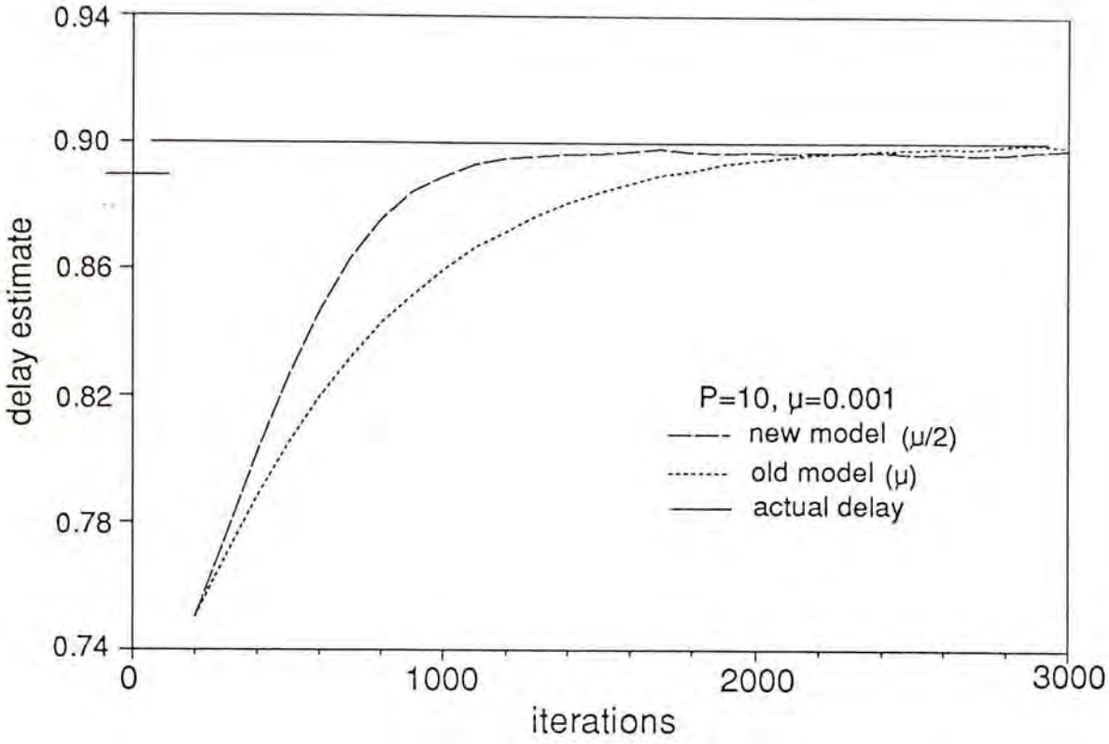


Fig. 4.16 Comparison of convergence rate for constrained adaptation without noise

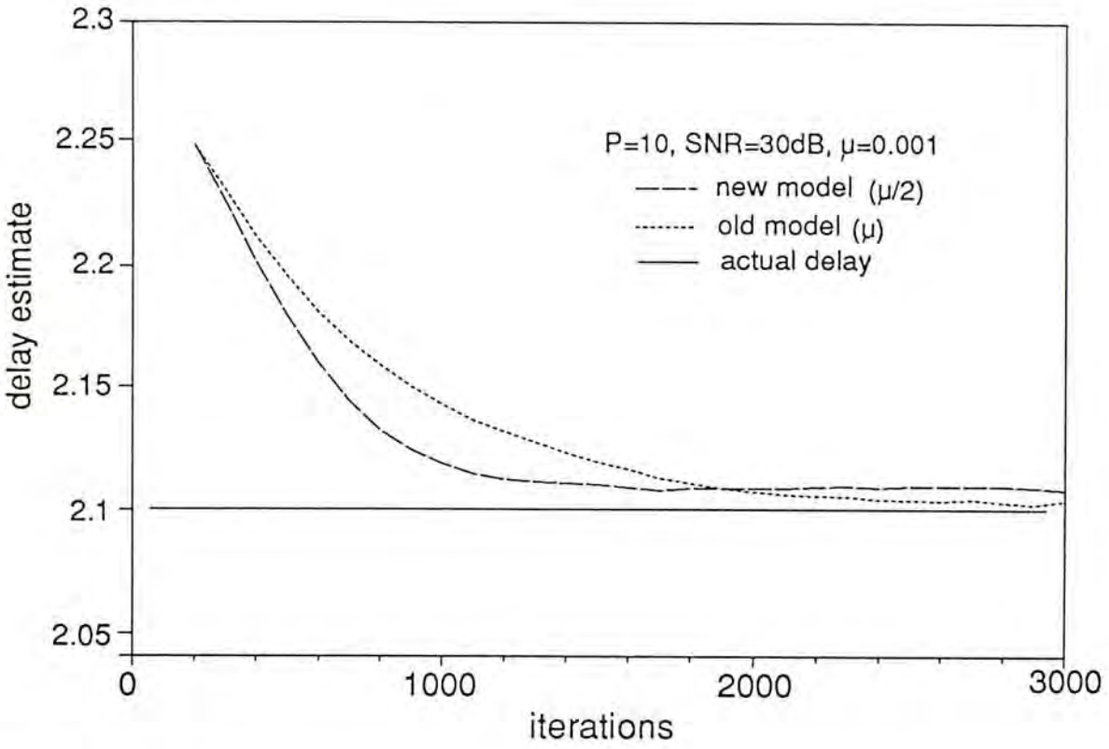


Fig. 4.17 Comparison of convergence rate for constrained adaptation with additive noise

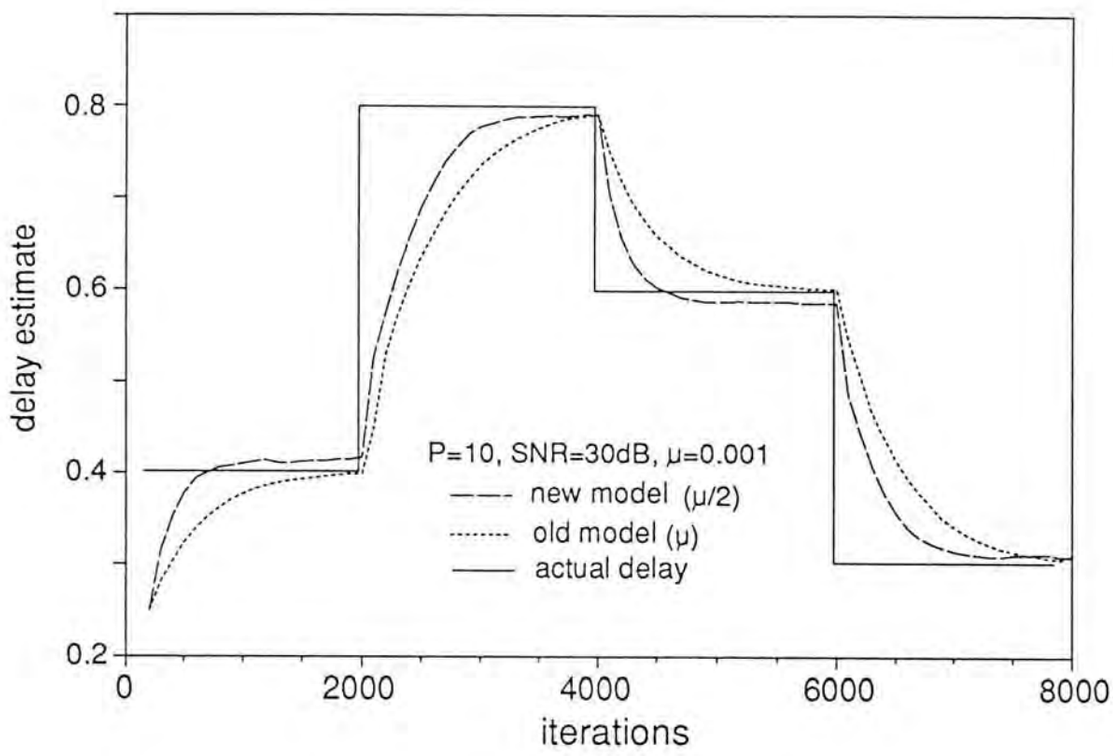


Fig. 4.18 Comparison of convergence rate for constrained adaptation with step change in delay

the step changes and were able to give accurate delay estimates after the initial transients. And again, the new model can track the non-stationary delay more quickly and accurately.

As the coefficients of $A(z)$ are known to converge to samples of the *sinc* function, constrained adaptation which restricts the filter weights to *sinc* function samples can be used in the new adaptive system to exclude the other non-desired solutions (like Figure 4.9) to occur and, in addition, to achieve a further convergence speed up. This phenomenon has confirmed by simulation results.

4.3 CHAPTER SUMMARY

The effect on dynamic adaptation characteristics incurred by serial splitting an adaptive filter is studied. It has been proved that it is particularly useful in representing an adaptive filter by two subunits connected in cascade for linear prediction application. However, it is in general not appropriate for joint process estimation.

The serial split idea is developed further to devise a new model for adaptive TDE. Conceptually, it contains an adaptive filter in both channels, with one filter inserting a time delay of $\frac{D}{2}$ in the upper channel while another providing a time advance of $\frac{D}{2}$ in the lower channel, where D is the actual delay between the two sensor measurements. They are adapted simultaneously to minimize the output MSE. A two-fold improvement in adaptation speed can be achieved compared with the conventional configuration. However, the actual hardware implementation of this new configuration only requires a single adaptive filter if proper arrangement is made. Analysis and proof for convergence speed up are given. The theoretical development is verified through simulation examples which also contain an

application to noise cancellation.

A deficiency of the new model is that the steady state solution is not unique. Nevertheless, it can be overcome by imposing *sinc* function constraint to the filter weights. Constrained adaptation on the new TDE model together with its performance analysis is investigated. Experimental results demonstrates that the new configuration has a considerably better performance for adaptive delay estimation in terms of convergence speed.

REFERENCES

[1] Y. T. Chan, R. V. Hattin, and J. B. Plant, "The least squares estimation of time delay and its use in signal detection," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 26, pp. 217-222, June 1978.

[2] Special Issue on Time Delay Estimation, *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, June 1981.

[3] P. Bolon and J. L. LaCoume, "Speed measurement by crosscorrelation - theoretical aspects and applications in paper industry," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1374-1378, Dec. 1983.

[4] B. V. Hamon and E. J. Hannan, "Spectral estimation of time delay for dispersive and non-dispersive systems," *Appl. Statist.*, vol. 23, 1974.

[5] C. H. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-24, pp. 320-327, Aug. 1976.

[6] J. C. Hassab and R. E. Boucher, "Optimum estimation of time delay by a generalized correlator," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 27, pp. 373-380, Aug. 1979.

[7] G. C. Carter, "Coherence and time delay estimation," *Proc. IEEE*, vol. 75, pp. 236-255, Feb. 1987.

[8] F. A. Reed, P. L. Feintuch and N. J. Bershad, "Time delay estimation using the LMS adaptive filter - static behavior,", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 561-571, June 1981.

[9] P. L. Feintuch, N. J. Bershad and F.A. Reed, "Time delay estimation using the LMS adaptive filter - dynamic behaviour," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 571-576, June 1981.

[10] Y. T. Chan, J. M. F. Riley and J.B. Plant, "Modeling of time delay and its application to estimation of nonstationary delays," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 577-581, June 1981.

[11] D. M. Etter and S. D. Stearns, "Adaptive estimation of time delays in sampled data systems," *IEEE Trans, Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 582-587, June 1981.

[12] D. H. Youn, N. Ahmed and G. C. Carter, "On using the LMS algorithm for time delay estimation," *IEEE Trans., Acoust., Speech, Signal Processing*, vol. ASSP-30, pp. 798-801, Oct. 1982.

[13] K. C. Ho and P. C. Ching, "A novel filter structure for adaptive time delay estimation," in *Proc. IEE Computer Architecture, Digital Signal Processing*, Hong Kong, Oct. 1989, pp. 150-154.

[14] K. C. Ho, P. C. Ching and Y. T. Chan, "Convergence speed up in adaptive time delay estimation," in *Proc. IEEE ICASSP-90*, New Mexico, Apr. 1990, pp. 1417-1420.

[15] G. M. Jenkins and D. G. Watts, *Spectral Analysis and Its Applications*. San Francisco: Holden-Day, 1968.

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

[17] P. C. Ching and Y. T. Chan, "Adaptive time delay estimation with constraints," *IEEE Trans., Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 599-602, Apr. 1988.

[18] K. C. Ho and P. C. Ching, "A new constrained least mean square time delay estimation system," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 1060-1064, Aug. 1990.

5 EXTENSION OF THE SPLIT ADAPTIVE SYSTEMS

We have already examined the properties and characteristics of an adaptive system which is constructed either by splitting an adaptive filter into two subunits in parallel or in series. Apparently, it may be possible to factorize an adaptive filter into more than two components. Such generalization of the two splitting methods will be studied in this chapter. In addition, performance comparison between the parallel and serial split system will be considered. It is shown that both the parallel and serial split predictor are superior to the transversal form configuration. The possibility of merging the two predictors to achieve further improvement in performance will also be investigated. Finally, the combined model will be used for speech analysis to demonstrate its suitability for practical applications.

5.1 THE GENERALIZED PARALLEL SPLIT SYSTEM

In chapter 3, we have only examined separating an adaptive filter $W(z)$ into two linear phase filters connected in parallel. In general, it can be decomposed into L subunits $W_i(z)$,

$$W(z)=\sum_{i=1}^{L} W_i(z) \tag{5.1}$$

Notice that at least one of the subunits must have an order greater than or equal to that of $W(z)$ so as to maintain order consistency. This adaptive system is depicted in Figure 5.1. All $W_i(z)$ have a common input $x(k)$ and their outputs are added to

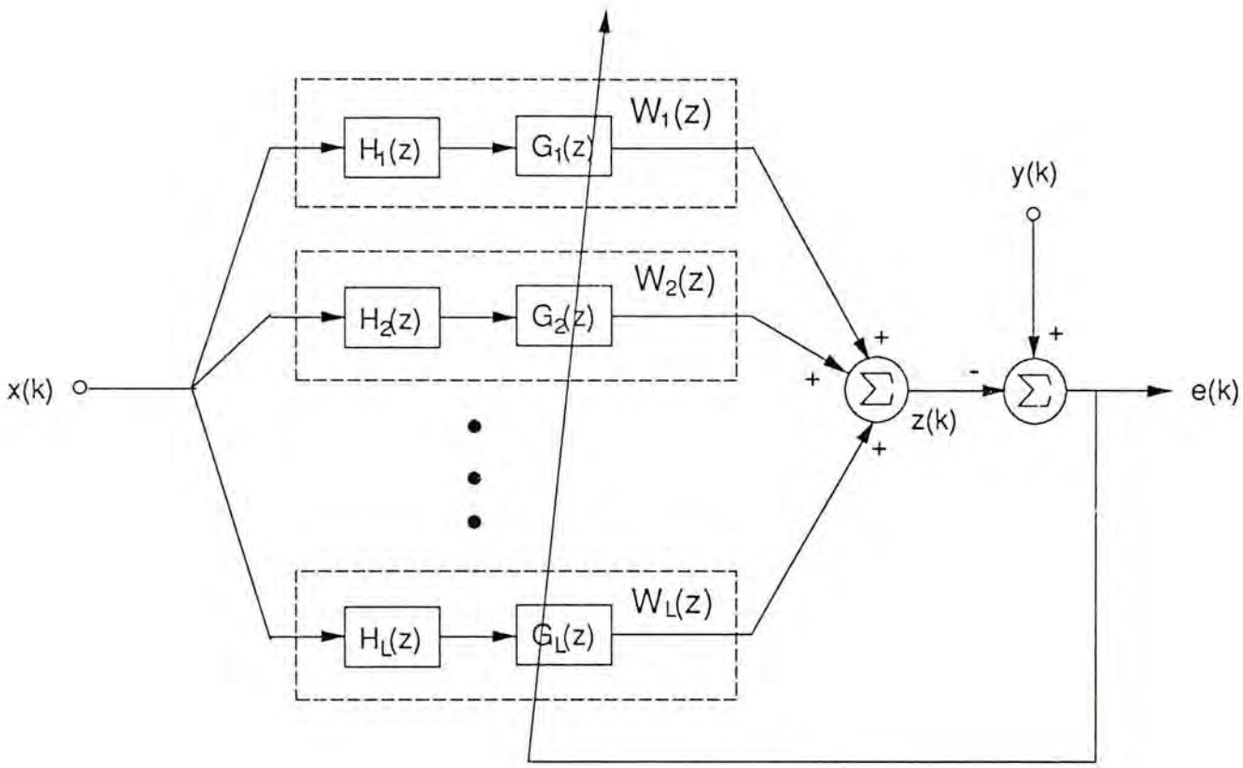


Fig. 5.1 The generalized parallel split adaptive system

form the total filter output $z(k)$. The error signal given by the difference between the desired response $y(k)$ and the total filter output $z(k)$ is employed to adjust each of the subunits.

We now consider how to select the subunits $W_i(z)$. When $L=2$, they are chosen to be linear phase filters to generate orthogonal adaptation sequences for them. Different adaptation step sizes can then be used for the two subunits to improve convergence speed. This gives us an insight that for L greater than 2, the subunits $W_i(z)$ should also be chosen so that their adaptation input sequences are also roughly orthogonal to each other. A simple method to achieve this requirement is to take $W_i(z)$ to be

$$W_i(z)=H_i(z)\,G_i(z) \quad , \qquad i=1,2,\ldots,L \tag{5.2}$$

where $H_i(z)$ are fixed filters to generate approximately orthogonal sequences whilst $G_i(z)$ are adaptive filters. In this way the regression vectors for $G_i(z)$ are the outputs of $H_i(z)$ which are about orthogonal to each other and different step sizes can then be utilized for each $G_i(z)$ to achieve a better convergence behavior. When L is equal to the filter length of $W(z)$, the generalized model becomes the transform domain adaptive filter [1]. The filters $H_i(z)$ is now used to produce the transform coefficients which are to be weighted by adaptive gains $G_i(z)$ to form the filter output.

It is anticipated that the generalized model can achieve faster convergence than the split-path adaptive configuration described in chapter 3. It requires, however, more computations for generating the approximately orthogonal adaptation sequences. Another drawback of this model is that for linear prediction, the backward prediction knowledge cannot be incorporated easily in the adaptation process as in the split-path model to reduce gradient noise.

5.2 THE GENERALIZED SERIAL SPLIT SYSTEM

It is trivial that the split filters $A(z)$ and $C(z)$ in the serial split adaptive system as shown in Figure 4.1 can further be decomposed into subsections, that means an adaptive filter $W(z)$ can be configured as cascade of more than two subunits,

$$W(z)=\prod_{i=1}^{L} H_i(z) \tag{5.3}$$

When the number of subsections becomes larger, greater computational burden is involved. In the linear prediction case, it is particularly useful to select $H_i(z)$ to be second order sections,

$$H_i(z)=(1+h_{1,i}z^{-1}+h_{2,i}z^{-2}) \quad , \tag{5.4}$$

so that the zero locations of the inverse filter can be easily tracked and constraints about the locations of the zeros can also be incorporated in the estimation procedure.

The generalized serial split predictor is depicted in Figure 5.2. The input signal $x(k)$ is fed into the first second order section and its output is applied to the second section and so on. The error $e(k)$ is taken as the output of the last section and each of the adaptive subunits $H_i(z)$ are adjusted to minimize the ultimate mean-square error. In accordance with the LMS adaptation algorithm [2], the updating equations for the filter parameters can be formulated as

$$h_{l,i}(k+1)=h_{l,i}(k)-2\mu e(k)g_{l,i}(k) \quad , \qquad l=1,2; \quad i=1,2,\ldots,L \tag{5.5}$$

where $g_{l,i}(k)$ is the partial derivative of $e(k)$ with respect to $h_{l,i}(k)$. It can be shown [3]-[4] that the gradient $g_{l,i}(k)$ can be obtained by passing the input sequence through all but the ith cascade filter section and delaying the output by l samples. Under the assumption that the step size μ is small such that the adjustment of the

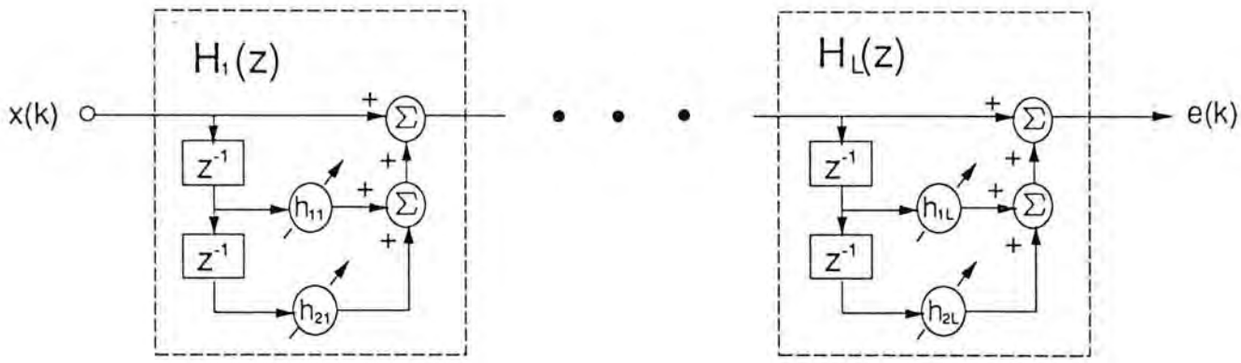


Fig. 5.2 The generalized serial split adaptive system

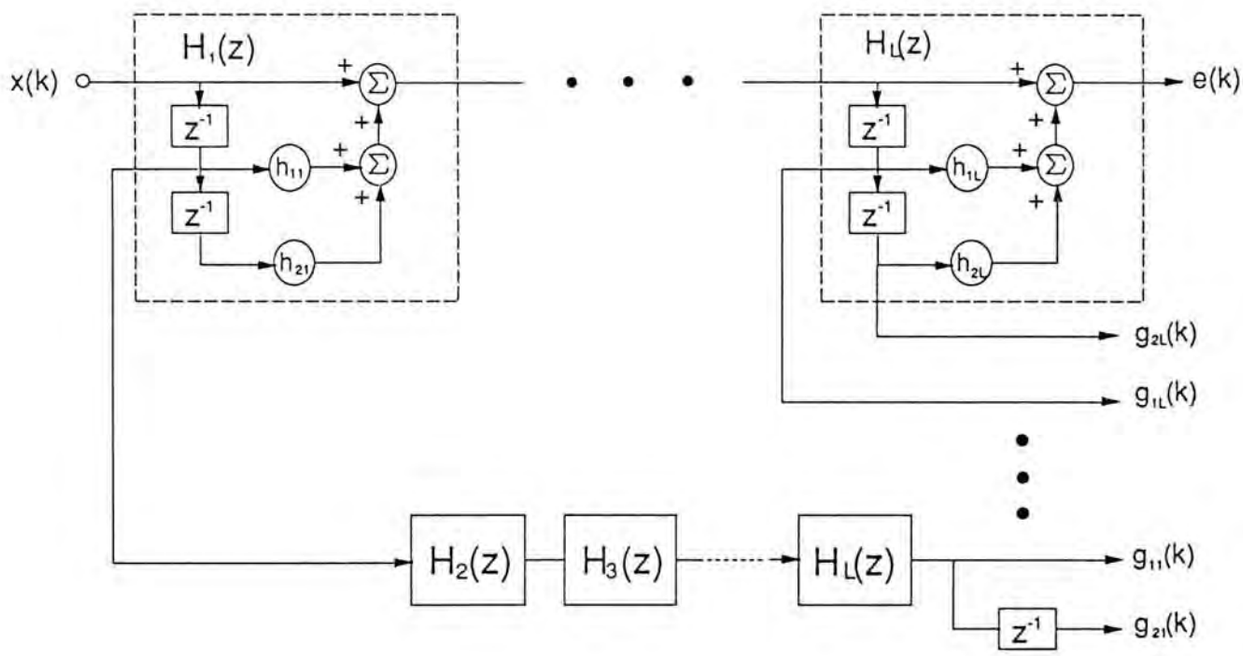


Fig. 5.3 Generation of gradient sequences for adaptive filter constructed in 2nd order cascade sections

parameters during each iteration is small, we have $g_{2,i}(k) = g_{1,i}(k-1)$. The details of the gradient calculation is illustrated in Figure 5.3. The transfer function to generate gradient components $g_{l,i}(k)$ can be expressed as

$$T(z) = \frac{G_{l,i}(z)}{X(z)} = z^{-l} \prod_{\substack{j=1 \\ j \neq i}}^{L} (1 + h_{1,j} z^{-1} + h_{2,j} z^{-2})$$

$$= z^{-l} \frac{W(z)}{H_i(z)} \quad , \qquad l = 1, 2; \qquad i = 1, 2, \ldots, L \tag{5.6}$$

where $G_{l,i}(z)$ and $X(z)$ designate the Z-transform of the gradient sequence $g_{l,i}(k)$ and the input signal $x(k)$. Equation (5.6) provides a computationally efficient method for generating the gradient sequences from the residual sequence $e(k)$ by passing it through an IIR filter which is the inverse of an appropriate second order section, as demonstrated in Figure 5.4.

Comparing with the case of simply splitting a predictor $W(z)$ into only two subunits, the system with second order cascade sections can further improve the adaptation speed since the autocorrelation matrices that determine the convergence behavior of the cascade sections have smaller eigenvalue spreads. Nevertheless, an increase in system complexity in this case is inevitable.

5.3 COMPARISON BETWEEN THE PARALLEL AND THE SERIAL SPLIT ADAPTIVE SYSTEM

In both parallel and serial split model, speed-up in adaptation rate is achieved by reducing the eigenvalue spread of the input process. However, the mechanism for the two split methods to accomplish this objective is different. In the parallel split model, the eigenvalues of the input correlation matrix is partitioned into two sets, leading to a decrease in eigenvalue spread. Notice that the eigenvalue spreads

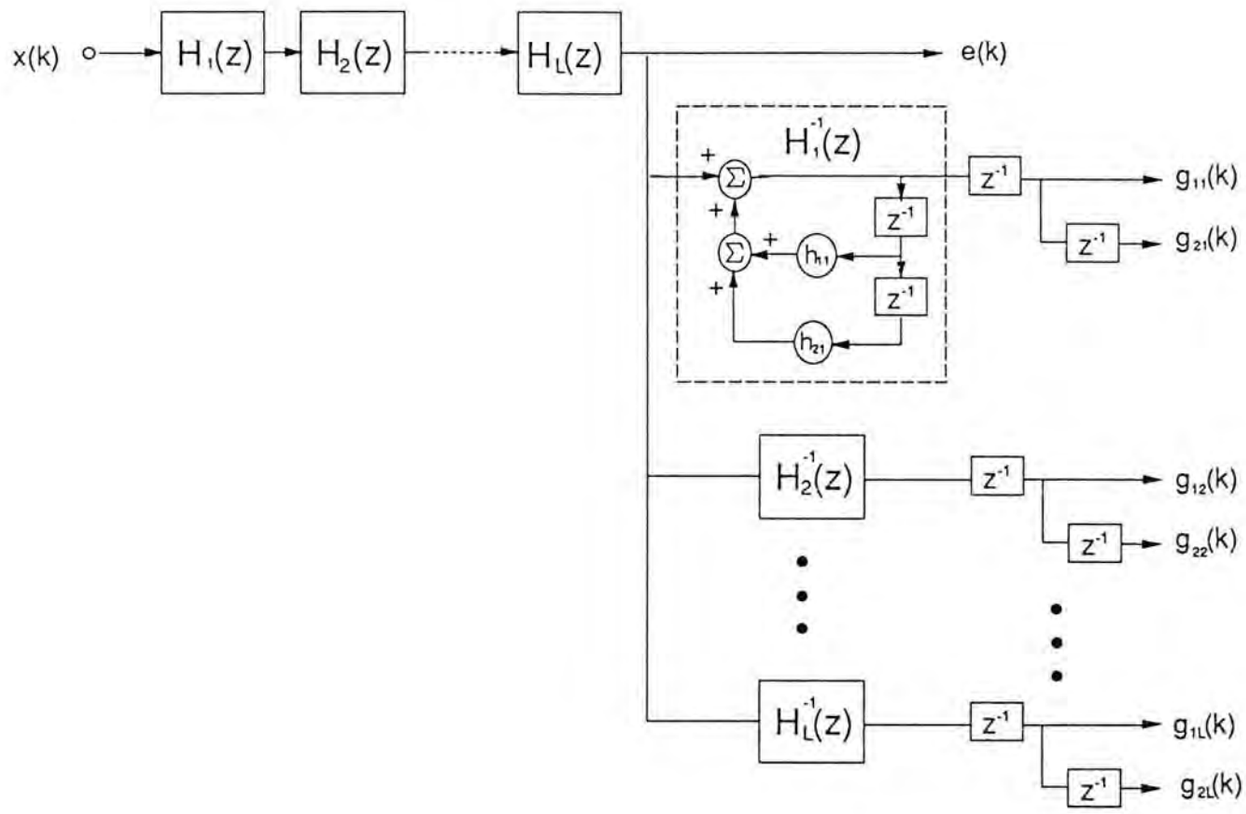


Fig. 5.4 Generation of gradient sequences using recursion method for 2nd order cascade adaptive filter

in this case are constant and will not change as adaptation proceeds. On the other hand, the serial split model changes the eigenvalue spread for each adaptive component by filtering the input sequence through the other adaptive subunit. As the adaptive components of the serial split model are varying during adaptation, the eigenvalue spreads for the subunits are not fixed and modification of their values is continued until steady state is reached.

In joint process estimation, the parallel split model is superior to the serial split configuration. The partition of eigenvalues in the parallel system allows selecting different step sizes for the two split-paths for a faster convergence. In the serial system, the filtering characteristics of the adaptive subunits depends on the desired response. The eigenvalue spread for the two subunits is therefore governed not only by the input sequence but also by the desired response. Consequently, there is no guarantee for reduction in eigenvalue spread and a better performance is not assured.

In the linear prediction problem, both models are superior to their non-split counterparts. It is appealing that the parallel model can incorporate the backward prediction naturally to the adaptation rule to diminish gradient noise so that larger step sizes can be chosen for the two split filters to further increase the adaptation rate, apart from the enhancement due to reduction in eigenvalue spread. In the serial model because the role of the two subunits are to flattening the input spectrum, the eigenvalues for the two subunits will continue to decrease until steady state is reached. Therefore a faster convergence speed is guaranteed.

Simulations were conducted to compare the adaptation characteristics of the serial and the parallel split predictor. The AR process shown in (4.28) with an eigenvalue spread of 6432 and that in (4.32) having an eigenvalue spread of 33.1 were to be whitened by these two models. In the parallel split predictor, the

adaptation step sizes for the two paths were chosen according to (3.132). While in the serial split predictor, the two subunits $A(z)$ and $C(z)$ had initial values set according to (4.30). The step sizes for both systems were chosen to maintain equal excess MSE. The corresponding learning characteristics were depicted in Figure 5.5 and Figure 5.6 respectively. It can be observed that for the input process (4.28), the serial split system outperformed the parallel system and the MSE reached a value of 4dB at about 200 iterations, whilst the parallel split model required 1100 iterations instead. However, when using the input process described by (4.32), the parallel split model was better. The MSE reduced to 1.5dB at about 50 iterations which is 70 iterations faster. Different initial conditions for the serial split predictor were tried but it was still found to be inferior to the parallel predictor.

These observations can be explained as follows. When the input process has a large eigenvalue spread, partitioning the eigenvalues in general could not reduce the spread all that many. In this case, the serial model is better due to its ability to continuously decrease the eigenvalue spread. On the other hand, when the input process has a relatively small eigenvalue spread, the serial split model could not decrease the eigenvalue spread substantially. Whereas partitioning the correlation matrix in this case can effectively diminish the eigenvalue spread. This merit together with its capability of reducing gradient noise makes the parallel split predictor much superior. As a consequence, the serial predictor is appropriate for input statistics which has a large eigenvalue spread while the parallel predictor is preferred when the input process has a relatively small eigenvalue spread.

Regarding system complexity, the parallel split system is desired because it requires extra additions only, which unlike the serial system where additional multiplications is required as well.

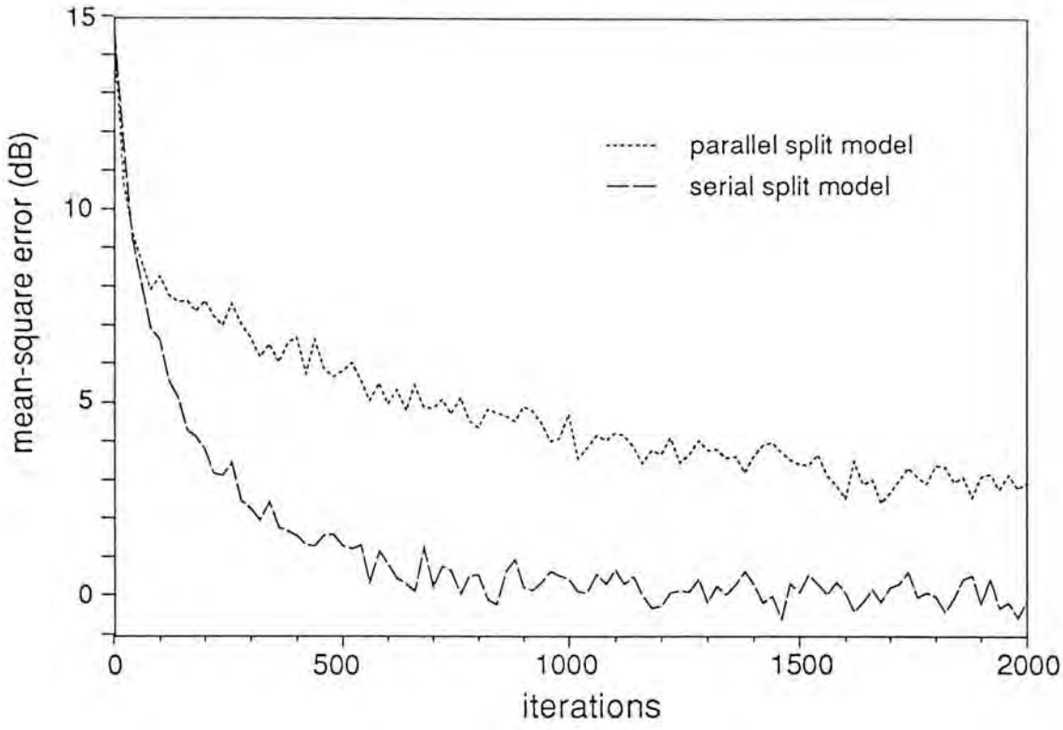


Fig. 5.5 Comparison of convergence rate for the parallel and the serial split model with large eigenvalue spread input

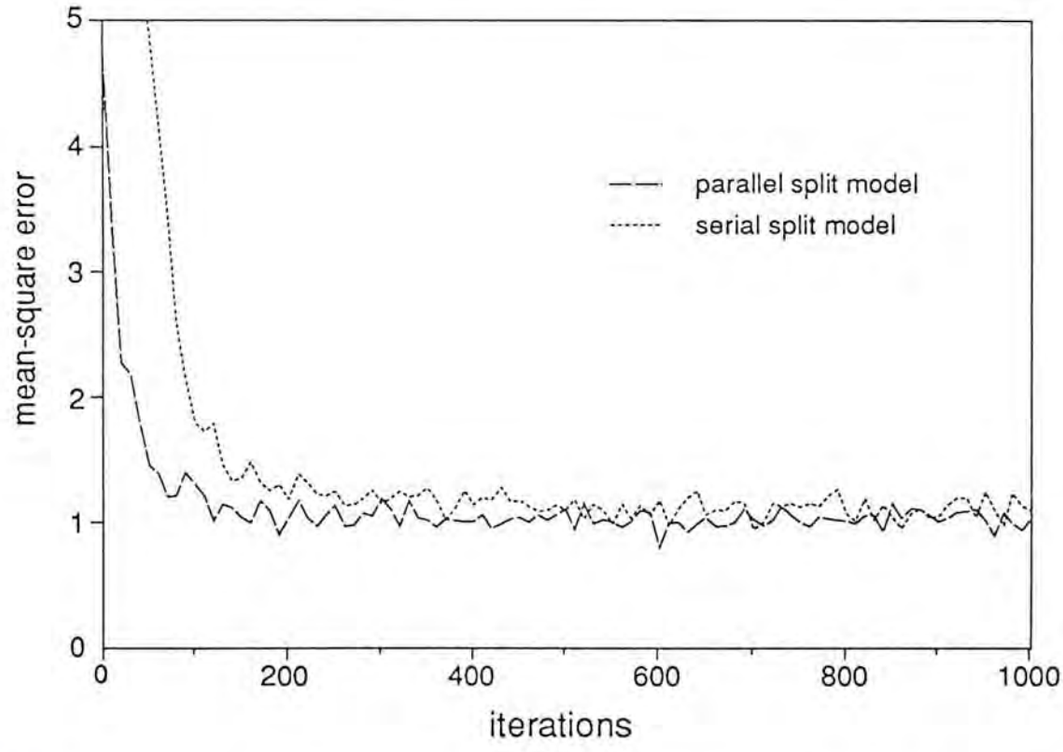


Fig. 5.6 Comparison of convergence rate for the parallel and the serial split model with small eigenvalue spread input

5.4 INTEGRATION OF THE TWO FORMS OF SPLIT PREDICTORS

We have separately developed the parallel split and serial split adaptive systems. Both of them have proven to operate better than their respective non-split counterparts in the context of linear prediction. This motivates yet another novel idea on whether the two different types of split predictors can be merged together to form a new adaptive structure to possess all the advantages inherent to these two systems for providing further improvement in terms of adaptation speed as well as dynamic convergence characteristics.

The simplest way to combine the two split methods is, first of all, parallel split an adaptive predictor $W(z)$ with independent adaptation. This provides a two-fold increase in adaptation speed by reduction of gradient noise. Then, serial split can be applied to the two linear phase filters. An additional improvement in convergence time can be acquired because the eigenvalue spreads for the adaptive subunits resulted from the serial split are reduced as adaptation goes on. The block diagram of this newly constructed model is shown in Figure 5.7. Note that one can choose different step sizes for the two split-paths to achieve additional gain in convergence rate.

Experiments were carried out to investigate the dynamic convergence characteristics of this new model. The input sequence generated by (4.28) was to be identified separately by the new model shown in Figure 5.7, the serial split model in Figure 4.1 and the parallel split model in Figure 3.11. The adaptation step sizes for the new system and the serial model were set to 0.0008 and that for the parallel system were chosen to be 0.0002 in order to keep the same level of misadjustment in all three cases. The average square output error for the three models are depicted in Figure 5.8. It is observed that the combined structure had the best performance

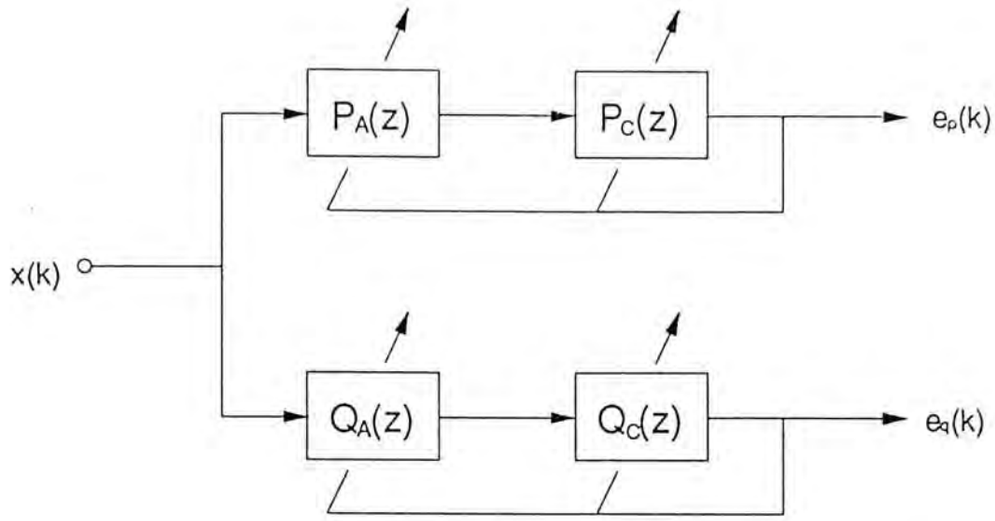


Fig. 5.7 The combined model of a parallel and a serial split predictor

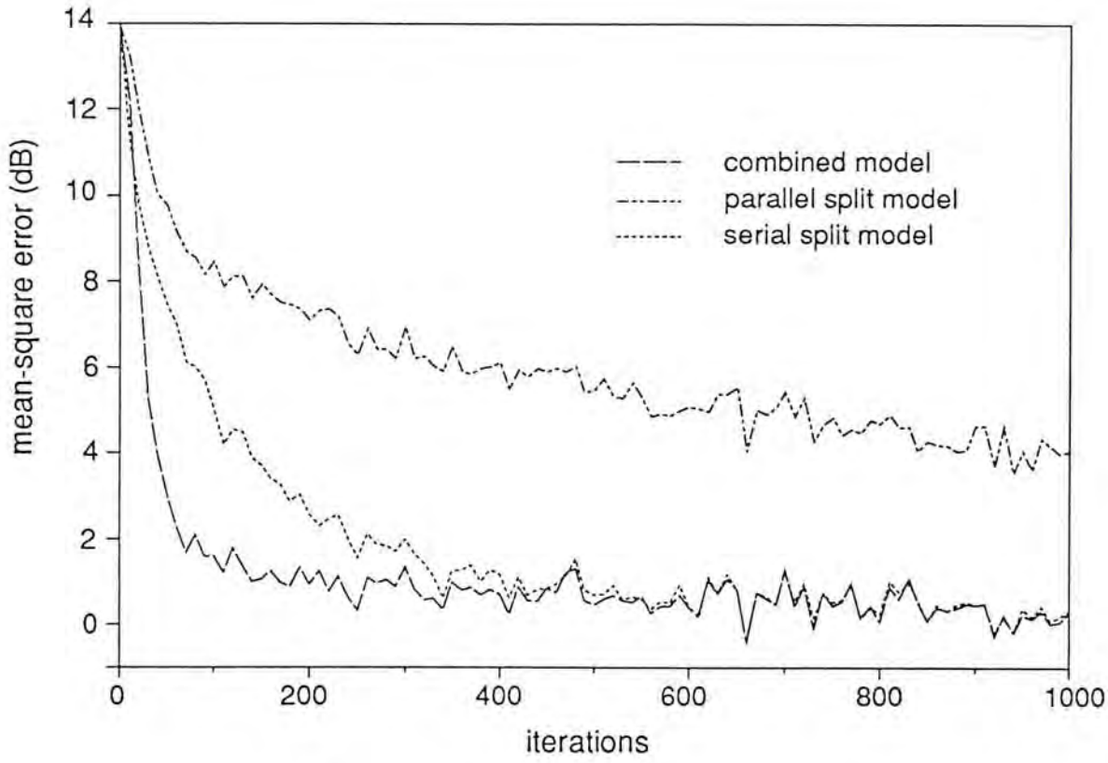


Fig. 5.8 Comparison of convergence rate

and it obtained an average square error of 2dB at about 80 iterations, which is 200 iterations less than the serial split system. Other simulation tests were also undertaken and they all verified the same result.

Next, we have applied this integrated model to the application of speech analysis. Besides the improvement in adaptation rate, it will also be demonstrated that when the adaptive subunits in Figure 5.7 are constructed as second order sections, the filter parameters are equivalent to the line spectral pair (LSP) coefficients, which are known to be especially appropriate for low bit rate speech coding.

5.5 APPLICATION OF THE INTEGRATED SPLIT MODEL TO SPEECH ENCODING [5]

Digital coding techniques have been widely used for low bit rate transmission of speech signals in the past decade. This is due not only to the rapid development of low-cost digital hardware but also to the invention of digital algorithms that accurately model the speech production mechanism. In particular, the introduction of linear prediction has opened up a new area of research in the analysis and synthesis of speech which has attracted numerous efforts from scientists and engineers.

Digital speech production model always assumes that the speech signal $s(k)$ is generated by an autoregressive (AR) function of order $2N$ which can be expressed as

$$s(k) = -\sum_{i=1}^{2N} w_i^* s(k-i) + e(k) \tag{5.7}$$

where $e(k)$ is referred to as the driving source that can often be approximated by either a periodic impulse train with period P for voiced sounds, or by a random

noise having a flat spectrum for unvoiced sounds. Figure 5.9 shows a schematic block diagram of a simple discrete analysis model in which the speech production mechanism is being modelled by an adaptive transversal filter. The system mainly concerns with finding iteratively a set of prediction coefficients, i.e. w_i^* in (5.7), that will best characterize the properties of the speech signal. There are many parameter sets that can equivalently represent the prediction coefficients. The choice of a particular set is usually based on the following criterion. First of all, a simple test on the parameters should be available to check whether the analysis filter has minimum phase or not such that stability of the respective synthesis filter can be guaranteed. Secondly, the parameter set should not be prone to quantization error to enable efficient transmission.

It is commonly recognized that PARCOR coefficients of the lattice filter [6] will generally satisfy the above requirements. Recently, it has been shown that [7]-[8] the line spectral pair (LSP) parameters are superior to the PARCOR coefficients with respect to their quantization and interpolation properties as a function of spectral distortion. Experimental results also indicate that high-quality synthesized speech can be obtained using the LSP parameters at relatively low transmission rates. However, conventional approach for calculating LSP parameters always involves computation of the prediction coefficients and then followed by a discrete cosine transform [9]. This complicated numerical procedure has made real-time implementation of the LSP analysis-synthesis speech codec very difficult, if not impossible. It is thus necessary to devise a procedure that can obtain the LSP parameters efficiently.

As speech production process can be modelled by an AR process, speech analysis can be accomplished by the split-path adaptive filter. When the split-path filter is configured as a cascade of second order sections, it will be shown that the

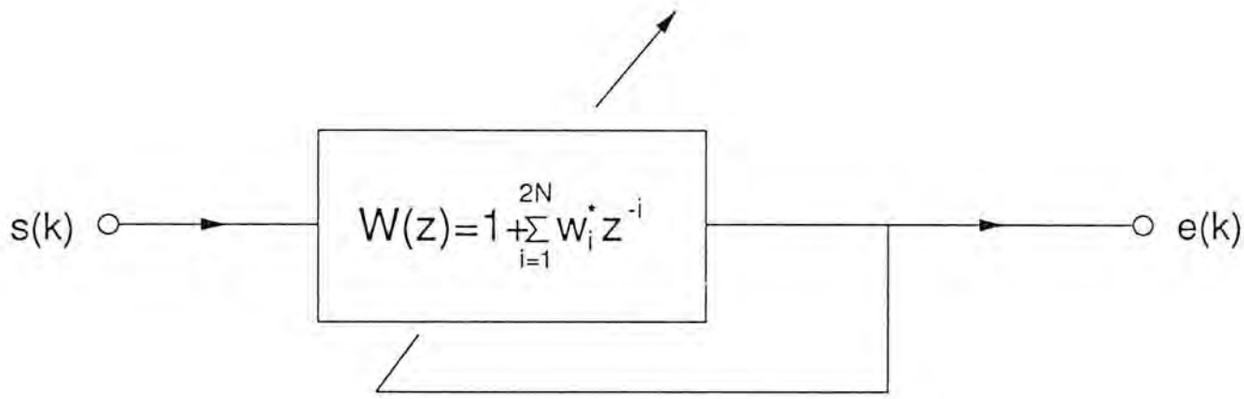


Fig. 5.9 A simple discrete speech analysis system

filter parameters are essentially equivalent to the LSP parameters. When the filter parameters are modified on a sample by sample basis, the adaptation speed will be greatly improved compared to the transversal filter and the LSP parameters can be obtained fairly easily, thereby reducing the system complexity considerably.

A speech analysis system using the split-path adaptive filter in cascade form is depicted in Figure 5.10. The analysis model is now separated into two filters, $P(z)$ and $Q(z)$, connected in parallel. $P(z)$ has anti-symmetric linear phase property whereas $Q(z)$ has symmetric linear phase property. The filter coefficients of $P(z)$ and $Q(z)$ are obtained iteratively by minimizing the expectation of their output mean-square errors $E[e_p^2(k)]$ and $E[e_q^2(k)]$ independently. Recall from chapter 3 that the transfer function of $P(z)$ and $Q(z)$ are given by

$$P(z) = 1 + \sum_{i=1}^{N} p_i \left(z^{-i} - z^{-2N-1+i} \right) - z^{-2N-1} \tag{5.8a}$$

and

$$Q(z) = 1 + \sum_{i=1}^{N} q_i \left(z^{-i} + z^{-2N-1+i} \right) + z^{-2N-1} \tag{5.8b}$$

Factorizing (5.8) yields

$$P(z) = (1 - z^{-1}) \prod_{i=1}^{N} (1 + c_i z^{-1} + z^{-2}) \tag{5.9a}$$

and

$$Q(z) = (1 + z^{-1}) \prod_{i=1}^{N} (1 + d_i z^{-1} + z^{-2}) \tag{5.9b}$$

where $c_i = -2\cos\omega_i$ and $d_i = -2\cos\theta_i$ for $i = 1, 2, \ldots, N$. It is trivial that the roots of $P(z)$ and $Q(z)$ can actually be obtained from $e^{\pm j\omega_i}$ and $e^{\pm j\theta_i}$ respectively. For simplicity but without loss of generality, assume that $\omega_1 < \omega_2 < \ldots < \omega_N$ and $\theta_1 < \theta_2 < \ldots < \theta_N$. Then it can be proved that they are alternate to each other on the unit circle [9]-[10], that is,

$$0 < \theta_1 < \omega_1 < \theta_2 < \ldots < \theta_N < \omega_N < \pi \tag{5.10}$$

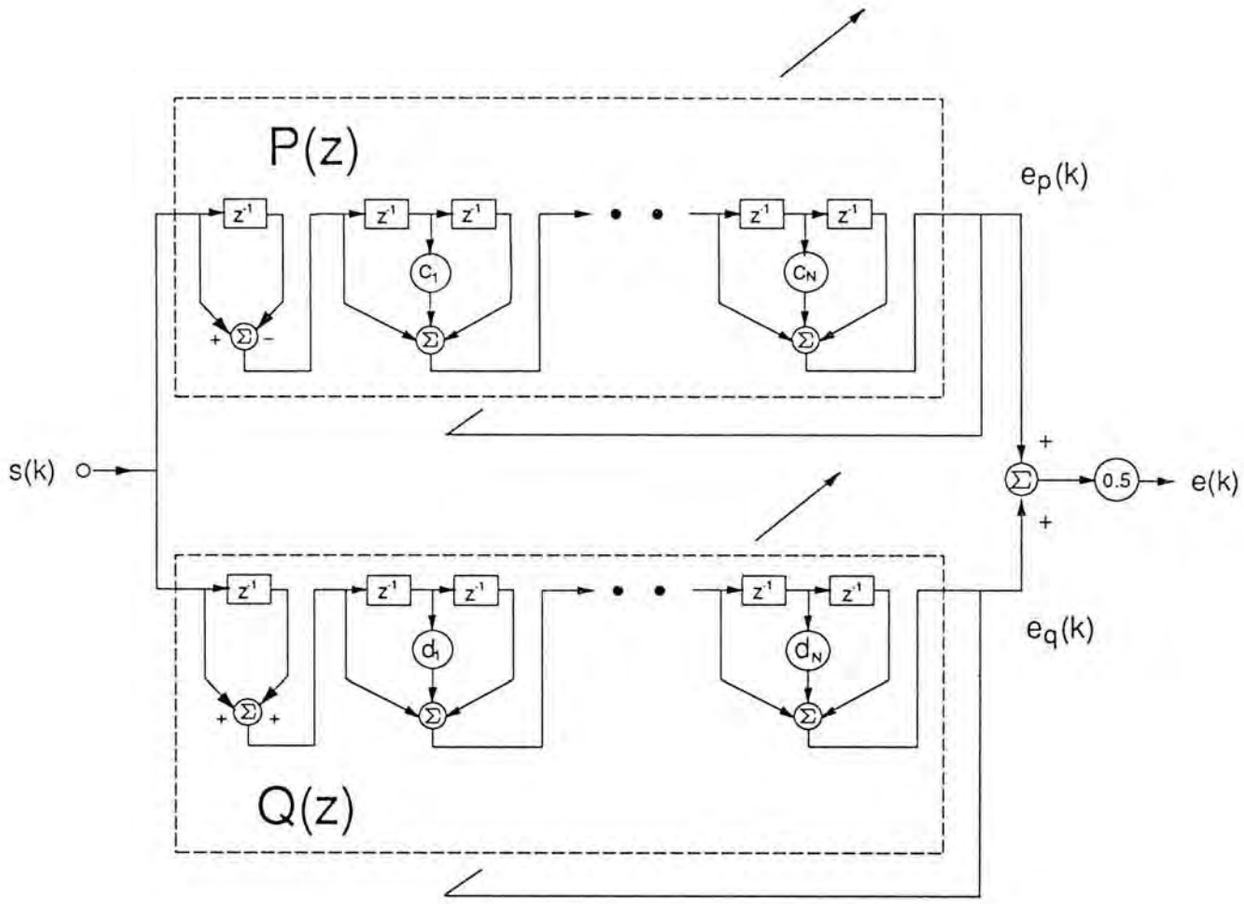


Fig. 5.10 The split-path speech analysis system

The parameters $\{ \theta_1, \omega_1, \theta_2, \omega_2, \ldots, \theta_N, \omega_N \}$ are commonly referred to as the line spectral pair (LSP) parameters. Since these coefficients are closely related to the zeros of the prediction transfer function and hence to the formant structure of the speech signal, they can be used to characterize the analysis model. These coefficients have the advantage of easier to quantize because of their uniform sensitivity across the frequency spectrum. It has been shown that [10] as long as equation (5.10) is satisfied, stability of the synthesis filter can be assured.

In the split-path analysis filter, the parameters c_i and d_i are adapted sequentially using LMS [2] algorithm. To simplify the system complexity to make real time processing feasible, the parameters are restricted to take on some uniformly pre-quantized values. Using pre-quantized values, only the sign of the noisy gradient is necessary to determine the direction for adjusting the LSP coefficients in the adaptive process. To summarize, the parameter updating equations are given as follows,

$$\begin{aligned} c_i(k+1) &= c_i(k) - \delta_p \, sgn\left\{ \frac{\partial e_p^2(k)}{\partial c_i(k)} \right\} \\ &= c_i(k) - \delta_p \, sgn\{ e_p(k) \} \, sgn\{ g_{p,i}(k) \} \quad , \quad i = 1, \ldots, N \end{aligned} \tag{5.11a}$$

and

$$\begin{aligned} d_i(k+1) &= d_i(k) - \delta_q \, sgn\left\{ \frac{\partial e_q^2(k)}{\partial d_i(k)} \right\} \\ &= d_i(k) - \delta_q \, sgn\{ e_q(k) \} \, sgn\{ g_{q,i}(k) \} \quad , \quad i = 1, \ldots, N \end{aligned} \tag{5.11b}$$

where δ_p and δ_q are the fixed increment for the parameters, $sgn(x)$ is the signum operation given by

$$sgn(x) = \begin{cases} 1 \, , & x > 0 \\ 0 \, , & x = 0 \\ -1 \, , & x < 0 \end{cases} \tag{5.12}$$

$g_{p,i}(k)$ and $g_{q,i}(k)$ are, respectively, the first derivative of $e_p(k)$ and $e_q(k)$ with respect to c_i and d_i. Referring to Figure 5.10 and using (5.9a), the Z-transform of $e_p(k)$ is given by

$$E_p(z)=S(z)P(z)=S(z)(1-z^{-1})\prod_{i=1}^{N}(1+c_i z^{-1}+z^{-2}) \tag{5.13}$$

where $E_p(z)$ and $S(z)$ are the Z-domain representation of $e_p(k)$ and $s(k)$ respectively. Let the Z-transform of $g_{p,i}(k)$ be $G_{p,i}(z)$. Taking the derivative of $E_p(z)$ with respect to c_i gives

$$\begin{aligned} G_{p,i}(z) &= S(z)z^{-1}(1-z^{-1})\prod_{\substack{j=1\\ j\neq i}}^{N}(1+c_j z^{-1}+z^{-2}) \\ &= E_p(z)\frac{z^{-1}}{1+c_i z^{-1}+z^{-2}} \end{aligned} \tag{5.14}$$

It is noted from (5.14) that the gradient for the ith section can be derived simply by passing the unit sample delayed output to a recursive filter which is just the inverse of the ith cascade section. Thus, the gradients in time domain can be computed from

$$g_{p,i}(k)=e_p(k-1)-c_i g_{p,i}(k-1)-g_{p,i}(k-2), \quad i=1,\ldots,N \tag{5.15a}$$

Similarly, $g_{q,i}(k)$ can be obtained from

$$g_{q,i}(k)=e_q(k-1)-d_i g_{q,i}(k-1)-g_{q,i}(k-2), \quad i=1,\ldots,N \tag{5.15b}$$

At the beginning of adaptation, it should be rated that the values of c_i and d_i must be initialized to different values. Otherwise, the update equation specified in (5.11) for each c_i and each d_i will be essentially identical and global convergence will be impossible due to system lock-up. In the simulation results given below, c_i and d_i were all initialized to their mean values obtained from [10]. Since the IIR filter

shown in (5.14) have poles on the unit circle, the gradient generation process may become unstable. To alleviate this difficulty, a constant which is very close to but less than unity can be multiplied to the terms $g_{p,i}(k-2)$ and $g_{q,i}(k-2)$ in (5.15) to maintain stability. When adaptation proceeds, the following condition (similar to (5.10)) has to be incorporated in the adaptive algorithm to ensure the minimum-phase of the analysis filter,

$$-2 < d_1 < c_1 < d_2 < c_2 < \ldots < d_N < c_N < 2 \tag{5.16}$$

Once the values of c_i and d_i are determined, the LSP coefficients can be obtained from an inverse cosine table, which is simply a table look-up operation and does not involve any extra computation. The intriguing properties of the LSP coefficients have made them one of the best feature parameter sets for digital speech transmission. Indeed, many efficient speech coding schemes have been designed to operate at low to medium data rates. For instance, Soong and Juang [10] has developed a differential quantization scheme for LSP parameters encoding while a study of quantizer design for these coefficients was reported in [11].

In order to evaluate the performance of the new predictor, a 10th order digital filter of three different forms namely the split-path cascade, the lattice and the transversal filter were used to adapt a "speech like" signal which was generated by exciting an 8th order all pole digital filter with spectrally flat Gaussian noise. The 4 pole pairs were chosen to be

$$0.95e^{\pm j30^\circ},\ 0.9e^{\pm j45^\circ},\ 0.85e^{\pm j80^\circ},\ 0.8e^{\pm j120^\circ}$$

In the split path cascade structure, δ_p and δ_q were both set to 0.01 whereas the normalized gradient lattice algorithm [12] was employed for lattice filter and the conventional Widrow's LMS algorithm was used for the transversal filter. The step sizes were being chosen to achieve their best performance. The learning curves

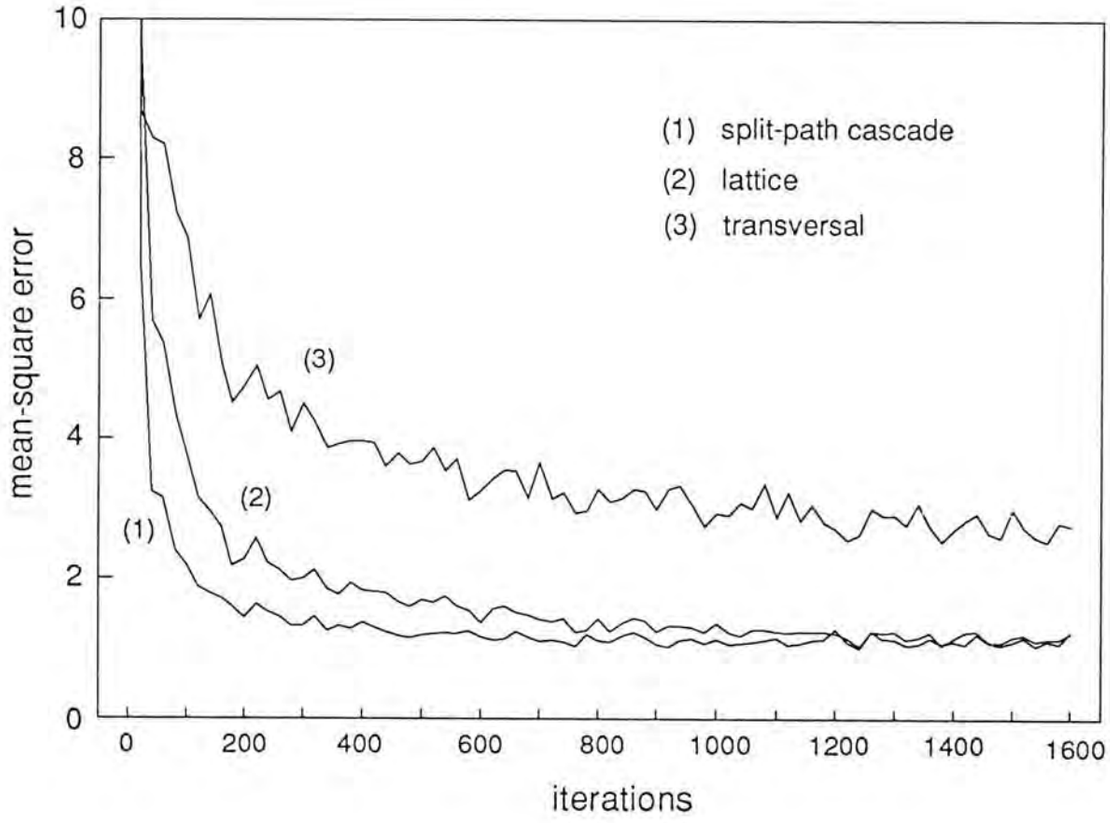


Fig. 5.11 Comparison of learning curves for different speech analysis systems

giving the convergence characteristics for these three different structures were shown in Figure 5.11. It can be seen from the simulation results that the split-path cascade method has a performance significantly better than the transversal form and offers an adaptation rate which is comparable or even superior to that of a lattice filter but requiring much less computations. Indeed, the split-path analysis filter requires only $4N$ multiplications per iteration but the lattice system needs $22N$ multiplications per iteration instead. Of course, for speech analysis, a faster convergence rate means that the adaptive method has a better tracking ability of the nonstationarity of the speech waveform.

5.6 CHAPTER SUMMARY

Generalization of the parallel and serial split adaptive systems is studied. The generalized models can achieve faster adaptation speed at the expense of a larger computational load. The performance of the parallel and serial split adaptive predictor is contrasted. The serial split predictor is preferred if the input process has a large eigenvalue spread whilst the parallel split predictor is favoured when the eigenvalue spread of the input process is small.

The dynamic convergence characteristics of a new adaptive structure formed by merging the parallel and serial predictor is investigated. It is found that the proposed new structure can achieve a much better performance. Its variant with the serial split subunits configured as cascade of second order sections was employed to extract the LSP coefficients of the speech signal. Experimental results demonstrate that its performance is comparable to the gradient lattice filter but the system complexity is greatly reduced.

REFERENCES

[1] S. S. Narayan, A. M. Peterson and M. J. Narasimha, "Transform domain LMS algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 609-615, Dec. 1983.

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

[3] L. B. Jackson and J. Bertrand, "An adaptive inverse filter for formant analysis of speech," in *Proc. IEEE ICASSP-76*, Apr. 1976, pp. 84-86.

[4] L. B. Jackson and S. L. Wood, "Linear prediction in cascade form," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, pp. 518-528, Dec. 1978.

[5] P. C. Ching and K. C. Ho, "An efficient adaptive LSP method for speech encoding," in *Proc. IEEE TENCON-90*, Hong Kong, Sept. 1990, pp. 324-328.

[6] Markel and Gray, *Linear Prediction of Speech*. Springer Verlag, 1976.

[7] N. Sugamura and F. Itakura, "Speech analysis and synthesis methods developed at ECL in NTT-From LPC to LSP", *Speech Commun.*, vol. 5, pp. 199-215, June 1986.

[8] P. E. Papamichalis, *Practical Approaches to Speech Coding*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[9] F. Itakura, T. Kobayashi and M. Honda, "A Hardware implementation of a narrow to medium band speech coding," in *Proc. ICASSP-82*, 1982, pp. 1964-1967.

[10] F. K. Soong and B. H. Juang, "Line spectrum pair (LSP) and speech data compression", in *Proc. ICASSP-84*, 1984, pp. 1.10.1-1.10.4.

[11] N. Sugamura and N. Farvardin, "Quantizer design in LSP speech analysis - synthesis," *IEEE J. Select. Areas Commun.*, vol. 6, Feb. 1988.

[12] S. T. Alexander, *Adaptive Signal Processing*. Springer-Verlag, 1986.

6 CONCLUSIONS

Adaptive filter has found applications in many areas such as plant modeling, channel equalization, noise cancellation etc. Its popularity is due to its "intelligent" nature of processing signals and the emergence of a family of powerful digital signal processors. There are two well known adaptive algorithms for FIR digital filters, namely the RLS and the LMS algorithm. The RLS algorithm has a property that the adaptation speed is independent of input signal statistics but it requires a large computational burden for its realization. Whilst the LMS algorithm, which is an approximation of the method of steepest descent, is relatively simple and easy to implement but it suffers from an inherent drawback that its adaptation characteristics is dependent on the input signals. When the autocorrelation matrix of the input process has a large eigenvalue spread, the convergence speed of the adaptation process will be significantly degraded. It is therefore required to seek an alternative adaptive filter structure that can improve the convergence speed but without a considerable increase in system complexity.

With the LMS algorithm, the performance of an adaptive filter is affected by its form of realization. It is understood that the lattice structure of an FIR filter can achieve a rapid convergence speed compared with the transversal ladder form because the input signal is successively orthogonalized in each stage of the lattice so that the correlation matrix of the effective sequence to adjust the filter parameters will become diagonal. As a result, the eigenvalue spread for the adaptation process is greatly reduced and convergence rate is greatly enhanced. Nevertheless, the cost for the improvement is a substantial increase in computational complexity. It can be shown that there exist some filter structures that can effectively diminish the

eigenvalue spread for the adaptation process to achieve a faster convergence while still requiring less computations. In this thesis, the effect on adaptation performance incurred by two simple realization schemes of an adaptive filter, namely the parallel and serial representation, is investigated.

When an adaptive filter is split into two linear phase filters that are connected in parallel, one antisymmetric while the other symmetric, the eigenvalues of the input signal are divided into two sets, each corresponds to an individual filter path. Due to the partition of eigenvalues, the eigenvalue spread of the two linear phase filters is decreased and thus improving the performance of the system. In the application of linear prediction, it is interesting to note that the backward prediction information can be incorporated naturally in the adaptation process by extending the filter length of the antisymmetric and symmetric linear phase filter by one unit with the extra coefficients fixed to -1 and 1 respectively. In this case, the gradient noise inherent in the LMS adaptation can be reduced. Consequently, in addition to the improvement due to a decrease in eigenvalue spread, convergence rate can further be doubled with a fixed tolerance level of excess MSE. The system complexity, however, is still comparable to the transversal model since only roughly M+1 extra additions are necessary for each iteration, where M is the filter order.

If an adaptive filter is represented by a cascade of two subunits, it is found that there is no particular advantage for joint process estimation. On the contrary, improvement can always be acquired for linear prediction. This is because the effective input sequence for each subunit is the filtered output of the input signal through the other subunit which has an effect of flattening the input power spectral density, which in turn reduces the eigenvalue spread for its adaptation. Experimental results illustrate that serial split of an adaptive filter is very efficient for enhancing convergence rate especially when the eigenvalue spread of the input process is large.

In addition, by setting appropriate initial conditions for the two subunits can also accelerate the adaptation speed. The extra computation in the serial split adaptive system is approximately $(M\text{-}1)/2$ multiplications and additions.

Serial split adaptive filter can also be used in the context of adaptive time delay estimation. In this application, one of the subunit in the serial split model can be placed in the desired input channel while the other in the remaining channel. The two adaptive subunits are now adjusted by minimizing the mean-square error with the error formed by the difference between their outputs. Theoretical studies of the new TDE model show that a two-fold improvement in adaptation speed can be obtained. The expense is only M addition operations. By imposing the condition that the filter parameters have the values of a sinc function, this TDE system can further improve the convergence rate. Experimental results demonstrate the feasibility and superiority of the application of constraint in adaptive TDE. An additional feature with constrained adaptation is that the possibility of convergence to the non-desired solution in the new model is avoided.

The parallel and the serial split adaptive filter can both be generalized to contain more than two subunits. In such cases, the generalized parallel split system becomes the transform domain adaptive system and the generalized serial split predictor turns out to be a linear predictor in second order cascade form. Comparison between the two splitting methods are given. The parallel split method is appropriate for joint process estimation as well as linear prediction. On the other hand, the serial split method is only applicable for linear prediction. The adaptation behavior of the parallel and serial split predictor are contrasted. When the input signal has a large eigenvalue spread, the serial predictor is desirable. If the input process has a relatively small eigenvalue spread, the parallel predictor is preferred.

As the two types of split predictor models are both superior to the transversal predictor, the possibility of merging them to form a new adaptive structure is investigated. It is found that the combined model can take the benefits of each individual predictor and provide a much better performance. The newly configured predictor is utilized for speech encoding to demonstrate its usefulness in practical applications. Simulation results show that its performance is comparable to the gradient lattice filter, but requiring much less computational burden.

The complexity of the split algorithms for joint process estimation and linear prediction are summarized in Table 6.1 and 6.2. The computational load for some other popular algorithms are also provided for comparison. Whether the increase in complexity of the split algorithms is significant or not depends on actual implementation. Nevertheless, in the worse case when the computational time for multiplication and addition operation are identical, the parallel and the serial split algorithm will never increase 25% complexity of the LMS algorithm. In addition, the combined predictor requires only extra 50% computations and the new speech analysis system needs just 50% additional operations. Comparing with the gradient lattice and the fast least square algorithm, the split algorithms are much more simpler. The split algorithms are thus an efficient adaptation rule which can substantially improve the convergence speed and can maintain simplicity for implementation.

As there are many other realizations of a FIR filter, their effects on adaptation performance need to be further studied. Some of them may be more promising than those examined in this thesis. Due to the advent of high speed digital signal processor, RLS adaptive algorithm might be preferred for applications where short filter length is required although it involves tremendous amount of computations.

	+	*	/	relative complexity (*=+)
LMS	2M+2	2M+2	0	1
PS_LMS	3M+2.5	2M+2.5	0	1.25
SS_LMS	2.5M+2.5	2.5M+2.5	0	1.25
Gradient Lattice	10M+3	11M+3	2M+1	>5
RLS	$O(M^2)$	$O(M^2)$	$O(M^2)$	O(M)
FRLS	~(7M - 11M)	~(7M - 11M)	~(3 - M)	~(3.5 - 5.5)

Table 6.1 Comparison of complexity in the application of joint process estimation

	+	*	/	relative complexity (*=+)
LMS	2M+1	2M	0	1
PS_LMS	3M+0.5	2M+0.5	0	1.25
SS_LMS	2.5M+0.5	2.5M-0.5	0	1.25
PSS_LMS	3.5M+1	2.5M	0	1.5
PSS_LMS (2ndOC)	4M+3	3M	0	1.5
Gradient Lattice	6M	6M	M	>3
RLS	$O(M^2)$	$O(M^2)$	$O(M^2)$	O(M)
FRLS	~(5M - 8M)	~(5M - 8M)	~(2 - M)	~(2.5 - 4)

Table 6.2 Comparison of complexity in the application of linear prediction

It is anticipated that different implementations will also affect the system complexity of the RLS algorithm. It might be interesting to evaluate the effect of a RLS adaptive filter with various types of realization.

Besides FIR filter, IIR can also be implemented in many different forms. Alternative realizations of direct form IIR filter has been recently examined by Nayeri and Jenkins [1]. They conclude that different filter structures lead to a change in the characteristics of the corresponding error surface, and hence to a change in the respective convergence rate and minimum mean-square error. Specifically, Shynk [2] factorizes an adaptive IIR filter into a sum of first order sections and investigates its performance in a system identification application. It is demonstrated that the parallel form IIR filter can provide robust convergence properties and robust stability monitoring with less complexity than that of the direct form. However, further research is necessary to fully explore the impact of different realization structures on an adaptive system.

REFERENCES

[1] M. Nayeri and W. K. Jenkins, "Alternate realizations to adaptive IIR filters and properties of their performance surfaces," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 485-496, Apr. 1989.

[2] J. J. Shynk, "Adaptive IIR filtering using parallel-form realizations," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 519-533, Apr. 1989.