National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Canada

Gradient Adaptive Digital Filtering: Problems and Solutions

by

Khaled A. Mayyass

A Thesis
Presented to the University of Ottawa
in Partial Fulfillment of the
Requirement for the Degree of
Doctor of Philosophy
In
Electrical Engineering
Department of Electrical Engineering
Faculty of Science and Engineering

Ottawa-Carleton Institute for Electrical Engineering

Department of Electrical Engineering
Faculty of Engineering
University of Ottawa
July, 1995

ISBN 0-612-11580-1

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

I hereby declare that I am the sole author of this thesis. I authorize the University of Ottawa to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Khaled Mayyass

I further authorize the University of Ottawa to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Khaled Mayyas

Abstract

The LMS adaptive algorithm has always been attractive to researchers in the field of adaptive signal processing due to its inherent conceptual and implementational simplicity. Unfortunately, this elegant simplicity is undermined by problems associated with the direct use of the LMS algorithm. One of the main disadvantages of the LMS is its relatively slow convergence. We deal with this problem for FIR adaptive filters by proposing two algorithms based on different approaches. The first algorithm relies on the time-varying step size approach. The step size of the algorithm is adjusted according to an error autocorrelation function. As a result, the algorithm can efficiently sense the adaptation state while maintaining the immunity against independent noise disturbance. The second algorithm is a gradient-based one that combines time- and order-updating when searching the bottom of the MSE surface, thus resulting in more efficient use of the available information. Moreover, two possibilities for the order update are considered: straightforward sequential or selective schemes. Approximate analysis of convergence and steady state performance of the two algorithms are provided.

The slow convergence problem of the LMS algorithm is also investigated for IIR adaptive filters based on output-error formulation. A new adaptive algorithm is proposed. The algorithm combines the least mean square (LMS) method with its low complexity and the least squares method with its fast convergence into a coupled LMS-LS adaptive scheme. Simulation examples indicate that the proposed scheme converges significantly faster than the LMS with minimal increase in complexity.

Next, we consider the Leaky LMS algorithm as an LMS variant proposed to deal with numerous problems that arise in direct application of LMS, including: lack of persistent excitation in the input sequence, stalling, bursting, etc. However, despite the wide spread usage of the Leaky LMS, there has been no detailed study of its performance. We present an analytical treatment of the mean square error for zero-mean Gaussian input data. Exact expressions for the second moment of the coefficient vector, the algorithm misadjustment, and rigorous conditions for MSE convergence are derived.

Finally, we consider one of the common applications of the LMS algorithm, echo cancellation in telephone networks. We investigate the presence of bursting on a back-to-back hybrid connection. Based on the essential fact that the high crosscorrelation between the input to the adaptive echo canceler and the transmitted signal at the near-end is the root cause of the bursting problem, we modify the conventional echo canceler such that under bursting circumstances the cross-correlation is substantially reduced and bursting is averted. The proposed system

ensures normal operation is not affected. Implementation details of the proposed system are studied.

To My Parents and Family

Acknowledgment

I wish to express my gratitude and deep appreciation to my advisor Dr. T. Aboulnasr, whose constructive guidance and direction brought this research into existence. I am most thankful for her moral encouragement which often went beyond academic interests.

My thanks are extended to the other members of my committee for their interest in the research.

I am grateful to the *Jordan University of Science and Technology* for awarding me the scholarship to pursue my Ph.D. degree.

I shall always be indebted to my parents, whose endless support continues to flow during all stages of my life. I thank all my sisters and brothers for their encouragement.

Finally, I feel indebted to all those who were helpful in many ways in completing this research.

Contents

List of Tables

List of Figures

List of Abbreviations

ADPCM	Adaptive differential pulse code modulation
AR	Autoregressive
ARMA	Autoregressive and moving average
BLMS	Time domain block least mean square
DCT	Discrete cosine transform
DFT	Discrete Fourier transform
DHT	Discrete Hartley transform
DRLMS	Data reusing least mean square
DTD	Double talk detector
ES-NLMS	Exponential step size normalized least mean square
ERLE	Echo return loss enhancement
FAEST	Fast a posteriori error sequential technique
FDBLMS	Frequency domain block least mean square
FFT	Fast Fourier transform
FIR	Finite impulse response
FSS	Fixed step size
FTF	Fast transversal filters
HA	Homogeneous adaptation
IA	Individual adaptation
IIR	Infinite impulse response
IIR SER	Infinite impulse response sequential regression algorithm
KLT	Discrete Karhunen-Loeve transform
LAV	Least absolute value
LMS	Least mean square
LMS-LS	Least mean square and least squares
LRS	Linear random search
LS	Least squares
MA	Moving average
MSE	Mean squared error

MVSS	Modified variable step size
NLMS	Normalized least mean square
RML	Recursive maximum likelihood
RLS	Recursive least squares
SFTF	Stabilized fast transversal filters
SGA-GAS	Stochastic gradient algorithm with gradient adaptive step size
SLOU	Selective order updating
SNR	Signal-to-noise ratio
SQOU	Sequential order updating
TDLMS	Transform domain least mean square
TW-LMS	Time-weighted least mean square
UFDLMS	Unconstrained frequency domain least mean square
VSS	Variable step size

List of Symbols

a_j	Used as a substitute for a quantity in the analysis of the leaky LMS algorithm
$a_j(n)$	jth coefficient of the AR part of the IIR adaptive filter
$\mathbf{A}$	Matrix used in the analysis of the leaky LMS algorithm
$\mathbf{A}(n)$	Coefficient vector of the AR part of the IIR adaptive filter
$A(q^{-1},n)$	Short form of the operator $\sum_{j=1}^{M} a_j(n)q^{-j}$
$A^*(q^{-1})$	Short form of the operator $\sum_{j=1}^{M} a_j^* q^{-j}$, where a_j^* is the optimal coefficient
$b_j(n)$	jth coefficient of the MA part of the IIR adaptive filter
b_1	Used as a substitute for $\sum_{i=1}^{N} \frac{1}{\mu_i}$ in the analysis of the leaky LMS algorithm
b_2	Used as a substitute of $\frac{1}{2}\sum_{i\neq}^{N}\sum_{j}^{N} \frac{1}{\mu_i}\frac{1}{\mu_j}$ in the analysis of the leaky LMS algorithm
$\mathbf{B}$	Matrix used in the analysis of the leaky LMS algorithm
$\mathbf{B}(n)$	Coefficient vector of the MA part of the IIR adaptive filter
$b(n)$	Binary random sequence $b(n) = \pm 1$
$B(q^{-1},n)$	Short form of the operator $\sum_{j=0}^{N-1} b_j(n)q^{-j}$
$B^*(q^{-1})$	Short form of the operator $\sum_{j=0}^{N-1} b_j^*$, where b_j^* is the optimal coefficient
c	Constant used to initialize the recursive computation of the inverse matrix in the LMS-LS IIR algorithm
$c(n)$	Output of the adaptive echo canceler filter
$\mathbf{C}$	Matrix used in the analysis of the leaky LMS algorithm
$\mathbf{C}(n)$	Vector used in the formulation of the fast version of the LMS-LS IIR algorithm
$d(n)$	Desired signal
$\hat{d}(n)$	Output of the adaptive IIR filter
$\hat{d}_f(n)$	Output of the filter $\frac{1}{1-A(q^{-1},n)}$, where the input is $\hat{d}(n)$
$\tilde{d}(i)$	Used as a substitute for $d(i) - \mathbf{B}^{\mathrm{T}}(i-1)\mathbf{X}(i)$

$\hat{d}(i/n)$	Output of the adaptive IIR filter at time iteration i computed using the coefficient vector at time n
$\mathbf{D}$	Matrix used in the analysis of the leaky LMS algorithm
$\mathbf{D}(n)$	Desired signal vector
$\hat{\mathbf{D}}(n)$	Output vector of the adaptive IIR filter
$\hat{\mathbf{D}}_f(n)$	Vector of the output of the filter $\frac{1}{1-A(q^{-1},n)}$, where the input is $\hat{d}(n)$
$\hat{\mathbf{D}}(i/n)$	Vector of the output of the adaptive IIR filter at time iteration i computed using the coefficient vector at time n
$\hat{\mathbf{D}}_f(i/n)$	Vector of the output of $\frac{1}{1-A(q^{-1},n)}$ at time iteration i computed using the coefficient vector at time n
$e(n)$	Error signal
$e^*(n)$	Optimal steady state error which is zero-mean and white sequence, where $E\{e^{*2}(n)\} = \epsilon_{min}$
$e_1(n)$	Error signal of the decorrelator filter
$e_2(n)$	Error signal in the proposed echo canceler, where $e_2(n) = e_1(n) - c(n)$
$e_i(n)$	$i$the stage partial a posteriori error
$e_p^b(n)$	Backward prediction error of the pth stage of the lattice filter
$e_p^f(n)$	Forward prediction error of the pth stage of the lattice filter
$E\{\ \}$	Expectation operator
$\mathbf{f}(n)$	Coefficient vector of the forward prediction filter in the fast LMS-LS IIR algorithm
$\mathbf{G}(n)$	Vector used in the formulation of the fast version of the LMS-LS IIR algorithm
$\mathbf{G}(n)$	Vector used in the analysis of the leaky LMS algorithm, where $\mathbf{G}(n) = \mathbf{B}\mathbf{Z}_2(n)$
$G_j(n)$	jth component of $\mathbf{G}(n)$
h^*	Singie optimal coefficient of a hybrid
$\hat{h}(n)$	Coefficient error; $\hat{h}(n) = h^* - h(n)$
$\mathbf{h_N}(n)$	Coefficient vector of the adaptive echo canceler filter
$H(z)$	Transfer function
$\mathbf{I}$	Identity matrix
$J(n)$	Objective function minimized by the leaky LMS algorithm
$k_j(n)$	jth coefficient of the MA filter $\frac{1}{Q(q^{-1},n)}$, where $j = 0, \ldots, \infty$
$k_p(n)$	Reflection coefficient of the pth stage of the lattice filter
$\mathbf{k}(n)$	Vector used in the formulation of the fast version of the LMS-LS IIR algorithm
$\mathbf{K}$	Matrix used in the analysis of the leaky LMS algorithm

$m(n)$	Scalar used in the formulation of the fast version the LMS-LS IIR algorithm
M	Misadjustment
M	Order of the linear predictor or the AR part of the IIR adaptive filter
$M(n)$	Diagonal matrix containing the effective time-variant step sizes in the SQOU algorithm
$\hat{M}$	Steady state diagonal matrix of the step size matrix $M(n)$
n	Current time index
N	Order of the FIR adaptive filter or the MA part of the adaptive IIR filter
$\mathbf{N}(n)$	Vector of gradient noise
$p(n)$	Time-averaging estimate of the autocorrelation of $e(n)$ and $e(n-1)$ in the MVSS algorithm
$\mathbf{P}$	Crosscorrelation vector between the input signal $x(n)$ and the desired signal $d(n)$
$\dot{\mathbf{P}}$	Rotated crosscorrelation vector between the input signal $x(n)$ and the desired signal
$\mathbf{P}(n)$	Vector used in the formulation of the fast version of the LMS-LS IIR algorithm
$\dot{P}_j$	jth component of the rotated crosscorrelation vector
q	Scalar used as substitute of a quantity in the analysis of the MVSS algorithm
q^{-j}	Delay operator; $q^{-j}x(n) = x(n-j)$
$\mathbf{Q}$	Matrix of eigenvectors of the input data autocorrelation matrix $\mathbf{R}$
$\mathbf{Q}(n)$	Matrix used in the derivation of the LMS-LS IIR algorithm
$Q(q^{-1}, n)$	Short form of the operator $1 - \sum_{i=1}^{M} a_i(n)q^{-i}$
$Q^*(q-1, n)$	Short form of the operator $1 - \sum_{i=1}^{M} a_i^* q^{-i}$
$r(n)$	Output error of the adaptive echo canceler
$\mathbf{R}$	Input data autocorrelation matrix
$s(n)$	Transmitted near-end signal in an adaptive hybrid
s_1	Used as a substitute of b_1 in the analysis of the leaky LMS algorithm
s_2	used as a substitute of $b_1^2 - 2b_2$ in the analysis of the leaky LMS algorithm
$v(n)$	Near-end signal in an adaptive hybrid
$\mathbf{V}(n)$	Coefficient error vector
$\dot{\mathbf{V}}(n)$	Rotated coefficient error vector
$\dot{V}_j(n)$	jth component of the rotated coefficient error vector

$w(n)$	Far-end signal in an adaptive hybrid
$w_i(n)$	ith weight of the coefficient vector $\mathbf{W}(n)$
$\mathbf{W}(n)$	Coefficient vector of the adaptive FIR filter
$\mathbf{W}^*$	Optimal coefficient vector
$\dot{\mathbf{W}}^*$	Rotated optimal coefficient vector
$\dot{W}_j^*$	jth component of the rotated optimal coefficient vector $\dot{\mathbf{W}}^*$
$\mathbf{W}^i(n)$	Weight vector of the ith stage in the formulation of the *SQOU* algorithm
$x(n)$	Input data sequence
$\mathbf{X}(n)$	Input data vector
$\dot{\mathbf{X}}(n)$	Rotated input data vector
$y(n)$	Output of the hybrid
y	Used as a substitute of a quantity in the analysis of the MVSS algorithm
$\mathbf{Y}$	Diagonal matrix used in the analysis of the leaky LMS algorithm
$\dot{\mathbf{Y}}$	Vector used in the analysis of the leaky LMS algorithm
z^{-1}	Unit-sample delay used in the z-transform
$\mathbf{Z}$ (n)	State vector used in the analysis of the leaky LMS algorithm
$\mathbf{Z_1}(n)$	Second moment vector of $\dot{\mathbf{V}}(n)$
$\mathbf{Z_2}(n)$	Mean vector of $\dot{\mathbf{V}}(n)$
α	Forgetting factor used in the step size equation of the MVSS algorithm, chapter 3
α	Used as a substitute for a quantity in the analysis of the SQOU algorithm, chapter 4
α	Echo attenuation value through the telephony loop, chapter 7
$\chi(\mu)$	Function of the step size μ used in the analysis of the leaky LMS algorithm
β	Exponential weighting parameter used in the step size equation of the MVSS algorithm, chapter 3
β	Used as a substitute of $\frac{1}{1+2\mu\sigma_x^2}$ in the analysis of the SQOU algorithm, chapter 4
β	Adaptation step size of the gradient lattice algorithm, chapter 7
γ	Parameter used in the step size equation of the MVSS algorithm chapter 3
γ	Leakage factor, chapter 6
γ^*	Optimal value for γ for the MVSS algorithm in nonstationary environment, chapter 3
δ	Constant used in the initialization of the fast version of the LMS IIR algorithm

$\epsilon(n)$	Mean squared error performance function
$\epsilon(n)$	Scalar used in the formulation of the fast version of the LMS-LS IIR algorithm, chapter 5
$\epsilon_{LS}(n)$	Least squares performance function
ϵ_{min}	Minimum value of the mean squared error
$\epsilon_{ex}(n)$	Excess mean squared error
$\epsilon_f(n)$	Scalar used in the formulation of the fast version of the LMS-LS IIR algorithm
$\epsilon_0(n)$	Scalar used in the formulation of the fast version of the LMS-LS IIR algorithm
$\eta(n)$	Stationary white noise process of zer-mean
$\eta_0(n)$	Scalar used in the formulation of the fast version of the LMS-LS IIR algorithm
Γ	Vector of the eigenvalues of the autocorrelation matrix $\mathbf{R}$
λ	Exponential weighting factor in the RLS and the LMS-LS IIR algorithms
λ_j	jth eigenvalue of the autocorrelation matrix $\mathbf{R}$
λ_{max}	Maximum eigenvalue of the autocorrelation matrix $\mathbf{R}$
λ_{min}	Minimum eigenvalue of the autocorrelation matrix
Λ	Diagonal matrix of the eigenvalues of the autocorrelation matrix
μ	Convergence step size in the gradient search algorithms
$\mu(n)$	Time varying step size
μ_i	Roots of $\chi(\mu)$ in the analysis of the leaky LMS algorithm
$\xi(n)$	Independent zero-mean noise
ρ_j	Poles of $\chi(\mu)$ in the analysis of the leaky LMS algorithm
μ^*	Step size value that determines stability for MSE and mean algorithm convergence
σ_n^2	Variance of the noise process $\eta(n)$
σ_x^2	Variance of the input signal
τ_i	Time constant of the ith mode of the adaptive algorithm
$\nabla(n)$	True gradient of the performance function
$\hat{\nabla}(n)$	Estimate of the true gradient

Chapter 1

Introduction

The ability of adaptive filters to identify unknown physical systems and to track changes in time-varying environments have made them a powerful tool for numerous practical applications. These include: inverse modeling [113], channel equalization [87], echo cancellation [50, 79], noise cancellation [110], adaptive line enhancement [32], adaptive waveform coding [64], adaptive spectrum estimation [7], adaptive array processing [41], etc. Popular adaptive algorithms can be generally classified into two main categories: gradient search and least squares (LS) [1]. The LS category encompasses the family of recursive LS (RLS) algorithms which offer faster convergence compared with algorithms from the gradient class. This has made them attractive in applications that need rapid convergence, such as adaptive differential code modulation (ADPCM), fast start-up of modem equalizers, and fast initialization of echo cancelers for full-duplex data transmission [19]. However, results show that the RLS family is not always the best [16], since it

requires high computational and implementational costs, and generally suffers stability complications caused by the accumulation of finite precision errors in the algorithm variables. These complications often lead to unstable systems in finite precision environments, particularly in low signal-to-noise ratio cases and when the input signal is highly colored [18, 23, 61, 100]. For these reasons and due largely to its simplicity, robustness, low complexity, and ease of implementation, the gradient search class, particularly the LMS algorithm, remains widely adopted and used in numerous areas including signal processing, communications, control, etc. Those advantages, however, are overshadowed by some inherent shortcomings of the LMS algorithm and other problems associated with its direct use in some applications.

1.1 Research motivation

Considerable effort has been directed towards resolving some of the LMS algorithm limitations and problems in different environments and applications, where numerous techniques and variants of the LMS have been proposed. The work here is a continuum to this track, focusing on several of those important practical problems and limitations. Our aim is to present robust solutions to some of these problems, along with analytical investigations of algorithms proposed to deal with them.

It is known that the convergence speed of the standard LMS algorithm is largely dependent on the input data statistics, leading to slow and nonuniform convergence when the input eigenvalue disparity is high [1, 7, 46, 111, 113]. This necessitates

the usage of a large step size value in the LMS to speed up the convergence process. Unfortunately, the penalty incurred as a result of using a large step size value is a corresponding increase in the steady state mean squared error (MSE). As a result, researchers have constantly looked for alternative means to improve its performance. One common approach is to employ a time-varying step size in the standard LMS weight update recursion, [11, 45, 56, 68, 100]. The motivation is to use large step size values when the algorithm is far from optimal solution, thus speeding up algorithm convergence rate. When the algorithm is near the optimum, small step size values are used to compromise between low level of misadjustment and acceptable tracking. Experimental results show that the performance of these existing variable step size (VSS) algorithms is highly sensitive to the noise disturbance [26, 100]. Additionally, their advantageous performance over the LMS algorithm is generally attained only in high signal-to-noise environment. Since noise is a fact in any practical system, the usefulness of any adaptive algorithm is judged by its performance *in the presence of this noise.*

In the first part of this research, we will propose a new VSS LMS-type algorithm with new features making it a powerful alternative to the current VSS algorithms in practical applications [72]. The step size of the new algorithm is controlled by an error autocorrelation function and, as a result, the algorithm can effectively adjust the step size while maintaining the immunity against independent noise

disturbances. Another significant feature of the proposed algorithm is that the addition of a new parameter pertaining to error autocorrelation function allows controlling misadjustment and convergence time more independently without the need to compromise between them as in other existing VSS algorithms.

Next, we proceed to investigating alternative means for establishing faster convergence algorithm to deal with the LMS main problem of slow convergence when the input signal is correlated. A gradient search algorithm is proposed in the second part of this research. The algorithm is based on the method of steepest descent but utilizes a new technique in searching for the global minimum of the MSE surface. The aim of this technique is to make efficient use of the available data. In essence, the algorithm has a stage-wise structure where information is carried on from stage to stage via a weight vector and a *partial a posteriori* error. The result is a rapid convergent algorithm for correlated inputs, and fast tracking capabilities of time-varying environments. The algorithm mean convergence and steady state MSE will be analyzed for zero-mean white Gaussian inputs. The analysis results will highlight desirable features of the algorithm. Further, enhancement of the algorithm performance will be presented by suggesting a selective updating approach instead of the conventional sequential one. These two approaches will be examined and assessed under different operational conditions.

The above work concentrated on FIR adaptive filters. These are very common

because of their guaranteed stability and the availability of clear analysis of their convergence. However, IIR filters are recognized as a powerful alternative to FIR filters due their inherent advantages of low complexity and the ability to model physical systems efficiently. As such, they have generally received considerable attention as a research topic [2, 3, 27, 37, 42, 43, 49, 75, 82, 97, 103]. Existing algorithms that are based on the output-error formulation generally explicitly or implicitly follow a gradient direction path in searching the output-error surface. Due to the recursive nature of this formulation, the resulting performance function is highly nonlinear leading to numerous convergence problems.

In the third part of this work, we demonstrate that it is possible to get around this problem by separating the search of the numerator and denominator coefficients of the adaptive IIR filter. The moving average part of the adaptive filter (numerator) defines a quadratic error function and, therefore, the simple LMS algorithm is used in updating it. On the other hand, the autoregressive part defines a nonlinear error function and is searched using a LS approach. The increase in complexity due to the use of the LS is minimal since the number of poles is generally fairly low. With some approximations, we will derive a recursive solution for the denominator coefficients. The suggested approximations are only true after convergence, nevertheless, extensive simulations show the fast convergence of the proposed algorithm in all cases. This is the main advantage of the proposed

algorithm. However, it will also be shown that the convergence rate is highly unaffected by the input signal statistics or the initial values of the adaptive filter weights. This is to be expected as a result of the use of the LS approach in the update of the denominator coefficients. Simulations will also show the ability of the algorithm to achieve global convergence for all cases except the white input, insufficient order case.

In the fourth part of this research, we consider one of the LMS variants proposed to improve its performance in practical applications. Experience with the standard LMS behavior in practical applications indicates that problems exist in the LMS leading to performance degradation when the input signal lacks persistence in excitation. The leaky LMS algorithm was found to have the capacity to alleviate this common problem [91]. Leaky LMS was also successfully applied to various other problems such as, stalling problem [7], bursting problem in telephone networks [36], and the self-cancellation problem in adaptive antenna [102]. In all above applications, the use of the leaky LMS has been based on simulation. Elementary attempts to provide an analytical framework for algorithm performance have been limited to analysis of weight vector mean convergence. Simulation experiments, however extensive, cannot provide us with the same insight achievable through analysis. The impact of input parameters as signal statistics, step size value, leakage factor value, and adaptive filter length on algorithm performance

are also better comprehended and quantified through analytical understanding. It is also recognized that conditions for weight vector mean convergence do not guarantee mean squared weight vector convergence nor do they ensure finite mean square error (MSE). An analytical investigation of the leaky LMS algorithm in the mean square is thus beneficial and is the subject of the fourth part of this research.

We will derive a recursive equation characterizing the transient evolution of the weight vector covariance matrix [73]. We will then be able to describe an exact equation of the second moment of the weight vector, and derive necessary and sufficient conditions for stability of this equation that will also guarantee MSE convergence. Further, we will carry out the difficult task to transform these conditions to tangible measures, where we will be able to provide new direct bounds on the adaptation step size of the algorithm. Additionally, exact expressions for the steady state MSE and algorithm misadjustment are derived. All these derivation are done without making any assumptions about the nature of the autocorrelation matrix of the input data and, hence, our results describe closely the behavior of the practical system.

Finally, we consider echo cancellation in telephone networks as one of the applications where the adaptive LMS echo canceler has been successfully applied to remove a significant portion of the echo returning to the near-end speaker. However, it was found that under certain circumstances of high correlation between

the input to the adaptive echo canceler and the near end signal combined with the inherent feedback structure of the telephone loop, the adaptive echo canceler behaves improperly leading the overall system to becoming unstable. This has been referred to as bursting. This problem has been the focus of much attention where it has been recently addressed in [24, 36, 50, 92, 93, 109]. Due to the complexity of analysis, only very simple cases of first order hybrids and echo cancelers were addressed in [24, 28, 92, 93]. Solutions provided in these cases generally required knowledge of the hybrid and were not generalizable to higher orders. A more practical case was considered in [109], nevertheless, their solution restricted the nature of the near-end signal to be a tone signal and, also, was based on some predetermined thresholds.

We will exploit the required conditions under which the adaptive echo canceler operates properly. We will then propose a modification to the conventional adaptive system setup that basically ensures these conditions persist even under bursting circumstances. As such, the proposed adaptive hybrid can inherently avoid bursting and, yet, normal operation is not affected. The approach is directly generalizable for practical hybrids and does not require knowledge of system or preselection of thresholds.

1.2 Thesis organization

This thesis is organized as follows. Chapter 2 provides a review of popular adaptive techniques as well as algorithms and structures associated with them.

In the first part of chapter 3, we will study the performance properties of the VSS algorithm developed in [56] in *a noisy environment*. We will then propose a new VSS LMS algorithm and highlight several of its aspects that make it favorable over the current VSS algorithms. Convergence and steady state analysis are introduced to establish an analytical framework of the proposed algorithm. Simulation examples are presented to compare algorithm performance with the standard LMS and other VSS algorithms, and also to verify the derived results.

In chapter 4, we present a new gradient technique to search the bottom of the mean squared error (MSE) surface. The algorithm is formulated and a complete weight update recursion of the proposed algorithm is established. Then we study the limit on the step size used for coefficients update through conducting approximate steady state performance evaluation of the algorithm. The performance of the proposed approach is further enhanced by careful selection of the order in which the weights are updated. Simulation examples are then presented to verify the performance of the proposed algorithm.

Chapter 5 deals with the application of the LMS adaptive algorithm to the IIR filters. We will discuss various problems that immediately cause the very

slow convergence speed associated with utilizing gradient-based algorithms. A new adaptive scheme for the output-error approach will be formulated. In this approach, the adaptation of the moving average (MA) part and the autoregressive part of the adaptive IIR filter is based on different minimization criteria, namely the mean squared error (MSE) and the least squares (LS) respectively. Then, two implementations of the same algorithm using different sequences of computations will be presented, along with a detailed discussion of their differences. Finally, several examples are carried out to illustrate the potential of the proposed scheme in speeding up the convergence of the IIR adaptive filters.

Chapter 6 presents a MSE detailed analytical investigation of the leaky LMS algorithm. We start by describing the leaky LMS algorithm and then evaluate its convergence in the mean. Mean square convergence analysis is also carried out where new conditions for MSE stability and an exact expression for the misadjustment are derived. Simulation examples are presented to verify theoretical results.

In the first part of chapter 7, we introduce a description of the bursting phenomenon in telephone networks with adaptive hybrids accompanied by a review of the current methods developed to solve this problem. By identifying the root causes of bursting in adaptive hybrids, we propose a new adaptive hybrid setup that can prevent bursting. The operation of the proposed system is studied along

with different issues related to its implementation. An analytical understanding of the operation of the new setup under normal conditions of persistent excitation and no disturbance is also given. Finally, examples are provided to verify system performance.

Finally, chapter 8 includes the conclusions of the thesis as well as some ideas and suggestions for further research.

Chapter 2

Review of Adaptive Techniques and Algorithms

This chapter reviews different approaches for adaptive filtering. Our purpose is to cover concisely the basics of adaptive techniques needed as a background for the understanding of this thesis. Mainly, we will discuss two techniques: gradient search and least squares. The review will also include algorithms and related variants emerging from these techniques.

2.1 Gradient search technique

Gradient-based algorithms are designed to minimize the mean squared error (MSE), which is one of the more commonly used performance criteria. The MSE is given by

$$\epsilon(n) = E\{e^2(n)\} \tag{2.1}$$

where

$$e(n) = d(n) - \mathbf{W}^{\mathrm{T}}(n)\mathbf{X}(n) \tag{2.2}$$

is the scalar output error at time instant n, $d(n)$ is the scalar desired signal, $\mathbf{X}(n) = [x(n), \ldots, x(n-N+1)]^{\mathrm{T}}$ is the input signal vector and $\mathbf{W}(n) = [w_1(n), \ldots, w_N(n)]^{\mathrm{T}}$ is the weight vector. The minimum MSE problem is to locate the optimal weight vector $\mathbf{W}^*$ resulting in the best estimate of the desired signal in the MSE sense; in mathematical terms

$$\frac{\partial \epsilon(n)}{\partial \mathbf{W}(n)} \mid_{\mathbf{W}(n)=\mathbf{W}^*} = 0 \tag{2.3}$$

Assuming that $x(n)$ and $d(n)$ are jointly stationary signals, and using Eqs. (2.1) and (2.2), $\epsilon(n)$ can then be expanded in the form

$$E\{e(n)^2\} = \sigma_d^2 - 2\mathbf{W}^{\mathrm{T}}(n)\mathbf{P} + \mathbf{W}^{\mathrm{T}}(n)\mathbf{R}\mathbf{W}(n) \tag{2.4}$$

where σ_d^2 is the mean square power of $d(n)$, $\mathbf{R} = E\{\mathbf{X}(n)\mathbf{X}^{\mathrm{T}}(n)\}$ is the input signal autocorrelation matrix, and $\mathbf{P} = E\{d(n)\mathbf{X}(n)\}$ is the crosscorrelation vector between the input and the desired signal. It is obvious from Eq.(2.4) that $E\{e^2(n)\}$ for FIR filters is a quadratic function with respect to the weight vector $\mathbf{W}(n)$ forming a bowl-shaped surface commonly called the MSE surface. The MSE surface has a unique minimum located at $\mathbf{W}^*$. This essential property is a straightforward consequence of a positive definite autocorrelation matrix $\mathbf{R}$ of the input signal, a condition that naturally holds in physical applications [46, 113]. A direct method to find $\mathbf{W}^*$ is to solve Eq.(2.3) for $\mathbf{W}^*$, this yields

$$\mathbf{R}\mathbf{W}^* = \mathbf{P} \tag{2.5}$$

This set of equations is often called the normal equation [1]. Hence, from Eq.(2.5)

$$\mathbf{W}^* = \mathbf{R}^{-1}\mathbf{P} \tag{2.6}$$

We note from Eq.(2.6) that uniqueness of $\mathbf{W}^*$ is an immediate consequence of the nonsingularity of $\mathbf{R}$. The evaluation of Eq.(2.6) requires matrix inversion which is generally computationally intensive. Another way to find $\mathbf{W}^*$ is to search for the bottom of the MSE surface and is accomplished in a time-recursive manner. This is done by iteratively updating $\mathbf{W}(n+1)$ in terms of $\mathbf{W}(n)$ according to the following relation [1, 46, 113]

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \mu\frac{\partial\epsilon(n)}{\partial\mathbf{W}(n)} \tag{2.7}$$

where

$$\begin{aligned}\frac{\partial\epsilon(n)}{\partial\mathbf{W}(n)} &= -2\mathbf{P} + 2\mathbf{R}\mathbf{W}(n) \\ &= -2E\{e(n)\mathbf{X}(n)\}\end{aligned} \tag{2.8}$$

and μ is the adaptation step size. Eq.(2.7) is intuitively understood by viewing the MSE surface as a function of a 2×1 coefficient vector in three-dimensional space as shown in Fig. 2.1. It indicates that $\mathbf{W}(n)$ can reach $\mathbf{W}^*$ at the bottom of the surface by moving an appropriate step along a direction opposite to the gradient at each time instant n, hence the name "gradient-based techniques".

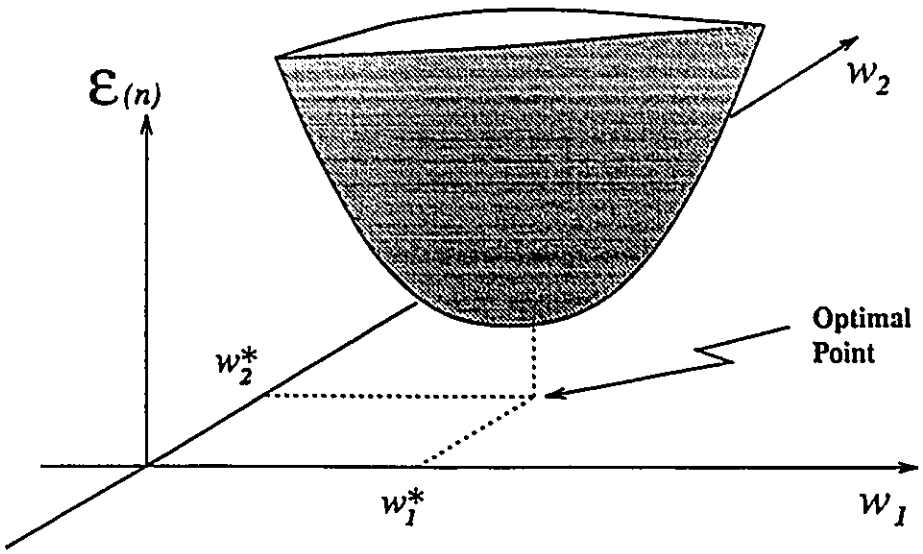

Fig. 2.1 The MSE surface in three-dimensional space.

2.1.1 The LMS algorithm

Using Eq.(2.7) requires an estimate of both $\mathbf{R}$ and $\mathbf{P}$ at each time iteration n which bears a high computational load and is impractical for real-time signal processing. For this reason, the conventional least mean squares (LMS) is one of the most widely used algorithms. This algorithm replaces $E\{e(n)\mathbf{X}(n)\}$ with the instantaneous gradient estimate $e(n)\mathbf{X}(n)$. This leads to the recursion [111]

$$\mathbf{W}(n+1) = \mathbf{W}(n) + 2\mu e(n)\mathbf{X}(n) \tag{2.9}$$

The simplicity of the LMS is evident from Eq.(2.9), since it depends only on the acquired data to compute the recursion. Additionally, the computational and storage requirements are low. The step size in Eq.(2.9) plays a crucial role in determining the stability, convergence rate, and final steady state MSE of the algorithm. It is easily shown from Eq.(2.9) that LMS converges exponentially in the mean if μ is bounded by [1, 46, 111, 113]

$$0 < \mu < \frac{1}{\lambda_{max}} \tag{2.10}$$

where λ_{max} is the maximum of the eigenvalues λ_i, $i = 1, \ldots, N$, of the autocorrelation matrix $\mathbf{R}$. It is also easy to find that for a small μ, the time constant associated with the mean of the ith uncoupled mode of the first order difference equation (2.9) is [1, 46, 111, 113]

$$\tau_i \approx \frac{1}{2\mu\lambda_i} \tag{2.11}$$

and, therefore, the slowest uncoupled mode is the one associated with minimum eigenvalue λ_{min}. We are only concerned with the actual mean weight $E\{\mathbf{W}(n)\}$ which is a linear combination of the uncoupled modes. Therefore, the convergence time of $E\{\mathbf{W}(n)\}$ will be inversely proportional to the smallest eigenvalue λ_{min}. Additionally, we note from Eq.(2.11) that the convergence time of $E\{\mathbf{W}(n)\}$ is also inversely proportional to the selected value of μ. Since μ cannot be chosen greater than $\frac{1}{\lambda_{max}}$, then we can intuitively conclude that the convergence time of $E\{\mathbf{W}(n)\}$ is directly proportional to $\frac{\lambda_{max}}{\lambda_{min}}$, which is often called the eigenvalue disparity or the condition number of $\mathbf{R}$. This reveals an important but undesirable inherent property of the gradient-based algorithms namely that their convergence rate is considerably dependent on the input signal statistics. When the input is white, corresponding to eigenvalue disparity of one, they provide their fastest convergence speed. Another interpretation is that the gradient vector always points to the bottom of the MSE surface for white inputs leading to the fastest convergence rate of algorithms from the gradient class. For high eigenvalue spread, or equivalently, when the input signal is spectrally nonflat, these algorithms suffer from very slow convergence.

Another limitation of the LMS algorithm is its inability to satisfy the conflicting goals of rapid convergence and low misadjustment; a fact that is obvious from recalling that the algorithm convergence time is inversely proportional the step

size μ. Larger μ results in fast convergence, however, this in turn causes larger fluctuations in steady state and, consequently, larger misadjustment levels [45].

2.1.2 LMS related algorithms and techniques

Variants of the LMS algorithm have been introduced in an attempt to overcome some of its drawbacks while maintaining its attractive properties of robustness, simplicity, and low complexity. In the following, we present a brief description of some of those widely used techniques and algorithms.

2.1.2.1 The Normalized LMS and its variants

The first modified LMS algorithm was introduced in [80], which was then named as the "fundamental method". The algorithm is now known in the literature as the Normalized LMS (NLMS) [8](sometimes it is called αLMS [94]). Simply, the NLMS modifies the LMS through normalizing the LMS fixed step size by an estimate of the input signal power so as to gain robustness and insensitivity to variations in the input signal power. Additionally, the NLMS has been shown to outperform the LMS in convergence speed, an argument that is drawn from the projection concept which can be applied on the NLMS as one of the projection algorithms for parameter estimation of deterministic systems [39] (p. 50-55).

In 1986, Mikhael et al. [77] rederived the NLMS version for equation-error based adaptive filters by proposing constraints on the choice of the adaptation step size.

They called it Homogeneous Adaptation (HA) algorithm since the algorithm uses the derived time-varying step size for all filter coefficients. They also extended the algorithm to Individual Adaptation (IA) where each filter coefficient uses different step size.

In [65], Makino et al. proposed the exponentially weighted step size NLMS (ES-NLMS) algorithm for acoustic echo cancelers. The algorithm is based on the observation that room impulse responses attenuate exponentially. Accordingly, each weight uses different time-invariant step size and, also, those step sizes are exponentially decaying similar to the room characteristics. Simulation results showed that the ES-NLMS algorithm provide a better convergence speed than the NLMS in acoustic echo applications [65, 114].

2.1.2.2 Variable step size (VSS) LMS algorithms

A different approach has been applied to device efficient LMS-type algorithms [11, 45, 56, 68, 100]. The idea is to use large step size value in Eq.(2.9) when the adaptive filter weight vector is far from its optimal value to achieve fast convergence. Near optimality, smaller step size value is used to obtain lower misadjustment level. The objective is to solve the inherent inability of the standard LMS of satisfying the opposing goals of fast convergence and low misadjustment due to employing a fixed step size in its update recursion.

Algorithms based on this approach are usually referred to as variable step size

(VSS) algorithms. These algorithms can be distinguished by the criterion that each algorithm utilizes to measure the adaptation process state in order to adjust its step size. The performance of the algorithm is significantly dependent on the associated criterion used. These algorithms may also use different adaptation step size for each filter weight. This was done in [45] where the algorithm monitors the sign changes of the gradient component in the direction of each filter weight. If it keeps the same sign for m_1 successive number of samples, then the algorithm concludes that it is still far from optimality, and accordingly, the step size associated with that weight is increased to speed up adaptation. Conversely, when the gradient component changes its sign for m_0 consecutive samples, the algorithm expects it has reached the optimal weight and started to fluctuate around it. Thus, the step size is decreased to obtain less excess MSE. However, results in [26] indicate that the step size evolution of the individual weights follow roughly the same trend and can be approximated by a single step size for all weights. This result is also found to be true for the algorithm in [68]. As thus, not much gain is expected from utilizing different step size for each weight using this criterion.

A major drawback of the current VSS algorithms is that their performance is highly sensitive to noise, a fact that limits their usefulness for practical applications [26]. This subject is to be discussed in more details in chapter 3.

2.1.2.3 Algorithms using additional terms in the LMS recursion

The weight update equation of all of the above mentioned algorithms can collectively be described by

$$\mathbf{W}(n+1) = \mathbf{W}(n) + 2\mathbf{M}(n)e(n)\mathbf{X}(n) \tag{2.12}$$

where the $\mathbf{M}(n)$ is a diagonal matrix of step sizes of individual weights, which has the form

$$M(n) = \text{diag}\{\mu_1(n),\ \mu_2(n), \ldots,\ \mu_N(n)\} \tag{2.13}$$

Each algorithm is recognized by the way it defines Eq.(2.13). For example, the NLMS algorithm uses $\mu_1(n) = \mu_2(n) = \ldots = \mu_N(n) = \frac{\alpha}{\mathbf{X}^T(n)\mathbf{X}(n)}$.

A different approach to improve the LMS performance is to add a new term to its update equation in Eq.(2.9). The algorithms reported utilizing this technique are: the Momentum LMS algorithm [86, 88], the LMS-Sine algorithm [53], and the Leaky LMS algorithm [35].

The Momentum LMS algorithm has the following weight update expression,

$$\mathbf{W}(n+1) = \mathbf{W}(n) + 2\mu e(n)\mathbf{X}(n) + \alpha(\mathbf{W}(n) - \mathbf{W}(n-1)) \tag{2.14}$$

where α is a scalar satisfying $|\alpha| < 1$. The Momentum LMS was initially introduced in [86], where it was thought that the algorithm possesses better speed characteristics than the standard LMS. It was shown later in [88] that the algorithm does

not provide any improvement in convergence speed over the LMS, however, it does have a smoothing property that makes it useful in some applications [86].

While the Momentum LMS modifies the regular LMS by adding a linear term to its recursion, the LMS-Sine algorithm incorporates a nonlinear term that is a function of the instantaneous gradient and weight vector [53]. The nonlinear term is governed by an exponentially decaying function that diminishes to zero when n exceeds N, N being the order of the adaptive filter. Consequently, the advantages of this algorithm are only expected in limited cases when the input signal is white and the unknown system is time invariant.

As we have discussed earlier, a positive autocorrelation matrix $\mathbf{R}$ of the input signal is an essential requirement for convergence of the LMS algorithm to the unique optimal solution. Lack of persistence in excitation of the input sequence results in a singular matrix $\mathbf{R}$. This leads to the convergence of the regular LMS to different solutions. The incorporation of a leakage term in the LMS update recursion solves this problem at the expense of increased misadjustment and coefficients bias [35, 46, 57, 113]. A complete coverage of the Leaky LMS algorithm will be discussed chapter 6.

2.1.2.4 Transform domain LMS adaptive filters

Transform domain LMS adaptive filters have also attracted the interest of the signal processing community. In these algorithms, the input signal is transformed before being applied to the adaptive structure using some kind of transform, usually or-

thogonal one, such as the discrete Fourier transform (DFT), the discrete cosine transform (DCT), the discrete Hartley transform (DHT), the discrete Karhunen-Loeve transform (KLT), etc. The purpose of this transformation is to generate a signal of approximately uncorrelated samples. However, implementation of these algorithms with fixed step sizes will not bring about any improvement over the time domain LMS algorithm. It may actually result in lower performance since the power in the transformed sequence will still vary from sample to sample (for correlated input signals) [40, 99]. Therefore, a self-orthogonalization technique should be applied to obtain a white-like input signal. This technique was introduced for the first time in the structure proposed in [81], in which the algorithm was referred to as the transform domain LMS (TDLMS) algorithm.

Transform domain LMS algorithms require large computational overhead if adaptation is performed on a sample-by-sample basis. Alternatively, considerable computational saving is expected for a block-wise adaptation when a fast discrete transform is used (like the FFT), and when the adaptive filter has a very long impulse response as in acoustic echo cancellation application [52] as well as other applications [83, 107]. Such block-type algorithms, e.g. the unconstrained frequency domain LMS (UFDLMS) algorithm [66], which are suboptimal implementations of the time domain block LMS (BLMS) algorithm [21]. They use circular convolution instead of linear convolution. An efficient implementation of the BLMS based on

the DFT is found in [22, 29]. A block-type version of the FDLMS has also been introduced under the name of frequency domain BLMS (FDBLMS) [58].

Since the block-type algorithms incremented their weight vector only at the end of every block, the constant step size stability boundaries are reduced [22, 30]. This limits their convergence speed capabilities. Moreover, the algorithm weight vector will not be able to track instantaneous variations in nonstationary systems resulting in degraded performance [96].

2.1.2.5 Subband techniques

The adaptive filter is required to model an unknown system with a very long impulse response. This is typical in some applications, e.g. acoustic echo cancellation. In this case, the adaptive filter needs more than 1000 taps to achieve a significant reduction in the echo level. Moreover, the adaptation process is controlled by a speech signal rather than white signal and the room response can be nonstationary. Those problems result in a very heavy computational cost and convergence complications that make the realization of a full band adaptive algorithm extremely difficult.

Subband adaptive filtering has provided an elegant solution to the above problems and therefore has received a considerable attention from researchers [34, 78, 85]. It has mostly been applied to acoustic echo cancellation [15, 34, 89] since it transforms the problem of modeling a very long filter into the task of identifying

several smaller filters at a lower rate. Also, the technique has been applied recently to acoustic noise cancellation [105, 106].

A subband adaptive structure is shown in Fig. 2.2. In subband adaptive filters, the input and the desired signals are split into several frequency bands via a set of filters known as analysis filters, where in each band the signal is decimated before the adaptive filtering process is applied to that band [78]. This leads to a reduction in the computational requirements. Also, the power spectral density in each band can be smoother than in the full band case thus leading to a smaller eigenvalue spread. Hence, faster convergence speed is expected especially when the step size in each band is normalized by the power in that band. The subband error signals are then upsampled and passed through a set of filters known as synthesis filters, and summed to produce the full band error.

However, the use of downsampling results in undesirable aliasing effects especially when the filter banks are critically downsampled. In order to avoid the aliasing problem, several measures were introduced such as decimation by a factor smaller than the number of bands [85]. Another approach consists of adding adaptive cross terms filters between subbands [34]. This, however, will increase the computational complexity and reduce convergence speed. Another drawback of adaptation in subbands is the delay introduced by the analysis and the synthesis filters.

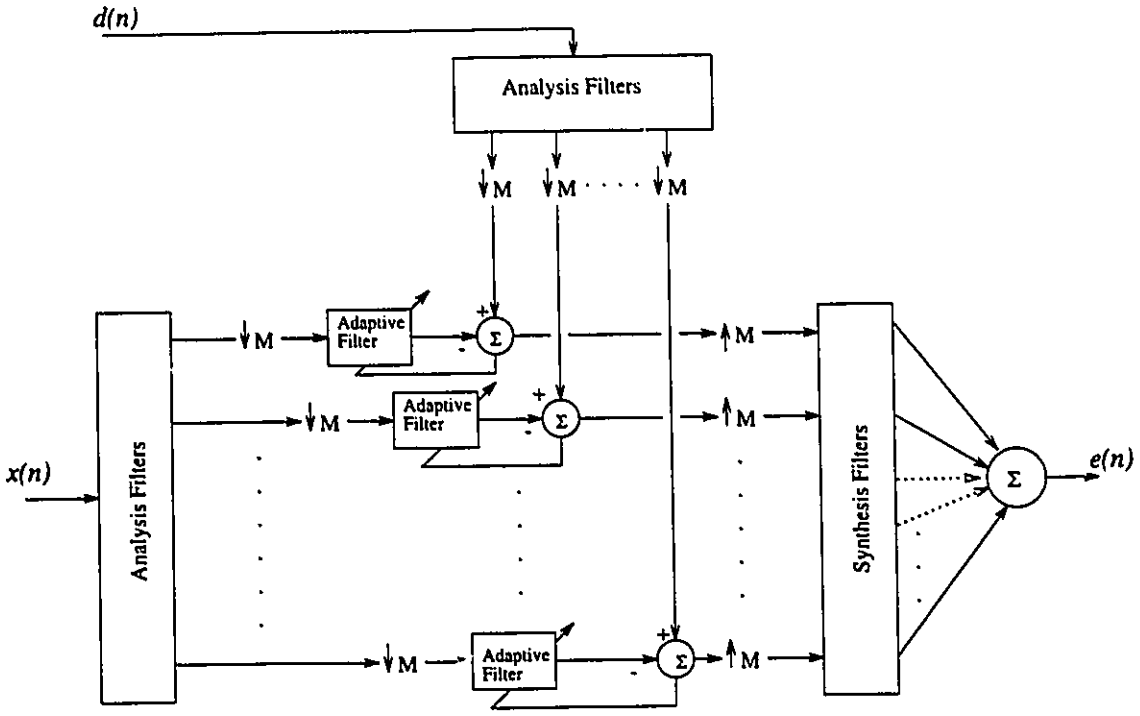

Fig. 2.2 Subband adaptive filtering structure.

2.1.2.6 Other methods and algorithms

Several other gradient methods and algorithms that do not fall in the previous categories exist: Data Reusing method [7, 54, 70, 80, 94], the Conjugate Gradient techniques [10], the Fast Newton algorithms [69, 84], the LMS algorithm with the reciprocal of the eigenvalues as step sizes [13], the Time-Weighted LMS algorithm [55], the LMS Coupled adaptive prediction and system identification scheme [74], and the LMS-Based algorithm for correlated input signals [71].

2.2 Least squares technique

In the RLS-type algorithms, the optimal weight vector $\mathbf{W}^*$ is obtained such that the cumulative squared error [1, 46]

$$\epsilon_{LS}(n) = \sum_{i=0}^{n} \lambda^{n-i} e^2(i \mid n) \tag{2.15}$$

is minimized. The constant λ is a data exponential weighting factor, where $0 < \lambda \leq 1$, and the error $e(i \mid n)$ is given by

$$e(i \mid n) = d(i) - \mathbf{W}_{\mathrm{N}}^{\mathrm{T}}(n)\mathbf{X}_{\mathrm{N}}(i) \tag{2.16}$$

where $\mathbf{X}_{\mathrm{N}}(i) = [x(i), \ldots, x(i-N+1)]^{\mathrm{T}}$.

It is clear that the LS method is a deterministic approach since it minimizes a function of the acquired data. More importantly, the method calculates at each time instant n a new optimal weight vector (in the LS sense) for the received data,

whereas the gradient-type algorithms iteratively approach the optimal solution (in the MSE sense) that is solely a function of the data statistics. This illustrates the potential of the RLS-type algorithms for higher convergence speed compared to their gradient-search counterparts. Basically, the gradient-type algorithms converge to the optimal solution with the progression of time while the RLS-type algorithms find the optimal solution at each sample time.

The original RLS algorithm was proposed to solve the least squares problem recursively in conjunction with the matrix inversion lemma [62]. This algorithm has the same mathematical form as the Kalman algorithm derived from applying Kalman filter theory to the adaptive filtering problem [19, 46, 90]. The computational load of RLS algorithm is about $2N^2 + 6N$ multiplications, which is too high for most practical applications, despite its fast initial convergence. This motivated further research to reduce its computational complexity.

By taking into account that the current data vector $\mathbf{X}(n)$ can be obtained from $\mathbf{X}(n-1)$ by shifting its elements and adding a new entry, Ljung et al. [60] derived the fast Kalman RLS adaptive algorithm with a computational burden of about $11N$ multiplications. A subsequent reduction of the fast Kalman RLS complexity to about $7N$ multiplications was attained by the introduction of the fast a posteriori error sequential technique (FAEST) algorithm proposed by Carayannis et al. [12]. A novel approach that applies the vector space formulation of the LS problem in

the derivation of a LS lattice adaptive filter was first presented by Lee et al. [59]. Motivated by the several advantages of the transversal filter structure over the lattice one and by its usage in most applications, Cioffi extended the vector space approach along with the projection concept to the transversal filter and developed the fast RLS transversal filters (FTF) [20]. Although the FTF algorithm requires the same computational count as that of the FAEST algorithm, the geometric approach used in the development of the FTF has simplified its derivation and, more importantly, provided insight and interpretation of various variables and quantities used in the derivation of the FAEST algorithm.

Ardalan also extended the work by Ljung et al. [60] in conjunction with the geometric approach to the derivation of an equation-error form RLS algorithm for the estimation of pole/zero (ARMA) modeled system [4, 5]. The algorithm allows for different numbers of poles and zeroes.

The FTF algorithm in [20] can easily be extended to a multichannel case, however, the order of the adaptive filter for each channel is constrained to be equal. Feber et al. derived a general order multichannel transversal RLS algorithm with the flexibility to specify the order of each channel independently and arbitrarily [28]. The vector space concept was also used to derive a block-type FTF algorithm [17]. The algorithm calculates the exact optimal solution in the LS sense for each block of data, which results in order update recursion of the algorithm variables.

Even though it appears that the FTF-type algorithms have solved the high computational requirements problem of the original RLS algorithm, it was discovered later that these algorithms diverge in finite precision environments; particularly for high filter orders, in low signal-to-noise ratio environment or when the input is highly correlated [18, 61, 63]. This is because numerical errors tend to propagate and accumulate in the algorithms variables. This, along with the large number of coupled variables used inside these algorithms, can lead to algorithm divergence. Some rescue methods were proposed in [20, 63]. However, they result in suboptimal performance. Slock [98] exploited the redundancy[1] in the FTF algorithm [20]. This allowed the introduction of an error feedback mechanism in the FTF that lead to a numerically stable FTF (SFTF) algorithm [98].

2.3 Other techniques

The least squares and gradient methods are the two main widespread approaches developed for adaptive systems. Other criteria include the Least Absolute Value (LAV), which is adopted by the Sign algorithm [67]. Also, other search techniques exist such as the random search. This covers a number of techniques like the linear random search (LRS) [112]. This class of algorithms (random search) generally suffers very slow convergence in comparison with LMS, nevertheless, they have a global optimization property that makes them more desirable over the LMS in

[1] By redundancy we mean the ability to change the order of the algorithm internal recursions without affecting algorithm solution.

problems with multimodal performance surfaces [25].

Chapter 3

A Robust Variable Step Size LMS-Type Algorithm: Analysis and Simulations

It is well known that the final excess mean square error (MSE) is directly proportional to the adaptation step size of the standard LMS while the convergence time increases as the step size value decreases [1, 7, 46, 111, 113]. This inherent limitation of the LMS necessitates a compromise between the opposing fundamental requirements of fast convergence rate and small misadjustment demanded in most adaptive filtering applications. Employing a variable step size [11, 45, 56, 68, 100] is one of the promising methods to improve the convergence characteristics of the LMS algorithm while preserving the steady state performance. This can be performed by adjusting the step size value in accordance with a certain criterion that can provide an approximate measure of the adaptation process state. Several criteria have been used: the squared instantaneous error [56], the sign changes of

successive samples of the gradient [45], or attempting to reduce the squared error at each instant [68, 95].

Our simulation results as well as other results in [26] and [100] indicate that the performance of the existing VSS algorithms [45, 56, 68] is affected considerably by the presence of noise. Good performance is obtained only in high signal-to-noise ratio cases. This is intuitively obvious by noting that the criteria controlling step size updating of these algorithms are directly obtained from the instantaneous error that is contaminated by the disturbance noise. This undesirable shortcoming prohibits the use of these algorithms in many practical adaptive filters applications where high level of disturbances can arise such as adaptive noise cancellation, echo cancellation, channel equalization, adaptive line enhancement, biomedical applications, etc.

As an example of the sensitivity of the current algorithms to noise, the algorithm in [56] will be studied in noisy conditions. This algorithm outperforms the fixed step size LMS algorithm and also exhibits favorable performance over other existing VSS algorithms [56]. It will be shown that its performance deteriorates in the presence of noise. We then propose a new VSS LMS algorithm where the step size of the algorithm is adjusted according to the square of time-averaged estimate of the autocorrelation of $e(n)$ and $e(n-1)$ [72]. As a result, the algorithm can efficiently track the adaptation process state while being insensitive to indepen-

dent noise disturbances. Moreover, changes in the noise level will be regarded as changes in the adaptation process in the current VSS algorithms which will lead to undesirable performance, whereas the proposed algorithm is totally protected in such situation since the error autocorrelation estimate will absorb these changes. It will also be shown that the proposed algorithm allows more direct control of misadjustment and convergence time without the need to compromise one for the other as in other VSS algorithms.

3.1 Algorithm formulation

In [56], the authors propose a variable step size that is adjusted according to the instantaneous squared error. The weight update recursion of the algorithm is of the form

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mu(n)e(n)\mathbf{X}(n) \tag{3.1}$$

and the step size update expression is

$$\mu(n+1) = \alpha\mu(n) + \gamma e^2(n) \tag{3.2}$$

where $0 < \alpha < 1, \gamma > 0$, and $\mu(n+1)$ is set to μ_{min} or μ_{max} when it falls below or above one of them, respectively. The constant μ_{max} is normally selected near the point of instability of the conventional LMS to provide the maximum possible convergence speed. The value of μ_{min} is chosen to provide a trade off between the desirable level of misadjustment and the required tracking capabilities of the

algorithm. The parameter γ affects the convergence time as well as the level of misadjustment of the algorithm. The algorithm has preferable performance over the fixed step size LMS: at early stages of adaptation the error is large causing the step size to increase to provide faster convergence speed. When the error decreases, the step size decreases thus yielding smaller misadjustment near the optimum. However, using the instantaneous error energy as a measure to sense the adaptation process state does not perform as well as expected in the presence of measurement noise. This can be deduced by examining Eq.(3.2). The output error of the identification system is

$$e(n) = d(n) - \mathbf{X}^{\mathrm{T}}(n)\mathbf{W}(n) \tag{3.3}$$

where the desired signal , $d(n)$, is given by

$$d(n) = \mathbf{X}^{\mathrm{T}}(n)\mathbf{W}^{*}(n) + \xi(n) \tag{3.4}$$

where $\xi(n)$ is a zero-mean independent disturbance and $\mathbf{W}^{*}(n)$ is the time-varying optimal weight vector. Substituting Eqs.(3.3) and (3.4) in the step size recursion,

$$\mu(n+1) = \alpha\mu(n)+\gamma\mathbf{V}^{\mathrm{T}}(n)\mathbf{X}(n)\mathbf{X}^{\mathrm{T}}(n)\mathbf{V}(n)+\gamma\xi^{2}(n))-2\gamma\xi(n)\mathbf{V}^{\mathrm{T}}(n)\mathbf{X}(n) \tag{3.5}$$

where $\mathbf{V}(n) = \mathbf{W}(n) - \mathbf{W}^{*}(n)$ is the translated error vector. The input signal autocorrelation matrix, defined as $\mathbf{R} = E\{\mathbf{X}(n)\mathbf{X}^{\mathrm{T}}(n)\}$, can be represented as $\mathbf{R} = \mathbf{Q}\Lambda\mathbf{Q}^{\mathrm{T}}$ where Λ is the eigenvalues matrix, and $\mathbf{Q}$ is the modal matrix of $\mathbf{R}$.

Using $\dot{\mathbf{V}}(n) = \mathbf{Q}^{\mathrm{T}}\mathbf{V}(n)$, and $\dot{\mathbf{X}}(n) = \mathbf{Q}^{\mathrm{T}}\mathbf{X}(n)$, then the statistical behavior of $\mu(n+1)$ is determined by taking the expected average of Eq.(3.5)

$$E\{\mu(n+1)\} = \alpha E\{\mu(n)\} + \gamma(E\{\xi^2(n)\} + E\{\dot{\mathbf{V}}^{\mathrm{T}}(n)\mathbf{\Lambda}\dot{\mathbf{V}}(n)\}\ ) \qquad (3.6)$$

where we have made use of the common independence assumption of $\mathbf{W}(n)$ and $\dot{\mathbf{X}}(n)$ [47]. This assumption will also be used in the forthcoming chapters. It is true only for white input data which is the case in some applications such as data transmission and adaptive array processing. Nevertheless, analysis employing this assumption has produced reliable results that agree well with simulation results in other applications where this assumption is not met [9]. Clearly, the term $E\{\dot{\mathbf{V}}^{\mathrm{T}}(n)\mathbf{\Lambda}\dot{\mathbf{V}}(n)\}$ influences how much the adaptive system is close to the optimal solution, accordingly $\mu(n+1)$ is adjusted. However, due to $E\{\xi^2(n)\}$, the step size update deteriorates at all stages of adaptation which will reduce the efficiency of the algorithm significantly. More specifically, close to the optimum, $\mu(n)$ will still be large due to the presence of the noise term $E\{\xi^2(n)\}$. This results in large misadjustment due to the large fluctuations around the optimum.

To ensure robustness against noise, a different approach is proposed to control step size adaptation attempting to enhance the performance of the algorithm in the presence of high independent disturbance noise, while retaining faster convergence characteristics. The objective is to ensure large $\mu(n)$ when the algorithm is far from optimality with $\mu(n)$ decreasing as we approach the optimum *even in the presence*

of noise. This is achieved by using an estimate of the autocorrelation between $e(n)$ and $e(n-1)$ to control step size updating. The estimate is a time-averaged of $e(n)e(n-1)$ described as

$$p(n) = \beta p(n-1) + (1-\beta)e(n)e(n-1) \tag{3.7}$$

and the step size update equation is

$$\mu(n+1) = \alpha\mu(n) + \gamma p(n)^2 \tag{3.8}$$

where limits on $\mu(n+1)$, α, and γ are the same as those of the VSS LMS algorithm in [56]. The positive constant β $(0 < \beta < 1)$ is an exponential weighting parameter that governs the averaging time constant, i.e the quality of the estimation. In stationary environments, previous samples contain information relevant in determining an accurate measure of adaptation state, i.e. the proximity of the adaptive filter coefficients to optimal ones. Therefore, β should be ≈ 1. For nonstationary optimal coefficients, the time averaging window should be small enough to allow forgetting the deep past and adapting to the current statistics, i.e. $\beta < 1$.

The use of $p(n)$ in the update of $\mu(n)$ serves two objectives; firstly, the autocorrelation is an efficient measure of the proximity to the optimum, secondly, it rejects the independent noise sequence effect on step size. When the adaptation process is active, the autocorrelation is large resulting in a large $\mu(n)$. As we approach the optimum, the autocorrelation approaches zero resulting in a smaller step size.

This provides the fast initial convergence due to large $\mu(n)$ while ensuring low misadjustment near optimum *even in the presence of* $\xi(n)$. This can be seen if we rewrite the step size in Eq.(3.8) as

$$\mu(n+1) = \alpha\mu(n) + \gamma E^2\{e(n)e(n-1)\} \tag{3.9}$$

where we have assumed perfect estimation of the autocorrelation of $e(n)$ and $e(n-1)$. Using Eqs.(3.3) and (3.4), we get

$$\begin{aligned} E\{e(n)e(n-1)\} &= E\{\xi(n)\xi(n-1) - \xi(n)\mathbf{V}^{\mathbf{T}}(n-1)\mathbf{X}(n-1) - \xi(n-1)\mathbf{V}^{\mathbf{T}}(n)\mathbf{X}(n) \\ &\quad + \mathbf{V}^{\mathbf{T}}(n)\mathbf{X}(n)\mathbf{X}^{\mathbf{T}}(n-1)\mathbf{V}(n-1)\} \end{aligned} \tag{3.10}$$

Under the assumption of zero-mean independent noise sequence,

$$E\{e(n)e(n-1)\} = E\{\mathbf{V}^{\mathbf{T}}(n)\mathbf{X}(n)\mathbf{X}^{\mathbf{T}}(n-1)\mathbf{V}(n-1)\}\} \tag{3.11}$$

Therefore,

$$\mu(n+1) = \alpha\mu(n) + \gamma\left[E\{\mathbf{V}^{\mathbf{T}}(n)\mathbf{X}(n)\mathbf{X}^{\mathbf{T}}(n-1)\mathbf{V}(n-1)\}\right]^2 \tag{3.12}$$

We note that owing to the averaging operation, the instantaneous behavior of the step size will be smoother. It is also clear from Eq.(3.12) that the update of $\mu(n)$ is dependent on how far we are from the optimum and is not affected by independent disturbance noise. This was possible by choosing a lag of one between the error samples, i.e. $e(n)$ and $e(n-1)$. Increasing the lag above one will have no effect on the algorithm performance in the presence of independent noise, however, it is

obvious from the second term of Eq.(3.9) that the algorithm ability to measure the adaptation state efficiently is reduced as the lag increases between the error samples.

Finally, the proposed algorithm involves two additional update equations (Eq.(3.7) and Eq.(3.8)) compared to the standard LMS algorithm. Therefore, the added complexity is 6 multiplications per iteration. These multiplications can be reduced to a shift if the parameters α, β, γ are chosen as power of 2. Compared to the VSS LMS algorithm in [56], the proposed algorithm adds a new equation (Eq.(3.7)), and a corresponding parameter β.

3.2 Performance analysis of the proposed algorithm

A performance analysis of the proposed algorithm will now be considered when operating in stationary and nonstationary environments. The input signal is assumed a zero-mean, stationary Gaussian. The weight update recursion of the modified algorithm has the form

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mu(n)e(n)\mathbf{X}(n) \tag{3.13}$$

The time-varying optimal weight vector is assumed to be generated by a random walk model [46] as

$$\mathbf{W}^*(n) = \mathbf{W}^*(n-1) + \eta(n-1) \tag{3.14}$$

where $\eta(n)$ is a stationary noise process of zero-mean and correlation matrix $\sigma_n^2\mathbf{I}$ that accounts for the nonstationarity of the physical system. In a stationary circumstances $\sigma_n^2 = 0$ and $\mathbf{W}^*(n) = \mathbf{W}^*$. Substituting Eqs.(3.3), (3.4) and (3.14) in (3.13) results in

$$\dot{\mathbf{V}}(n+1) = [\mathbf{I} - \mu(n)\dot{\mathbf{X}}(n)\dot{\mathbf{X}}^{\mathbf{T}}(n)]\dot{\mathbf{V}}(n) + \mu(n)\xi(n)\dot{\mathbf{X}}(n) - \eta(n) \tag{3.15}$$

Normally γ is chosen to be a very small value, hence $\mu(n)$ is slowly varying when compared with $e(n)$ and $X(n)$. This will justify the independence assumption of $\mu(n)$ and $\mu^2(n)$ with $e(n)$, $\mathbf{W}(n)$ and $\mathbf{X}(n)$. Accordingly, the following condition can be obtained from Eq.(3.15) to ensure convergence of the weight vector mean,

$$0 < E\{\mu(n)\} < \frac{2}{\lambda_{max}} \tag{3.16}$$

where λ_{max} is the maximum eigenvalue of $\mathbf{R}$. However convergence in the mean cannot guarantee convergence of the mean square error. Therefore, we need to determine the necessary and sufficient conditions for stability in the mean square. To evaluate the performance of the system, approximate expressions for misadjustment are derived. This will lead to tight conditions for the selection of the parameters α, β, and γ. More specifically, since we have presented some guidelines to the choice of α and β, the foregoing analysis will constrain the selection of γ to guarantee MSE convergence as well as to produce the desirable misadjustment level.

The MSE is given by [111]

$$\begin{aligned} E\{e^2(n)\} &= \epsilon_{min} + \epsilon_{ex}(n) \\ &= \epsilon_{min} + E\{\dot{\mathbf{V}}^{\mathbf{T}}(n)\mathbf{\Lambda}\dot{\mathbf{V}}(n)\} \end{aligned} \tag{3.17}$$

where $\epsilon_{ex}(n)$ is the excess MSE and $\epsilon_{min} = E\{\xi^2(n)\}$ is the minimum value of the MSE. Eq.(3.17) shows that the MSE is immediately related to the diagonal elements of $E\{\dot{\mathbf{V}}(n)\dot{\mathbf{V}}^{\mathbf{T}}(n)\}$. Consequently, convergence of the MSE is ensured by the convergence of these elements. Postmultiplying both sides of Eq.(3.15) by $\dot{\mathbf{V}}^{\mathbf{T}}(n+1)$, and then taking the expected value yields

$$\begin{aligned} E\{\dot{\mathbf{V}}(n+1)\dot{\mathbf{V}}^{\mathbf{T}}(n+1)\} &= E\{\dot{\mathbf{V}}(n)\dot{\mathbf{V}}^{\mathbf{T}}(n)\} - E\{\mu(n)\}E\{\dot{\mathbf{V}}(n)\dot{\mathbf{V}}^{\mathbf{T}}(n)\}\mathbf{\Lambda} \\ &- E\{\mu(n)\}\mathbf{\Lambda}E\{\dot{\mathbf{V}}(n)\dot{\mathbf{V}}^{\mathbf{T}}(n)\} \\ &+ 2E\{\mu^2(n)\}\mathbf{\Lambda}E\{\dot{\mathbf{V}}(n)\dot{\mathbf{V}}^{\mathbf{T}}(n)\}\mathbf{\Lambda} \\ &+ E\{\mu^2(n)\}\mathbf{\Lambda}tr(\mathbf{\Lambda}E\{\dot{\mathbf{V}}(n)\dot{\mathbf{V}}^{\mathbf{T}}(n)\}) \\ &+ E\{\mu^2(n)\}\epsilon_{min}\mathbf{\Lambda} + \sigma_n^2\mathbf{I} \end{aligned} \tag{3.18}$$

Note that in Eq.(3.18), we have used the Gaussian factoring theorem to simplify the expression $E\{\dot{\mathbf{X}}(n)\dot{\mathbf{X}}^{\mathbf{T}}(n)\dot{\mathbf{V}}(n)\dot{\mathbf{V}}^{\mathbf{T}}(n)\dot{\mathbf{X}}(n)\dot{\mathbf{X}}^{\mathbf{T}}(n)\}$ into a sum of second order moments [47]. From Eq.(3.7), $p(n)$ can be solved recursively as

$$p(n) = (1-\beta)\sum_{i=0}^{n-1}\beta^i e(n-i)e(n-i-1) \tag{3.19}$$

and

$$p(n)^2 = (1-\beta)^2 \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \beta^i \beta^j e(n-i)e(n-i-1)e(n-j)e(n-j-1) \qquad (3.20)$$

The following analysis is aimed at studying the *steady state* performance of the proposed algorithm. Therefore, we will assume in our analysis that the algorithm has converged. In this case, the samples of the error $e(n)$ can be assumed uncorrelated, i.e, $E\{e(n-i)e(n-j)\} = 0$ for $\forall\, i \neq j$ leading to the small $\mu(n)$ after convergence. Using Eqs.(3.8), (3.19) and (3.20), the mean and the mean-square behavior of the step size $\mu(n)$ are

$$E\{\mu(n+1)\} = \alpha E\{\mu(n)\} + \gamma(1-\beta)^2 \sum_{i=0}^{n-1} \beta^{2i} E\{e^2(n-i)\} E\{e^2(n-i-1)\} \quad (3.21)$$

and

$$\begin{aligned} E\{\mu^2(n+1)\} &= \alpha^2 E\{\mu^2(n)\} + 2\alpha\gamma E\{\mu(n)\}(1-\beta)^2 \sum_{i=0}^{n-1} \beta^{2i} E\{e^2(n-i)\} E\{e^2(n-i-1)\} \\ &+ \gamma^2 E\{p^4(n)\} \end{aligned} \qquad (3.22)$$

Since γ is small, the last term in Eq.(3.22) involving γ^2 is too small compared to the other terms and can be discarded. Thus

$$E\{\mu^2(n+1)\} \approx \alpha^2 E\{\mu^2(n)\} + 2\alpha\gamma E\{\mu(n)\}(1-\beta)^2 \sum_{i=0}^{n-1} \beta^{2i} E\{e^2(n-i)\} E\{e^2(n-i-1)\} \qquad (3.23)$$

The misadjustment is defined as [111]

$$M = \frac{\epsilon_{ex}(\infty)}{\epsilon_{min}} \qquad (3.24)$$

Following the same argument in [56], a sufficient condition that ensures convergence of the MSE is

$$0 < \frac{E\{\mu^2(\infty)\}}{E\{\mu(\infty)\}} \leq \frac{2}{3tr(\mathbf{R})} \tag{3.25}$$

Assuming that the condition in Eq.(3.25) holds, we can use Eqs.(3.18) and (3.17) in (3.24) to find the following expression for algorithm misadjustment [56]

$$M = \frac{\sum_{j=1}^{N} \frac{E\{\mu^2(\infty)\}\lambda_j}{2E\{\mu(\infty)\}-2E\{\mu^2(\infty)\}\lambda_j} + \sum_{j=1}^{N} \frac{\sigma_n^2}{(2E\{\mu(\infty)\}-2E\{\mu^2(\infty)\}\lambda_j)\epsilon_{min}}}{1 - \sum_{j=1}^{N} \frac{E\{\mu^2(\infty)\}\lambda_j}{2E\{\mu(\infty)\}-2E\{\mu^2(\infty)\}\lambda_j}} \tag{3.26}$$

where $E\{\mu(\infty)\}$ and $E\{\mu^2(\infty)\}$ are the steady state values of $E\{\mu(n)\}$ and $E\{\mu^2(n)\}$. We recall that

$$E\{e^2(\infty)\} = \epsilon_{min} + \epsilon_{ex}(\infty) \tag{3.27}$$

then from Eq.(3.21)

$$E\{\mu(\infty)\} = \frac{\gamma(1-\beta)(\epsilon_{min} + \epsilon_{ex}(\infty))^2}{(1-\alpha)(1+\beta)} \tag{3.28}$$

Substituting Eq.(3.28) in (3.23) yields

$$E\{\mu^2(\infty)\} \approx \frac{2\gamma^2\alpha(1-\beta)^2(\epsilon_{min} + \epsilon_{ex}(\infty))^4}{(1-\alpha^2)(1-\alpha)(1+\beta)^2} \tag{3.29}$$

In our analysis we will assume that $\epsilon_{ex}(\infty) \ll \epsilon_{min}$ which is practically valid in stationary environments and in slow time-varying environments for the nonstationary case. Then from Eqs.(3.28) and (3.29) we have

$$\frac{E\{\mu^2(\infty)\}}{E\{\mu(\infty)\}} \approx y \tag{3.30}$$

where

$$y = \frac{2\gamma\alpha\epsilon_{min}^2(1-\beta)}{(1-\alpha^2)(1+\beta)} \tag{3.31}$$

Noting that $\gamma > 0$, $0 < \beta < 1$, and $0 < \alpha < 1$, then

$$y < \frac{2\gamma\epsilon_{min}^2(1-\beta)}{1-\alpha^2} \tag{3.32}$$

From Eq.(3.25), (3.30), and (3.32) the following condition is imposed on γ, α, and β to guarantee stability of the MSE

$$0 < \frac{\gamma\epsilon_{min}^2(1-\beta)}{1-\alpha^2} \leq \frac{1}{3tr(\mathbf{R})} \tag{3.33}$$

Substituting Eq.(3.30) in (3.26) yields the following expression for algorithm misadjustment

$$M \approx \frac{\sum_{j=1}^{N} \frac{y\lambda_j}{2-2y\lambda_j} + \sum_{j=1}^{N} \frac{\sigma_n^2}{E\{\mu(\infty)\}(2-2y\lambda_j)\epsilon_{min}}}{1 - \sum_{j=1}^{N} \frac{y\lambda_j}{2-2y\lambda_j}} \tag{3.34}$$

In the event of small values of misadjustment so that $\sum_j y\lambda_j \ll 1$

$$M \approx \frac{y}{2}tr(\mathbf{R}) + \frac{N\sigma_n^2}{2E\{\mu(\infty)\}\epsilon_{min}} \tag{3.35}$$

where

$$E\{\mu(\infty)\} \approx \frac{\gamma(1-\beta)}{(1-\alpha)(1+\beta)}\epsilon_{min}^2 \tag{3.36}$$

In a stationary environment $\sigma_n^2 = 0$ and the misadjustment is determined by

$$M \approx \frac{y}{2}tr(\mathbf{R}) \tag{3.37}$$

Practically, α, γ and β are selected to produce the same MSE attained by the fixed step size LMS (FSS) while achieving the increase in convergence speed. Accordingly, α, γ and β could be selected to satisfy $y \leq \mu_{FSS}$, where μ_{FSS} is the adaptation step size of the FSS algorithm. In a stationary environment, a large number of samples would be used in the estimation of $E\{e(n)e(n-1)\}$, i.e. the exponential weighting parameter β should be close to one. From Eqs.(3.37) and (3.36), this value for β results in a reduced misadjustment. Thus, for the same level of misadjustment a larger γ can be used while maintaining the stability of the algorithm, Eq.(3.33). A larger γ results in a larger step size in the initial stages of adaptation, i.e. faster convergence. Thus, the introduction of the β along with γ has provided the algorithm with an extra degree of freedom that facilitates better simultaneous control of both convergence speed and final excess MSE. When operating in a nonstationary environment, the choice of β becomes crucial. To provide good tracking capabilities β should be small to cope with the time-varying statistics of the environment. Although this will decrease the second term in the misadjustment expression in Eq.(3.35), it will at the same time increase the first term. Therefore, the selection of β should achieve a compromise between tracking speed and excess MSE.

Since the first term in Eq.(3.35) is directly proportional to γ and the second term is inversely proportional to γ, we can optimize the choice of γ for given α

and β to obtain the minimum M by equating the two terms. Therefore, from Eqs.(3.31) ,(3.36), and (3.35), the optimal value of γ is

$$\gamma^* = \sqrt{\frac{N\sigma_n^2}{q\epsilon_{min}tr(\mathbf{R})}} \tag{3.38}$$

where

$$q = \frac{2\alpha\epsilon_{min}^4(1-\beta)^2}{(1-\alpha^2)(1-\alpha)(1+\beta)^2} \tag{3.39}$$

3.3 Simulation

Here, the proposed variable step size LMS (MVSS) algorithm is used under different case studies of stationary and nonstationary models of system identification. The performance of the algorithm is compared with the: variable step size LMS (VSS) algorithm [56], the stochastic gradient algorithm with gradient adaptive step size (SGA-GAS) [68], and the fixed step size LMS (FSS) algorithm [111]. Parameters of these algorithms are selected to produce a comparable level of misadjustment in all cases. Moreover, our choice of these parameters is also guided by the recommended values in their corresponding publication. For the VSS algorithm , α_{vss} was set to 0.97 in all experiments, while γ_{vss} is chosen to achieve the desired level of misadjustment according to the following approximate expression derived for stationary cases [56]

$$M_{vss} = \frac{1-\left[1-2\frac{(3-\alpha_{vss})\gamma_{vss}\epsilon_{min}}{1-\alpha_{vss}^2}\right]^{1/2}}{1+\left[1-2\frac{(3-\alpha_{vss})\gamma_{vss}\epsilon_{min}}{1-\alpha_{vss}^2}\right]^{1/2}} \tag{3.40}$$

In [68], it was shown that for a zero-mean and white Gaussian signal, misadjustment can be approximately determined by

$$M_{SGA-GAS} = N\sigma_x^2\sqrt{\frac{\epsilon_{min}\rho}{2}} \tag{3.41}$$

where σ_x^2 is the input signal power, and ρ is a small positive constant for controlling the step size adaptation [68].

In all simulations presented here, the desired signal $d(n)$ is disturbed by zero-mean, uncorrelated Gaussian noise of ϵ_{min} variance. Results are obtained by averaging over 200 independent runs.

3.3.1 Example 1: stationary white input, low SNR

In the first example, the moving average system has four time-invariant coefficients, and the FIR adaptive filter is of equal order. Both are excited by a zero-mean, white Gaussian signal of unity variance. ϵ_{min} is equal to 1. This results in $SNR = 0$. The MVSS is used with the parameters $\alpha = 0.97$, which is found to be a good choice in stationary and nonstationary environments, and $\beta = 0.99$. In accordance with Eq.(3.37), γ is set to 1×10^{-3} to produce a steady state excess MSE of about -34 dB. Note that these parameter values fulfill the condition in Eq.(3.33) for MSE convergence. The VSS algorithm is used with $\alpha_{vss} = 0.97$, and using Eq.(3.40), γ_{vss} is chosen as 1×10^{-5} to obtain the same level of the steady state excess MSE. According to Eq.(3.41), we choose $\rho = 2 \times 10^{-8}$ to obtain a comparable misadjustment value with the other algorithms. For all previous algorithms, we

used $\mu_{max} = 0.1$, and $\mu_{min} = 1 \times 10^{-5}$ [56]. Also, we used $\mu_{FSS} = 3.5 \times 10^{-4}$ [111]. Fig. 3.1 shows that the MVSS algorithm provides the fastest speed of convergence among all other algorithms, while retaining the same small level of misadjustment. This is supported by plotting the mean behavior of one of the weights in Fig. 3.2, where the optimum coefficient is 2.0. Note that it is possible to determine the overall algorithm convergence rate based on one weight evolution because in this example the adaptive filter coefficients are decoupled and they all have the same instantaneous time constant.

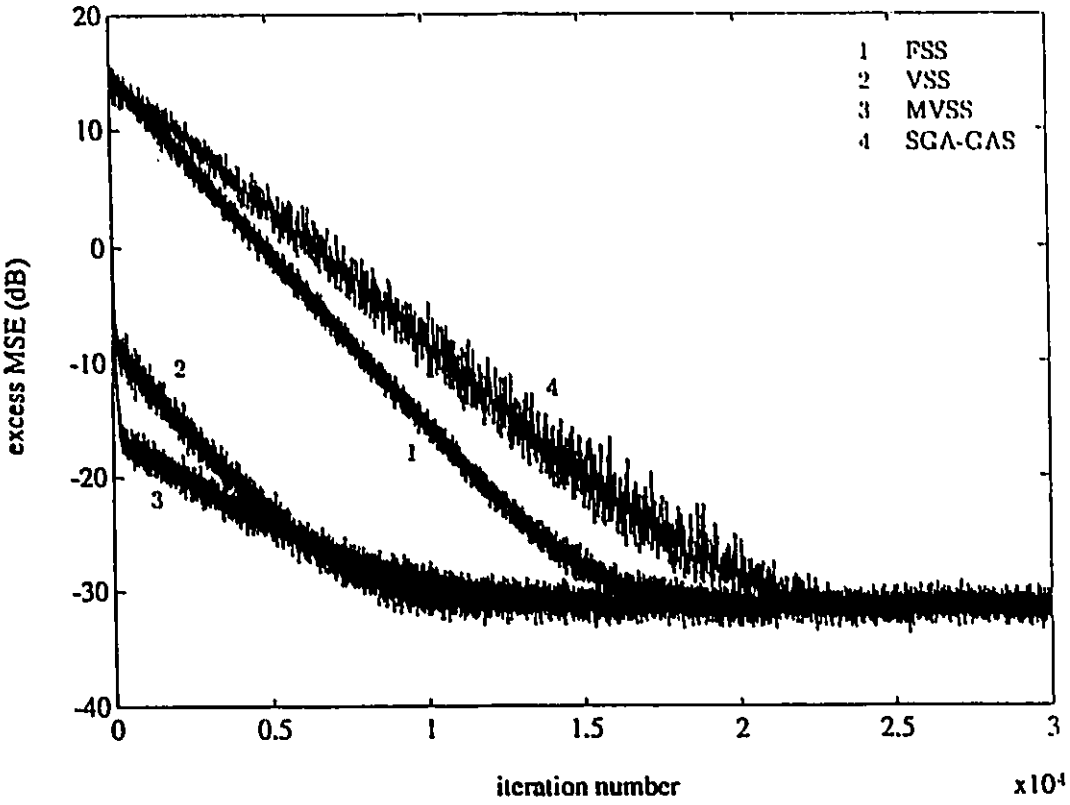

Fig. 3.1 Comparison of excess MSE of various adaptive algorithms for the white input case, when the added noise to the desired signal is of unity variance.

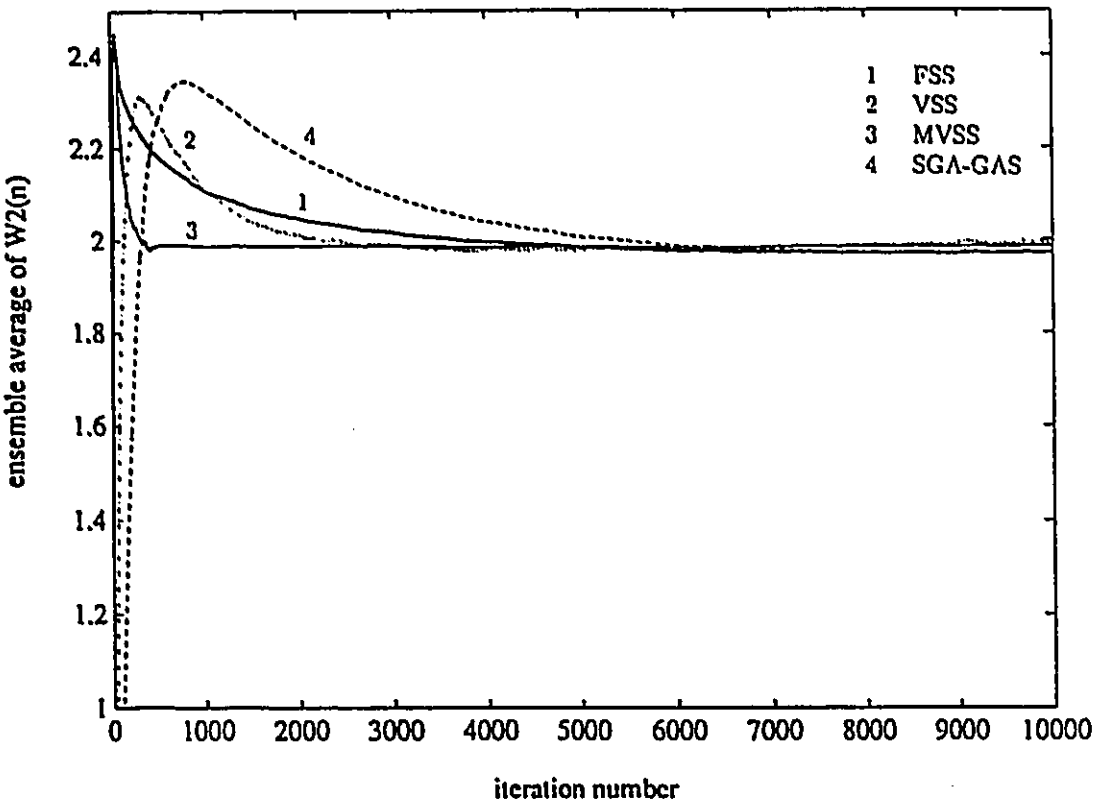

Fig. 3.2 Ensemble average of the second component of $\mathbf{W}(n)$ for the white input case, when the added noise to the desired signal is of unity variance.

3.3.2 Example 2: stationary correlated input, low SNR

In the second example, both the system and the adaptive filter are excited by a correlated signal $x(n)$ generated by [56]

$$x(n) = 0.9x(n-1) + a(n) \tag{3.42}$$

where $a(n)$ is a zero-mean, uncorrelated Gaussian noise of unity variance. This type of input results in flattened elliptical contours which usually cause difficulties in the tracking capabilities of gradient algorithms. ϵ_{min}, α, β are chosen as in example 1 while $\gamma = 8 \times 10^{-4}$ to obtain a final steady state excess MSE of about -28 dB. Note that for this example $tr(\mathbf{R}) = 21.0526$. The VSS algorithm is used with $\alpha_{vss} = 0.97$, and to obtain the same level of misadjustment, γ_{vss} is set to 8×10^{-6}. Since there is no theoretical analysis of misadjustment for SGA-GAS in the case of correlated input signals, we found experimentally that $\rho = 1 \times 10^{-7}$ provides the desired misadjustment. For all previous algorithms, we used $\mu_{max} = 0.008$, and $\mu_{min} = 1 \times 10^{-4}$. The FSS algorithm is used with $\mu_{FSS} = 3 \times 10^{-4}$.

Fig. 3.3 shows that for correlated input signals, the MVSS is superior in its convergence rate to the VSS, SGA-GAS and the FSS algorithms, while providing the same steady state MSE. This can be seen from the step size evolution in Fig. 3.4: the step size of MVSS remains near the μ_{max} value until the algorithm is fairly close to steady state, where it automatically decreases to its minimum value to produce low misadjustment. On the other hand, the SGA-GAS step size stays high even

after algorithm convergence and decreases slowly to its steady state value after 50,000 iterations. Therefore, the excess MSE of the algorithm will achieve almost the same level as other algorithms only after 50,000 samples. On the other hand, the VSS algorithm is still showing dramatic improvement over the FSS algorithm in such environments.

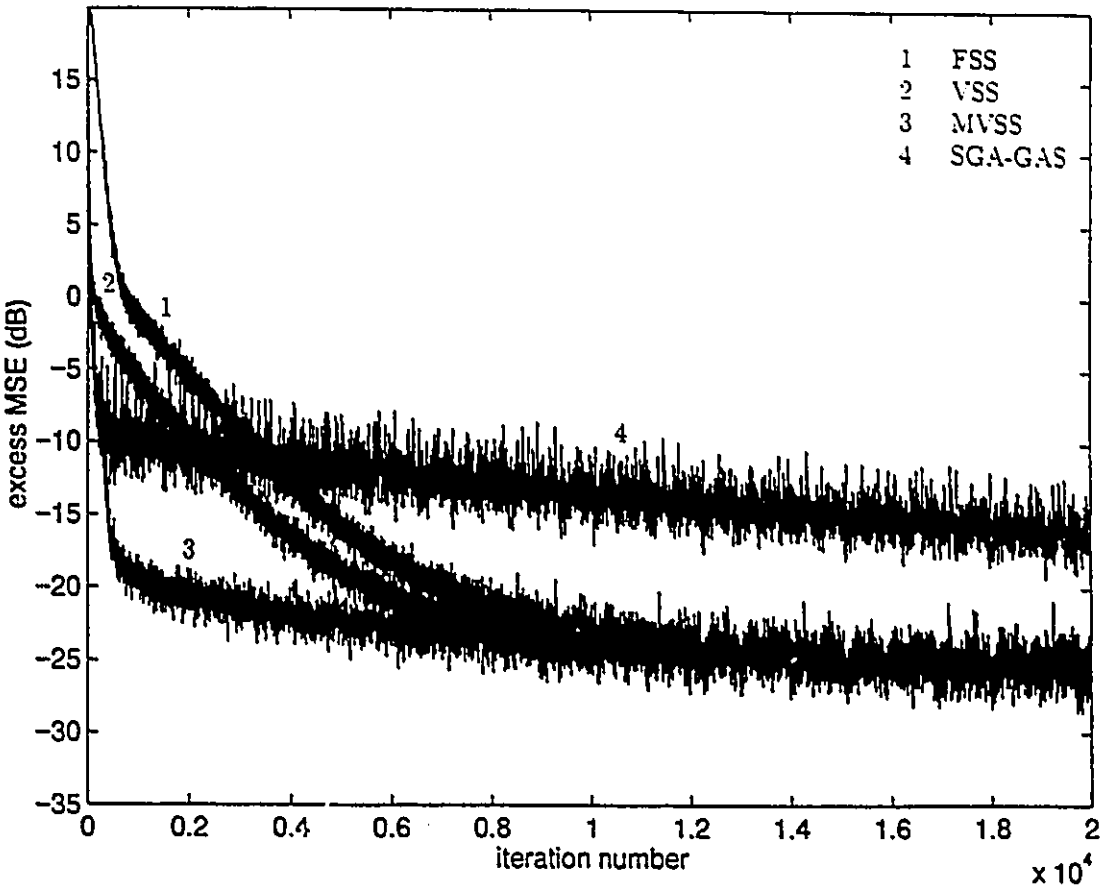

Fig. 3.3 Comparison of excess MSE of various adaptive algorithms for the correlated input case, when the added noise to the desired signal is of unity variance.

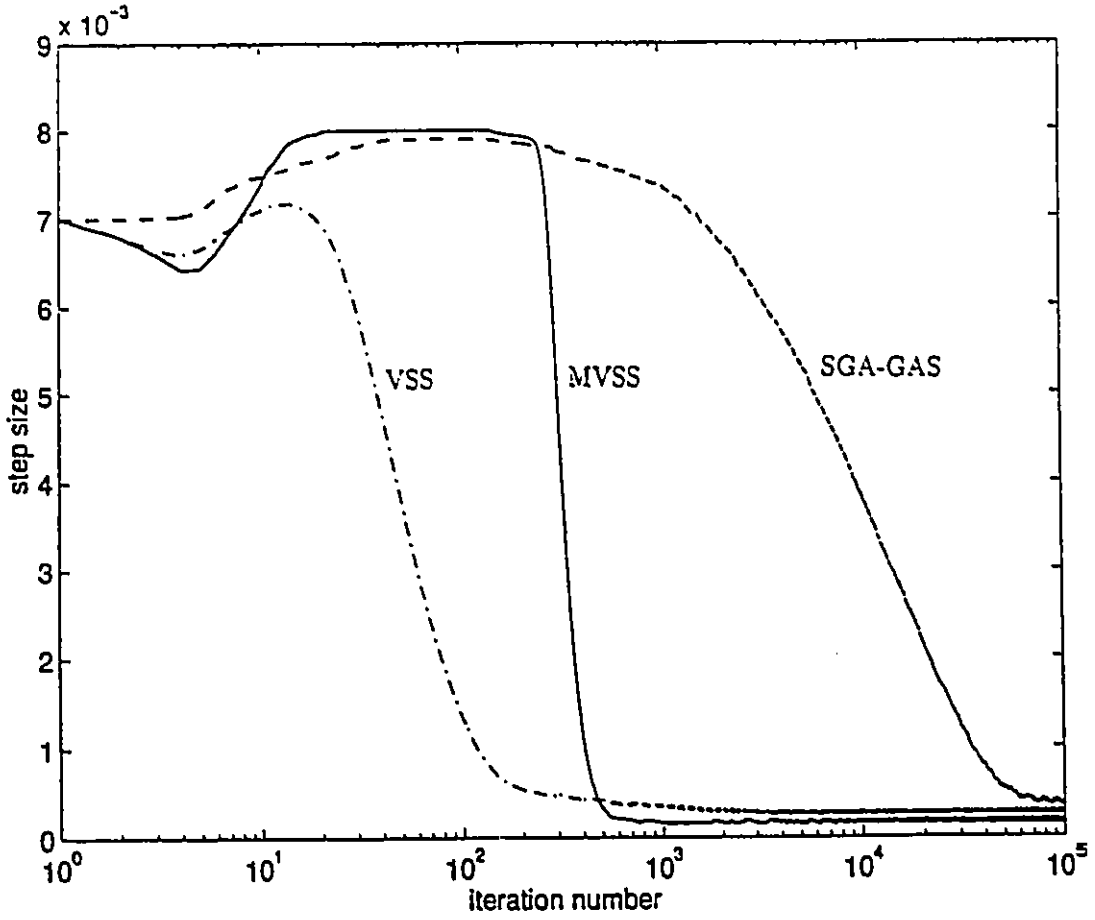

Fig. 3.4 Comparison of step size mean evolution of the VSS, SGA-GAS, and the MVSS algorithms for the correlated input case, when the added noise to the desired signal is of unity variance.

3.3.3 Example 3: stationary correlated input (perturbed system)

The second example is applied to the MVSS with the same parameter values, except that at iteration 20,000, system coefficients were all switched to their corresponding negative values. Figures 3.5 and 3.6 demonstrate the ability of the algorithm to respond quickly to abrupt environment changes by setting the step size value to μ_{max} to provide the fastest speed to track changes in the system. Note also that the convergence rate of MVSS after the change in the system remains the same as at the initial stage.

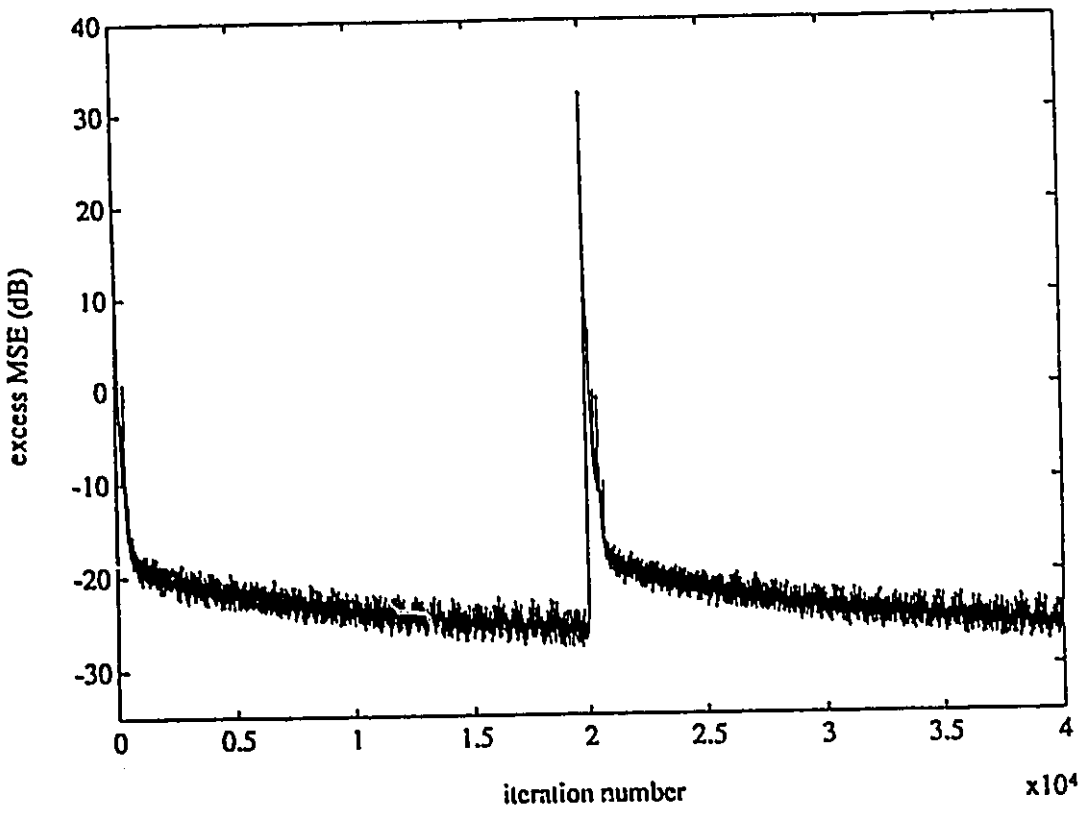

Fig. 3.5 Excess MSE of the MVSS algorithm when there is an abrupt change in the system.

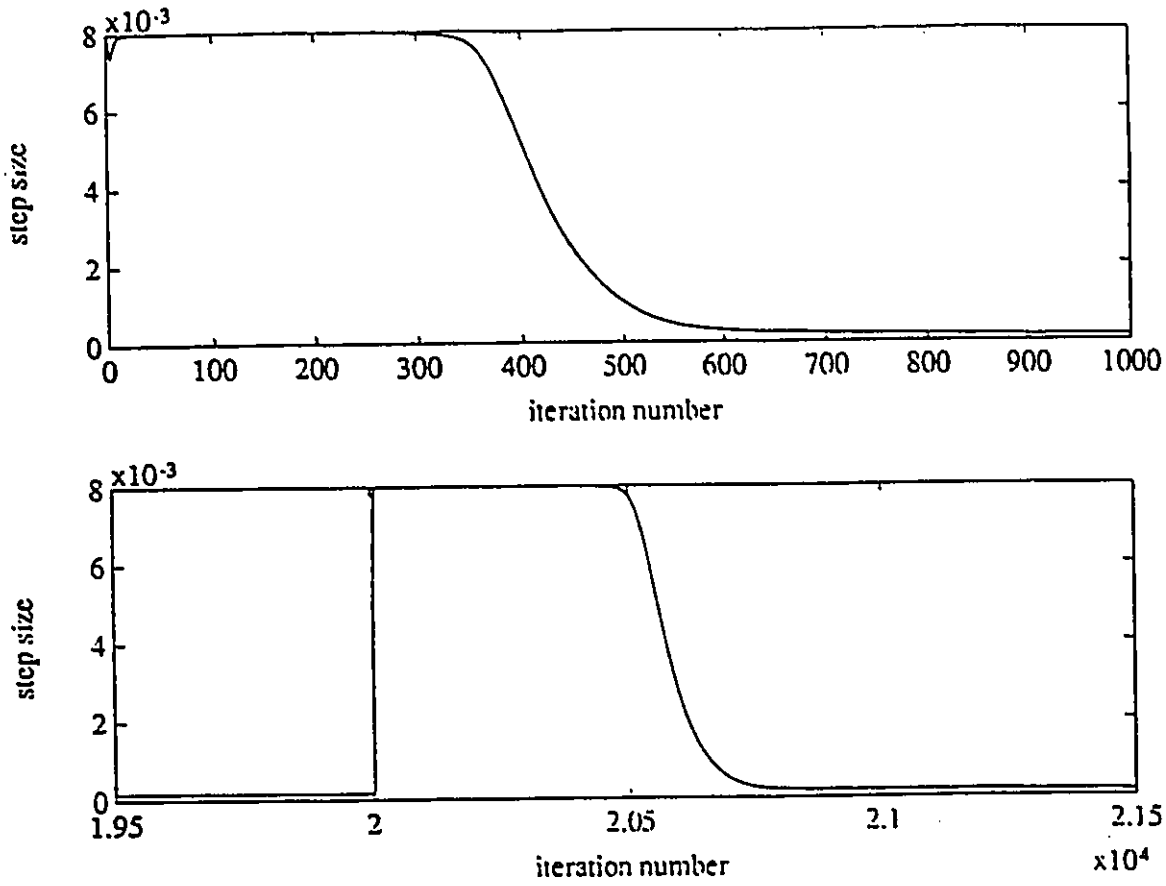

Fig. 3.6 Step size mean evolution of the MVSS algorithm when there is an abrupt change in the system.

3.3.4 Example 4: high SNR

Examples 1 and 2 examined the performance of the various algorithms under low signal-to-noise ratio (SNR) conditions. These two examples are now repeated for higher SNR using $\epsilon_{min} = 0.001$.

Example 1 is repeated with the parameters: $\mu_{max} = 0.1$, $\mu_{min} = 5 \times 10^{-4}$, $\alpha = 0.97$, $\beta = 0.99$, $\gamma = 1$, $\alpha_{vss} = 0.97$, $\gamma_{vss} = 0.02$, $\rho = 0.001$, and $\mu_{FSS} = 5 \times 10^{-4}$. Those parameters are chosen to achieve about -60 dB steady state excess MSE. Fig. 3.7 shows the result of comparison the various algorithms. It should be noted that the SGA-GAS algorithm takes about 70,000 iterations in this example to converge to the same steady state excess MSE of the other algorithms.

Example 2 is repeated using the parameters: $\mu_{max} = 0.008$, $\mu_{min} = 1 \times 10^{-3}$, $\alpha = 0.97$, $\beta = 0.99$, $\gamma = 0.5$, $\alpha_{vss} = 0.97$, $\gamma_{vss} = 0.01$, $\rho = 0.001$, and $\mu_{FSS} = 1 \times 10^{-3}$. The parameters are selected to achieve a steady state excess MSE of about -50 dB. Comparison of various algorithms is shown in Fig. 3.8. Note that in Fig. 3.8, the VSS and SGA-GAS algorithms exhibit similar behavior in this case. Comparing Figs. 3.1, 3.3 and Figs. 3.7, 3.8 respectively; we can see that the performance of the VSS and SGA-GAS deteriorated significantly as the noise was increased from 0.001 in Figs. 3.7, 3.8 to 3.1 in Figs. 3.1, 3.3. On the other hand, the MVSS algorithm maintained the same level of performance despite the high level of noise in Figs. 3.1, 3.3. With a low level of noise, it still showed same

improvement compared to the other algorithms.

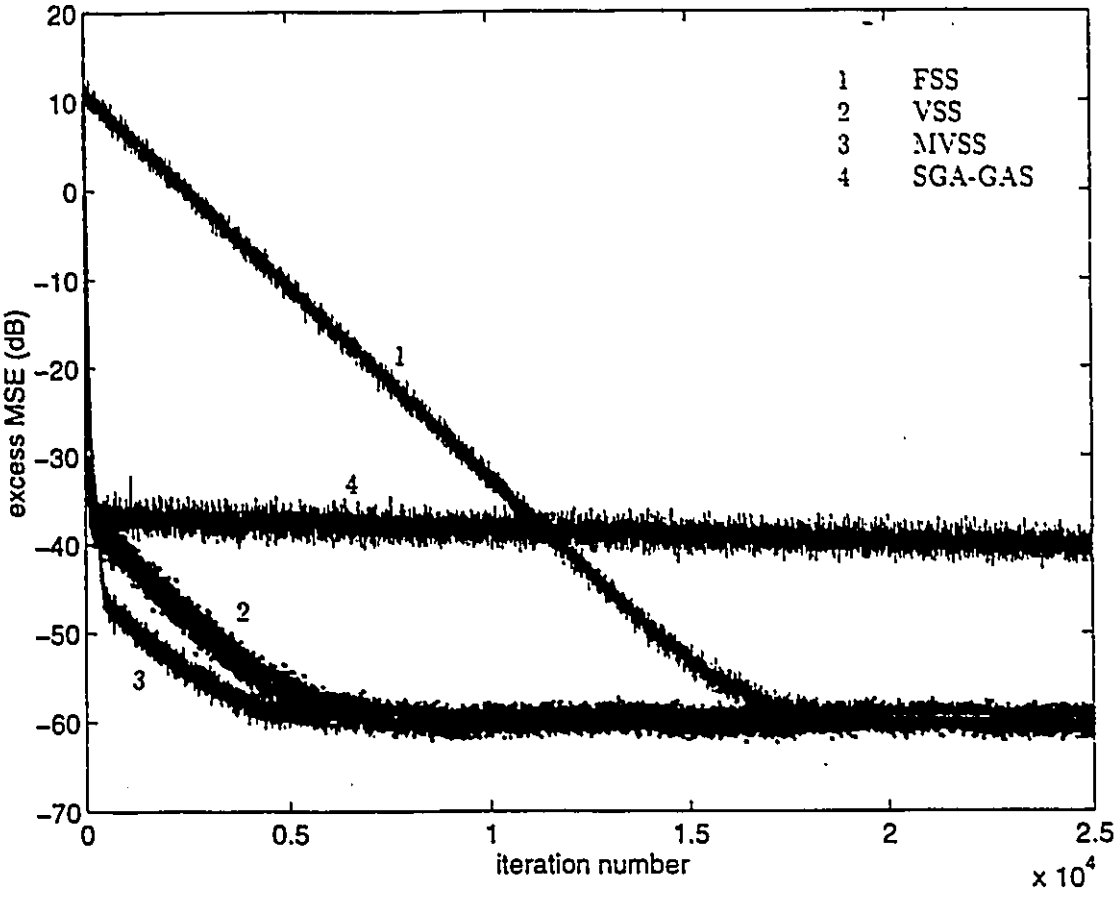

Fig. 3.7 Comparison of excess MSE of various adaptive algorithms for the white input case, when the added noise to the desired signal is of 0.001 variance.

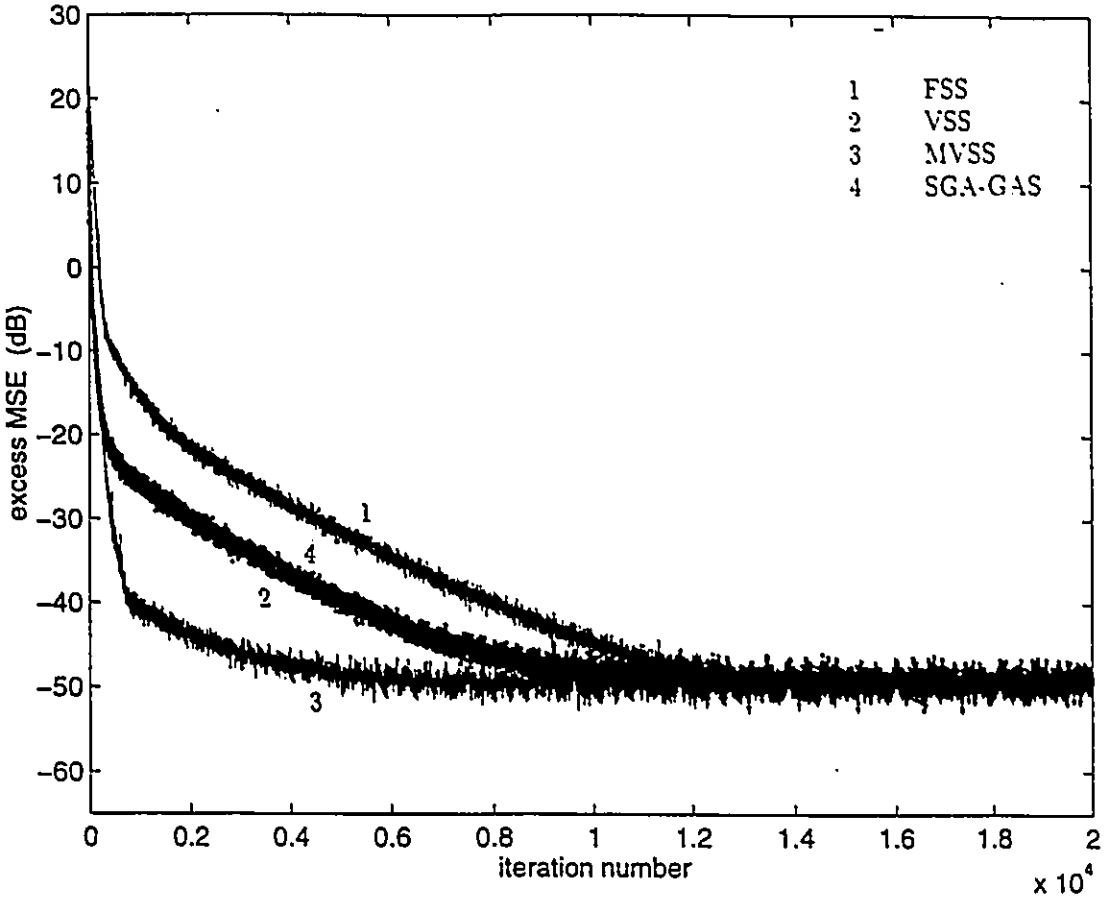

Fig. 3.8 Comparison of excess MSE of various adaptive algorithms for the correlated input case, when the added noise to the desired signal is of 0.001 variance.

3.3.5 Example 5: nonstationary optimal weight vector

In this example, we utilized a similar model to that used in the first example, except that the optimal weight vector is nonstationary and generated according to the random walk model [46],

$$\mathbf{W}^*(n) = \mathbf{W}^*(n-1) + \eta(n-1) \tag{3.43}$$

where $\eta(n)$ is a stationary process of zero-mean and a correlation matrix $\sigma_n^2\mathbf{I}$. Fig. 3.9 compares the four algorithm with $\sigma_n^2 = 0.001$. Optimal parameters for a given level of nonstationarity were calculated to achieve minimum misadjustment. The MVSS algorithm is used with $\beta = 0.6$, $\alpha = 0.97$, and according to Eq.(3.38), $\gamma^* = 3.8 \times 10^{-3}$. Parameters used for the VSS algorithm are $\alpha_{vss} = 0.97$, and $\gamma_{vss} = 7.65 \times 10^{-4}$ which were utilized in [56]. For this level of nonstationarity, the optimal step size for the FSS algorithm is found to be $\mu_{FSS}^* = 0.0316$ [111]. Since the excess MSE of SGA-GAS is relatively insensitive to the selection of ρ in nonstationary environments [68], we used $\rho = 1 \times 10^{-4}$ which was chosen experimentally to obtain the best performance of the algorithm.

Fig. 3.9 illustrates the ability of the MVSS to operate as well as the FSS and the VSS in nonstationary environment. Fig. 3.10 shows the step size trajectory for the MVSS, VSS and SGA-GAS algorithms. Comparing the step size trajectory of MVSS with its MSE evolution in Fig. 3.9 demonstrates the effectiveness of its step size adjustment in tracking the MSE changes accurately.

In Table 3.1 we compare the MVSS to the other algorithms under different level of nonstationarity. Moreover, Table 3.1 compares theoretical results of misadjustment obtained from Eq.(3.35) with the results of simulation. When $\sigma_n^2 = 0.01$, the MVSS is used with $\alpha = 0.97$, $\beta = 0.4$, and we found that $\gamma^* = 6.9 \times 10^{-3}$. The VSS and the SGA-GAS algorithms were used with $\alpha_{vss} = 0.97$, $\gamma_{vss} = 1 \times 10^{-3}$, and $\rho = 1 \times 10^{-4}$ which were chosen experimentally to provide the minimum level of misadjustment. The FSS algorithm is used with the optimum step size $\mu_{FSS}^* = 0.1$. For $\sigma_n^2 = 0.0001$, the MVSS is used with $\alpha = 0.97$, $\beta = 0.8$, and $\gamma^* = 2.7 \times 10^{-3}$. The VSS and SGA-GAS were used with $\alpha_{vss} = 0.97$, $\gamma_{vss} = 1 \times 10^{-4}$, and $\rho = 1 \times 10^{-4}$. The FSS is used with $\mu^* = 0.01$. It is obvious from Table 3.1 that the MVSS algorithm can be as good as the FSS and the VSS algorithms even when operating under different levels of nonstationarity. Furthermore, it can be seen that misadjustment predicted in Eq.(3.35) agrees well with simulation results.

Table 3.1

Comparison of theoretical and experimental various VSS LMS algorithms misadjustment

	Calculated Misadjustment		Measured Misadjustment			
σ_n^2	MVSS(Eq.(3.35)	FSS	MVSS	SGA-GAS	FSS	VSS
0.01	0.416	0.4	0.57	0.59	0.56	0.55
0.001	0.129	0.126	0.157	0.19	0.136	0.137
0.0001	0.04	0.04	0.044	0.065	0.039	0.049

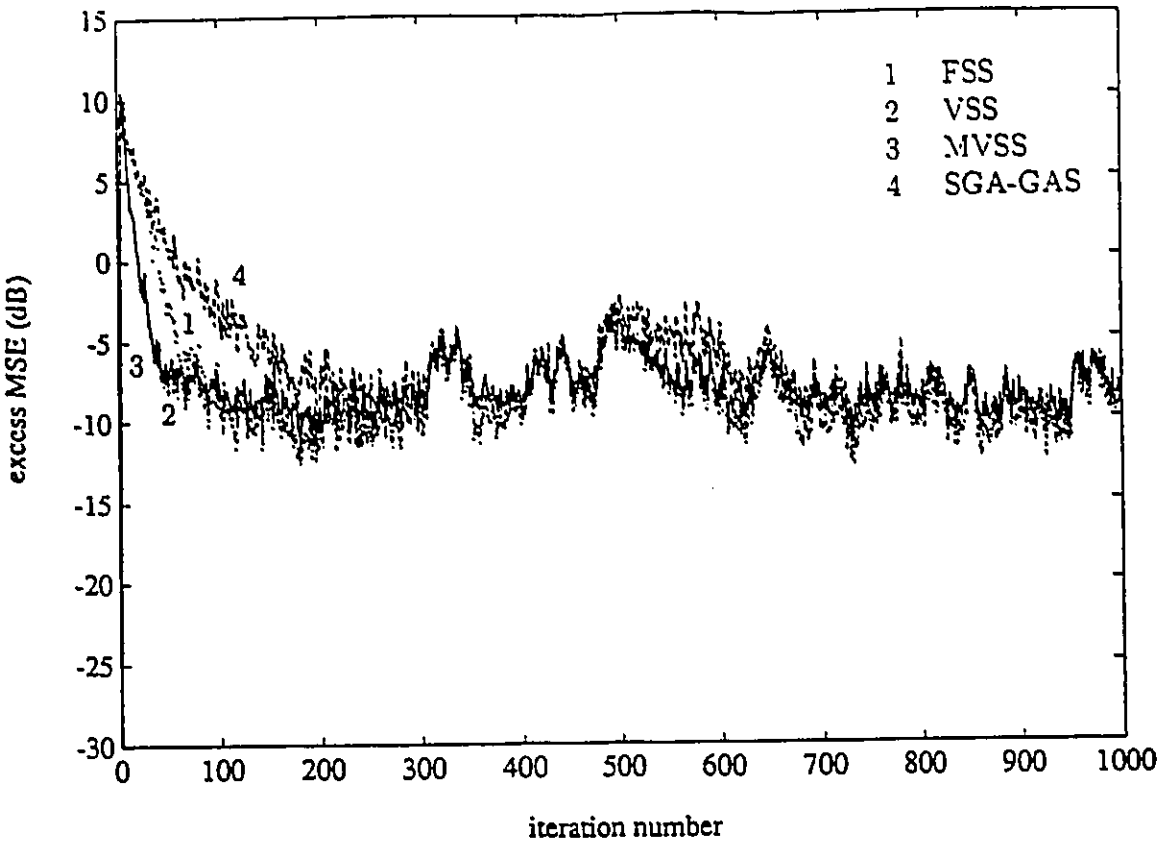

Fig. 3.9 Comparison of excess MSE of various adaptive algorithms for a non-stationary optimal weight vector.

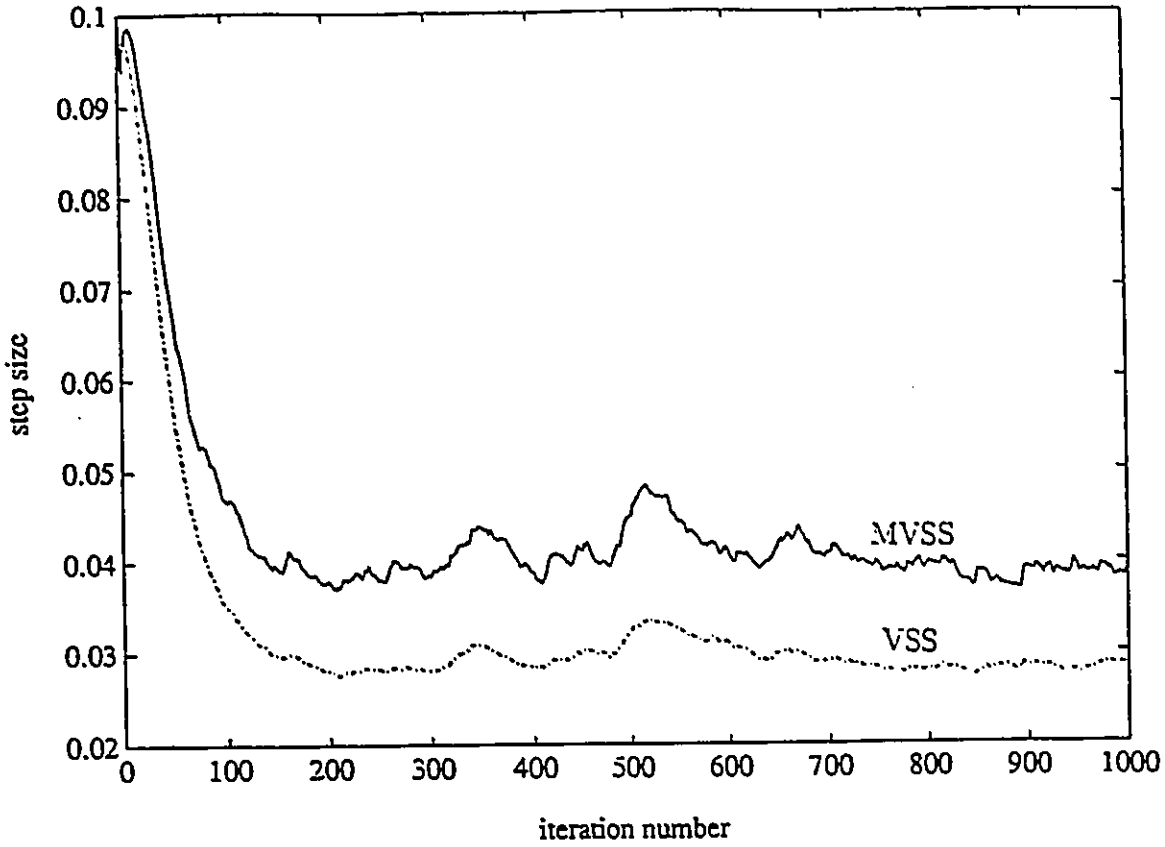

Fig. 3.10 (a) Comparison of step size evolution of the VSS algorithm and the MVSS algorithm for a nonstationary optimal weight vector.

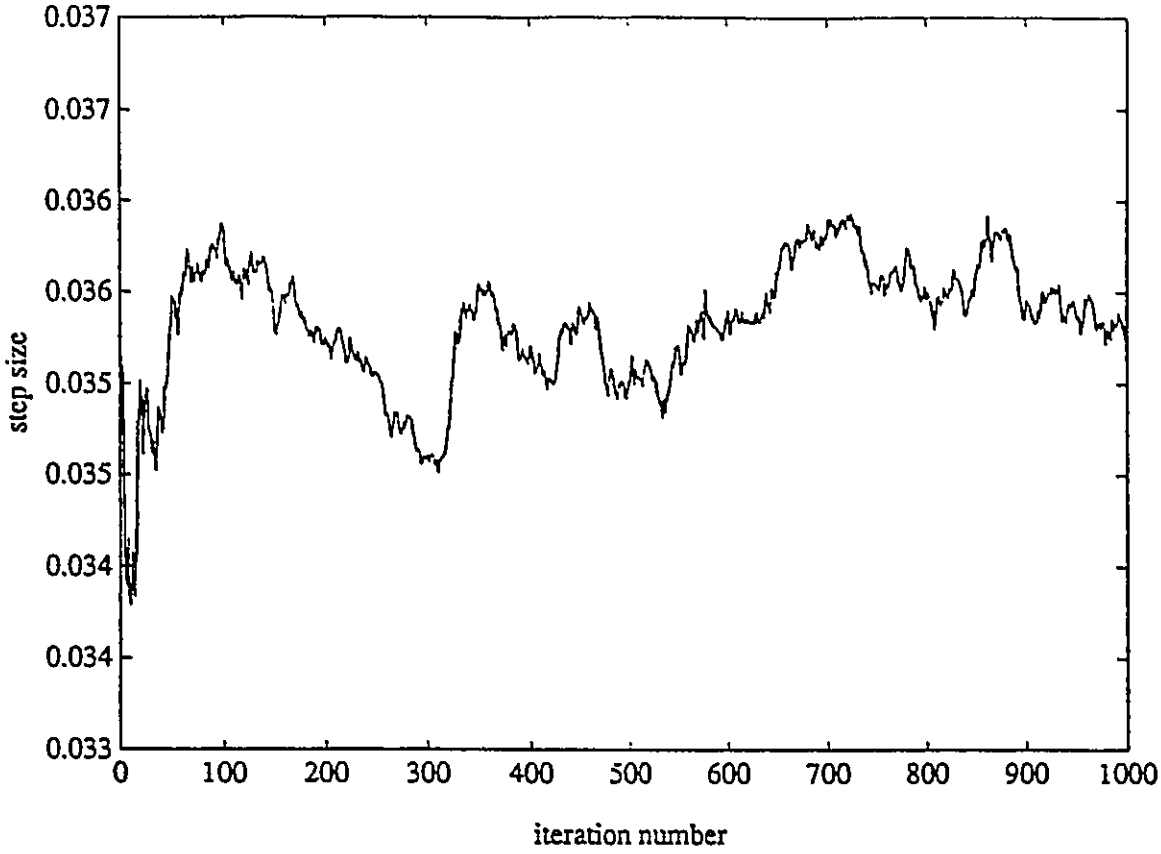

Fig. 3.10 (b) Step size evolution of the SGA-GAS algorithm for a nonstationary optimal weight vector.

The value of γ^* in Eq.(3.38) providing minimum misadjustment in nonstationary environment is verified through simulations for a given α, β, and level of misadjustment σ_n^2. Fig. 3.11 displays the excess MSE of the MVSS algorithm with $\alpha = 0.97$, $\beta = 0.8$, $\gamma_1 = \gamma^* = 2.7 \times 10^{-3}$, $\gamma_2 = 1 \times 10^{-4}$, and $\gamma_3 = 8 \times 10^{-3}$ when $\sigma_n^2 = 0.0001$. Note that $\gamma_2 < \gamma^* < \gamma_3$. It is clear from Fig. 3.11 that the optimal value $\gamma^* = \gamma_1$ obtained fror. Eq.(3.38) indeed provides the minimum possible misadjustment for a given level of nonstationarity.

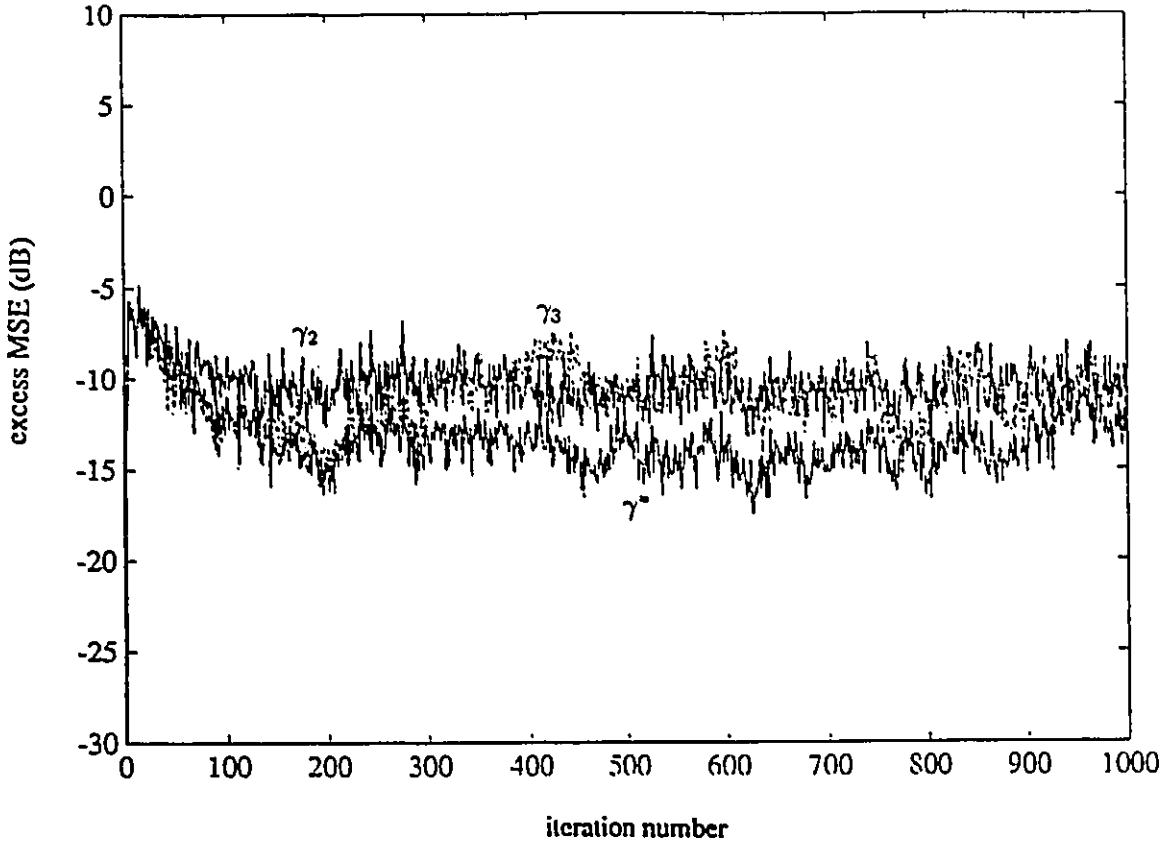

Fig. 3.11 Comparison of excess MSE of the MVSS algorithm for a nonstationary optimal weight vector.

3.3.6 Example 6: nonstationary input signal (speech)

The MVSS is applied here to the cancellation of echo produced by a real hybrid. The input signal is a real speech of a male voice obtained at a sampling rate of 8kHz, Fig. 3.12. The impulse response of a real hybrid was measured at a sampling rate of 8kHz and is shown in Fig. 3.13. In order to cope with the changes in the speech signal energy, the step size of the MVSS algorithm is normalized so that its recursion becomes as

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \frac{\mu(n)}{\delta + \mathbf{X}^{\mathrm{T}}(n)\mathbf{X}(n)} e(n)\mathbf{X}(n) \tag{3.44}$$

where δ is a small positive number. Note that the asymptotic convergence of Eq.(3.44) is assured by [46]

$$0 < E\{\mu(n)\} < 2 \tag{3.45}$$

It is shown in [77] that the fastest speed of convergence of the normalized LMS (NLMS) algorithm is obtained when $\mu = 1$. This value, however, results in high misadjustment value. Accordingly, μ_{max} for the normalized MVSS is set to 1, and μ_{min} is chosen to attain the required level of misadjustment. In this experiment, the number of the FIR filter coefficients are selected to be 30 to achieve 20 dB steady state echo return loss enhancement (ERLE). The ERLE is defines as [76]

$$ERLE = 10log_{10}\left(\frac{E\{d^2(n)\}}{E\{e^2(n)\}}\right) \tag{3.46}$$

Note that the undermodeling will result in some "colourness" in the noise. For the normalized MVSS, we used $\alpha = 0.97$, $\beta = 0.95$, $\gamma = 1 \times 10^{-5}$, and $\mu_{min} = 0.04$. The performance of the algorithm is compared with the NLMS algorithm with two step sizes, $\mu = 1$ and $\mu = 0.04$. In Fig. 3.14, we plotted $tr(\mathbf{V}^{\mathrm{T}}(n)\mathbf{V}(n))$; the squared norm of the weight error vector. This measure is directly proportional to algorithm misadjustment and therefore appeared to be more sensitive to and illustrative of the impact of the step size value on the algorithm performance compared to the ERLE. Fig. 3.14 shows that for a speech input the normalized MVSS behaves better than the NLMS algorithm in the sense that it can compromise simultaneously between the two states of the NLMS of fast convergence (attained when $\mu = 1$) and low error in the adaptive filter coefficients (attained when $\mu = 0.04$).

It should be mentioned that the theoretical and simulation results presented for the proposed algorithm assumed uncorrelated noise sequence. This assumption is generally true in most applications. However, in some specific applications, the noise can be correlated. For example, in acoustic echo cancellation, the acoustic noise can be a correlated one.

In this situation, the term $E\{\xi(n)\xi(n-1)\}$ in Eq.(3.10) will not vanish, and it will appear in the step size equation (3.12). Consequently, the improvement provided by the MVSS algorithm over the VSS algorithm, Eq.(3.6), will depend on the relative ratio between $E\{\xi(n)\xi(n-1)\}$ and $E\{\xi^2(n)\}$. On the other hand,

if the correlation between the noise samples is known to decrease as time difference between them increases, we could in this case use the measure $e(n)e(n-D)$ such that $D \ll N$. Deterioration in the performance of the algorithm in the presence of correlated noise is reduced since the effect of $E\{\xi(n)\xi(n-D)\}$ is guaranteed to be inferior to that of $E\{\xi(n)\xi(n-1)\}$.

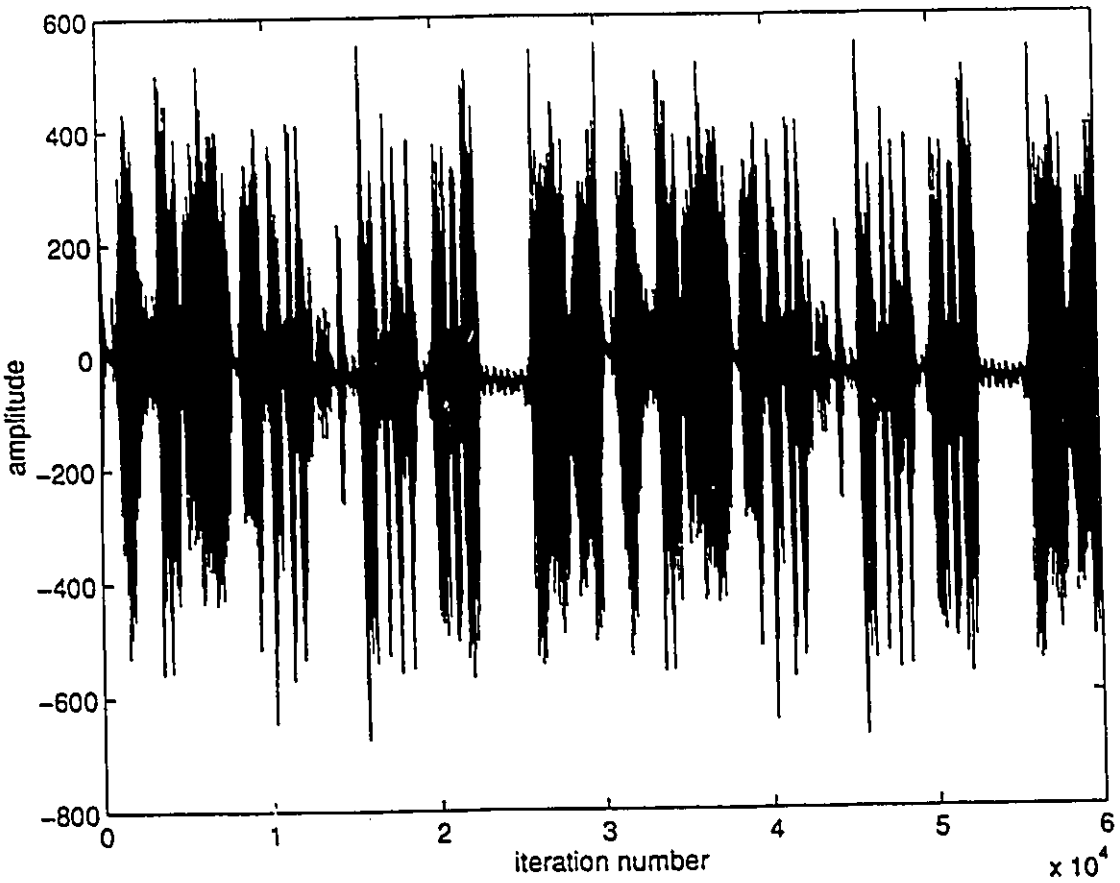

Fig. 3.12 Real speech signal of a male voice. The sampling frequency is 8kHz.

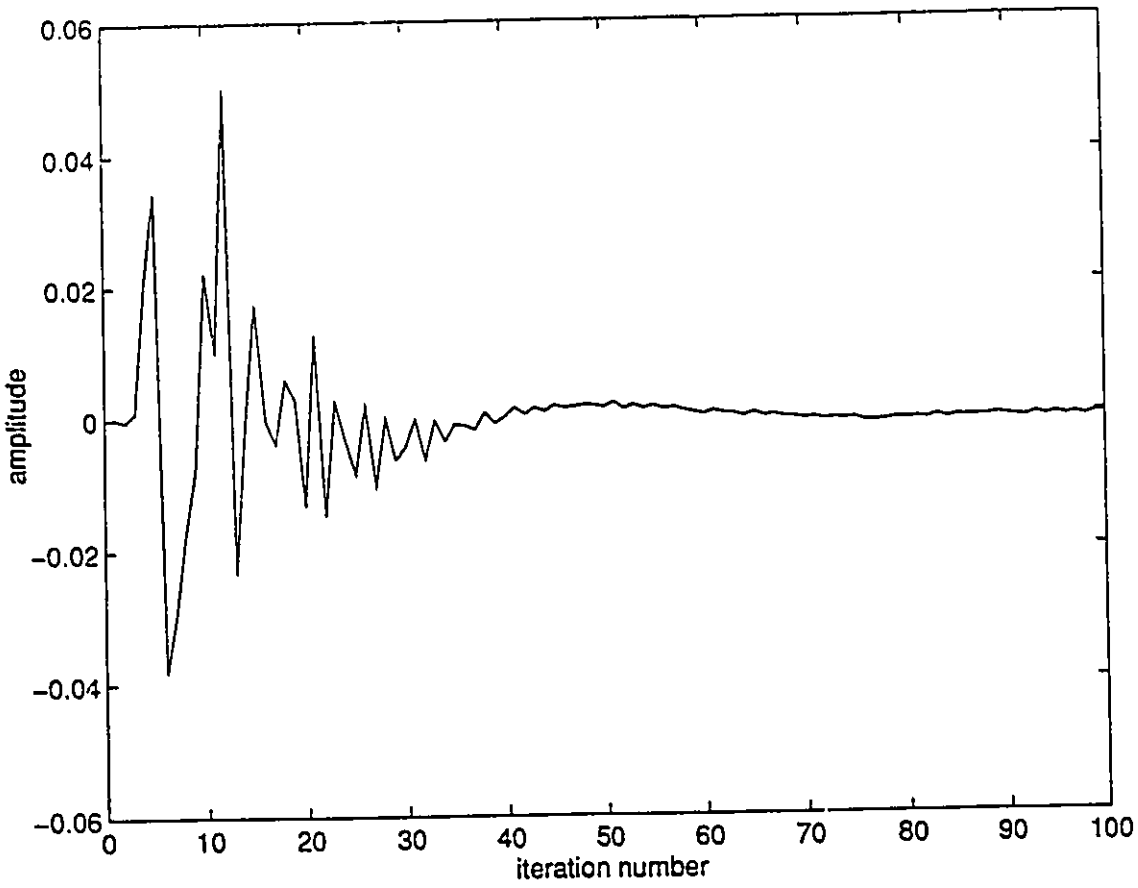

Fig. 3.13 Measured impulse response of a real hybrid. The sampling frequency is 8kHz.

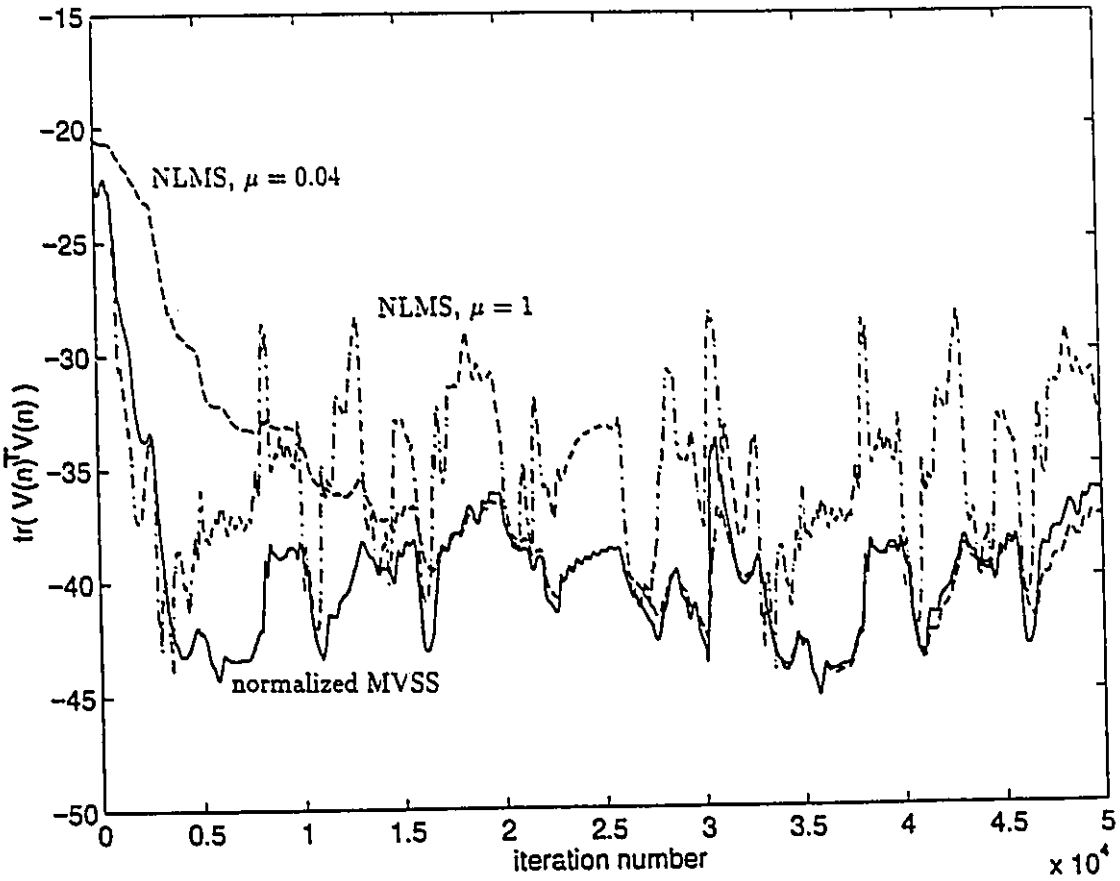

Fig. 3.14 Comparison of performance of the normalized MVSS and the NLMS algorithms for example 5.

3.4 Conclusion

This chapter presented a new variable step size algorithm for adaptive FIR filters. The proposed algorithm provides rapid convergence and small steady state misadjustment even in the presence of noise disturbances. The step size adjustment of the algorithm is based upon a function of the crosscorrelation between $e(n)$ and $e(n-1)$. This has rendered the algorithm insensitive to uncorrelated noise. We presented a theoretical steady state performance analysis of the algorithm under stationary and nonstationary conditions. The algorithm was also extended for applications to nonstationary input signals and applied to echo cancellation with real speech and hybrid. Simulation results showed that the algorithm has a significant convergence rate improvement over those algorithms in stationary environment for the same excess MSE in both high and low SNR environments. The performance in nonstationary cases is comparable to the standard LMS algorithm.

Chapter 4

Time/Order Update Gradient-Based Adaptive Algorithm

The LMS algorithm [46] is very popular in adaptive signal processing due to its inherent simplicity. However, the main disadvantage of the LMS has been its relatively slow convergence and its sensitivity to the input signal statistics. In this chapter, we propose an algorithm based on the iterative steepest descent technique as the LMS [111, 113] but utilizing a new search method to reach the bottom of the MSE surface. The goal is to use time- and order-updating procedures simultaneously so as to ensure efficient use of the available data and the most recent information about the location of the weight vector relative to the optimum. At time instant n, the algorithm uses the instantaneous mean square *partial a posteriori* error to increment the first component of the weight vector leaving the remaining components unchanged. Geometrically, this is equivalent to searching

for the minimum of a single variable quadratic function, the variable being the first component of the weight vector (since all other components are fixed). The resulting new location of the weight vector will have the first component closer to the global minimum of the MSE surface. Given that improved weight vector, a new error is calculated and used to obtain a better gradient estimate to update the second component of the weight vector. The process is repeated until all components of the weight vector are updated. In the standard LMS algorithm, all weight vector components are updated simultaneously with the same *a priori* error [1].

The LMS algorithm has also been used more than once at each sample time to improve the speed of adaptation, leading to the data reusing LMS algorithm [7, 70, 94]. Again, in this approach the current data are utilized to increment all weight vector components simultaneously. Then, the resulting new weight vector and the same current data are used to update the new weight vector. The process continues for M times, where M is the number of times the current data are used. In [71], individual taps were updated based on the *a priori* error. The proposed algorithm, however, follows a recursive-order procedure to update the components of its weight vector where each stage takes advantage of the new information available from the previous stages and the *partial a posteriori* error to perform its updating process. Intuitively, the proposed algorithm is expected to exhibit

improved convergence speed over the LMS algorithm when the input signal is highly correlated leading to a highly flattened MSE surface with a ridge bottom rather than the typical bowl shaped surface [1, 113].

4.1 A posteriori error-based order update of the LMS algorithm

The setup considered here is the basic system identification problem. At time instant n, the available information consists of the input data vector $\mathbf{X}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^T$, the adaptive FIR filter weight vector $\mathbf{W}(n-1) = [w_1(n-1), w_2(n-1), \ldots, w_N(n-1)]^T$, and the desired signal $d(n)$. The error at this instant is given by

$$e_0(n) = d(n) - \mathbf{X^T}(n)\mathbf{W}(n-1) \tag{4.1}$$

This is referred to here as the *a priori* error at n, i.e, error at instant n prior to updating any of the weights. The *partial a posteriori* error is defined as the error following the update of the first weight vector component $w_1(n)$ and is given by

$$e_1(n) = d(n) - \mathbf{X^T}(n)\mathbf{W^1}(n) \tag{4.2}$$

where

$$\mathbf{W^1}(n) = [w_1(n), w_2(n-1), \ldots, w_N(n-1)]^T \tag{4.3}$$

Assuming for the time being that $e_1(n)$ can be calculated before actually obtaining $w_1(n)$, then $\frac{\partial e_1^2(n)}{\partial w_1(n)}$ would be a better gradient estimate to be used in the update of

$w_1(n-1)$ than that based on the *a priori* error $e_0(n)$, i.e,

$$w_1(n) = w_1(n-1) - \mu \frac{\partial e_1^2(n)}{\partial w_1(n)} \tag{4.4}$$

We will show later how $e_1(n)$ is to be obtained in a causal fashion. Repeating the same procedure, we now update the second component $w_2(n-1)$ as

$$w_2(n) = w_2(n-1) - \mu \frac{\partial e_2^2(n)}{\partial w_2(n)} \tag{4.5}$$

where

$$e_2(n) = d(n) - \mathbf{X^T}(n)\mathbf{W^2}(n) \tag{4.6}$$

and

$$\mathbf{W^2}(n) = [w_1(n), w_2(n), w_3(n-1), \ldots, w_N(n-1)]^T \tag{4.7}$$

Note that $\mathbf{W^2}(n)$ holds more information about the optimal weight vector than $\mathbf{W^1}(n)$ and therefore was used to obtain a better gradient estimate to update this component. We proceed in updating the remaining components of $\mathbf{W}(n-1)$ in a similar manner till the Nth component of $\mathbf{W}(n-1)$ is updated

$$w_N(n) = w_N(n-1) - \mu \frac{\partial e_N^2(n)}{\partial w_N(n)} \tag{4.8}$$

where

$$e_N(n) = d(n) - \mathbf{X^T}(n)\mathbf{W^N}(n) \tag{4.9}$$

and

$$\mathbf{W^N}(n) = [w_1(n), w_2(n), \ldots, w_N(n)]^T \tag{4.10}$$

At this point, all components of $\mathbf{W}(n)$ will have been computed. Each has been updated with the best information available at the point of update.

We now address the issue of using the *partial a posteriori* error $e_i(n) = d(n) - \mathbf{X}^{\mathrm{T}}(n)\mathbf{W}^{\mathrm{i}}(n)$ to update $w_i(n-1)$, $i = 1, \ldots, N$.

Given Eqs.(4.2), (4.6), (4.9), we can see that

$$\frac{\partial e_i^2(n)}{\partial w_i(n)} = -2e_i(n)x(n-i+1) \ , \quad i = 1, 2, \ldots, N \tag{4.11}$$

The question now is how to obtain $e_1(n)$ as given by Eq.(4.2) before actually updating $w_1(n)$. Recalling Eqs. (4.1) and (4.2), then eliminating $d(n)$ from both equations, we can see that

$$\begin{aligned} e_1(n) &= e_0(n) + \mathbf{X}^{\mathrm{T}}(n)[\, w_1(n) - w_1(n-1),\, 0, \ldots, 0\,]^T \\ &= e_0(n) + x(n)(\, w_1(n) - w_1(n-1)\,) \end{aligned} \tag{4.12}$$

Using Eqs.(4.4), (4.11) in (4.12), we find that $e_1(n)$, the *a posteriori* error following the update of $w_1(n)$, can be calculated from the knowledge of $e_0(n)$ and the new input data as

$$e_1(n) = \frac{e_0(n)}{1 + 2\mu x^2(n)} \tag{4.13}$$

Similarly, it can be shown that the *partial a posteriori* error of the *ith* stage is described by the following equations

$$e_i(n) = \frac{e_{i-1}(n)}{1 + 2\mu x^2(n-i+1)} \ , \quad i = 1, 2, \ldots, N \tag{4.14}$$

The complete weight update relation of the proposed algorithm can now be put in the form

$$\mathbf{W}(n) = \mathbf{W}(n-1) + 2\mu\mathbf{E}(n)\mathbf{X}(n) \tag{4.15}$$

where the *partial a posteriori* errors matrix $\mathbf{E}(n)$ is the diagonal matrix

$$\mathbf{E}(n) = \begin{bmatrix} e_1(n) & 0 & 0 & \dots & 0 \\ 0 & e_2(n) & 0 & \dots & 0 \\ 0 & 0 & e_3(n) & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & 0 \\ 0 & 0 & 0 & \dots & e_N(n) \end{bmatrix} \tag{4.16}$$

and $e_i(n)$, $i = 1, 2, \dots, N$ are given by Eq.(4.14).

Note that at each time instant n, one new input sample $x(n)$ is obtained and the oldest sample $x(n-N)$ is discarded. By storing the previous computed values $(\mu x(n-i))(x(n-i))$, $i = 1, 2, \dots, N-1$, only two multiplications are required per iteration to calculate $(\mu x(n))(x(n))$, only one if μ is a power of 2, N divisions are needed for evaluating $\frac{e_{i-1}(n)}{1+2\mu x^2(n-i+1)}$, $i = 1, 2, \dots, N$. Thus, the proposed algorithm adds two multiplications and N divisions per iteration to the complexity of the standard LMS algorithm [1].

4.2 Guidelines for the choice of the step size value

4.2.1 Convergence in the mean

As with the LMS-based algorithms, the choice of the step size greatly affects the algorithm's convergence. Here, we will consider establishing some guidelines for

the choice of the adaptation step size based on the study of the system steady state performance. From Eq.(4.14), the *ith* stage *partial a posteriori* error $e_i(n)$ can expressed as

$$e_i = \frac{e_0(n)}{\prod_{j=1}^{i}(1 + 2\mu x^2(n-j+1))} \quad , \quad i = 1, 2, \ldots, N \tag{4.17}$$

Accordingly, the algorithm weight recursion in Eq.(4.15) is rewritten as

$$\mathbf{W}(n) = \mathbf{W}(n-1) + 2\mathbf{M}(n)e_0(n)\mathbf{X}(n) \tag{4.18}$$

where the diagonal matrix $\mathbf{M}(n)$ is given by

$$\begin{aligned} \mathbf{M}(n) \;=\; & \mathrm{diag}\{ \frac{\mu}{(1 + 2\mu x^2(n))} \, , \, \frac{\mu}{(1 + 2\mu x^2(n))(1 + 2\mu x^2(n-1))} \, , \, \cdots \\ & \cdots \, , \, \frac{\mu}{\prod_{j=1}^{N}(1 + 2\mu x^2(n-j+1))} \} \end{aligned} \tag{4.19}$$

Using Eq.(4.1), Eq.(4.18) can be rearranged as

$$\mathbf{W}(n) = (\mathbf{I} - 2\mathbf{M}(n)\mathbf{X}(n)\mathbf{X}^{\mathrm{T}}(n))\mathbf{W}(n-1) + 2\mathbf{M}(n)d(n)\mathbf{X}(n) \tag{4.20}$$

To make the analysis tractable, we use the common assumption of zero-mean white Gaussian input. Moreover, $\mathbf{M}(n)$ is assumed to be statistically independent of $\mathbf{X}(n)$ and $e_0(n)$. This assumption is practically valid for small μ so that the statistics of $\mathbf{M}(n)$ are slowly varying with respect to those of $\mathbf{X}(n)$ and $e_0(n)$ [45, 68] (note that the data dependent terms, $2\mu x^2(n)$, ..., $2\mu x^2(n-N+1)$, in $\mathbf{M}(n)$ are multiplied by μ, which strengthens the validity of our assumption). Taking

the expected value of both sides of Eq.(4.20) and using the common independence assumption of $\mathbf{X}(n)$ and $\mathbf{W}(n-1)$ [47], we find that

$$E\{\mathbf{W}(n)\} = (\mathbf{I} - 2\hat{\mathbf{M}}\mathbf{R})E\{\mathbf{W}(n-1)\} + 2\hat{\mathbf{M}}\mathbf{P} \tag{4.21}$$

where $\mathbf{R} = \sigma_x^2\mathbf{I}$ is the autocorrelation matrix of the input data, σ_x^2 is the input data power, $\mathbf{P}$ is the crosscorrelation vector between the input and the desired signal, and $\hat{\mathbf{M}}$ is the diagonal matrix

$$\hat{\mathbf{M}} = \text{diag}\{\ \mu\beta,\ \mu\beta^2,\ \mu\beta^3,\ldots,\ \mu\beta^N\} \tag{4.22}$$

where $\beta = \frac{1}{1+2\mu\sigma_x^2}$. It is straightforward to see that all modes of Eq.(4.21) are stable (i.e, that the weight vector will converge in the mean sense) if

$$0 < \mu\beta < \frac{1}{\sigma_x^2} \tag{4.23}$$

Note that if this condition is satisfied, then the matrix $\hat{\mathbf{M}}$ is guaranteed to be positive definite and, consequently, $E\{\mathbf{W}(n)\}$ in Eq.(4.21) will converge to the optimal vector $\mathbf{W}^* = \mathbf{R}^{-1}\mathbf{P}$. From Eq.(4.23) it can be seen that a sufficient condition on the choice of the step size μ for mean convergence of the algorithm is

$$\mu > 0 \tag{4.24}$$

Despite the fact that in Eq.(4.15), μ was identified as the adaptation step size; it is not the actual step size used in the coefficient update of the proposed algorithm. Recalling Eq.(4.18), we can see that the matrix $\mathbf{M}(n)$ provides the actual control

on the step size used in the coefficient update, μ being just a parameter in that update. Even though μ does not have an upper limit, $\mathbf{M}(n)$ in Eq.(4.19) is inherently bounded $\forall\, \mu > 0$ thus ensuring the stability of the algorithm in the mean sense for all μ. However, mean convergence of the proposed algorithm does not imply necessarily a satisfactory performance or even convergence in the mean square sense. A large step size μ or large input power will result in very high variance of the algorithm coefficients and subsequent large steady state error [1, 111, 113]. Since there is always a limit on the tolerable level of misadjustment for a certain application, an explicit expression for algorithm misadjustment is essential to provide a restrictive upper bound on the actual step size value.

4.2.2 Misadjustment

The misadjustment of the proposed algorithm is now derived by extending the approach developed by Widrow [111, 113].

The proposed algorithm defined by Eq.(4.18) uses an estimated gradient in its recursion, consequently the algorithm can be represented as [111, 113]

$$\mathbf{W}(n) = \mathbf{W}(n-1) - \mathbf{M}(n)(\ \nabla(n) + \mathbf{N}(n)\ ) \tag{4.25}$$

where $\nabla(n)$ is the true gradient and $\mathbf{N}(n)$ is the gradient estimation noise vector which has a zero mean and independent components. We introduce the error vector $\mathbf{V}(n) = \mathbf{W}(n) - \mathbf{W}^*$, where $\mathbf{W}^*$ is the optimal weight vector. Recall that the true gradient is given by $2\mathbf{R}\mathbf{V}(n)$ [1, 111, 113], and accordingly, Eq.(4.25) can

be written in the form

$$\mathbf{V}(n) = (\ \mathbf{I} - 2\mathbf{M}(n)\mathbf{R}\ )\mathbf{V}(n-1) - \mathbf{M}(n)\mathbf{N}(n) \tag{4.26}$$

Postmultiplying both sides of Eq.(4.26) by $\mathbf{V}^{\mathbf{T}}(n)$, and taking the expected value results in

$$\begin{aligned} E\{\mathbf{V}(n)\mathbf{V}^{\mathbf{T}}(n)\} &= E\{\mathbf{V}(n-1)\mathbf{V}^{\mathbf{T}}(n-1)\} - 2E\{\mathbf{M}(n)\}\mathbf{R}E\{\mathbf{V}(n-1)\mathbf{V}^{\mathbf{T}}(n-1)\} \\ &-2E\{\mathbf{V}(n-1)\mathbf{V}^{\mathbf{T}}(n-1)\}\mathbf{R}E\{\mathbf{M}(n)\} + 4E\{\mathbf{M}(n)\mathbf{R}\mathbf{V}(n-1)\mathbf{V}^{\mathbf{T}}(n-1)\mathbf{R}\mathbf{M}(n)\} \\ &+E\{\mathbf{M}(n)\mathbf{N}(n)\mathbf{N}^{\mathbf{T}}(n)\mathbf{M}(n)\} \end{aligned} \tag{4.27}$$

Note that $\mathbf{M}(n)$ and $\mathbf{R}$ are diagonal matrices. Defining $\mathbf{Z_1}(n)$ as a $N \times 1$ second moment vector whose entries are the diagonal elements of $E\{\mathbf{V}(n)\mathbf{V}^{\mathbf{T}}(n)\}$, then $\mathbf{Z_1}(n)$ can be expressed as

$$\begin{aligned} \mathbf{Z_1}(n) &= (\mathbf{I} - 4E\{\mathbf{M}(n)\}\mathbf{R} + 4E\{\mathbf{M^2}(n)\}\mathbf{R^2}\ )\mathbf{Z_1}(n-1) \\ &\quad +E\{\mathbf{M^2}(n)\}\mathrm{diag}\left\{E\{\mathbf{N}(n)\mathbf{N}^{\mathbf{T}}(n)\}\right\} \end{aligned} \tag{4.28}$$

After convergence, the true gradient $\nabla(n)$ is zero and

$$\mathbf{N}(n) = -2e_0^*(n)\mathbf{X}(n) \tag{4.29}$$

where $e_0^*(n)$ is the optimal steady state error which is uncorrelated with $\mathbf{X}(n)$ and assumed zero-mean and independent [111]. Thus, at convergence

$$E\{\mathbf{N}(n)\mathbf{N}^{\mathbf{T}}(n)\} = 4E\{e_0^{*2}\}\mathbf{R}$$

$$= 4\epsilon_{min}\mathbf{R} \tag{4.30}$$

where ϵ_{min} is the minimum MSE. Recall that the excess MSE is given by

$$\begin{aligned} \epsilon_{ex}(n) &= \epsilon(n) - \epsilon_{min} \\ &= E\{\mathbf{V}^{\mathrm{T}}(n)\mathbf{R}\mathbf{V}(n)\} \\ &= tr(\,\mathbf{R}\mathbf{Z}_1(n)\,) \end{aligned} \tag{4.31}$$

where $\epsilon(n)$ is the MSE. The misadjustment is defined as [46, 111, 113]

$$M = \frac{\epsilon_{ex}(\infty)}{\epsilon_{min}} \tag{4.32}$$

From Eq.(4.28) and Eq.(4.30), we find that

$$\mathbf{Z}_1(\infty) = \left(E\{\mathbf{M}(\infty)\}\mathbf{R} - E\{\mathbf{M}^2(\infty)\}\mathbf{R}^2\right)^{-1} E\{\mathbf{M}^2(\infty)\}\epsilon_{min}\mathbf{R} \tag{4.33}$$

where $E\{\mathbf{M}(\infty)\} = \hat{\mathbf{M}}$, where $\hat{\mathbf{M}}$ is defined in Eq.(4.22). We can also show that

$$E\{\mathbf{M}^2(\infty)\} = \mathrm{diag}\{\ \mu^2\alpha,\ \mu^2\alpha^2,\ \mu^2\alpha^3, \ldots,\ \mu^2\alpha^N\ \} \tag{4.34}$$

where

$$\alpha = \frac{1}{(1 + 2\mu\sigma_x^2)^2 + 8\mu^2\sigma_x^4} \tag{4.35}$$

If we use Eqs.(4.31),(4.33), (4.34) in Eq.(4.32), we obtain the following expression for the proposed algorithm misadjustment

$$M = \sum_{i=1}^{N} \frac{\mu\sigma_x^2\alpha^i}{\beta^i - \mu\sigma_x^2\alpha^i} \tag{4.36}$$

To provide a clear insight into the algorithm steady state characteristics in comparison with standard LMS, a simple estimate of the misadjustment in Eq.(4.36) is obtained as follows. We note from Eq.(4.35) that

$$\alpha = \frac{\beta^2}{1 + 8\mu^2\sigma_x^4\beta^2} \tag{4.37}$$

For small values of μ, $8\mu^2\sigma_x^4\beta^2 \ll 1$ and $\alpha \simeq \beta^2$. Thus

$$M \simeq \sum_{i=1}^{N} \frac{\mu\sigma_x^2\beta^i}{1 - \mu\sigma_x^2\beta^i} \tag{4.38}$$

Also, if μ is sufficiently small such that $\mu\sigma_x^2\beta^i \ll 1 \; \forall \; i = 1, \; 2, \ldots, \; N$, then

$$M \approx \mu\sigma_x^2 \sum_{i=1}^{N} \beta^i \tag{4.39}$$

Note that since $\mu > 0$, then $0 < \beta < 1$, and consequently

$$M \approx \mu\sigma_x^2 \frac{1 - \beta^N}{1 - \beta} \tag{4.40}$$

For the LMS algorithm, the misadjustment is given by [111, 113]

$$M_{LMS} = \mu_{LMS}\sigma_x^2 N \tag{4.41}$$

To obtain the same misadjustment as the standard LMS, i.e., $M = M_{LMS}$, it can be seen from Eq.(4.40) and Eq.(4.41) that

$$\mu = \mu_{LMS} \frac{N\,(1 - \beta)}{(1 - \beta^N)} \tag{4.42}$$

Owing to the fact that $\frac{1-\beta^N}{1-\beta} < N$, then we can deduce from Eq.(4.42) that for the same misadjustment

$$\mu \; > \; \mu_{LMS} \tag{4.43}$$

Eqs.(4.40) and (4.42) present some guidelines fo the selection of the adaptation step size of the proposed algorithm for given misadjustment. Finally, from Eq.(4.43) we can see that the proposed algorithm will result in faster convergence for the same misadjustment as expected.

4.3 Selective order updating

The proposed algorithm as explained earlier conducts its order-updating procedure in a sequential manner, i.e., $w_1(n)$ then $w_2(n) \ldots$ up to $w_N(n)$. Given that updating $w_i(n)$ affects all updates of $w_j(n)$, $j > i$, and based on the fact that depending on the bowl shape, the error function is not equally sensitive to variations in all coefficients; then an improved updating technique that benefits from these two key features follows. The idea is to update the most important component of the weight vector first to obtain the greatest reduction in the error. This choice results in a weight vector following the first update that is closest to the optimal weight vector compared to starting the update with the first component $w_1(n)$. This provides the best initialization to the subsequent stages. The second stage, which benefits from information in the first stage, will update the next important component of the filter weight vector. The process continues in this manner till the Nth stage is reached.

A simple and direct criterion to measure the importance of the filter weight is the magnitude of the gradient estimate in the direction of that weight: larger

gradient leads to larger reduction in the error. The gradient estimate in the direction of the ith weight is $-2e_0(n)x(n-i+1)$, where $1 \leq i \leq N$. Clearly, since all gradient components involve the quantity $-2e_0(n)$, then the weight associated with the maximum $|x(n-i+1)|$ has to be updated first. Note that in low SNR environments, the error $e(n)$ can be very noisy leading to a noisy gradient estimate that can provide false information. However, with the elimination of $e(n)$ from the criterion, the algorithm criterion will be immune to the effect of noise disturbances.

In terms of multiplications/additions, the selective algorithm has the same complexity overhead of the sequential algorithm. However, in the selective algorithm, the acquired sample $x(n)$ magnitude should be, at most, compared with the previous stored $N-1$ elements to sort out the coefficients to be updated at that iteration. The comparison operation can be of the same complexity as a multiplication operation. However, dedicated DSP chips for FIR adaptive filtering have been highly popular on the market and may provide a potential solution to the added complexity of the selective algorithm.

4.4 Simulation Examples

To verify the expected performance of the proposed algorithm, various examples of stationary and nonstationary system identification and channel equalization are used. The two updating strategies, namely the sequential order-updating (SQOU) and the selective order-updating (SLOU), are examined and compared with the

standard LMS algorithm. In all simulations presented in this section, except for example 3 of case 1, and case 2, a white Gaussian noise with variance 0.0001 is added to the desired signal $d(n)$. All results are obtained by averaging over 100 independent runs.

4.4.1 Case 1: stationary models

4.4.1.1 Example 1: system identification (white input)

The all-zero system to be modeled has 128 fixed coefficients. The input signal $x(n)$ is a zero-mean white Gaussian signal of time-varying power. The variance of $x(n)$ is changed from 1 to 2 and back to 1 at iterations 5000 and 6000, respectively. Such a case arises in teleconferencing rooms where the adaptive algorithm has to adapt to a sudden change in the input signal power due to a change of speakers [100]. The FIR adaptive filter used has an equal order $N = 128$. Fig. 4.1 shows the MSE obtained from using the LMS, the SQOU and the SLOU algorithms for $\mu = \mu_{LMS} = 0.006$. This value of μ was chosen near the stability bound of the LMS algorithm for an input signal of variance 1 to obtain the fastest achievable convergence speed.

Fig. 4.1(a) shows that both the LMS and the SQOU algorithms have equivalent fast initial convergence rate, however, the SQOU algorithm results in a less steady state MSE than the LMS. This result is verified by the theoretical steady state conclusions in previous sections, where we can see from Eqs.(4.40) and (4.41) that

if the the LMS and the SQOU algorithms are used with the same step size, i.e., $\mu = \mu_{LMS}$, then

$$M < M_{LMS} \tag{4.44}$$

Fig. 4.1(a) also shows that the LMS diverges and never comes back to stability after 5000 samples due to the increase in the input signal power from 1 to 2 (while the step size was not changed), whereas the SQOU algorithm remains stable, indicating its robustness to input signal power variations.

The divergence problem of the LMS in this example can be simply solved by reducing the value of the step size or normalizing the LMS. On the other hand, our objective in presenting this example was to show that the proposed algorithm can choose μ close to μ_{max} of the standard LMS algorithm to obtain the fastest convergence speed in a white input environment while yielding less misadjustment than the LMS algorithm, bounded convergence, and robustness against unexpected power variations or disturbances in the input sequence. This agrees with the approximate theoretical results in section 4.2.

Fig. 4.1(b) illustrates that little improvement in performance is obtained from using the SLOU over the SQOU algorithm in this case where the input is white.

4.4.1.2 Example 2: system identification (correlated input)

In this example, we consider the case of highly correlated input signal. This type of signal produces flattened elliptical contours in the MSE surface resulting in

slower convergence for gradient-based algorithms in general. The input signal is obtained by passing a zero-mean white Gaussian signal of unity variance through the autoregressive filter [1]

$$H(z) = \frac{1}{1 - 1.588z^{-1} + 0.81z^{-2}} \tag{4.45}$$

The system to be modeled is a 32-coefficient FIR filter. The LMS is used with $\mu_{LMS} = 0.0005$, which was chosen close to the point of instability (i.e., $\mu_{LMS} = \mu_{max}$), to obtain the fastest convergence rate. The SQOU and SLOU algorithms are used with $\mu = 0.003$ to obtain the same level of MSE of that of the LMS algorithm. Fig. 4.2 indicates that for correlated input signals, SQOU and SLOU outperform the LMS in convergence speed. Also, the SLOU algorithm is seen to be superior in its convergence rate compared to SQOU in this correlated input signal environment.

4.4.1.3 Example 3: channel equalization

We consider a channel equalization application and is based on the example in [88]. The channel is assumed to have the impulse response [46, 88]

$$h(k) = \frac{1}{2}\left(1 + \cos(\frac{2\pi(k-1)}{s})\right), \quad k = 0,\ 1,\ 2 \tag{4.46}$$

where s is used to control the eigenvalue spread of the autocorrelation matrix $\mathbf{R}$. The input to the channel is a binary random sequence $b(n) = \pm 1$ of equal

probability. The adaptive equalizer has 11 taps. The input to the equalizer is

$$x(n) = \sum_{k=0}^{2} h(k)b(n-k) + v(n) \tag{4.47}$$

where $v(n)$ is a zero-mean white Gaussian noise of 0.001 variance, independent of $b(n)$. The desired signal is $d(n) = b(n-7)$ [88]. The LMS, SQOU and SLOU algorithms are used with $\mu = \mu_{LMS} = 0.03$. Furthermore, we choose $s = 4$, resulting in a high eigenvalue spread of 716.11 (ill-conditioned autocorrelation matrix). In Fig. 4.3, we can see the bursty nature of the LMS algorithm as mentioned in [88]. On the other hand, both proposed algorithms, SQOU and SLOU, result in a smoother output without performance loss. Another algorithm that exhibits this smoothing property is the momentum LMS algorithm [88]. However, this is the only advantage this algorithm provides, with no improvement in convergence speed [88], over the standard LMS algorithm while requiring N multiplications overhead.

It is worth noting that for the case of white inputs, where the MSE contours are circular, increasing μ for the new algorithm above μ_{max} of the LMS algorithm will not yield any improvement in convergence rate over the LMS. This is a result of the fact that the LMS gradient vector always points towards the global minimum of the MSE surface in the case of white inputs and hence the advantages of the order iterative technique suggested by the proposed algorithm will not apply in this case. However, the proposed algorithm will still exhibit the best tracking capabilities of the LMS algorithm (particularly when $\mu = \mu_{max}$) while maintaining the immunity

against divergence for a given input power variations or any input disturbances, as seen in the mean convergence analysis. On the other hand, the effect of a larger step size is more pronounced for the case of correlated inputs, where the proposed algorithm provides better performance than does the conventional LMS.

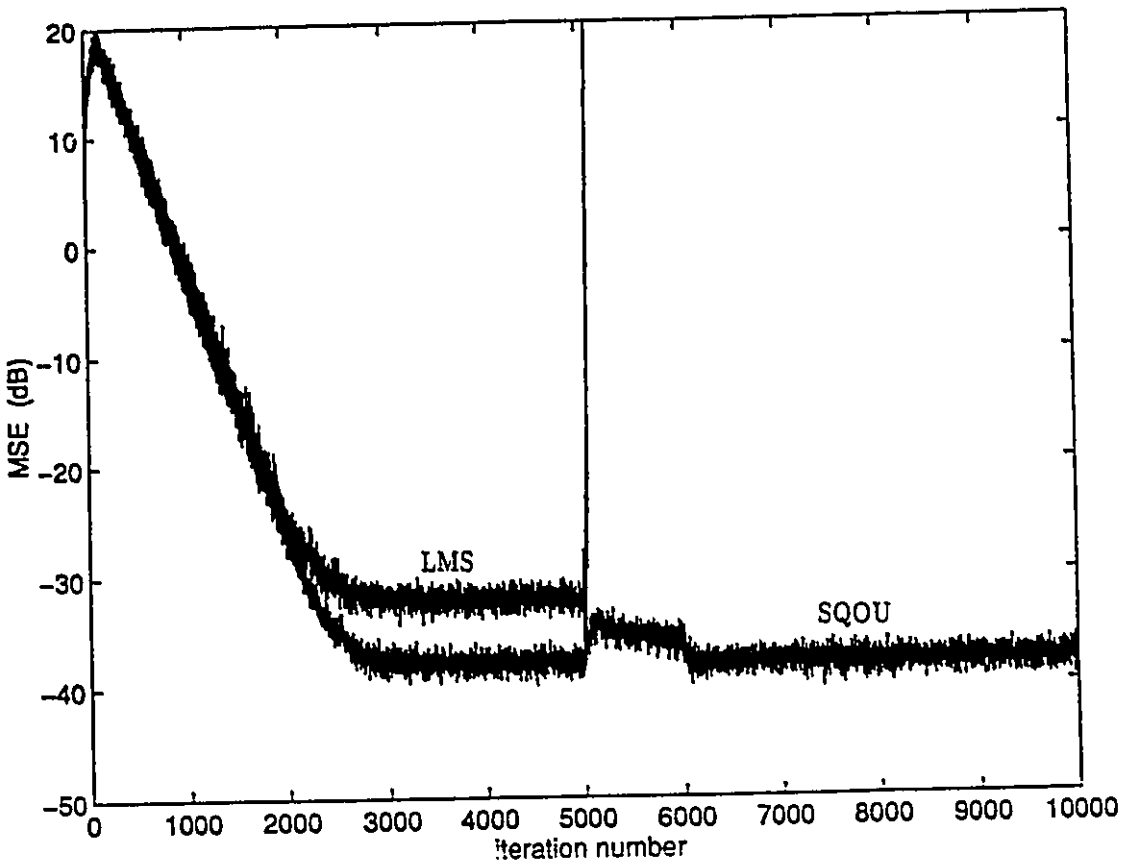

Fig. 4.1 (a) Comparison of MSE between LMS and SQOU algorithms for example 1 of case 1.

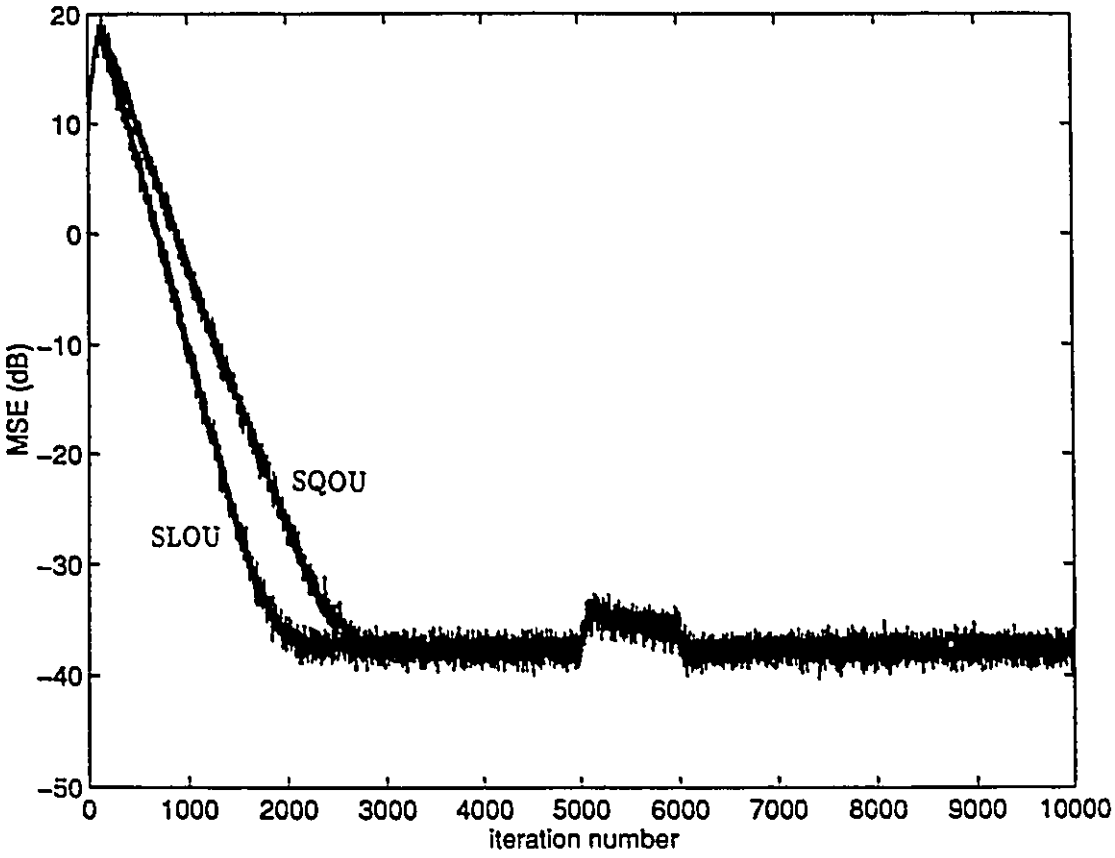

Fig. 4.1 (b) Comparison of MSE between SQOU and SLOU algorithms for example 1 of case 1.

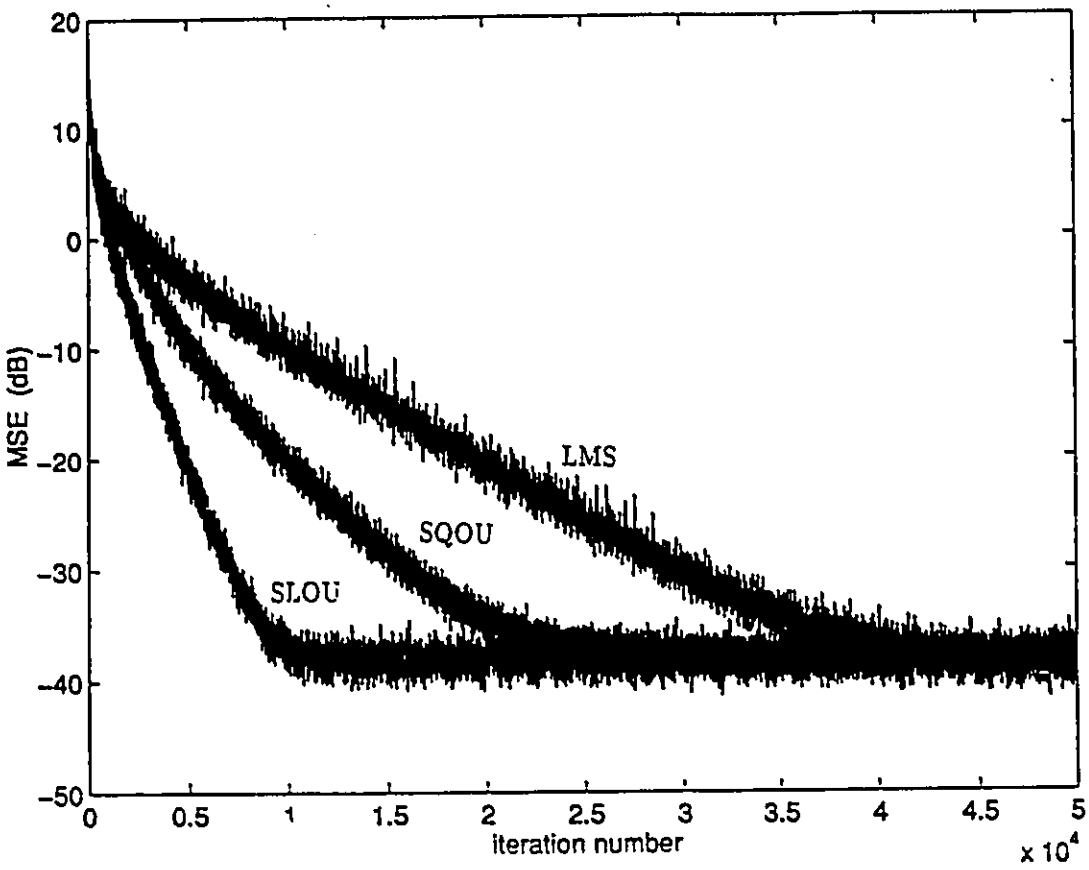

Fig. 4.2 Comparison of MSE between LMS, SQOU, and SLOU algorithms for example 2 of case 1.

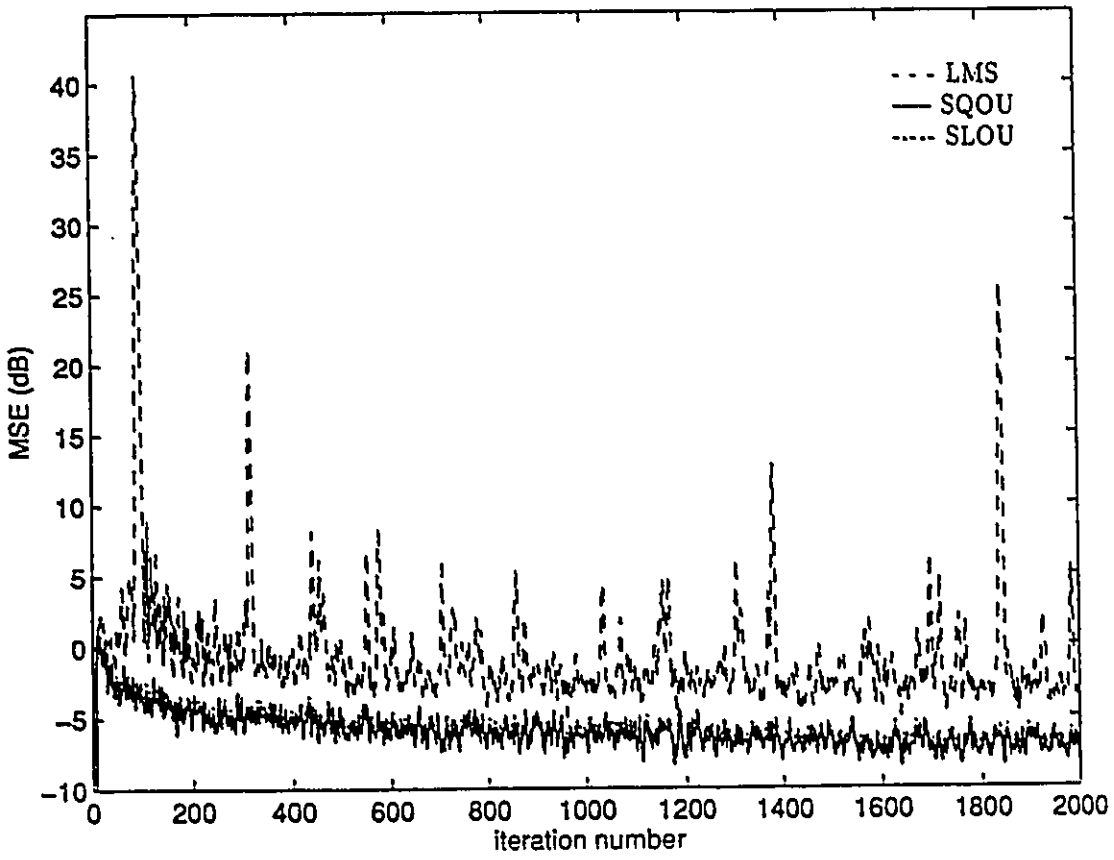

Fig. 4.3 Comparison of MSE between LMS, SQOU, and SLOU algorithms for example 3 of case 1.

4.4.2 Case 2: nonstationary model

The example used here is based on [111]. The purpose is to examine the ability of the proposed algorithm to operate in a nonstationary environment compared to the LMS algorithm. The LMS algorithm is known to have excellent tracking capabilities in nonstationary environments. The system to be identified is an FIR filter of length 15, where the time-varying optimal weight vector $\mathbf{W}^*(n)$ is generated from a stationary first order Markov process. Each component of $\mathbf{W}^*(n)$ is generated by passing an independent zero-mean white Gaussian sequence of variance of 0.05 through the IIR filter

$$H(z) = \frac{1}{1 - 0.987z^{-1}} \tag{4.48}$$

The FIR adaptive filter is of length $N = 15$. Both the adaptive filter and the unknown system are excited by a zero-mean independent Gaussian signal having a variance of 1. The desired signal $d(n)$ is corrupted with a zero-mean white Gaussian noise of unity variance, which is independent of the input signal. The adaptive algorithm should track the time-varying weight vector $\mathbf{W}^*(n)$.

Fig. 4.4(a) shows a comparison between the ensemble average of the first weight of the adaptive filter and weight number one of $\mathbf{W}^*(n)$ using the LMS. The LMS is used with $\mu_{LMS} = 0.03$, which was chosen near the stability bound. Figs. 4.4(b) and 4.4(c) show the same comparison for SQOU and SLOU with $\mu = 0.08$ (note

that the LMS diverges for this value of μ). Clearly, from Figs. 4.4(a) and 4.4(b), we note that the SQOU algorithm provides better tracking performance to the nonstationary optimal weight vector than the LMS. On the other hand, no performance gain is attained over the LMS by using the SLOU algorithm, which is obvious from comparing Fig. 4.4(a) and Fig. 4.4(c).

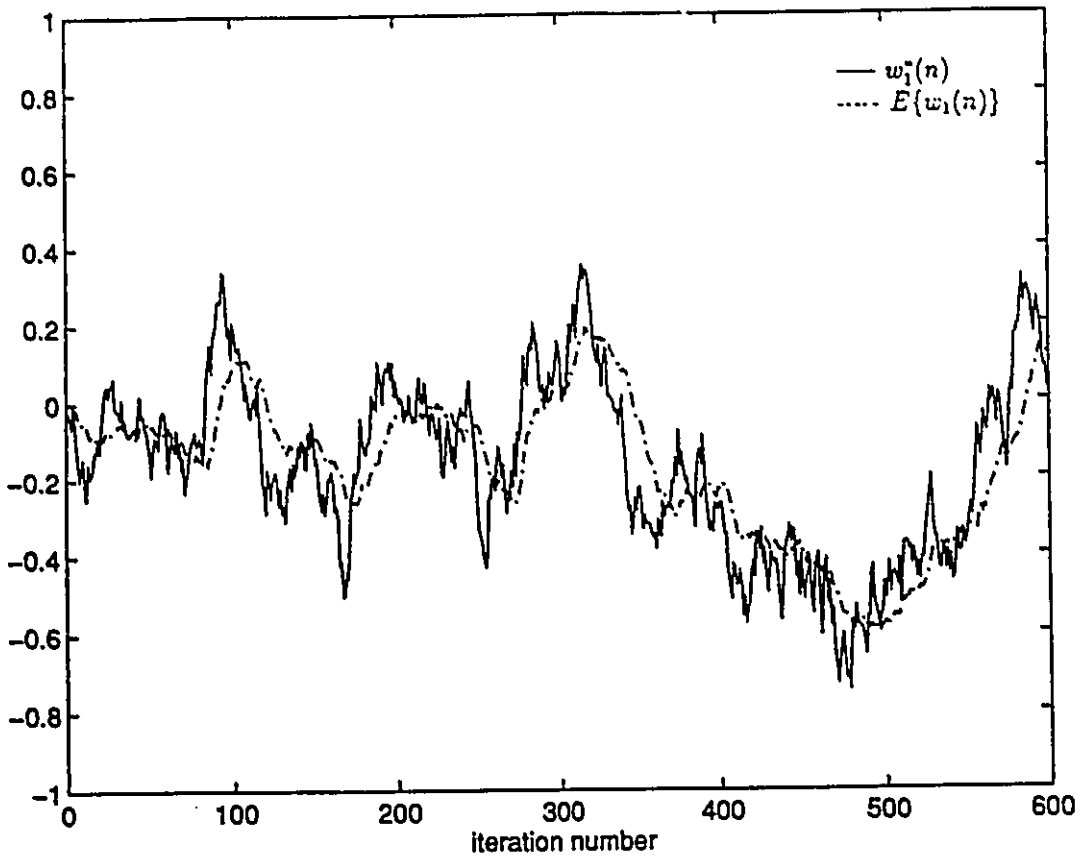

Fig. 4.4 (a) Comparison between $w_1^*(n)$ and the trajectory of $E\{w_1(n)\}$ using the LMS algorithm for case 2.

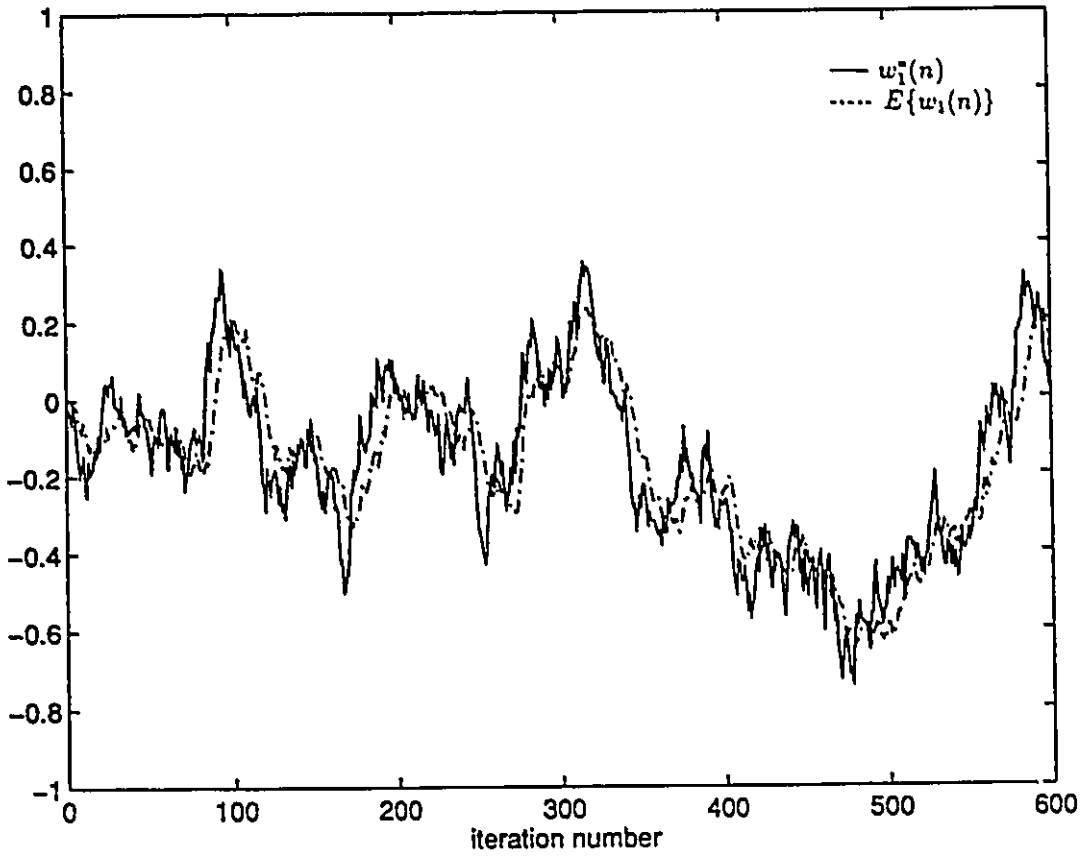

Fig. 4.4 (b) Comparison between $w_1^*(n)$ and the trajectory of $E\{w_1(n)\}$ using the SQOU algorithm for case 2.

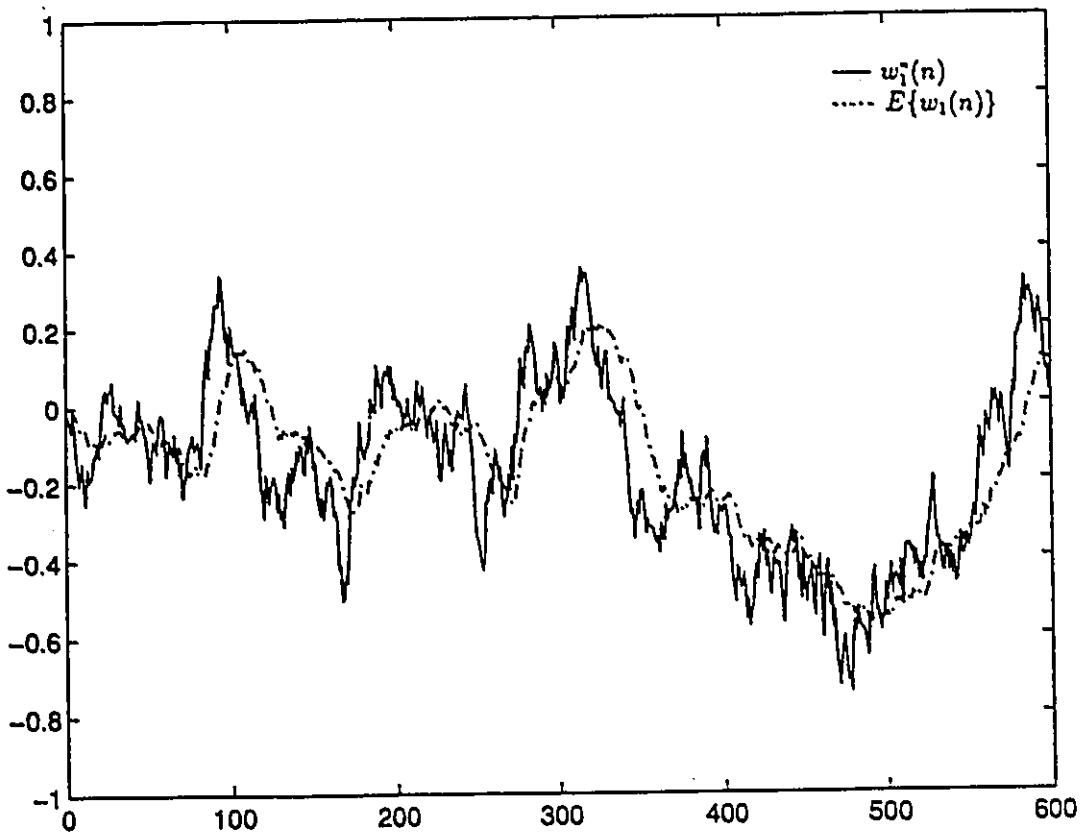

Fig. 4.4 (c) Comparison between $w_1^*(n)$ and the trajectory of $E\{w_1(n)\}$ using the SLOU algorithm for case 2.

4.5 Conclusion

This chapter presented a gradient-based adaptive algorithm that employs a new time and order updating technique to adjust filter coefficients resulting in a more efficient use of the available information. The algorithm is shown to have improved convergence speed for correlated input cases and to efficiently track nonstationary environments. We introduced steady state analysis under zero-mean white Gaussian inputs that led to guidelines to the choice of the adaptive step size. Two algorithms were proposed. One of them is the sequential order-updating (SQOU) algorithm and the other is the selective order-updating (SLOU) algorithm. The later gives more priority to coefficients contributing more to the error measure. Simulations indicated that the SQOU algorithm is a better choice for stationary white inputs and in nonstationary environments, however, the SLOU algorithm provides much better convergence speed when the input is highly correlated at expense of increased complexity.

Chapter 5

A Coupled LMS-LS Approach to Output-Error Adaptive IIR Filtering

FIR adaptive filters have traditionally been preferable to IIR ones because of the unimodality of their MSE surface and the stability of the adaptation (given proper choice of the adaptation parameters). However, practical physical systems have transfer functions of the pole-zero (ARMA) nature. The transversal filter is, in general, only a rough approximation of the actual system and can never estimate its impulse response exactly. Because of the presence of poles in the unknown systems, longer adaptive FIR filters are needed in many applications. This results in excessive complexity and severely reduces the convergence speed of all gradient-type adaptive algorithms.

For these reasons, and because of the improved performance and reduced complexity they can offer, IIR adaptive filters have been the subject of much research.

IIR adaptive filters have been applied in different applications, such as echo cancellation [2, 3], acoustic echo cancellation [42, 44, 75], speech compression in ADPCM and linear prediction coding [82, 103], adaptive array processing [37], etc.

The IIR filter adaptation can be based on the output error or the equation error [49, 97]. Algorithms based on the equation-error formulation use the equation error in the coefficients update as shown in Fig. 5.1. $x(n)$ is the input signal, $d(n)$ and $\hat{d}(n)$ are the desired and estimated output signals, respectively. This approach uses the well-understood transversal structure with the attractive quadratic MSE function but leads to a biased estimate of the adaptive filter coefficients; the bias depending on the power of the disturbance noise.

On the other hand, the output-error based adaptive IIR filter updates the filter coefficients based on the actual output error, as shown in Fig. 5.2. This approach results in unbiased coefficient estimates [49, 97]. Even though the error surface in this approach may have local minima, it was noted in [27] that as long as the adaptive filter of sufficient order , the IIR filter will outperform the FIR one. Moreover, it was recently shown that the stability of the filter can be efficiently monitored [97].

Gradient search algorithms, particularly the LMS algorithm, are the most commonly used update algorithms in the output-error IIR filter method [97]. However, the nonlinearity of the resultant MSE function complicates the convergence of the

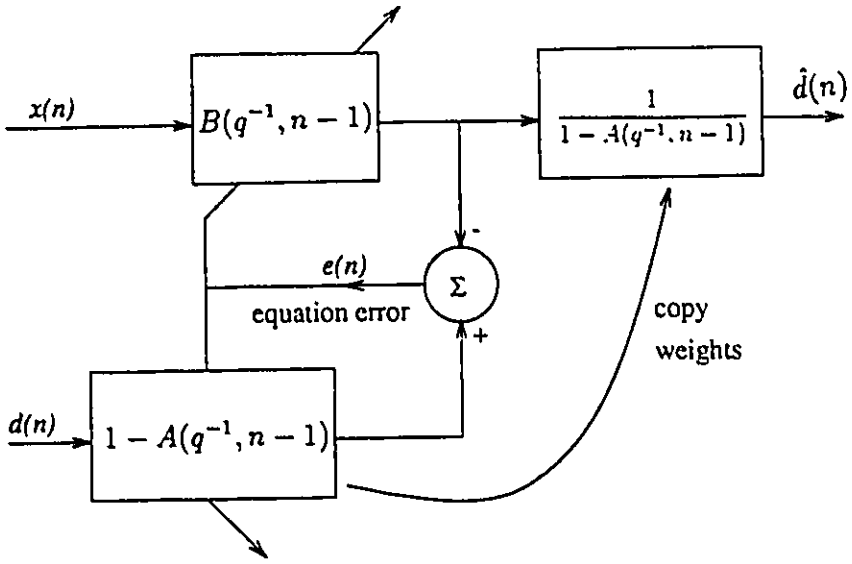

Fig. 5.1 Equation-error formulation.

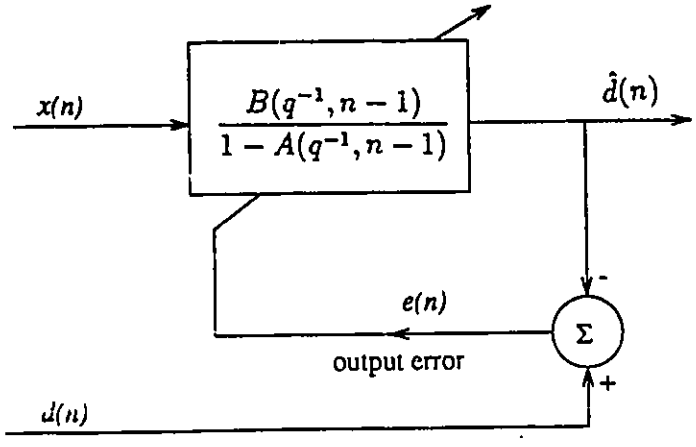

Fig. 5.2 Output-error formulation.

LMS leading the algorithm to converge very slowly and possibly to a local minimum. This fact is one of the major limitations of the output-error IIR approach restricting its application [27, 97].

This chapter studies the problem of slow convergence of the output-error adaptive IIR gradient algorithms. The goal is to improve the convergence speed while maintaining the IIR basic advantage of lower complexity. Our proposed solution mainly relies on the results in [101] studying the nature of the error surface of the adaptive IIR filters. On the one hand, it was shown that the error function is quadratic with respect to the numerator coefficients of the adaptive IIR filter and therefore has a bowl-shaped MSE surface as a function of these parameters. On the other hand, with respect to the denominator coefficients, the performance function is not quadratic and is nonlinear. These results clearly indicate that the slow convergence problem of the gradient-type IIR adaptive algorithms is essentially caused by the inefficient search for the optimum denominator coefficients which in turn delays the convergence of the numerator coefficients resulting in very slow convergence overall. These results also suggest that if the denominator coefficients' convergence rate is improved, then the simple LMS algorithm will have no problem in searching the numerator coefficients since they define a bowl-shaped surface with a unique minimum. The overall convergence speed of the adaptive

IIR algorithm will thus be significantly improved.

Based on the above discussion, we propose an adaptive algorithm for the output-error IIR filter in which the numerator and denominator coefficients are updated using different techniques: the LMS algorithm is used to update the numerator coefficients of the adaptive IIR filter while a least squares-type method is used to update the denominator coefficients. Two mathematically equivalent algorithms, but with different complexities, are presented. The complexity of the first algorithm is proportional to $O(M^2) + O(N)$, whereas the second one requires about $O(M) + O(N)$ multiplications, where M and N are the orders of the denominator and numerator, respectively. Though it appears that the $O(M^2)$ or $O(M)$ added multiplications might be expensive, in most practical applications only few poles are needed to account for long tails in the actual system impulse response. Thus, the overall increase in complexity is marginal compared with the case when the LMS is used to update both numerator and denominator coefficients.Also, it is considerably less than the case when the LS method is utilized to update both the numerator and denominator coefficients.

It should be noted that the least squares method was used in [62] to adjust the coefficients of the output-error IIR adaptive filter to speed up their convergence. This led to the derivation of the recursive maximum likelihood (RML) algorithm [62]. Also, the recursive LS technique was applied to the identifications of pole/zero

models in [2, 3, 5]. Particularly, the fast Kalman algorithm was used to cancel echo generated from an ARMA modeled echo path [2, 3]. In [5], fast transversal RLS filters were formulated as an extension of the work in [60]. The algorithms in [2, 3, 5] were formulated for the equation-error IIR filters. These algorithms, however, are computationally demanding since they require $O((N+M)^2)$ multiplications.

5.1 Coupled LMS-LS IIR output-error adaptive algorithm

The unknown system to be identified here is assumed to be a general ARMA system. The output error is used to update the coefficients as shown in Fig. 5.2. $\hat{d}(n)$ is the output of the adaptive filter given by

$$\begin{aligned}\hat{d}(n) &= \sum_{j=1}^{M} a_j(n-1)\hat{d}(n-j) + \sum_{j=0}^{N-1} b_j(n-1)x(n-j) \\ &= A(q^{-1}, n-1)\hat{d}(n) + B(q^{-1}, n-1)x(n) \\ &= \mathbf{A}^{\mathbf{T}}(n-1)\hat{\mathbf{D}}(n) + \mathbf{B}^{\mathbf{T}}(n-1)\mathbf{X}(n) \end{aligned} \tag{5.1}$$

where

$$\begin{aligned} A(q^{-1}, n-1) &= \sum_{j=1}^{M} a_j(n-1)q^{-j} \\ B(q^{-1}, n-1) &= \sum_{j=0}^{N-1} b_j(n-1)q^{-j} \end{aligned} \tag{5.2}$$

where q^{-j} is the delay operator, i.e., $q^{-j}x(n) = x(n-j)$, $a_j(n-1)$ and $b_j(n-1)$ are the coefficients of the adaptive IIR filter computed at time (n-1), and

$$\begin{aligned}
\mathbf{A}(n-1) &= [a_1(n-1), a_2(n-1), \ldots, a_M(n-1)]^T \\
\mathbf{B}(n-1) &= [b_0(n-1), b_2(n-1), \ldots, b_{N-1}(n-1)]^T \\
\hat{\mathbf{D}}(n) &= [\hat{d}(n-1), \hat{d}(n-2), \ldots, \hat{d}(n-M)]^T \\
\mathbf{X}(n) &= [x(n), x(n-1), \ldots, x(n-N+1)]^T
\end{aligned}$$

Note that in Eq.(5.1) the coefficient $a_j(n-1)$ is multiplied by the delayed output sample $\hat{d}(n-j)$. Since $\hat{d}(n-j)$ is a function of $A(q^{-1}, n-j-1)$ and $B(q^{-1}, n-j-1)$, then $\hat{d}(n)$ in Eq.(5.1) is a nonlinear function of the adaptive IIR filter coefficients. Our objective is to adjust the numerator coefficients of the adaptive IIR filter such that the mean square error $E\{e^2(n)\}$ is minimized, where

$$e(n) = d(n) - \hat{d}(n) \tag{5.3}$$

is the output error as shown in Fig. 5.2. It has been shown in [101] that the numerator coefficients produce a bowl-shaped error surface $E\{e^2(n)\}$ for given denominator coefficients. Thus, the standard LMS algorithm can be easily utilized to minimize $E\{e^2(n)\}$ relative to the elements of $\mathbf{B}(n-1)$ in a straightforward manner, i.e.,

$$\begin{aligned}
\mathbf{B}(n) &= \mathbf{B}(n-1) - \mu \frac{\partial e^2 n)}{\partial \mathbf{B}(n-1)} \\
&= \mathbf{B}(n-1) + 2\mu e(n) \frac{\partial \hat{d}(n)}{\partial \mathbf{B}(n-1)}
\end{aligned} \tag{5.4}$$

From Eq.(5.1), we have

$$\frac{\partial \hat{d}(n)}{\partial b_i(n-1)} = x(n-i) + \sum_{j=1}^{M} a_j(n-1)\frac{\partial \hat{d}(n-j)}{\partial b_i(n-1)} \quad , i = 0, \ ,1 \ , \ldots , N-1 \qquad (5.5)$$

Note that the relationship between the current coefficients $b_i(n-1)$ and the past quantities $\hat{d}(n-j)$ is not straightforward [49, 97]. This makes the evaluation of $\frac{\partial \hat{d}(n-j)}{\partial b_i(n-1)}$ in Eq.(5.5) not possible. However, the step size μ is usually small in the IIR filtering application such that, over a window of M samples the approximation $b_i(n-1) \approx b_i(n-2) \approx \ldots \approx b_i(n-M)$ is valid [97]. Consequently,

$$\frac{\partial \hat{d}(n-j)}{\partial b_i(n-1)} \approx \frac{\partial \hat{d}(n-j)}{\partial b_i(n-j)}, \ j = 1, \ 2, \ \ldots , M \qquad (5.6)$$

and Eq.(5.5) becomes

$$\frac{\partial \hat{d}(n)}{\partial \mathbf{B}(n-1)} \approx \mathbf{X}(n) + A(q^{-1}, n-1)\frac{\partial \hat{d}(n)}{\partial \mathbf{B}(n-1)} \qquad (5.7)$$

Substituting Eq.(5.7) in (5.4) gives the update recursion of the numerator coefficients,

$$\mathbf{B}(n) = \mathbf{B}(n-1) + 2\mu \mathbf{X}_f(n)e(n) \qquad (5.8)$$

where $\mathbf{X}_f(n) = \frac{1}{(1-A(q^{-1},n-1))}\mathbf{X}(n)$.

The high nonlinearity of the performance function $E\{e^2(n)\}$ as a function of the coefficients of the autoregressive (AR) part of the adaptive IIR filter severely reduces the convergence rate of the iterative gradient-based algorithms when used to adapt these coefficients. This problem becomes worse when the input signal

is a correlated one. In such input environments, the recursive LS technique is known to be a powerful alternative because of its rapid convergence as well as its insensitivity to the signal statistics. Accordingly, our objective is to calculate, at each time instant n, the denominator coefficient vector $\mathbf{A}(n)$ which minimizes, in the *least squares* sense, the output error function *given the numerator coefficients.* The objective function is thus given as

$$\epsilon_{LS}(n) = \sum_{i=1}^{n} \lambda^{n-i}(d(i) - \hat{d}(i/n)\ )^2 \tag{5.9}$$

where

$$\hat{d}(i/n) = \mathbf{A}^{\mathbf{T}}(n)\hat{\mathbf{D}}(i/n) + \mathbf{B}^{\mathbf{T}}(i-1)\mathbf{X}(i) \tag{5.10}$$

and λ is an exponential weighting parameter, $0 < \lambda \leq 1$, $\hat{d}(i/n)$ is the output of the adaptive filter at time i using the denominator coefficient vector $\mathbf{A}(n)$. Utilizing Eq.(5.10), we get

$$\frac{\partial \hat{d}(i/n)}{\partial a_l(n)} = \hat{d}((i-l)/n) + \sum_{j=1}^{M} a_j(n) \frac{\partial \hat{d}((i-j)/n)}{\partial a_l(n)} \ , \ l = 1, \ ,2 \ , \ldots , M \tag{5.11}$$

Observe that $\hat{d}((i-j)/n)$ is evaluated using $\mathbf{A}(n)$, then it is straightforward that

$$\frac{\partial \hat{d}(i/n)}{\partial \mathbf{A}(n)} = \frac{1}{(1 - A(q^{-1}, n))} \hat{\mathbf{D}}(i/n) \tag{5.12}$$

Using Eq.(5.12), it is easy to show that minimizing $\epsilon_{LS}(n)$ in Eq.(5.9) with respect to $\mathbf{A}(n)$ results in the following equation

$$\mathbf{R_f}(n)\mathbf{A}(n) = \mathbf{P_f}(n) \tag{5.13}$$

where

$$\begin{aligned}\mathbf{R_f}(n) &= \sum_{i=1}^{n} \lambda^{n-i} \hat{\mathbf{D}}_\mathbf{f}(i/n)\hat{\mathbf{D}}^\mathbf{T}(i/n) \\ \mathbf{P_f}(n) &= \sum_{i=1}^{n} \lambda^{n-i} \tilde{d}(i)\hat{\mathbf{D}}_\mathbf{f}(i/n)\end{aligned} \tag{5.14}$$

and $\tilde{d}(i)$, $\hat{\mathbf{D}}_\mathbf{f}(i/n)$ are defined as

$$\begin{aligned}\tilde{d}(i) &= d(i) - \mathbf{B}^\mathbf{T}(i-1)\mathbf{X}(i) \\ \hat{\mathbf{D}}_\mathbf{f}(i/n) &= \frac{1}{(1 - A(q^{-1}, n))}\hat{\mathbf{D}}(i/n)\end{aligned}$$

Note that the dependency of $\mathbf{R_f}(n)$ and $\mathbf{P_f}(n)$ on $A(q^{-1}, n)$ makes the recursive computation of $\mathbf{A}(n)$ in Eq.(5.13) generally impossible. In [62, p.26-29], this problem was bypassed by minimizing an expression that is a Talyer expansion of the LS objective function $\epsilon_{LS}(n)$ in Eq.(5.9). The approximations used to reach the solution were only true after convergence. In this work, we follow a different path by solving for $\mathbf{A}(n)$ in Eq.(5.13) using some approximations. Using the same earlier assumption that coefficients change slowly, then $a_i(n) \approx a_i(n-1) \approx \ldots a_i(n-M) \approx a_i(n-M-1)$, and the following essential approximations hold

$$\hat{\mathbf{D}}(i/n) \approx \hat{\mathbf{D}}(i/n-1) \tag{5.15}$$

and

$$\begin{aligned}\hat{\mathbf{D}}_\mathbf{f}(i/n) &\approx \hat{\mathbf{D}}_\mathbf{f}(i/n-1) \\ &= \frac{1}{(1 - A(q^{-1}, n-1))}\hat{\mathbf{D}}(i/n-1)\end{aligned} \tag{5.16}$$

The above assumptions lead to the following approximation

$$\begin{aligned}\hat{\mathbf{D}}(n/n-1) &\approx [\hat{d}(n-1/n-2), \hat{d}(n-2/n-3), \ldots, \hat{d}(n-M/n-M-1) \\ &= \hat{\mathbf{D}}(n) \end{aligned} \tag{5.17}$$

Accordingly

$$\begin{aligned}\hat{\mathbf{D}}_{\mathbf{f}}(n/n-1) &\approx \frac{1}{(1-A(q^{-1},n-1))}\hat{\mathbf{D}}(n) \\ &= \hat{\mathbf{D}}_{\mathbf{f}}(n) \end{aligned} \tag{5.18}$$

While those approximations may not be valid initially because of the fast convergence of the LS method, such approximations are generally valid after convergence[1] when $A(q^{-1},n) \approx A^*(q^{-1})$ and $B(q^{-1},n) \approx B^*(q^{-1})$. They are quite useful since they allow the recursive calculation of the vector $\mathbf{A}(n)$ in Eq.(5.13) assuming the knowledge of $\mathbf{A}(n-1)$, which is the optimal vector coefficient at time $n-1$. Based on these assumptions and using the matrix inversion lemma, $\mathbf{A}(n)$ in Eq.(5.13) can be calculated recursively using the following sequence of computations

$$\mathbf{k}(n) = \frac{\mathbf{Q}(n-1)\hat{\mathbf{D}}_{\mathbf{f}}(n)}{\lambda + \hat{\mathbf{D}}^{\mathbf{T}}(n)\mathbf{Q}(n-1)\hat{\mathbf{D}}_{\mathbf{f}}(n)} \tag{5.19}$$

$$\mathbf{A}(n) = \mathbf{A}(n-1) + \mathbf{k}(n)e(n) \tag{5.20}$$

$$\mathbf{Q}(n) = \lambda^{-1}\left[\mathbf{Q}(n-1) - \mathbf{k}(n)\hat{\mathbf{D}}^{\mathbf{T}}(n)\mathbf{Q}(n-1)\right] \; ; \; \mathbf{Q}(0) = c\mathbf{I}, \; c \gg 1 \tag{5.21}$$

[1]These assumptions were used in [62, p.26-29] in the derivation of the recursive maximum likelihood (RML) algorithm, and in [113, p.158-159] in the derivation of the IIR sequential regression (SER) algorithm.

Note that calculation of $\mathbf{X_f}(n)$ and $\hat{\mathbf{D}}_\mathbf{f}(n)$ is computationally expensive since updating each component of $\mathbf{X_f}(n)$ and $\hat{\mathbf{D}}_\mathbf{f}(n)$ involves a filter operation using $\frac{1}{(1-A(q^{-1},n-1))}$. A more usable approximation is to obtain the components of $\mathbf{X_f}(n)$ and $\hat{\mathbf{D}}_\mathbf{f}(n)$ from the following equations [49, 97]

$$x_f(n-j) = \frac{1}{(1-A(q^{-1},n-1-j))}x(n-j),\ j=0,\ldots,N-1 \quad (5.22)$$

$$\hat{d}_f(n-j) = \frac{1}{(1-A(q^{-1},n-1-j))}\hat{d}(n-j),\ j=1,\ldots,M \quad (5.23)$$

The proposed output-error IIR adaptive algorithm can be fully described by Eqs. (5.22), (5.23), (5.3), (5.8), (5.19), (5.20) and (5.21) and is summarized in Table 4.1. Calculations of Eq.(5.19), (5.20) and (5.21) require $O(M^2)$ multiplications mainly arising from updating the $M \times M$ matrix $\mathbf{Q}(n)$.

Table 4.1

Computational organization of the coupled LMS-LS IIR output-error adaptive algorithm

---

Initialization :

$$\mathbf{Q}(0) = c\mathbf{I},\ c \gg 1$$

for $n=1$ **to** n **final do** :

$$x_f(n-j) = \frac{1}{(1-A(q^{-1},n-1-j))}x(n-j), j=0,...,N-1 \quad (5.24)$$

$$\hat{d}_f(n-j) = \frac{1}{(1-A(q^{-1},n-1-j))}\hat{d}(n-j), j=1,...,M \quad (5.25)$$

$$e(n) = d(n) - \mathbf{A}^{\mathrm{T}}(n-1)\hat{\mathbf{D}}(n) + \mathbf{B}^{\mathrm{T}}(n-1)\mathbf{X}(n) \quad (5.26)$$

$$\mathrm{k}(n) = \frac{\mathbf{Q}(n-1)\hat{\mathbf{D}}_{\mathbf{f}}(n)}{\lambda + \hat{\mathbf{D}}^{\mathrm{T}}(n)\mathbf{Q}(n-1)\hat{\mathbf{D}}_{\mathbf{f}}(n)} \quad (5.27)$$

$$\mathbf{B}(n) = \mathbf{B}(n-1) + 2\mu\mathbf{X}_{\mathbf{f}}(n)e(n) \quad (5.28)$$

$$\mathbf{A}(n) = \mathbf{A}(n-1) + \mathrm{k}(n)e(n) \quad (5.29)$$

$$\mathbf{Q}(n) = \lambda^{-1}\left[\mathbf{Q}(n-1) - \mathrm{k}(n)\hat{\mathbf{D}}^{\mathrm{T}}(n)\mathbf{Q}(n-1)\right] \quad (5.30)$$

5.2 Fast coupled LMS-LS IIR output-error adaptive algorithm

It should be noted that by taking advantage of the shifting property[2] of $\hat{\mathbf{D}}(n)$ and $\hat{\mathbf{D}}_{\mathbf{f}}(n)$ in Eqs.(5.19) and (5.21), complexity of the calculation of the gain vector $\mathrm{k}(n)$ can be reduced to about $8M$ multiplications. This reduction is primarily based on results in [60, lemma 2]. A full derivation of the algorithm is given in [60] for $\lambda = 1$. However, it is not difficult to rederive the algorithm for a general λ $(0 < \lambda \leq 1)$. In the following, we outline the sequence of computations for fast calculation of the gain vector $\mathrm{k}(n)$ with exponential weighting factor λ:

---

[2] $\hat{\mathbf{D}}(n)$ can be obtained from $\hat{\mathbf{D}}(n-1)$ by shifting its elements and adding the new element $\hat{d}(n)$.

Initialization :

$$\mathbf{k}(0) = \mathbf{f}(0) = \mathbf{P}(0) = \mathbf{G}(0)$$

$$\epsilon_f(0) = \delta \quad (\ \delta \ \text{is a small positive number}\ )$$

for $n = 1$ **to** n **final do** :

$$\epsilon_0(n) = \hat{d}(n-1) + \mathbf{f}^{\mathbf{T}}(n-1)\hat{\mathbf{D}}(n-1) \tag{5.31}$$

$$\mathbf{f}(n) = \mathbf{f}(n-1) - \mathbf{k}(n-1)\epsilon_0(n) \tag{5.32}$$

$$\mathbf{P}(n) = \lambda\mathbf{P}(n-1) + \hat{\mathbf{D}}(n-1)\hat{d}_f(n-1) \tag{5.33}$$

$$\epsilon(n) = \hat{d}_f(n-1) - \mathbf{P}^{\mathbf{T}}(n)\mathbf{k}(n-1) \tag{5.34}$$

$$\epsilon_f(n) = \lambda\epsilon_f(n-1) + \epsilon(n)\epsilon_0(n) \tag{5.35}$$

$$\begin{pmatrix} \mathbf{C}(n) \\ \\ m(n) \end{pmatrix} = \begin{pmatrix} \epsilon_f^{-1}(n)\epsilon(n) \\ \\ \mathbf{k}(n-1) + \mathbf{f}(n)\epsilon_f^{-1}(n)\epsilon(n) \end{pmatrix} \tag{5.36}$$

$$\eta_0(n) = \hat{d}(n-M-1) + \mathbf{G}^{\mathbf{T}}(n-1)\hat{\mathbf{D}}(n) \tag{5.37}$$

$$\mathbf{G}(n) = [\mathbf{G}(n-1) - \mathbf{C}(n)\eta_0(n)]\,[1 - m(n)\eta_0(n)]^{-1} \tag{5.38}$$

$$\mathbf{k}(n) = \mathbf{C}(n) - \mathbf{G}(n)m(n) \tag{5.39}$$

Theoretically speaking, the above fast algorithm for evaluating $\mathbf{k}(n)$ is mathematically equivalent to that in Eqs.(5.19) and (5.21). However, the fast algorithm exploits the shifting property to replace the matrix computation in Eq.(5.19) and (5.21) with vector operations requiring $O(M)$ multiplications. On the other hand, the effect of finite-precision errors becomes apparent in the $O(M)$ case. This is due to the large number of internal variables involved in the algorithm recursions, and especially in low signal-to-noise environments and for large order filters. This is a

typical problem for all fast algorithms [18, 20]. Several techniques and rescue methods have been proposed to improve finite-precision performance at the expense of the optimality of the solution and some delayed convergence [18, 20]. In our case here, the assumed low order of the denominator reduces the effect of numerical errors. However, actual implementation of this fast algorithm is essential to verify its true numerical performance. Given that the complexity reduction form $O(M^2)$ to $O(M)$ is not that dramatic for small M, the usage of the $O(M^2)$ algorithm will be preferable from a numerical stability point of view when considering numerical complications for cases of few poles.

5.3 Discussion

Simulation-based studies of the LS error surfaces [104] and the mean square error surfaces [101] of the output-error formulation show that steepest descent directions can vary drastically over the error surface, particularly near stability boundaries. Moreover, gradient vectors can point away from the global minimum for non-quadratic functions [101, 104]. Therefore, adaptive algorithms based on steepest descent methods are forced to use very small step sizes, and sometimes tend to converge to local minima.

On the other hand, the proposed algorithm does not conduct a gradient direction search for the optimal poles and is believed to help the algorithm escape local minima in most of the examples considered in this work. Nevertheless, this

is not conclusive and cannot be generalized since our observations are simulation based, as is the case for other existing output-error IIR algorithms. Another advantage follows from decoupling of the pole/zero update in the proposed scheme, the $\mathbf{A}(n)$ update is not a function of step size parameter controlling the stability of the recursion. This results in an increase in the stability boundary of the LMS step size μ since the LMS algorithm now determines μ based on the numerator update independent of poles and their stability restrictions. This permits larger step size value to be used thus speeding up the convergence of the overall adaptive algorithm. Rapid convergence of the LS method in the denominator update, *irrespective of the initial value of the adaptive filter weights*, is another significant improvement of the proposed algorithm. When the LMS is used for both numerator and denominator update, convergence rate is highly dependent on the proximity of the initial value of the adaptive filter weights to the local or the global minima.

5.4 Simulation Examples

A number of simulation examples, based on the system identification model, will be discussed here. Our main objective is to illustrate the significant improvement of the convergence speed of the proposed LMS-LS algorithm compared to the case when LMS algorithm is used to update both the numerator and denominator coefficients. Moreover, we will present some typical examples to examine the performance of the proposed algorithm in cases of multimodal error surfaces. These

examples demonstrate the general tendency of the LMS-LS algorithm to converge to the global minimum of the error surfaces in those examples. However, it should be stressed that this is still simulation observations and cannot be claimed as an inherent property of the proposed algorithm. The $O(M^2)$ algorithm is used in all simulation examples, except the last one. The fast algorithm was found to produce identical simulation results (double precision floating point is used). For all examples, except the last one, the desired signal $d(n)$ has an additional zero-mean white Gaussian noise of 0.0001 variance. Results are obtained by averaging over 100 ensemble members.

5.4.1 Verification of improvement in convergence speed

In this part, we show the improvement in the convergence speed provided by the proposed algorithm over the LMS algorithm for white and correlated input signals. The system to be identified has the transfer function [51]

$$H(z) = \frac{0.05 - 0.4z^{-1}}{1 - 1.1314z^{-1} + 0.25z^{-2}} \tag{5.40}$$

and that of the adaptive filter is

$$H(z,n) = \frac{b_0(n) + b_1(n)z^{-1}}{1 - a_1(n)z^{-1} - a_2(n)z^{-2}} \tag{5.41}$$

All adaptive filter coefficients are initialized to zero.

5.4.1.1 Example 1: white input

In the first example, the system and the adaptive filter are excited by a zero-mean white Gaussian signal of unity variance. The LMS and the LMS-LS algorithms are compared for $\mu_{LMS} = \mu_{LMS-LS} = 0.001$. The LMS-LS algorithm is used with an exponential weighting factor $\lambda = 0.99$. Fig. 5.3 depicts the mean behavior of the denominator and numerator coefficients. Fig. 5.4 plots the mean square error (MSE). It is clear that though the numerator coefficients of both algorithms are updated using the same LMS method (as well as the same step size value), the employment of the LS approach to search the nonlinear error surface for the optimum poles has significantly increased the convergence speed of the denominator coefficients, Fig. 5.3(a), resulting in the overall adaptive filter converging faster compared to the case when the LMS approach is used to update both the numerator and denominator coefficients.

Since for the proposed coupled LMS-LS algorithm, the step size used by the LMS is no longer restricted to ensure pole stability , we can now increase the μ_{LMS-LS} beyond the limit of μ_{LMS} without affecting the stability of the algorithm. Figs. 5.5, 5.6 show the performance of the LMS-LS proposed algorithm for a much larger $\mu_{LMS-LS} = 0.005$. We can see further significant increase in the speed of convergence. However, this large step size would not be tolerated in the basic LMS algorithm since it results in unstable poles that keep moving outside the stability

region even after being reflected inside. Figures 5.5, 5.6 show the results for the basic LMS with $\mu_{LMS} = 0.001$.

5.4.1.2 Example 2: coloured input

In this example, the input signal is a correlated one generated by passing a zero-mean white Gaussian signal of unity variance through the filter

$$H(z) = \frac{1}{1 - 0.8z^{-1}} \tag{5.42}$$

In Figs. 5.7, 5.8, the LMS and the LMS-LS algorithms are used with the same step size $\mu_{LMS} = \mu_{LMS-LS} = 0.0003$, and with $\lambda = 0.99$ for the LMS-LS algorithm. Figs. 5.7 and 5.8 show 60,000 iteration of the LMS. Convergence to the global minimum is achieved after 110,000 while the LMS-LS converged in about 20,000 iterations. The example is repeated for the LMS-LS algorithm with a larger step size $\mu_{LMS-LS} = 0.0006$, and $\lambda = .99$. Figs. 5.9, 5.10 show the result along with the LMS for $\mu_{LMS} = 0.0003$. It is to be noted that $\mu_{LMS} = 0.0006$ results in instability (same as before) as was observed in example 1.

In the above examples, the improved speed of convergence of the LMS-LS algorithm in both coloured and white input environment was demonstrated.

It should be emphasized that the increased convergence speed is due to two contributing factors:

1. Use of LS to update poles. This is inherently faster than the LMS and is evident from examples using same step size for both approaches.

2. Allowing for a larger step size in the LMS update of the numerator coefficients since that step size no longer affects pole stability.

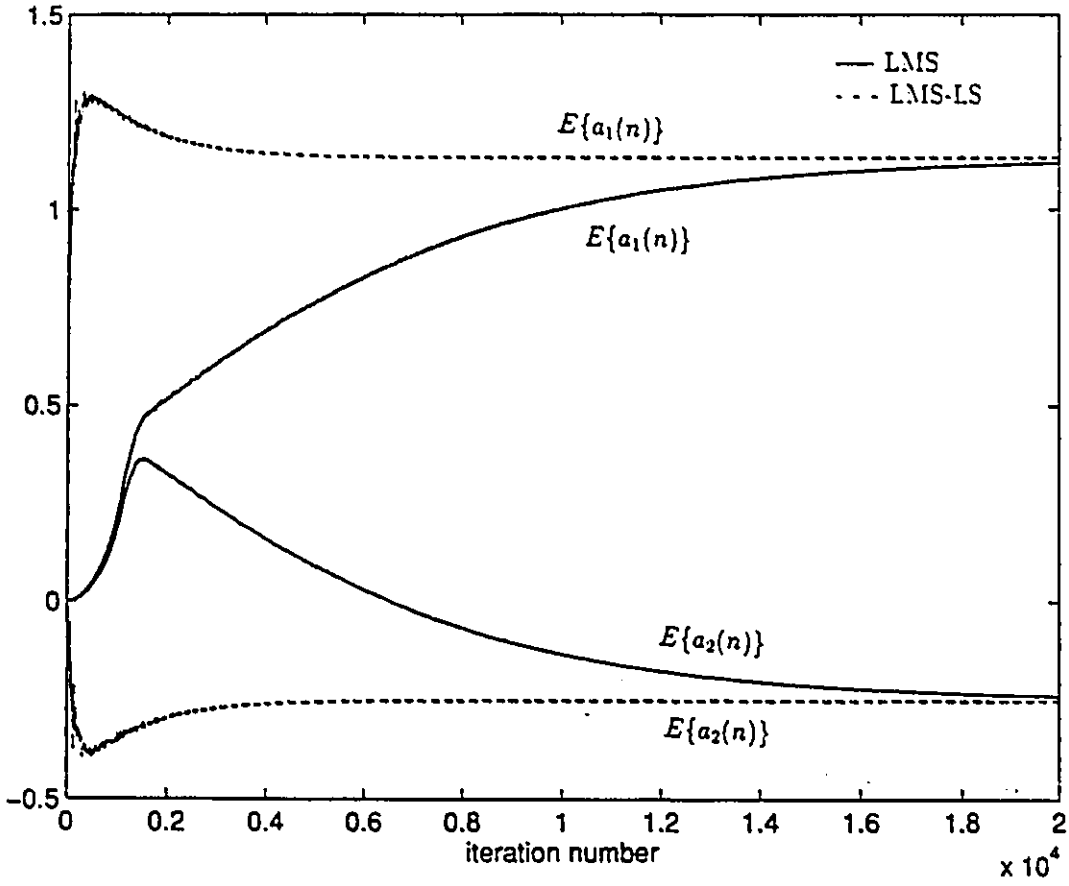

Fig. 5.3 (a) Comparison of ensemble average of $\mathbf{A}(n)$ between the LMS and the LMS-LS algorithms with $\mu_{LMS} = \mu_{LMS-LS} = 0.001$ for example 1.

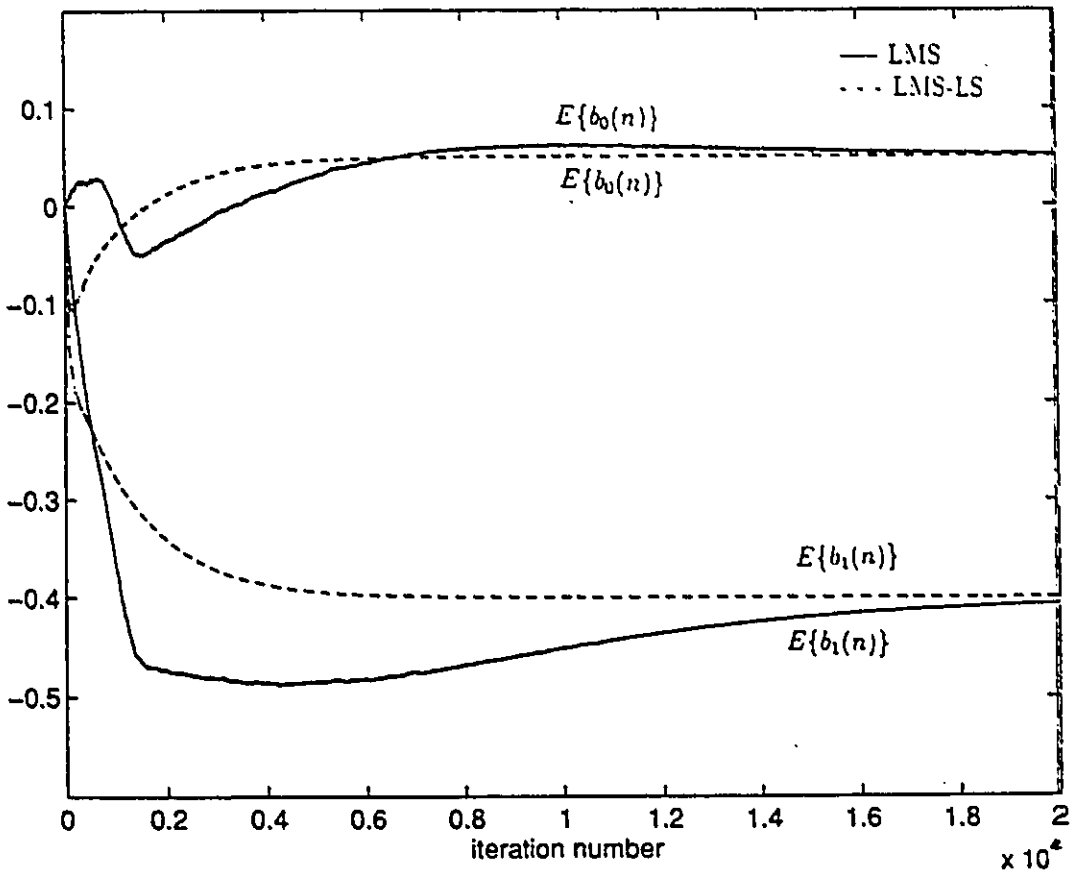

Fig. 5.3 (b) Comparison of ensemble average of $\mathbf{B}(n)$ between the LMS and the LMS-LS algorithms with $\mu_{LMS} = \mu_{LMS-LS} = 0.001$ for example 1.

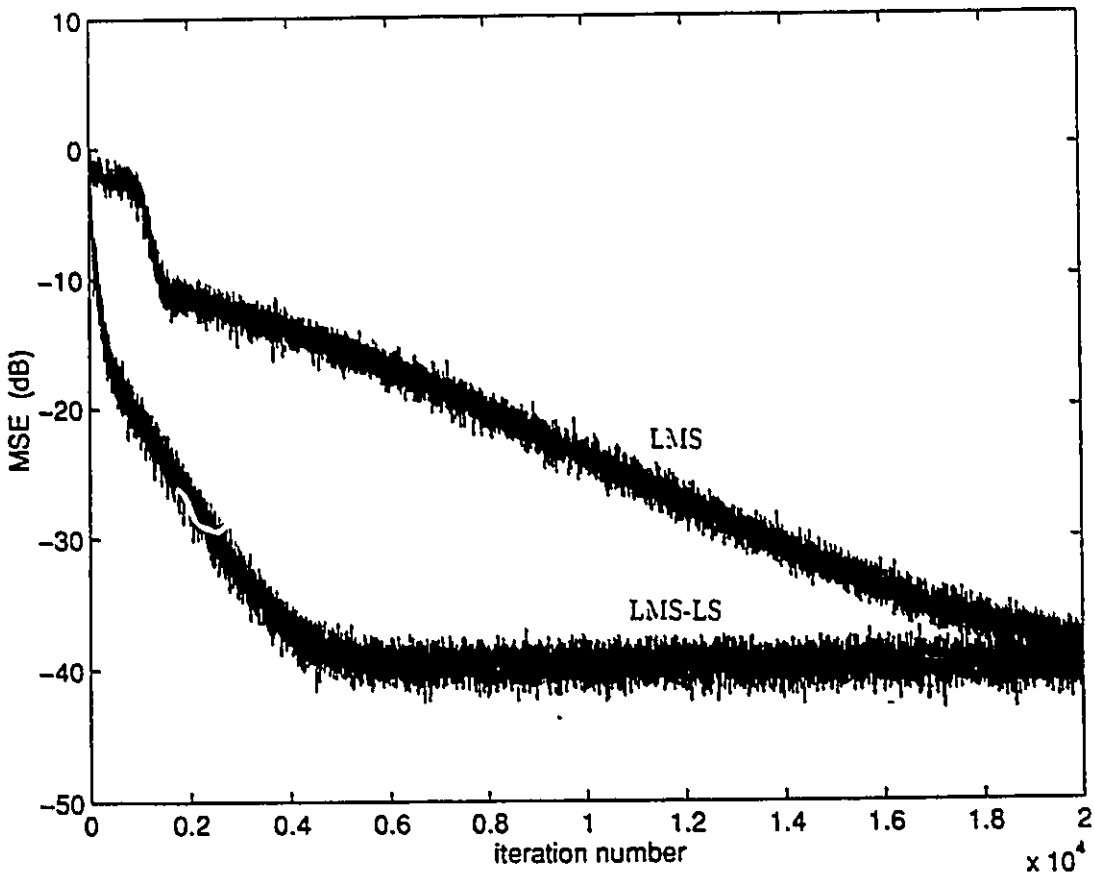

Fig. 5.4 Comparison of MSE between the LMS and the LMS-LS algorithms with $\mu_{LMS} = \mu_{LMS-LS} = 0.001$ for example 1.

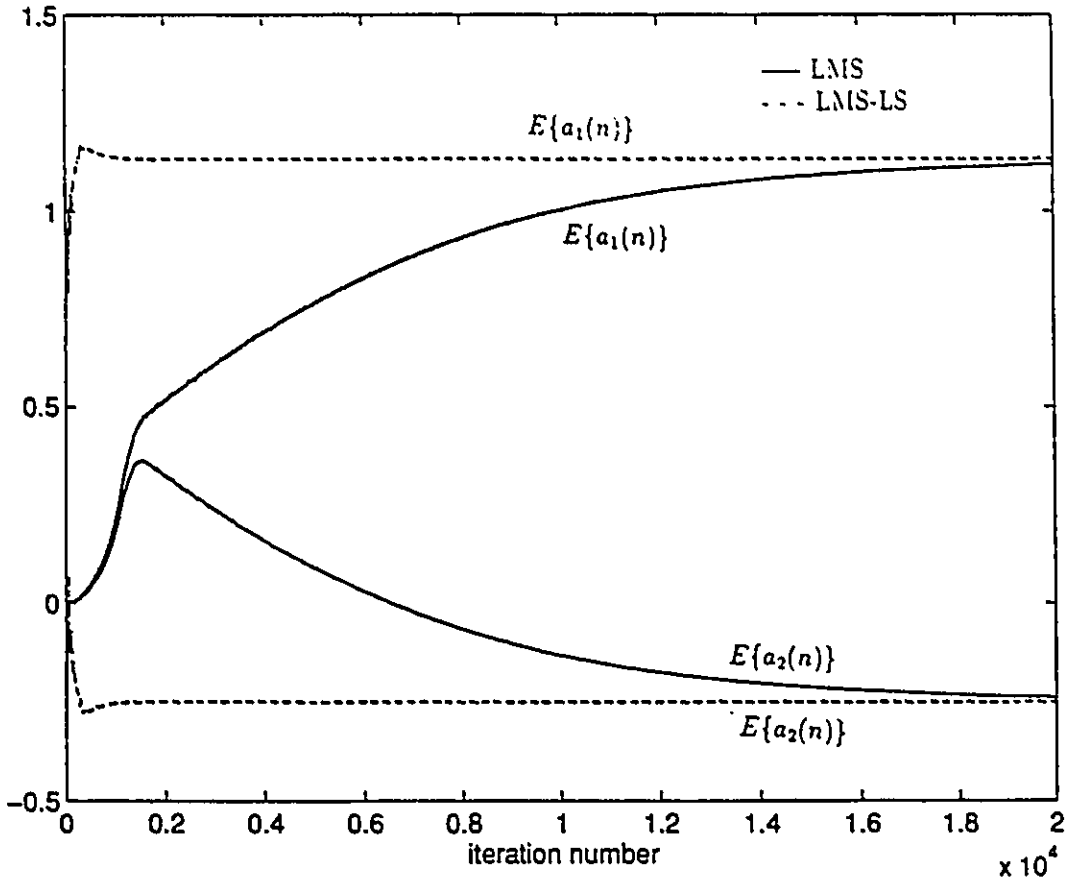

Fig. 5.5 (a) Comparison of ensemble average of $\mathbf{A}(n)$ between the LMS and the LMS-LS algorithms with $\mu_{LMS} = 0.001$ and $\mu_{LMS-LS} = 0.005$ for example 1.

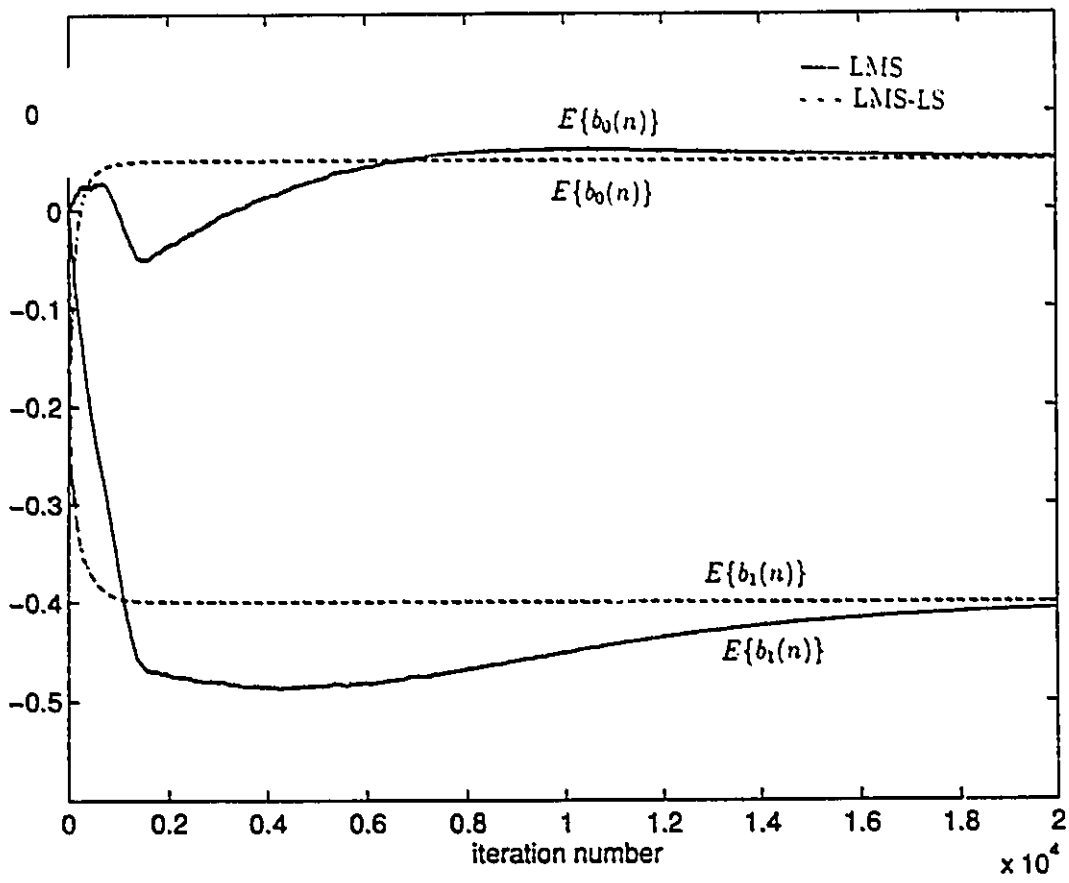

Fig. 5.5 (b) Comparison of ensemble average of $\mathbf{B}(n)$ between the LMS and the LMS-LS algorithms with $\mu_{LMS} = 0.001$ and $\mu_{LMS-LS} = 0.005$ for example 1.

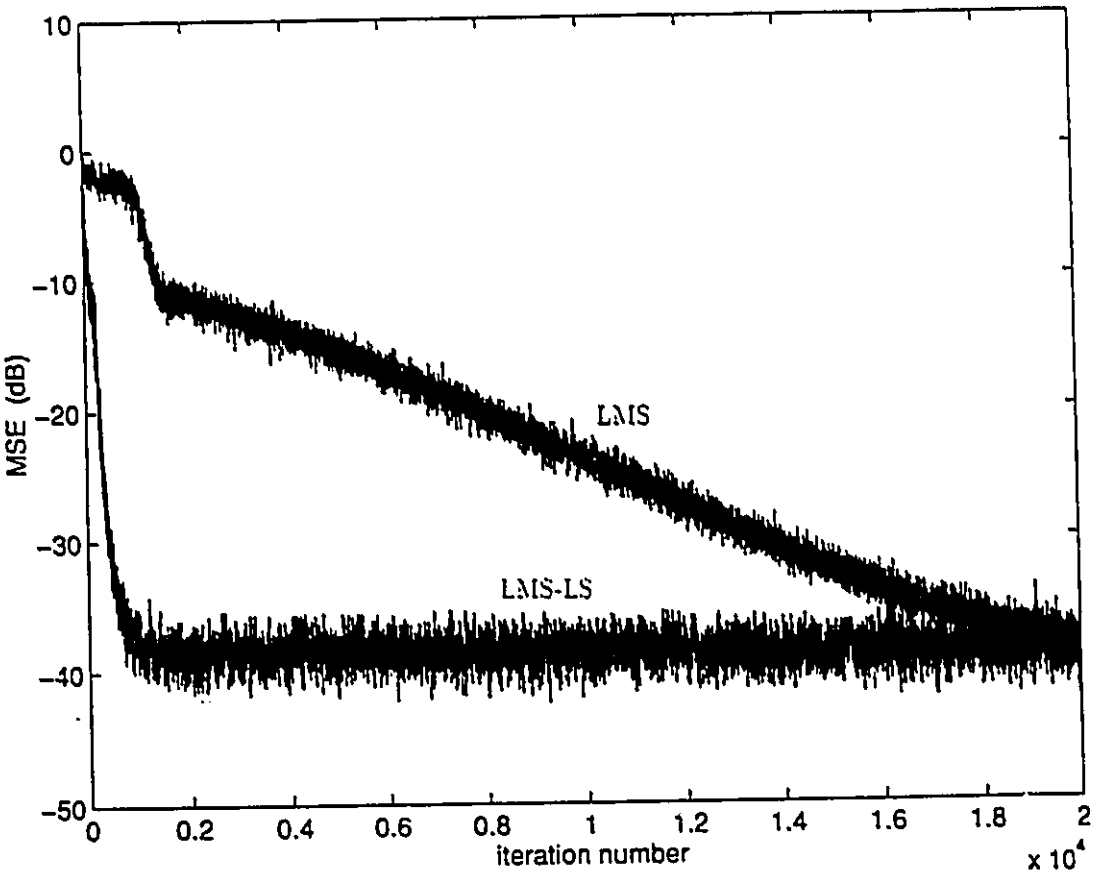

Fig. 5.6 Comparison of MSE between the LMS and the LMS-LS algorithms with $\mu_{LMS} = 0.001$ and $\mu_{LMS-LS} = 0.005$ for example 1.

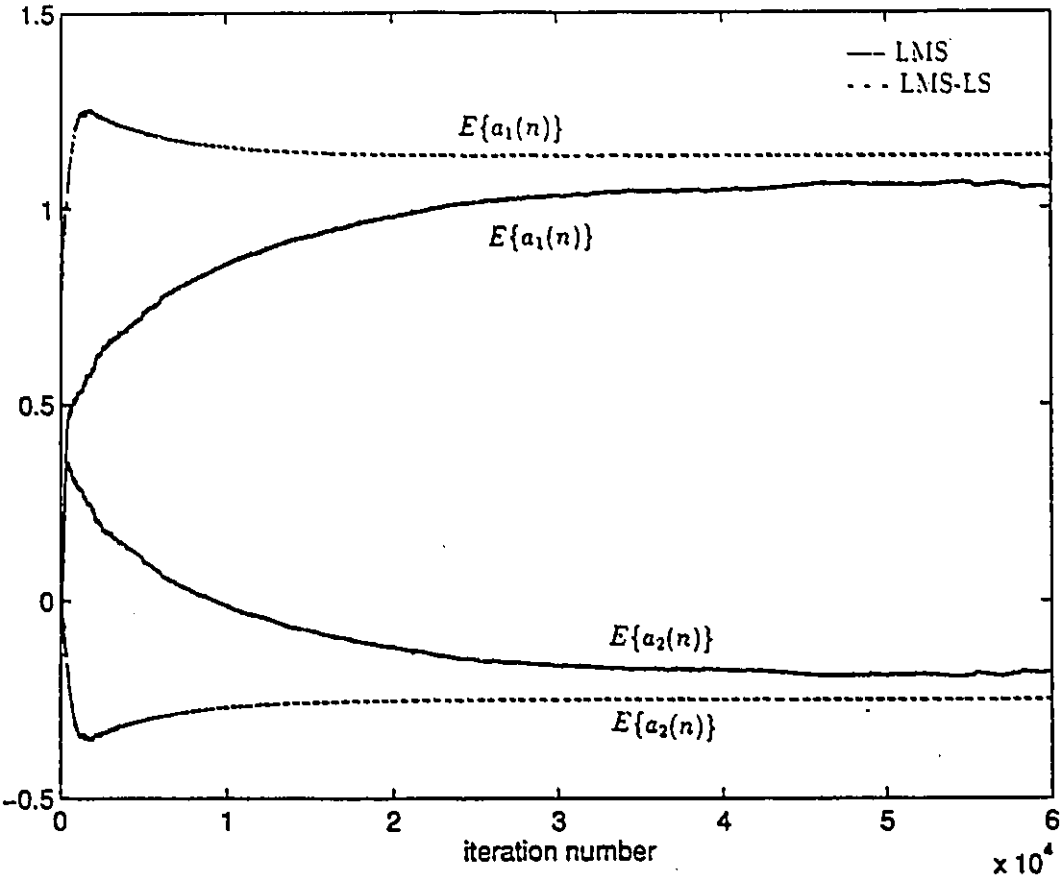

Fig. 5.7 (a) Comparison of ensemble average of $\mathbf{A}(n)$ between the LMS and the LMS-LS algorithms with $\mu_{LMS} = \mu_{LMS-LS} = 0.0003$ for example 2.

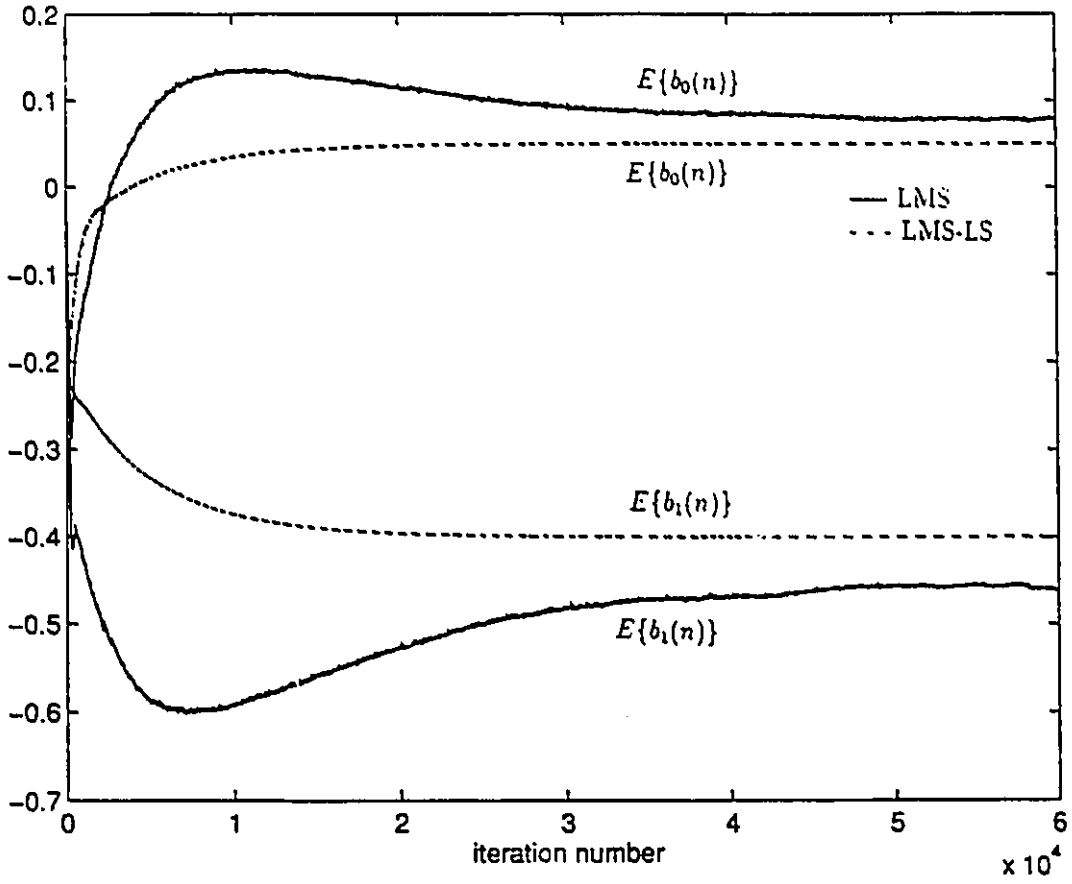

Fig. 5.7 (b) Comparison of ensemble average of $\mathbf{B}(n)$ between the LMS and the LMS-LS algorithms with $\mu_{LMS} = \mu_{LMS-LS} = 0.0003$ for example 2.

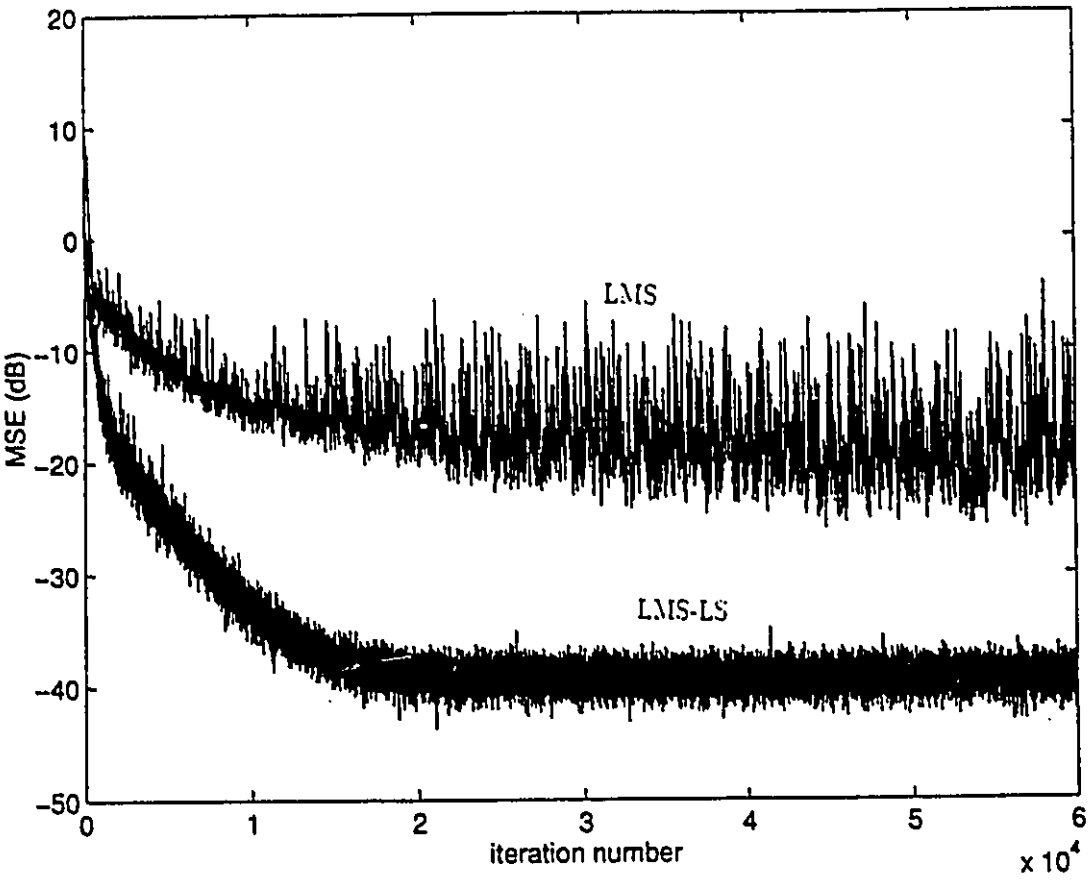

Fig. 5.8 Comparison of MSE between the LMS and the LMS-LS algorithms with $\mu_{LMS} = \mu_{LMS-LS} = 0.0003$ for example 2.

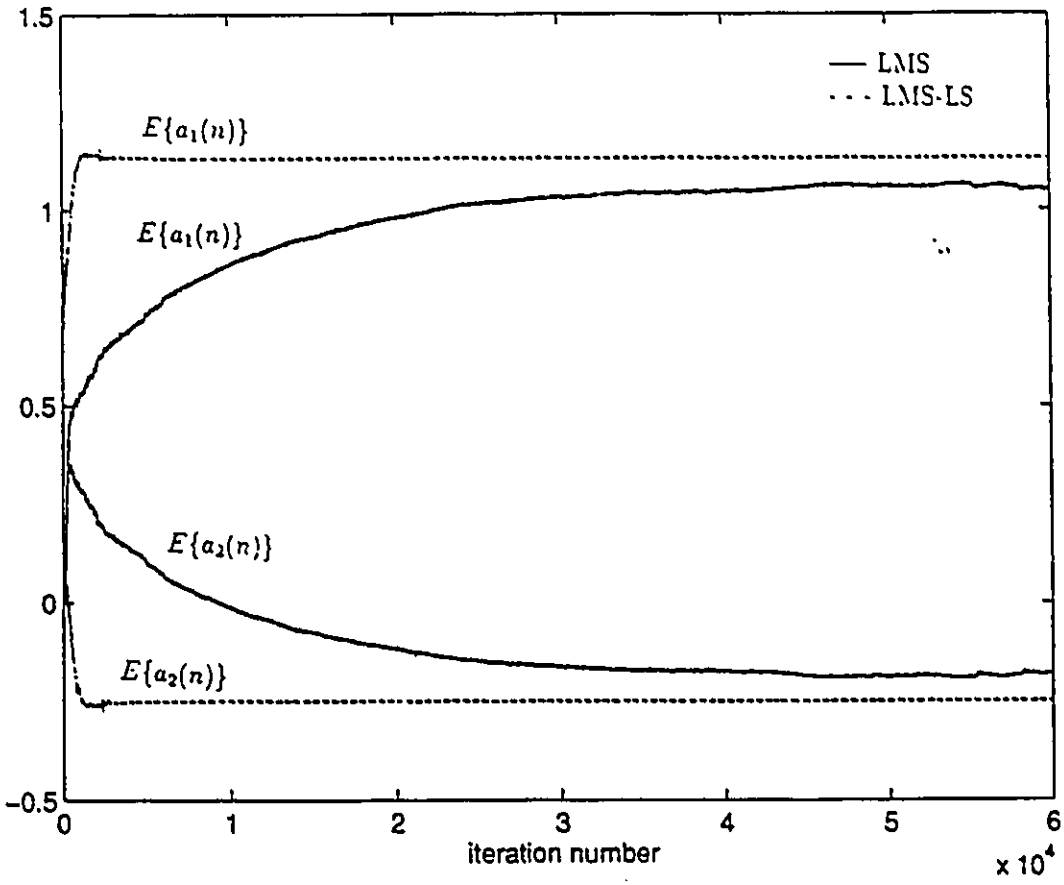

Fig. 5.9 (a) Comparison of ensemble average of $\mathbf{A}(n)$ between the LMS and the LMS-LS algorithms with $\mu_{LMS} = 0.0003$ and $\mu_{LMS-LS} = 0.0006$ for example 2.

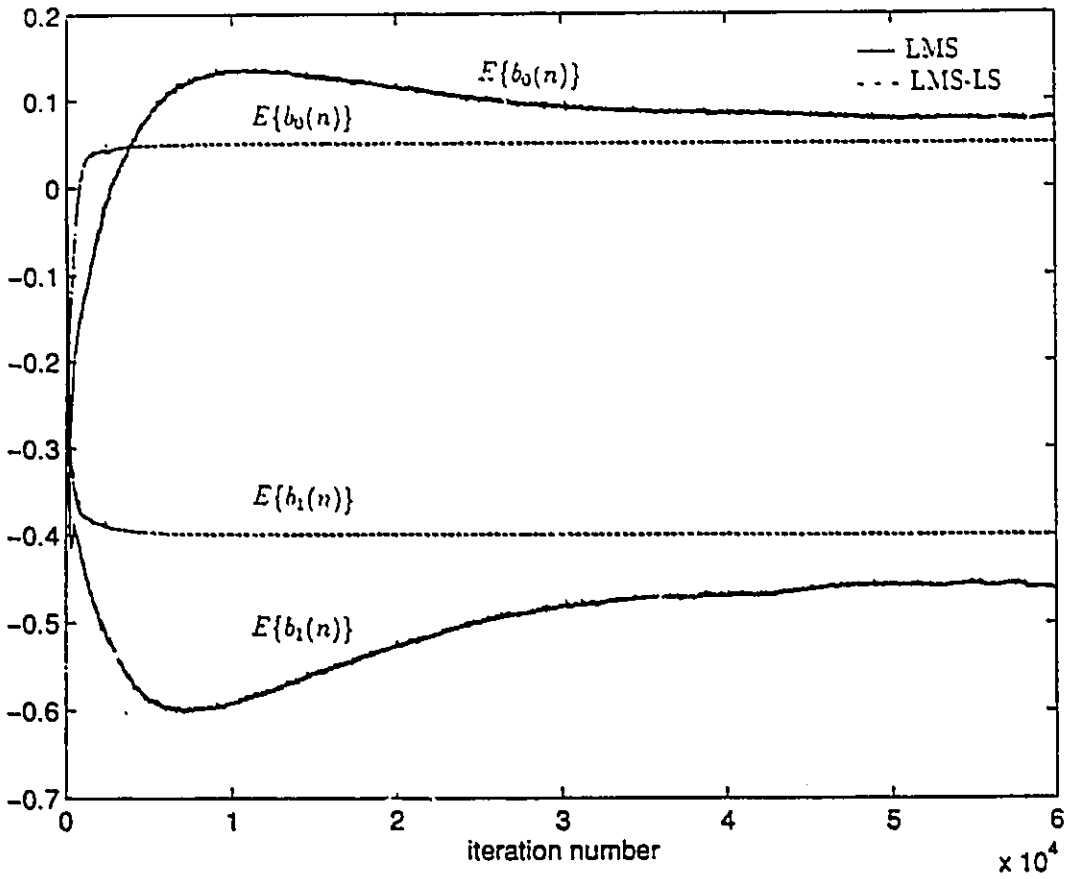

Fig. 5.9 (b) Comparison of ensemble average of $\mathbf{B}(n)$ between the LMS and the LMS-LS algorithm with $\mu_{LMS} = 0.0003$ and $\mu_{LMS-LS} = 0.0006$ for example 2.

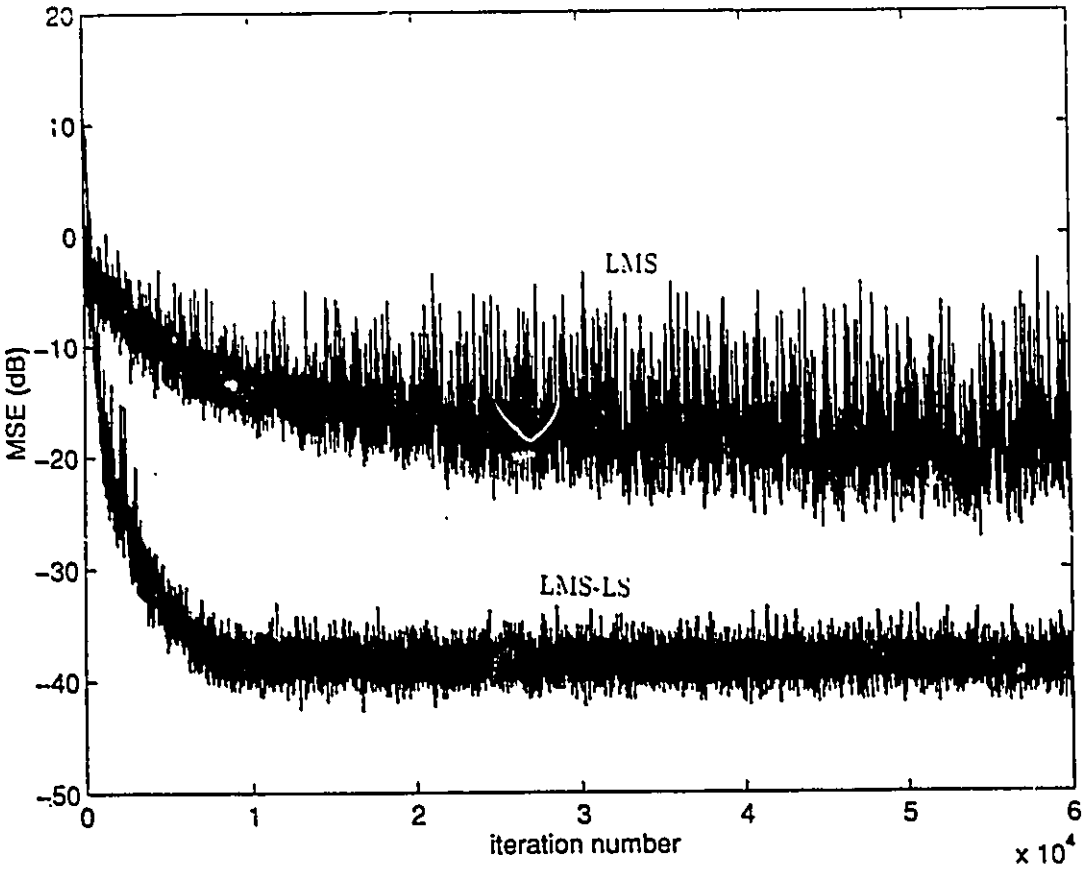

Fig. 5.10 Comparison of MSE between the LMS and the LMS-LS algorithms with $\mu_{LMS} = 0.0003$ and $\mu_{LMS-LS} = 0.0006$ for example 2.

5.4.2 Testing local and global convergence

In the following examples, local and global convergence properties of the LMS-LS algorithms are examined. Those examples are obtained from [27, 51, 101]. The error surface contours of these examples are plotted in reference [27]. The input signal in all next examples is a coloured one.

5.4.2.1 Example 3: sufficient order

The system to be modeled is [27]

$$H(z) = \frac{1}{1 - 1.4z^{-1} + 0.49z^{-2}} \tag{5.43}$$

and the adaptive filter transfer function is

$$H(z,n) = \frac{b_0(n)}{1 - a_1(n)z^{-1} - a_2(n)z^{-2}} \tag{5.44}$$

The input signal is obtained by passing a zero-mean uncorrelated Gaussian signal through the colouring filter $(1 - 0.7z^{-1})^2(1 + 0.7z^{-1})^2$. The resulting error surface has two minima (see Fig. 10 of reference [27]). The global minimum is located at $[b_0, a_1, a_2] = [1, 1.4, -0.49]$, and the local minimum is approximately at $[-0.22, -1.35, -0.49]$. Both the LMS and the LMS-LS algorithms are used with the same step size $\mu_{LMS} = \mu_{LMS-LS} = 0.0005$. λ was chosen to be 0.99. The initial point for both algorithms is given by $[b_0(0), a_1(0), a_2(0)] = [0, -1.4, -0.5]$ which was chosen near the local minimum. Results are shown in Fig. 5.11. The

LMS algorithm is seen to converge to the local minimum whereas the LMS-LS algorithm was able to escape the local minimum and converge to the global one.

5.4.2.2 Example 4: insufficient order

In this example, we consider the case when the input is coloured and the adaptive filter is of insufficient order. The transfer function of the system is given by [27]

$$H(z) = \frac{1}{1 - 1.8z^{-1} + 1.08z^{-2} - 0.216z^{-3}} \tag{5.45}$$

and that of the adaptive filter is

$$H(z,n) = \frac{b_0(n)}{1 - a_1(n)z^{-1} - a_2(n)z^{-2}} \tag{5.46}$$

The colouring filter is $(1 - 0.6z^{-1})^2(1 + 0.6z^{-1})^2$. The error surface for this example is bimodal (see Fig. 11 of reference [27]). The global minimum is located at approximately $[b_0, a_1, a_2] = [1.14, 1.44, -0.55]$, and the local minimum is at approximately $[-0.33, -0.9, -0.35]$. The LMS-LS algorithm is used with $\mu_{LMS-LS} = 0.001$, $\lambda = 0.99$, and the LMS algorithm is used with $\mu_{LMS} = 0.001$. The adaptive filter coefficients are initialized in the proximity of the the local minimum at the point $[b_0(0), a_1(0), a_2(0)] = [0, -.9, -0.6]$. From Fig. 5.12, it is clear that the LMS-LS algorithm converged to the global minimum while the LMS algorithm converged to the local minimum.

While we believe that the proposed algorithm tends to converge to the global optimum in general, it was noticed that this was not the case for white input,

insufficient filter order when the adaptive algorithm is initialized near the local minimum. In this case, convergence to the global optimum was not consistent over all runs; sometimes the algorithm converged to the local minimum without any apparent cause. However, it should be noted that in real applications, input signals are more likely to be coloured than white.

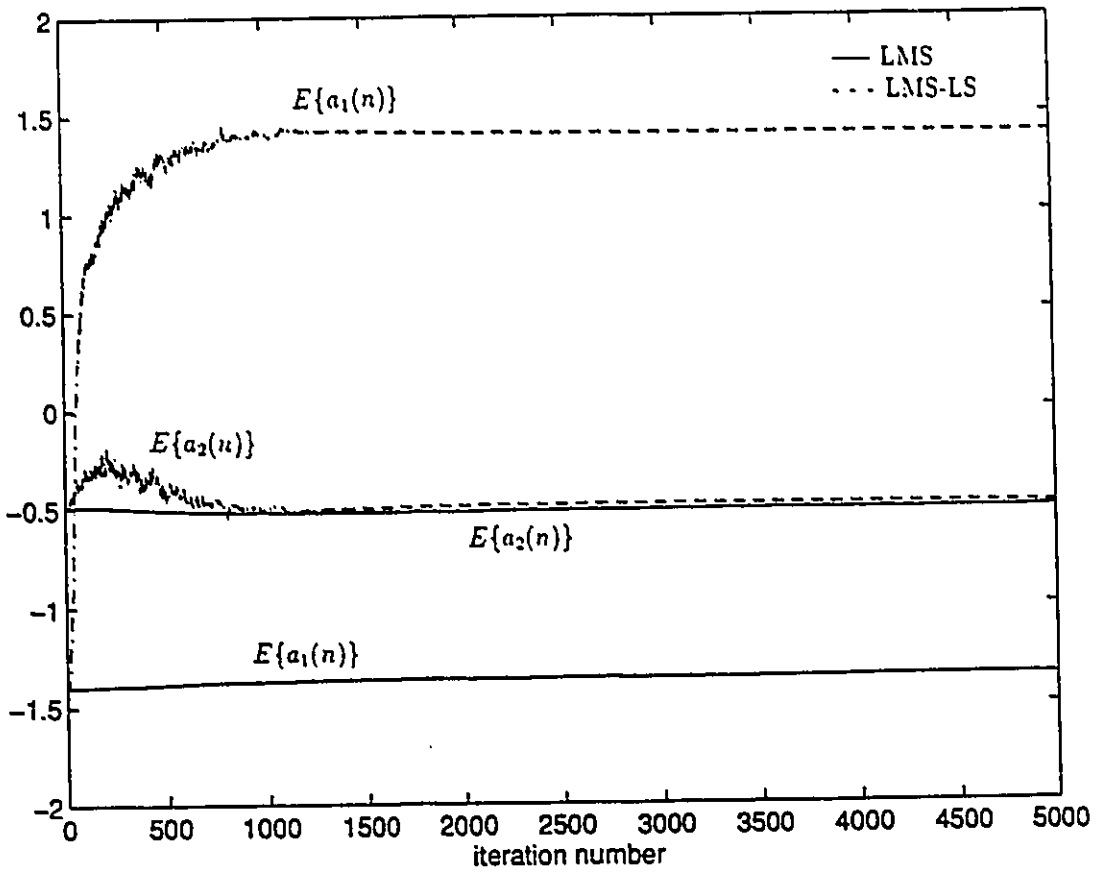

Fig. 5.11 (a) Comparison of ensemble average of $\mathbf{A}(n)$ between the LMS and the LMS-LS algorithms with $\mu_{LMS} = \mu_{LMS-LS} = 0.0005$ for example 3.

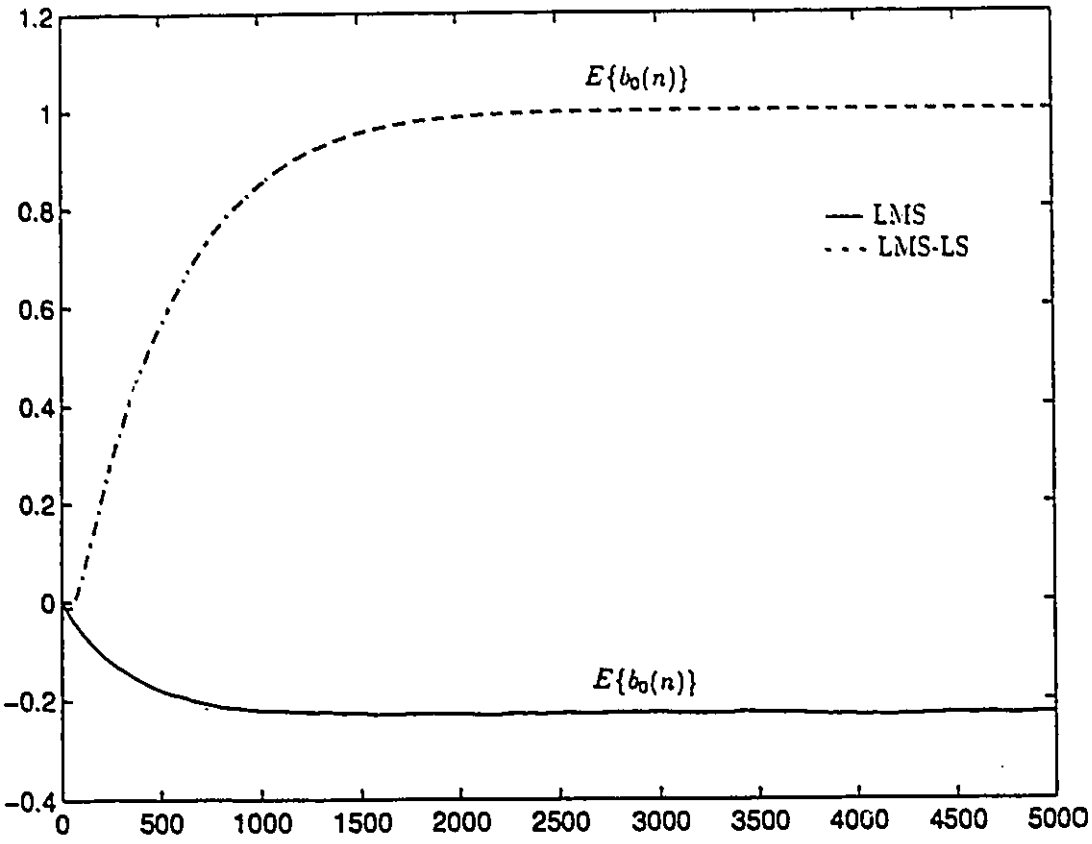

Fig. 5.11 (b) Comparison of ensemble average of $\mathbf{B}(n)$ between the LMS and the LMS-LS algorithms with $\mu_{LMS} = \mu_{LMS-LS} = 0.0005$ for example 3.

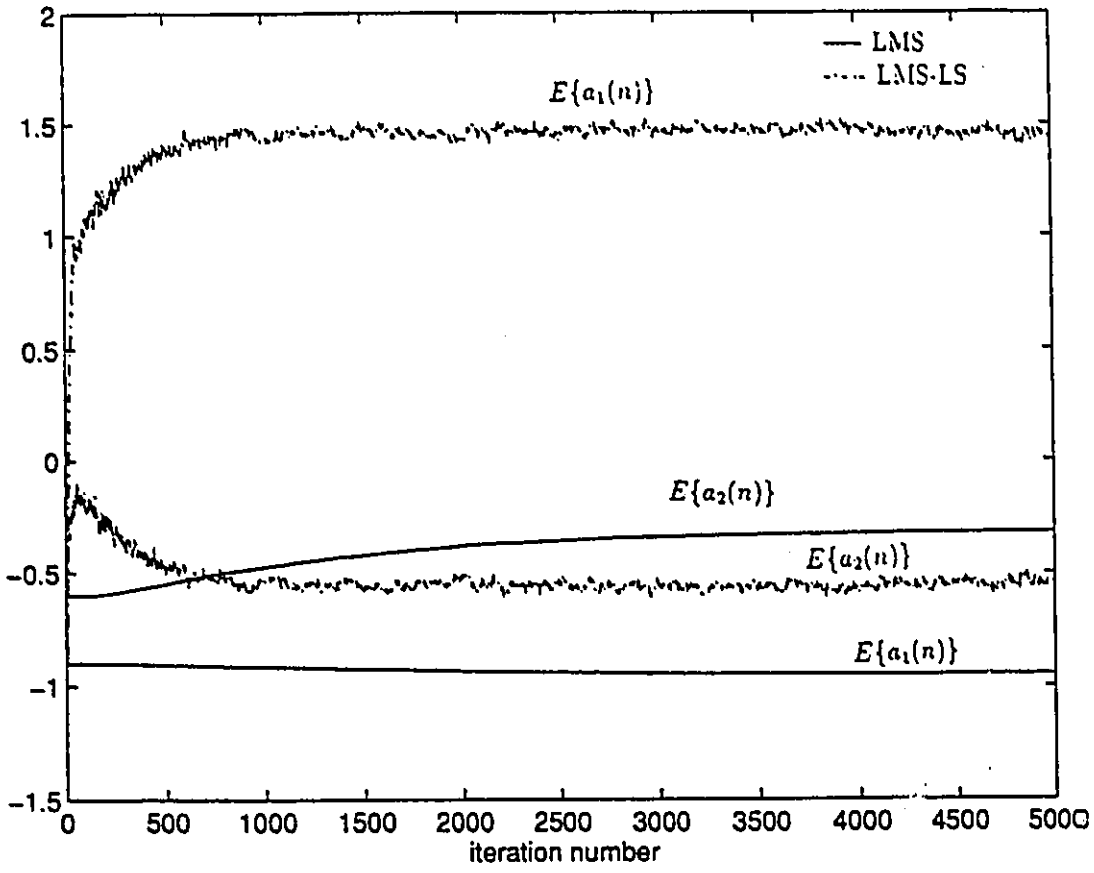

Fig. 5.12 (a) Comparison of ensemble average of $\mathbf{A}(n)$ between the LMS and the LMS-LS algorithms with $\mu_{LMS} = \mu_{LMS-LS} = 0.001$ for example 4.

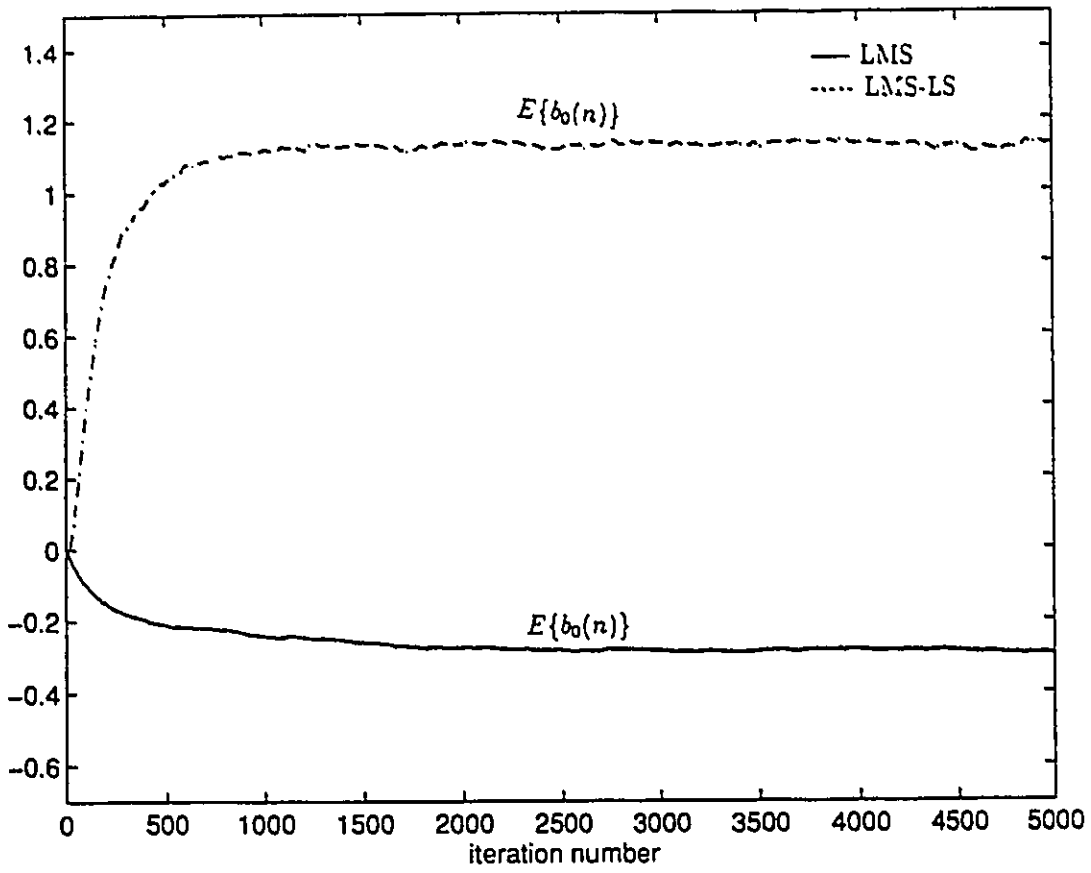

Fig. 5.12 (b) Comparison of ensemble average of $\mathbf{B}(n)$ between the LMS and the LMS-LS algorithms with $\mu_{LMS} = \mu_{LMS-LS} = 0.001$ for example 4.

5.4.3 Acoustic echo cancellation

In this section, the fast LMS-LS IIR, the LMS IIR, and the LMS FIR [113] algorithms are applied to the cancellation of acoustic echo in the example given in [14]. The echo path is modeled by pole/zero transfer function of 10-order numerator and 10-order denominator. The numerator coefficients are generated randomly whereas the denominator is given by

$$A(z^{-1}) = 1 - 0.8z^{-10} \tag{5.47}$$

The impulse response of the echo path is plotted in figure 5.13. The input signal $x(n)$ is an artificial speech generated as

$$x(n) = \frac{1}{1 - 0.7q^{-10}}\eta(n) \tag{5.48}$$

where $\eta(n)$ is zero-mean white Gaussian noise of unity variance. A zero-mean white noise of -50dB relative to the echo is added to the output of the echo path. The fast LMS-LS IIR and the LMS IIR algorithms are used with $M = 10$, $N = 10$, and $\mu = 0.0003$. For the fast LMS-LS IIR algorithm, δ is chosen to be 0.001. Stability monitoring of the IIR adaptive filter is performed based on the method in [14], where a sufficient condition for stability is to guarantee that

$$\sum_{j=1}^{M} |a_j(n)| < 1 \ , \ \forall n \tag{5.49}$$

From Fig. 5.14, the ERLE achieved by the fast LMS-LS IIR algorithm is approximately 50 dB. This follows from the ability of the IIR adaptive filter to model

the echo path exactly. Note, also, the extremely fast convergence of the LMS-LS IIR method compared to the LMS IIR. The LMS IIR algorithm converges to the same level of ERLE after 120,000 samples. The same experiment is carried out for the LMS FIR algorithm [113]. The number of weights in the FIR adaptive filter is 280. This is chosen to achieve the same steady state ERLE as the fast LMS-LS IIR algorithm. It is obvious from Fig. 5.14 that the LMS FIR algorithm needs approximately the same number of iterations as the fast LMS-LS IIR algorithm to achieve the same level of ERLE. On the other hand, the LMS FIR algorithm requires $2 \times 280 = 560$ multiplications at each iteration, while the fast LMS-LS IIR algorithm needs[3] $2 \times 10 + 14 \times 10 = 160$. The advantage of the fast LMS-LS IIR algorithm over the LMS FIR algorithm is clear in this case: it reduces the number of computations at each iteration by a factor of 3.5 without any loss in performance.

[3]Eq.(5.24)$\rightarrow M$, Eq.(5.25)$\rightarrow M$, Eq.(5.26)$\rightarrow M+N$, Eq.(5.28)$\rightarrow N$, Eq.(5.29)$\rightarrow M$, Eqs.(5.31)-(5.39)$\rightarrow 10M$. The total is $14M+2N$

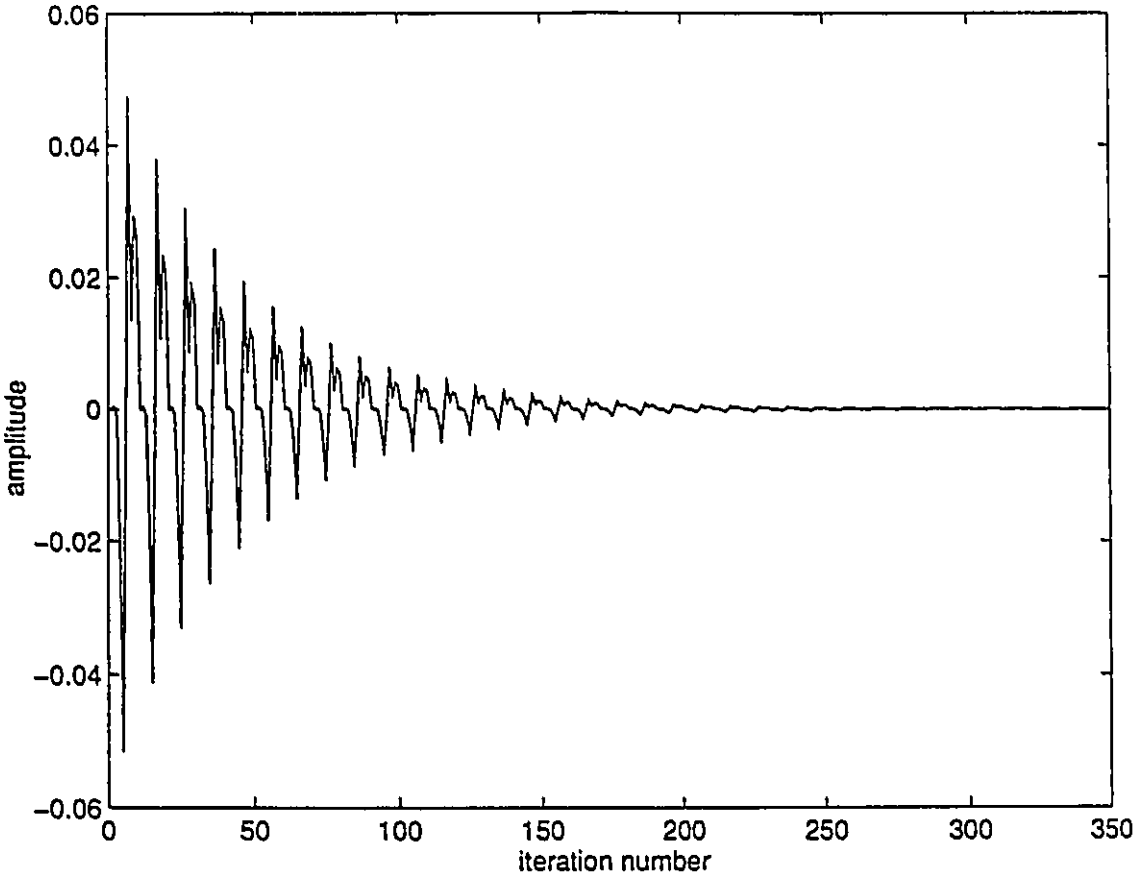

Fig. 5.13 Impulse response of the echo path.

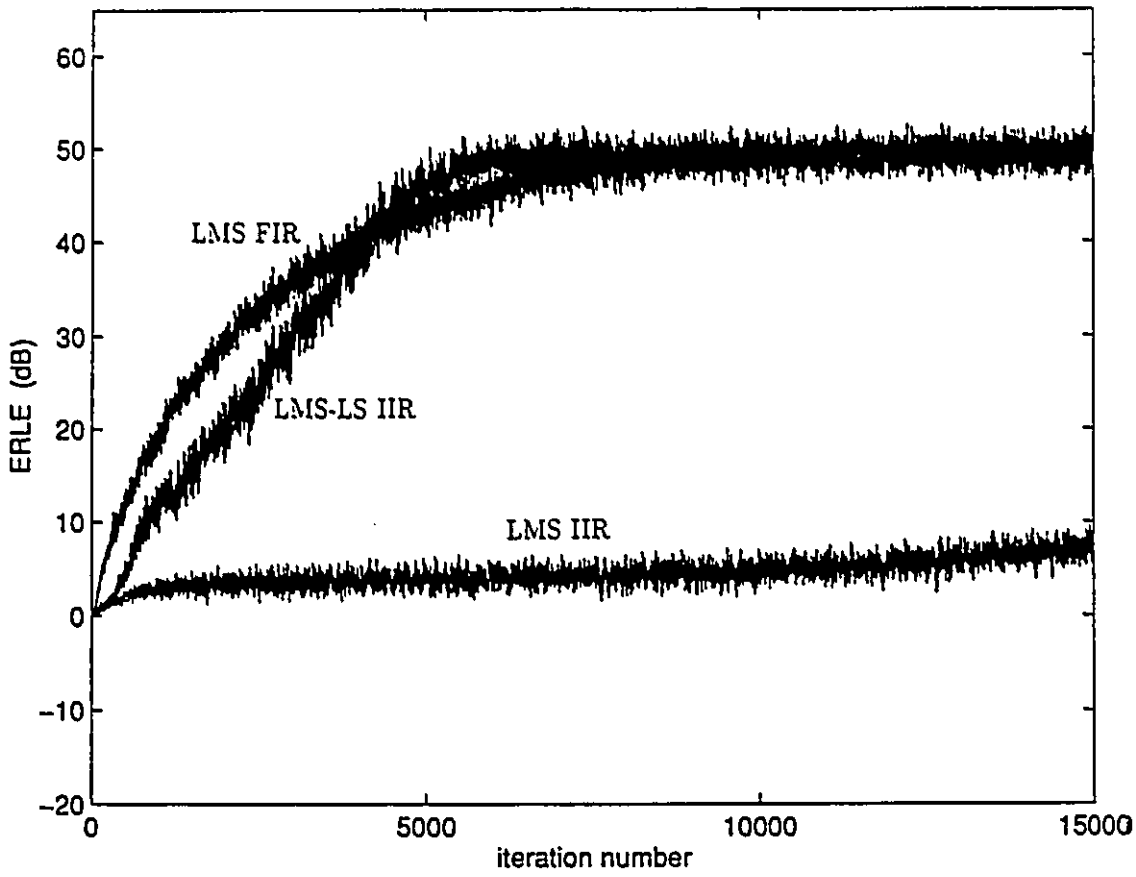

Fig. 5.14 Comparison of ERLE of the LMS-LS IIR, LMS IIR and LMS FIR algorithms.

5.5 Conclusion

This chapter presented a new adaptive algorithm for IIR filters based on output-error formulation. The proposed algorithm uses the LMS method to update the numerator coefficients of the adaptive filter, since they define a quadratic performance function, while a LS-type method is used to update the denominator coefficients, since they define a highly nonlinear error function. Simulation examples indicated that the proposed approach converges significantly faster than the LMS with minimal increase in complexity. This was to be expected as a result of the use of the LS method in the update of the denominator coefficients. Moreover, since the denominator update is not function of the step size parameter, the stability boundary of the numerator step size is increased. This permits a larger step size to be used to speed up adaptation. Examples were provided to compare the proposed algorithm to the LMS IIR adaptive algorithm. Simulation results showed that the proposed coupled LMS-LS algorithm tends to converge to the global minimum for sufficient order cases considered and some reduced order cases.

Chapter 6

Leaky LMS: MSE Analysis For Gaussian Data

The conventional LMS is widely used in numerous applications because of its simplicity and ease of implementation. However, it is generally known that direct implementation of the regular LMS can be problematic. Variants of the LMS, as the Leaky LMS, were proposed to deal with some of these problems. For example, inadequacy of excitation in the input sequence can result in unbounded parameter estimates [91]. This behavior can cause numerical problems due to overflow as well as degraded performance as a consequence of possibly unbounded prediction error. Introducing leakage in the LMS algorithm stabilizes the system.

When the gradient estimate becomes too small to adjust algorithm coefficients due to very low input signal, the coefficients are locked up causing a "stalling" problem and consequently biasing filter coefficients. In such cases, it might be preferable to have them return to zero [113], which can be achieved by employing

the Leaky LMS.

The employment of adaptive echo cancelers on short distance telephony highlighted the undesirable phenomenon of "bursting" that can lead to poor performance. It was shown in [36] that for a properly selected leakage factor value, bursting can be stopped in simple cases.

In adaptive antenna [102], the adaptive sidelobe canceler works well if the input signal is weak compared to the interference or jamming signals. However, the received signal can occasionally be strong enough to delete itself. The Leaky LMS is used to solve this problem through generating an inherent added interference [113].

Despite the fact that leaky LMS has been used in practice in numerous applications as indicated above, there has been no formal analysis of its performance. It is generally accepted that leakage results in coefficient bias and as such has to be maintained low. However, no relationship exists to exactly quantify the effect of the leakage factor on the MSE. Moreover, no analytical bounds on the adaptation step size were presented to ensure the convergence of the Leaky LMS algorithm.

In this chapter, we provide MSE analysis of the leaky LMS algorithm with the assumption that the input and the desired signals are jointly Gaussian and zero-mean processes [73]. We also will employ the common independence assumption that the present input data vector is statically independent of the present coefficient

weight vector [47]. The MSE is determined as a function of the leakage factor. Stability bounds on the step size, in the presence of leakage, are also determined

6.1 MSE Analysis

6.1.1 Convergence of the MSE

The leaky LMS is introduced to minimize the instantaneous objective function

$$J(n) = e^2(n) + \gamma \mathbf{W}^{\mathbf{T}}(n)\mathbf{W}(n) \tag{6.1}$$

where $\mathbf{W}(n)$ is the $N \times 1$ coefficient weight vector of the adaptive algorithm, γ is the leakage factor greater than zero, $e(n) = d(n) - \mathbf{X}^{\mathbf{T}}(n)\mathbf{W}(n)$ is the error signal, $d(n)$ is the desired signal, and $\mathbf{X}(n)$ is the $N \times 1$ input data vector. $J(n)$ in Eq.(6.1) is a quadratic function of the coefficient vector $\mathbf{W}(n)$. The existence and uniqueness of a minimum of $J(n)$ is ensured when the input $\mathbf{X}(n)$ is persistently exciting [91], along with the implicit assumption of stationary sequences $d(n)$ and $\mathbf{X}(n)$. This will guarantee the invertibility of the input data autocorrelation matrix $\mathbf{R} = E\{\mathbf{X}(n)\mathbf{X}^{\mathbf{T}}(n)\}$ (i.e, $\mathbf{R}$ is positive definite). Also, it allows the decomposition of the symmetric matrix $\mathbf{R}$ into $\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{\mathbf{T}}$, where $\mathbf{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$, $\lambda_N \leq \lambda_{N-1} \ldots \leq \lambda_1$, is the matrix of the eigenvalues and $\mathbf{Q}$ is the matrix of the eigenvectors of $\mathbf{R}$. The minimum of $J(n)$ can be sought recursively using the gradient method

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \frac{\mu}{2}\frac{\partial J(n)}{\partial \mathbf{W}(n)}$$

$$= (1 - \mu\gamma)\mathbf{W}(n) + \mu e(n)\mathbf{X}(n) \tag{6.2}$$

where μ is the adaptation step size. Introducing the translated weight vector $\mathbf{V}(n) = \mathbf{W}(n) - \mathbf{W}^*$, where $\mathbf{W}^* = \mathbf{R}^{-1}\mathbf{P}$ is the optimal weight vector, and defining the rotated vectors $\acute{\mathbf{V}}(n) = \mathbf{Q}^{\mathbf{T}}\mathbf{V}(n)$, $\acute{\mathbf{X}}(n) = \mathbf{Q}^{\mathbf{T}}\mathbf{X}(n)$, $\acute{\mathbf{W}}^* = \mathbf{Q}^{\mathbf{T}}\mathbf{W}^*$, then using the relation $d(n) = \mathbf{X}^{\mathbf{T}}(n)\mathbf{W}^* + e^*(n)$, where $e^*(n)$ is assumed zero-mean and white Gaussian sequence of ϵ_{min} variance, Eq.(6.2) can be rewritten as

$$\acute{\mathbf{V}}(n+1) = [\mathbf{I} - \mu(\gamma\mathbf{I} + \acute{\mathbf{X}}(n)\acute{\mathbf{X}}^{\mathbf{T}}(n))]\acute{\mathbf{V}}(n) - \mu\gamma\acute{\mathbf{W}}^* + \mu e^*(n)\acute{\mathbf{X}}(n) \tag{6.3}$$

Taking the expected value of both sides of Eq.(6.3) and using the common independence assumption of $\acute{\mathbf{V}}(n)$ and $\acute{\mathbf{X}}(n)$ yields[1]

$$E\{\acute{\mathbf{V}}(n+1)\} = [\mathbf{I} - \mu(\gamma\mathbf{I} + \mathbf{\Lambda})]E\{\acute{\mathbf{V}}(\mathbf{n})\} - \mu\gamma\acute{\mathbf{W}}^* \tag{6.4}$$

Clearly, from Eq.(6.4), boundness of expectation of all modes is guaranteed by the well-known condition on the step size μ [46]

$$0 < \mu < \frac{2}{\gamma + \lambda_{max}} \tag{6.5}$$

where λ_{max} is the largest eigenvalue of $\mathbf{R}$. The jth component of the steady state solution of Eq.(6.4) is

$$E\{\acute{V}_j(\infty)\} = \frac{-\gamma}{\lambda_j + \gamma}\acute{W}_j^* \tag{6.6}$$

[1]Note that $E\{e^*(n)\acute{\mathbf{X}}(n)\} = 0$, and since $e^*(n)$ and $\acute{\mathbf{X}}(n)$ are zero-mean Gaussian processes then they are independent.

Recalling that $\acute{V}_j(n)$ is the rotated error in the coefficient vector, it is clear from Eq.(6.6) that a nonzero leakage factor γ results in some nonzero steady state coefficient bias.

However, convergence in the mean does not guarantee convergence of the mean-square error. Therefore, we study next the necessary and sufficient conditions for stability in the mean square sense. The analysis is based on the assumption that $\mathbf{X}(n)$ and $d(n)$ are jointly Gaussian, zero-mean stationary signals and on the commonly employed independence assumption of the weight vector $\mathbf{W}(n)$ and the input signal vector $\mathbf{X}(n)$ [47].

The MSE is given by [113]

$$\epsilon(n) = \epsilon_{min} + E\{\acute{\mathbf{V}}^{\mathbf{T}}(n)\mathbf{\Lambda}\acute{\mathbf{V}}(n)\} \tag{6.7}$$

where

$$\epsilon_{min} = E\{d^2(n)\} - \mathbf{P}^{\mathbf{T}}\mathbf{W}^{*} \tag{6.8}$$

is the minimum mean-square error when $\gamma = 0$, and $\mathbf{P} = E\{d(n)\mathbf{X}(n)\}$ is the crosscorrelation vector. Eq.(6.7) shows that for the MSE to converge, it is necessary that the diagonal elements of $E\{\acute{\mathbf{V}}(n)\mathbf{\Lambda}\acute{\mathbf{V}}^{\mathbf{T}}(n)\}$ converge. We will thus proceed to determine the conditions guaranteeing the convergence of the diagonal elements of $E\{\acute{\mathbf{V}}(n)\acute{\mathbf{V}}^{\mathbf{T}}(n)\}$. Postmultiplying both sides of Eq.(6.3) by $\acute{\mathbf{V}}^{\mathbf{T}}(n+1)$, and taking the expected value results in

$$E\{\acute{\mathbf{V}}(n+1)\acute{\mathbf{V}}^{\mathbf{T}}(n+1)\} = (\mu^2\gamma\mathbf{\Lambda} - \mu\mathbf{\Lambda})E\{\acute{\mathbf{V}}(n)\acute{\mathbf{V}}^{\mathbf{T}}(n)\}$$

$$
\begin{aligned}
&+ E\{\acute{\mathbf{V}}(n)\acute{\mathbf{V}}^{\mathbf{T}}(n)\}(\mathbf{I} - \mu\mathbf{\Lambda} - 2\mu\gamma\mathbf{I} + \mu^2\gamma^2\mathbf{I} + \mu^2\gamma\mathbf{\Lambda}) \\
&+ 2\mu^2\mathbf{\Lambda}E\{\acute{\mathbf{V}}(n)\acute{\mathbf{V}}^{\mathbf{T}}(n)\}\mathbf{\Lambda} + \mu^2\mathbf{\Lambda}tr(\mathbf{\Lambda}E\{\acute{\mathbf{V}}(n)\acute{\mathbf{V}}^{\mathbf{T}}(n)\}) \\
&+ E\{\acute{\mathbf{V}}(n)\}(\mu^2\gamma^2\acute{\mathbf{P}}^{\mathbf{T}}\mathbf{\Lambda}^{-1} - \mu\gamma\acute{\mathbf{P}}^{\mathbf{T}}\mathbf{\Lambda}^{-1}) \\
&+ (\mu^2\gamma^2\mathbf{\Lambda}^{-1}\acute{\mathbf{P}} - \mu\gamma\mathbf{\Lambda}^{-1}\acute{\mathbf{P}})E\{\acute{\mathbf{V}}(n)\} \\
&+ \mu^2\gamma\mathbf{\Lambda}E\{\acute{\mathbf{V}}(n)\}\acute{\mathbf{P}}^{\mathbf{T}}\mathbf{\Lambda}^{-1} + \mu^2\gamma\mathbf{\Lambda}^{-1}\acute{\mathbf{P}}E\{\acute{\mathbf{V}}^{\mathbf{T}}(n)\}\mathbf{\Lambda} \\
&+ \mu^2\gamma^2\mathbf{\Lambda}^{-1}\acute{\mathbf{P}}\acute{\mathbf{P}}^{\mathbf{T}}\mathbf{\Lambda}^{-1} + \mu^2E\{e^{*2}(n)\}\mathbf{\Lambda} \qquad (6.9)
\end{aligned}
$$

where *tr(.)* is the trace operator. Note that in Eq. (6.9), the assumption of zero-mean stationary Gaussian inputs allowed the use of the Gaussian moment factoring theorem to simplify the expression $E\{\acute{\mathbf{X}}(n)\acute{\mathbf{X}}^{\mathbf{T}}(n)\acute{\mathbf{V}}(n)\acute{\mathbf{V}}^{\mathbf{T}}(n)\acute{\mathbf{X}}(n)\acute{\mathbf{X}}^{\mathbf{T}}(n)\}$ into a sum of second order moments [47]. Moreover, we employed the relation $\acute{\mathbf{W}}^{*} = \mathbf{\Lambda}^{-1}\acute{\mathbf{P}}$ where $\acute{\mathbf{P}} = \mathbf{Q}^{\mathbf{T}}\mathbf{P}$. Defining $\mathbf{Z_1}(n)$ as an $N \times 1$ second moment vector whose components are the diagonal elements of $E\{\acute{\mathbf{V}}(n)\acute{\mathbf{V}}^{\mathbf{T}}(n)\}$, $\mathbf{Z_2}(n) = E\{\acute{\mathbf{V}}(n)\}$, we construct the state vector as

$$
\mathbf{Z}(n+1) = \begin{bmatrix} \mathbf{Z_1}(n+1) \\ \mathbf{Z_2}(n+1) \end{bmatrix} \qquad (6.10)
$$

Using Eqs. (6.4), and (6.9), $\mathbf{Z}(n+1)$ can be expressed as

$$
\mathbf{Z}(n+1) = \mathbf{K}\mathbf{Z}(n) + \mathbf{C} \qquad (6.11)
$$

where

$$
\mathbf{K} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \qquad (6.12)
$$

and

$$\begin{aligned}\mathbf{A} &= [\,1-2\mu\gamma+\mu^2\gamma^2\,]\mathbf{I}-2\mu\mathbf{\Lambda}[\,\mathbf{I}-\mu\gamma\mathbf{I}\,] \\ &\quad +2\mu^2\mathbf{\Lambda}^2+\mu^2\mathbf{\Gamma}\mathbf{\Gamma}^{\mathbf{T}} \qquad (6.13)\\ \mathbf{B} &= \left([\,2\mu^2\gamma^2-2\mu\gamma\,]\mathbf{\Lambda}^{-1}+2\mu^2\gamma\mathbf{I}\right)\mathbf{Y} \qquad (6.14)\\ \mathbf{D} &= \mathbf{I}-\mu[\,\gamma\mathbf{I}+\mathbf{\Lambda}\,] \qquad (6.15)\end{aligned}$$

where the diagonal matrix $\mathbf{Y}=\mathrm{diag}\{\dot{p}_1,\dot{p}_2,\ldots,\dot{p}_N\}$, $\dot{p}_j$ is the jth component of $\dot{\mathbf{P}}$ and $\mathbf{\Gamma}=[\lambda_1,\lambda_2,\ldots,\lambda_N]^T$. $\mathbf{C}$ will not be needed in our analysis, however, it must be noted that it is bounded. We further make the following definition

$$a_j=1-2\mu(\gamma+\lambda_j)+\mu^2(\,(\gamma+\lambda_j)^2+\lambda_j^2) \qquad (6.16)$$

To study the convergence of $E\{\dot{\mathbf{V}}(n)\dot{\mathbf{V}}^{\mathbf{T}}(n)\}$, we consider Eq. (6.11) exponentially stable if the roots ρ_j of $\det[\mathbf{K}-\rho\mathbf{I}]$ lie inside the unit circle [38]. Therefore,

$$\det[\mathbf{K}-\rho\mathbf{I}]=\det[\mathbf{D}-\rho\mathbf{I}]\,\det[\mathbf{A}-\rho\mathbf{I}] \qquad (6.17)$$

where we can see from Eq.(6.15) that

$$\det[\mathbf{D}-\rho\mathbf{I}]=\prod_{j=1}^{N}(1-\mu(\gamma+\lambda_j)-\rho) \qquad (6.18)$$

Eqs.(6.17),(6.18) show that the first condition for convergence of $\mathbf{Z}(n)$ is the same as that found from convergence in the mean, Eq. (6.5). Also, knowing that $\det[\mathbf{\Gamma}\mathbf{\Gamma}^{\mathbf{T}}]=0$, then we get [33]

$$\det[\mathbf{A}-\rho\mathbf{I}] \;=\; \det[\mathrm{diag}\{a_1-\rho,\;a_2-\rho,\ldots,a_N-\rho\}\,](1+\mu^2\mathbf{\Gamma}^{\mathbf{T}}$$

$$\text{diag}\{a_1-\rho,\ a_2-\rho,\ldots,a_N-\rho\}^{-1}\Gamma\ )$$

$$= \left[\prod_{j=1}^{N}(a_j-\rho)\right]\left[1+\mu^2\sum_{j=1}^{N}\frac{\lambda_j^2}{a_j-\rho}\right] \tag{6.19}$$

Following the approach in [31], we can show that necessary and sufficient conditions for the roots of $\det[\mathbf{A}-\rho\mathbf{I}]$ to be inside the unit circle are

$$\left.\begin{array}{rcl} -1 & < & a_j < 1\ ,\ \ j=1,2,\ldots,N \\ 1 & + & \mu^2\sum_{j=1}^{N}\frac{\lambda_j^2}{a_j-1} > 0 \end{array}\right\} \tag{6.20}$$

Recalling a_j in Eq. (6.20) defined in Eq.(6.16) as a function of μ, we can see that it is a convex function and is greater than zero $\forall\,\mu$ and γ since it has a minimum nonnegative value of $\frac{\lambda_j^2}{(\gamma+\lambda_j)^2+\lambda_j^2}$ at $\mu=\frac{\gamma+\lambda_j}{(\gamma+\lambda_j)^2+\lambda_j^2}$. This leaves us with the following conditions

$$a_j < 1\ ,\ \ j=1,2,\ldots,N \tag{6.21}$$

and

$$1+\mu^2\sum_{j=1}^{N}\frac{\lambda_j^2}{a_j-1} > 0 \tag{6.22}$$

The condition in Eq.(6.21) results in

$$\mu(\ \mu(\ (\gamma+\lambda_j)^2+\lambda_j^2)-2(\gamma+\lambda_j)\ ) < 0 \tag{6.23}$$

Using Eq.(6.5), $\mu > 0$, then from Eq.(6.23) we get

$$0 < \mu < \frac{2(\gamma+\lambda_j)}{(\gamma+\lambda_j)^2+\lambda_j^2}\ ,\quad j=1,\ 2,\ldots,N \tag{6.24}$$

Eq. (6.22) leads to a second condition on μ for convergence in the mean square

$$\sum_{j=1}^{N}\frac{\mu\lambda_j^2}{2(\gamma+\lambda_j)-\mu(\ (\gamma+\lambda_j)^2+\lambda_j^2\ )}-1\ \ <\ 0 \tag{6.25}$$

To convert this condition into a direct bound on μ, we denote the left-hand side of Eq.(6.25) by $\chi(\mu)$ and note that $\chi(\mu)$ is a monotonically increasing function of μ since

$$\frac{\partial \chi(\mu)}{\partial \mu} = \sum_{j=1}^{N} \frac{2\lambda_j^2(\gamma + \lambda_j)}{(\ 2(\gamma + \lambda_j) - \mu(\ (\gamma + \lambda_j)^2 + \lambda_j^2)\ )^2} \tag{6.26}$$

Moreover, $\chi(\mu)$ has poles at $\rho_j = \frac{2(\gamma+\lambda_j)}{(\gamma+\lambda_j)^2+\lambda_j^2}$, $j = 1,\ 2,\ \ldots,\ N$, and is equal to -1 at $\mu = 0$, and $\lim_{\mu\to\infty} \chi(\mu) < 0$. If we write $\chi(\mu) = 0$ as

$$\chi(\mu) = \prod_{i=1}^{N}(\mu - \mu_i) = 0 \tag{6.27}$$

where $\mu_i,\ i = 1, \ldots,\ N$ are the roots of $\chi(\mu)$, and since $0 < \lambda_N \leq \lambda_{N-1} \leq \cdots \leq \lambda_1$, then $\chi(\mu)$ has the form shown in Fig. 6.1. It is obvious from Fig. 6.1 that

$$0 < \mu_1 < \rho_1 < \mu_2 < \rho_2 < \cdots < \mu_N < \rho_N \tag{6.28}$$

Therefore, for the conditions in Eqs.(6.21) and (6.22) to hold, μ should be bounded by

$$0 < \mu < \mu_1 \tag{6.29}$$

As such, the condition in Eq.(6.25) has provided us with a more stringent bound on μ as given in Eq.(6.29), than the one in Eq. (6.24), since $\mu_1 < \rho_1$. A closed form expression for μ_1 cannot be found. However, we will take advantage of a theorem established in [48, Appendix C, theorem 13] to obtain a tight lower bound on μ_1. For Eq.(6.27), where $0 < \mu_1 < \mu_2 < \ldots < \mu_N$, the smallest root μ_1 is lower

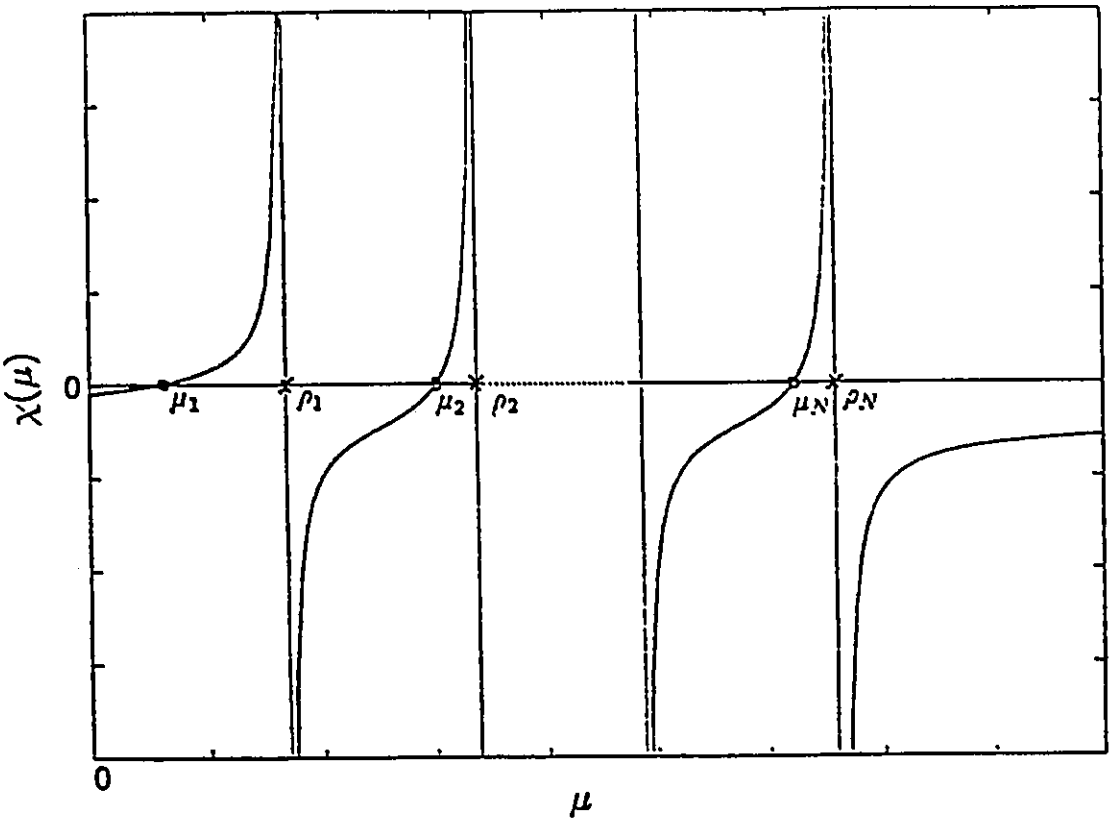

Fig. 6.1 A plot illustrating a general form of $\chi(\mu)$.

bounded by

$$\mu_1 \geq \frac{N}{s_{-1} + \sqrt{(N-1)(Ns_{-2} - s_{-1}^2)}} \tag{6.30}$$

where

$$s_{-1} = \sum_{i=1}^{N} \frac{1}{\mu_i} \tag{6.31}$$

and

$$s_{-2} = \sum_{i=1}^{N} \left(\frac{1}{\mu_i}\right)^2 = \left(\sum_{i=1}^{N} \frac{1}{\mu_i}\right)^2 - \sum_{i \neq}^{N} \sum_{j}^{N} \frac{1}{\mu_i} . \frac{1}{\mu_j} \tag{6.32}$$

Rewriting the equation $\chi(\mu) = 0$ as

$$\begin{aligned} &\prod_{i=1}^{N} \left(\frac{1}{\mu} - \frac{1}{\mu_i}\right) = 0 \\ &\left(\frac{1}{\mu}\right)^N - b_1 \left(\frac{1}{\mu}\right)^{N-1} + b_2 \left(\frac{1}{\mu}\right)^{N-2} \cdots + b_N(-1)^N = 0 \end{aligned} \tag{6.33}$$

then expanding the first part of Eq.(6.33) and comparing it with the second part of the same equation, we obtain

$$b_1 = \sum_{i=1}^{N} \frac{1}{\mu_i} \tag{6.34}$$

and

$$b_2 = \frac{1}{2} \sum_{i \neq}^{N} \sum_{j}^{N} \frac{1}{\mu_i} . \frac{1}{\mu_j} \tag{6.35}$$

Consequently,

$$\left.\begin{aligned} s_{-1} &= b_1 \\ s_{-2} &= b_1^2 - 2b_2 \end{aligned}\right\} \tag{6.36}$$

Now by comparing the left-hand side of the second part of Eq.(6.33) with the left-hand side of Eq.(6.25), it can be shown, after some manipulation

$$b_1 = \sum_{i=1}^{N} \frac{2\lambda_i^2 + (\gamma + \lambda_i)^2}{2\gamma + 2\lambda_i} \tag{6.37}$$

and

$$b_2 = \sum_{i\neq}^{N} \sum_{j}^{N} \frac{(3\lambda_i^2 + (\gamma + \lambda_i)^2)(\lambda_j^2 + (\gamma + \lambda_j)^2)}{2(2\gamma + 2\lambda_i)(2\gamma + 2\lambda_j)} \tag{6.38}$$

Substituting Eq.(6.36) in (6.30) results in

$$\mu_1 \geq \frac{N}{b_1 + \sqrt{b_1^2(N-1)^2 - 2b_2N(N-1)}} = \mu^* \tag{6.39}$$

Thus, to ensure convergence in the mean square, μ should be bounded by

$$0 < \mu < \mu^* \tag{6.40}$$

To make Eq.(6.40) more practical, we note that

$$\mu^* \geq \frac{1}{b_1} \tag{6.41}$$

then,

$$0 < \mu < \frac{1}{\sum_{j=1}^{N} \frac{2\lambda_j^2 + (\gamma + \lambda_j)^2}{2\gamma + 2\lambda_j}} \tag{6.42}$$

Eq.(6.42) provides a bound on the step size μ to guarantee convergence of the MSE of the LMS algorithm with leakage γ. The bounds on the step size μ in Eqs.(6.40) and (6.42) matches the ones derived in [31] for the LMS algorithm when γ is set to zero. From Eq.(6.42), if we put $\gamma = 0$, then we obtain the following condition on

the step size that ensures the MSE convergence of the LMS algorithm (note that [31] uses 2μ as a step size and, hence, the upper bound in the following equation is different from that in [31] by a factor of 2)

$$0 < \mu_{LMS} < \frac{2}{3\sum_{j=1}^{N} \lambda_j} \tag{6.43}$$

To provide an insight into the difference between the bound on the step size of the Leaky LMS algorithm in Eq.(6.42) and that of the LMS algorithm in Eq.(6.43), we note that in the event of a small γ such that $\gamma^2 \ll \lambda_i(3\lambda_i + 2\gamma)$, then the upper bound on μ in Eq.(6.42) for the Leaky LMS algorithm can be written as $\frac{2}{3\sum_{i=1}^{N} \lambda_i \left(\frac{\lambda_i + \frac{2}{3}\gamma}{\lambda_i + \gamma}\right)}$. Since $\frac{\lambda_i + \frac{2}{3}\gamma}{\lambda_i + \gamma} < 1$, then we observe that the Leaky LMS algorithm allows for a larger step μ than the LMS algorithm to be used to guarantee MSE convergence of the algorithm.

6.1.2 Steady State MSE

We proceed to determine the steady state performance of the MSE for the Leaky LMS. Recall that

$$\begin{aligned} \epsilon(n) &= \epsilon_{min} + E\{\hat{\mathbf{V}}^{\mathbf{T}}(n)\mathbf{\Lambda}\hat{\mathbf{V}}(n)\} \\ &= \epsilon_{min} + \mathbf{\Gamma}^{\mathbf{T}}\mathbf{Z_1}(n) \end{aligned} \tag{6.44}$$

We define the vectors $\hat{\mathbf{Y}} = [\hat{p_1^2}, \hat{p_2^2}, \ldots, \hat{p_N^2}]^T$, and $\mathbf{G}(n) = \mathbf{B}\mathbf{Z_2}(n)$. From Eqs.(6.9), (6.10) , and (6.11), the second moment vector $\mathbf{Z_1}(n+1)$ can be represented as

$$\mathbf{Z_1}(n+1) = \mathbf{A}\mathbf{Z_1}(n) + \mathbf{G}(n) + \mu^2\epsilon_{min}\mathbf{\Gamma} + \mu^2\gamma^2(\mathbf{\Lambda}^{-1})^2\hat{\mathbf{Y}} \tag{6.45}$$

The excess MSE, $\epsilon_{ex}(n)$, is defined as

$$\begin{aligned} \epsilon_{ex}(n) &= \epsilon(n) - \epsilon_{min} \\ &= \mathbf{\Gamma}^{\mathbf{T}}\mathbf{Z_1}(n) \end{aligned} \tag{6.46}$$

Assuming that μ is chosen such that the convergence conditions of the mean weight and the MSE is guaranteed, then the steady state $\epsilon_{ex}(\infty)$ is given by

$$\epsilon_{ex}(\infty) = \mathbf{\Gamma}^{\mathbf{T}}\mathbf{Z_1}(\infty) \tag{6.47}$$

where from Eq. (6.45)

$$\begin{aligned} \mathbf{Z_1}(\infty) &= [\,\mathbf{I} - \mathbf{A}\,]^{-1}\,[\,\mathbf{G}(\infty) + \mu^2\epsilon_{min}\mathbf{\Gamma} \\ &\quad + \mu^2\gamma^2(\mathbf{\Lambda}^{-1})^2\hat{\mathbf{Y}}\,] \end{aligned} \tag{6.48}$$

Contrary to the LMS algorithm, where the increase in the steady state MSE (or alternatively the steady state excess MSE) is solely attributed to the fluctuations of its coefficients around the optimal ones, the increase in the steady state MSE for the Leaky LMS algorithm, which is described by Eq(6.47), is due to the bias in the Leaky LMS algorithm coefficients, *as well as* the fluctuations of those coefficients around their mean values. To make a clear distinction in the resultant expression of $\epsilon_{ex}(\infty)$ between these two sources of error, we first define the bias vectors $\mathbf{L} = \acute{\mathbf{W}}^{*} - \acute{\mathbf{W}}_{\mathbf{L}} = [l_1, l_2, \ldots, l_N]^T$, and $\hat{\mathbf{L}} = [l_1^2, l_2^2, \ldots, l_N^2]^T$ where $\acute{\mathbf{W}}_{\mathbf{L}} = \mathbf{Q}^{\mathbf{T}}\mathbf{W}_{\mathbf{L}}$ and $\mathbf{W}_{\mathbf{L}}$ is the steady state mean weight vector of the Leaky LMS algorithm. From

Eq.(6.2), it is easy to see that $\mathbf{W_L} = E\{\mathbf{W}(\infty)\} = (\mathbf{R}+\gamma\mathbf{I})^{-1}\mathbf{P}$. Accordingly, we find that

$$l_j = \frac{\gamma p_j}{\lambda_j(\lambda_j+\gamma)} \quad , j = 1, 2, \ldots, N \tag{6.49}$$

Further, we notice that the steady state MSE ϵ_{Lmin} attained by the Leaky LMS algorithm (obtained by substituting $\mathbf{W_L}$ for $\mathbf{W}(n)$ in Eq.(6.7)) is given by

$$\begin{aligned} \epsilon_{Lmin} &= \epsilon_{min} + \mathbf{L^T \Lambda L} \\ &= \epsilon_{min} + \mathbf{\Gamma^T \hat{L}} \end{aligned} \tag{6.50}$$

(where ϵ_{min} as defined in Eq.(6.8) is the minimum MSE achievable by the LMS algorithm). Now, we rewrite Eq.(6.48) as

$$\begin{aligned} \mathbf{Z_1}(\infty) &= [\,\mathbf{I}-\mathbf{A}\,]^{-1}\,[\,\mathbf{G}(\infty) + \mu^2\epsilon_{Lmin}\mathbf{\Gamma} \\ &\quad + \mu^2\gamma^2(\mathbf{\Lambda}^{-1})^2\hat{\mathbf{Y}} - \mu^2\mathbf{\Gamma}\mathbf{\Gamma}^{\mathbf{T}}\hat{\mathbf{L}}\,] \end{aligned} \tag{6.51}$$

In Eq. (6.51), one needs to evaluate $[\mathbf{I}-\mathbf{A}]^{-1}$ and $\mathbf{G}(\infty)$. We can write $[\mathbf{I}-\mathbf{A}]$ as[2]

$$\begin{aligned} [\,\mathbf{I}-\mathbf{A}\,]^{-1} &= \Big[\,[2\mu\gamma-\mu^2\gamma^2]\mathbf{I} + 2\mu\mathbf{\Lambda}[\mathbf{I}-\mu\gamma\mathbf{I}] - 2\mu^2\mathbf{\Lambda}^2\,\Big]^{-1} \,.\,(\,\mathbf{I} + \\ &\quad \mu^2\mathbf{\Gamma}\mathbf{\Gamma}^{\mathbf{T}}\,.(1 - \\ &\quad \mu^2\mathbf{\Gamma}^{\mathbf{T}}\Big[\,[2\mu\gamma-\mu^2\gamma^2]\mathbf{I} + 2\mu\mathbf{\Lambda}[\mathbf{I}-\mu\gamma\mathbf{I}] - 2\mu^2\mathbf{\Lambda}^2\,\Big]^{-1}\mathbf{\Gamma})^{-1}\,. \\ &\quad \Big[\,[2\mu\gamma-\mu^2\gamma^2]\mathbf{I} + 2\mu\mathbf{\Lambda}[\mathbf{I}-\mu\gamma\mathbf{I}] - 2\mu^2\mathbf{\Lambda}^2\,\Big]^{-1}\,) \end{aligned} \tag{6.52}$$

[2]To find $[\mathbf{I}-\mathbf{A}]^{-1}$, we have used the matrix inverse lemma, which is given by
$(\mathbf{F}+\mathbf{bc}^{\mathbf{T}})^{-1} = \mathbf{F}^{-1} - \mathbf{F}^{-1}\mathbf{b}(1+\mathbf{c}^{\mathbf{T}}\mathbf{F}^{-1}\mathbf{b})^{-1}\mathbf{c}^{\mathbf{T}}\mathbf{F}^{-1}$
where F is a nonsingular matrix, b, and c are vectors.

Using Eq. (6.4) and (6.14), we can solve for $\mathbf{G}(\infty)$ whose jth component is given by

$$G_j(\infty) = \frac{2\mu\gamma^2\acute{p}_j^2(1-\mu(\gamma+\lambda_j))}{\lambda_j^2(\gamma+\lambda_j)} \tag{6.53}$$

If we use Eqs. (6.49), (6.51), (6.52), and (6.53) in (6.47), we obtain after some manipulation

$$\epsilon_{ex}(\infty) = \frac{\sum_{j=1}^{N} \frac{\lambda_j^2 G_j(\infty)+\mu^2\gamma^2(\acute{p}_j^2-\lambda_j^3 r-\acute{p}_j^2 k_j)}{(2(\gamma+\lambda_j)-\mu((\gamma+\lambda_j)^2+\lambda_j^2))\mu\lambda_j} + \sum_{j=1}^{N} \frac{\mu\lambda_j^2(\epsilon_{Lmin}+\lambda_j l_j^2)}{2(\gamma+\lambda_j)-\mu((\gamma+\lambda_j)^2+\lambda_j^2)}}{1-\sum_{j=1}^{N} \frac{\mu\lambda_j^2}{2(\gamma+\lambda_j)-\mu((\gamma+\lambda_j)^2+\lambda_j^2)}} \tag{6.54}$$

where $r = \sum_{i=1}^{N} \frac{\acute{p}_j^2}{\lambda_j(\gamma+\lambda_j)^2}$ and $k_j = \frac{\lambda_j^2}{(\gamma+\lambda_j)^2}$. Eq.(6.54) completely characterizes the steady state excess MSE of the Leaky LMS. Observe that the expression in Eq.(6.54) reduces to that derived for the LMS algorithm in [31] when[3] $\gamma = 0$

$$\epsilon_{LMSex}(\infty) = \frac{\sum_{j=1}^{N} \frac{\mu_{LMS}\lambda_j}{2-2\mu_{LMS}\lambda_j}}{1-\sum_{j=1}^{N} \frac{\mu_{LMS}\lambda_j}{2-2\mu_{LMS}\lambda_j}} \epsilon_{min} \tag{6.55}$$

To obtain a simple estimate of the relative level of misadjustment for the Leaky LMS algorithm, we note that for small μ and γ such that $\mu((\gamma+\lambda_j)^2+\lambda_j^2) \ll 2(\gamma+\lambda_j)$, $\mu\sum_{j=1}^{N} \frac{\lambda_j^2}{\gamma+\lambda_j} \ll 2$, $\mu(\gamma+\lambda_j)(\acute{p}_j^2+\lambda_j^3 r+\acute{p}_j^2 k_j) \ll 2\acute{p}_j^2$, and in the event that $\lambda_j l_j^2 \ll \epsilon_{Lmin}$, Eq.(6.54) can be approximated as

$$\epsilon_{ex}(\infty) \approx \gamma^2 \sum_{j=1}^{N} \frac{\acute{p}_j^2}{\lambda_j(\gamma+\lambda_j)^2} + \frac{\mu}{2} \sum_{j=1}^{N} \frac{\lambda_j^2}{\gamma+\lambda_j} \epsilon_{Lmin} \tag{6.56}$$

[3]Note that in [31] they use 2μ in the LMS recursion equation which accounts for the difference between Eq.(6.55) and that in [31].

If we substitute $\gamma = 0$ in Eq.(6.56), then we obtain the standard LMS algorithm steady state excess MSE [113]

$$\epsilon_{LMSex}(\infty) \approx \frac{\mu_{LMS}}{2} \sum_{j=1}^{N} \lambda_j \epsilon_{min} \tag{6.57}$$

Note that Eq.(6.54) (as well as Eq.(6.56)) quantifies the impact of all input parameters: input signal statistics, crosscorrelation between the input signal and the desired signal, step size value, and the leakage factor on the Leaky LMS algorithm steady state performance. More specifically, Eq.(6.54) (and Eq.(6.56)) shows that the steady state excess MSE $\epsilon_{ex}(\infty)$ is due to two sources. The first is due to the fluctuations of the algorithm coefficient vector about its steady state mean solution $\mathbf{W_L}$. Indeed, it is shown in Appendix A that the increase in Leaky LMS algorithm steady state MSE caused by the fluctuations of its coefficients around the steady state biased solution is

$$\epsilon_{Lex}(\infty) = \frac{\sum_{j=1}^{N} \frac{\mu\lambda_j^2(\epsilon_{Lmin}+\lambda_j l_j^2)}{2(\gamma+\lambda_j)-\mu((\gamma+\lambda_j)^2+\lambda_j^2)}}{1-\sum_{j=1}^{N} \frac{\mu\lambda_j^2}{2(\gamma+\lambda_j)-\mu((\gamma+\lambda_j)^2+\lambda_j^2)}} \tag{6.58}$$

Note that $\epsilon_{Lex}(\infty)$ in Eq.(6.58) matches exactly the second term in Eq.(6.54). Hence, the second term in Eq.(6.54) (and in Eq.(6.56)) serves as a separate measure of the contribution of the coefficients fluctuations to the algorithm steady state excess MSE. Again, for $\gamma = 0$ (LMS algorithm case), Eq.(6.58) reduces to ϵ_{LMSex} in Eq.(6.55).

In the analysis in Appendix A, the bias was initially removed from the coefficients. This led to identifying Eq.(6.58) (i.e, second term in Eq.(6.54)) as being a

result of the fluctuations around the biased solution. Eq.(6.54) provides the full error. Thus, it follows that the first term is the error component due to the bias. This interpretation of the source of the first term in Eq.(6.54) (and in Eq.(6.56)) is even more logical when we note that it is independent of ϵ_{Lmin} and is significantly dependent on $\acute{\mathbf{P}}$, the crosscorrelation between the input and the desired signals.

6.2 Simulations

Simulation results are presented here to verify the accuracy of the theoretical results derived in this chapter.

6.2.1 System Identification

1- *White input signal*

In the first example, zero-mean white Gaussian noise of ϵ_{min} variance is added to the desired signal $d(n)$. The unknown system is an FIR system with 4 coefficients $\{0.1, 0.3, 0.5, 0.3\}$ and the FIR adaptive filter has the same dimension N=4. Both the system and the adaptive filter are excited by a zero-mean uncorrelated Gaussian signal of unity variance. Results are obtained by averaging over 600 independent runs. The Leaky LMS algorithm is used with the adaptation step size $\mu = 0.01$. In Fig. 6.2 we plotted the evolution of the excess MSE $\epsilon_{ex}(n)$ obtained from theoretical analysis via Eq.(6.46), and from simulation results. The leakage factor is $\gamma = 0.01$ and $\epsilon_{min} = 0.001$. It can be seen that our analysis agrees well with

empirical results. Note that the independent assumption of $\mathbf{W}(n)$ and $\mathbf{X}(n)$, employed in the analysis, is true in this case since $x(n)$ is a white signal.

Table 6.1 compares theoretical results for the steady state excess MSE $\epsilon_{ex}(\infty)$ obtained from Eq.(6.54) and (6.56) with results of simulation for different values of γ and ϵ_{min}. It can be seen that Eq.(6.54) predicts very closely the actual $\epsilon_{ex}(\infty)$ as obtained from simulation. Since the assumptions beyond Eq.(6.56) hold for this particular example, Eq.(6.56) could provide an approximated measure to the expected level of $\epsilon_{ex}(\infty)$. From Table 6.1, we can conclude that $\epsilon_{ex}(\infty)$ of the Leaky LMS algorithm, in general, is insensitive to ϵ_{min}. This can be seen from Eq.(6.54) given that the first term is dominant .

Table 6.1

Comparison of theoretical and experimental Leaky LMS algorithm steady state excess MSE for the first example

		Predicted $\epsilon_{ex}(\infty)$ (dB)		Measured $\epsilon_{ex}(\infty)$ (dB)
γ	ϵ_{min}	Eq.(6.54))	Eq.(6.56)	
0	0.001	-46.86	-46.99	-47.67
0.001	0.001	-46.78	-46.90	-47.42
0.01	0.001	-41.90	-42.00	-41.8
0.1	0.001	-24.27	-24.36	-24.22
1	0.001	-9.53	-9.58	-9.54
1	0.01	-9.53	-9.57	-9.53
1	0.1	-9.50	-9.54	-9.47
1	1	-9.15	-9.20	-8.98
1	2	-8.80	-8.85	-8.51

2-*Correlated input signal*

In the second example, we use similar model to that utilized in the previous example, except that both the system and the adaptive filter are excited by a correlated signal generated by

$$x(n) = 0.9x(n-1) + a(n) \tag{6.59}$$

where $a(n)$ is a zero-mean, uncorrelated Gaussian noise of unity variance. The Leaky LMS algorithm is used with a step size $\mu = 0.001$. The evolution of the excess MSE $\epsilon_{ex}(n)$ is depicted in Fig. 6.3 for $\gamma = 0.01$, and $\epsilon_{min} = 0.001$. Fig. 6.3

also compares the theoretical prediction of the Leaky LMS algorithm for $\epsilon_{ex}(n)$, described in Eq.(6.46), to results of simulation. Note that the common independence assumption of $\mathbf{X}(n)$ and $\mathbf{W}(n)$ is not valid in this case. This is because $\mathbf{X}(n)$ and $\mathbf{X}(n-1)$ are correlated, and since $\mathbf{W}(n)$ is dependent on $\mathbf{X}(n-1)$, then $\mathbf{W}(n)$ is correlated with $\mathbf{X}(n)$. In spite of that, simulation and theoretical results in Fig. 6.3 match very well. This is also confirmed in Table 6.2, where theoretical results of $\epsilon_{ex}(\infty)$ calculated from Eq.(6.54) and Eq.(6.56) are compared to simulation results using different values of γ and ϵ_{min}. Once again, Table 6.2 shows well agreement between analytical and empirical results for the correlated input signal case.

Table 6.2

Comparison of theoretical and experimental Leaky LMS algorithm steady state excess MSE for the second example

		Predicted $\epsilon_{ex}(\infty)$ (dB)		Measured $\epsilon_{ex}(\infty)$ (dB)
γ	ϵ_{min}	Eq.(6.54))	Eq.(6.56)	
0	0.001	-49.67	-49.78	-49.59
0.001	0.001	-49.51	-49.72	-49.51
0.01	0.001	-46.35	-46.04	-46.33
0.1	0.001	-30.19	-29.21	-30.10
1	0.001	-14.54	-12.52	-14.30
1	0.01	-14.53	-12.53	-14.29
1	0.1	-14.42	-12.46	-14.19
1	1	-13.47	-11.85	-13.28
1	2	-12.63	-11.27	-12.55

6.2.2 Channel Equalization

The third example is based on the example in [46]. The linear dispersive channel is assumed to have a raised cosine impulse response

$$h(k) = \frac{1}{2}\left(1 + cos(\frac{2\pi(k-1)}{s})\right), \quad k = 0,\ 1,\ 2 \tag{6.60}$$

where s is used to control the eigenvalue spread of the autocorrelation matrix $\mathbf{R}$. The input to the channel is a binary random sequence $b(n) = \pm 1$ of equal

probability. The 11 taps adaptive equalizer is excited by

$$x(n) = \sum_{k=0}^{2} h(k)b(n-k) + v(n) \tag{6.61}$$

where $v(n)$ is a zero-mean white Gaussian noise of 0.001 variance, independent of $b(n)$. The desired signal is $d(n) = b(n-7)$ [46]. The leakage factor γ is set to 0.1. Furthermore, we choose $s = 4$, resulting in an eigenvalue spread of 716.11. Results are obtained by averaging over 200 ensemble members. It is evident from Fig. 6.4 and Fig. 6.5 that the measured μ_1 is 0.056 for which the weight vector mean converges and the MSE does not. For μ less than μ_1, e.g. $\mu = 0.05$, convergence of both the mean and the mean square of the weight vector is guaranteed. In Table 6.3, we compare theoretical results for μ^*, the lower bound on μ_1 obtained from Eq.(6.39), and the lower bound $\frac{1}{b_1}$ on μ^* in Eq.(6.41) to results of simulation. We also include in Table 6.3 the upper bound on the step size μ in Eq.(6.5) given by the mean convergence. Note that though Eq.(6.39) does not predict exactly with a lower bound on the measured μ_1, it nevertheless provides a good indication of its value. Moreover, Table 6.3 illustrates that $\frac{1}{b_1}$ in Eq.(6.41) represents a good lower bound on the measured μ_1. Since $\frac{1}{b_1}$ is the upper bound of Eq.(6.42), then the selection of μ according to Eq.(6.42) can always assure convergence in the mean square. We note that the upper bound on μ predicted by Eq.(6.5) is much larger than the measured step size μ_1, hence, Eq.(6.5) provides a loose bound on the region from which the step size of the Leaky LMS algorithm can be selected to

ensure MSE convergence.

Table 6.3

Comparison of theoretical and experimental Leaky LMS algorithm stability bounds for example three

			Predicted			Measured
s	Eigenvalue Spread	γ	Eq.(6.39) μ^*	Eq.(6.41) $\frac{1}{b_1}$	Eq.(6.5) upper bound on μ	μ_1
4	716.11	0.1	0.0859	0.0408	0.5031	0.056
3.5	46.82	0.1	0.1037	0.0473	0.6308	0.07
3.1	11.123	0.1	0.1245	0.0535	0.8077	0.099
2.9	6.782	0.1	0.137	0.0565	0.9392	0.105

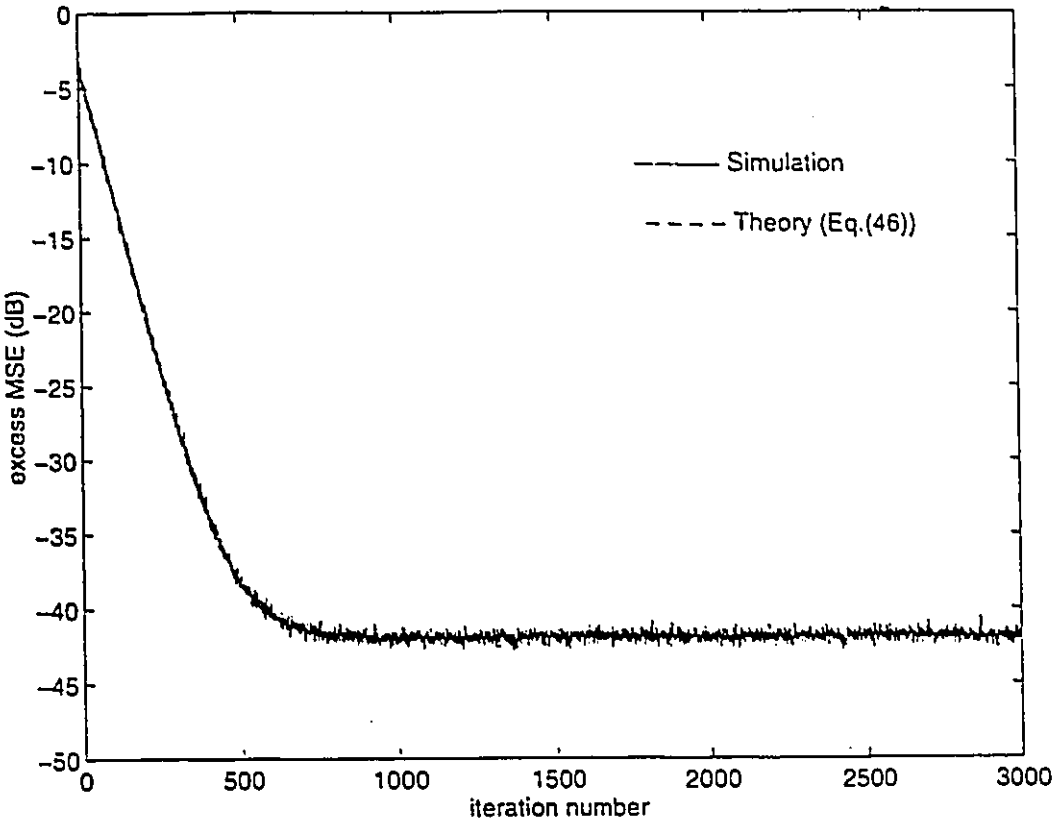

Fig. 6.2 Comparison of the theoretical and simulation results of the excess MSE $\epsilon_{ex}(n)$ of the Leaky LMS algorithm for the first example.

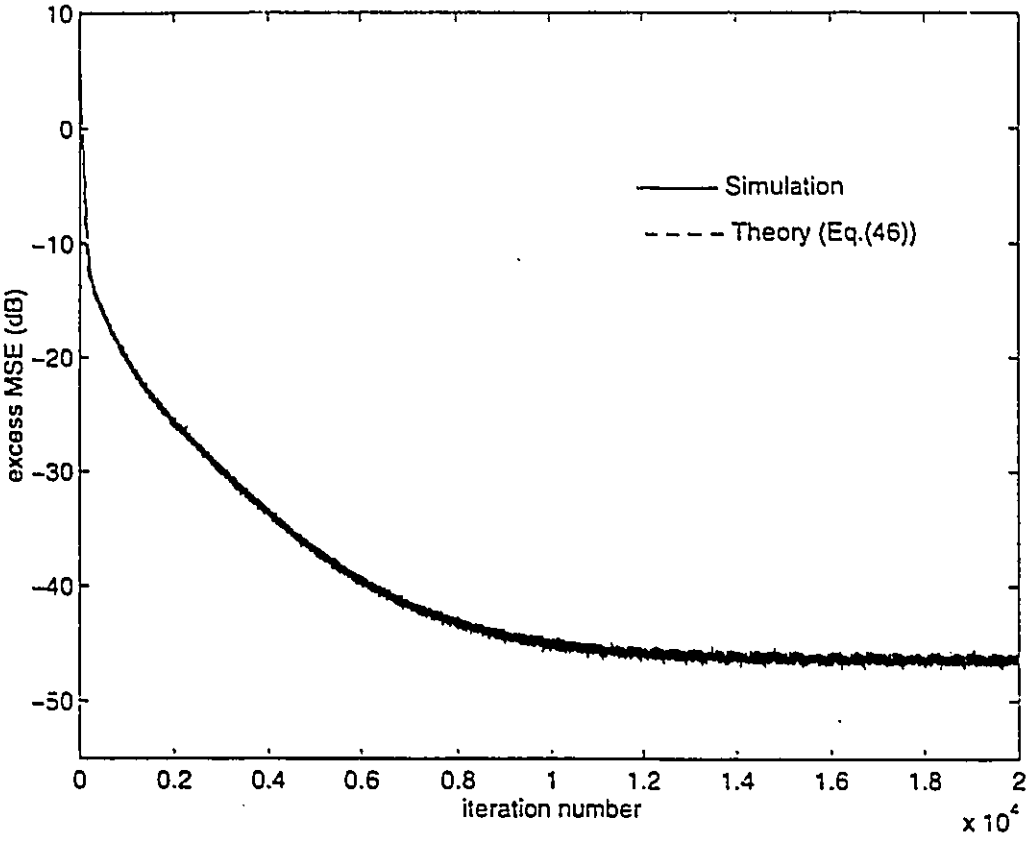

Fig. 6.3 Comparison of the theoretical and simulation results of the excess MSE $\epsilon_{ex}(n)$ of the Leaky LMS algorithm for the second example.

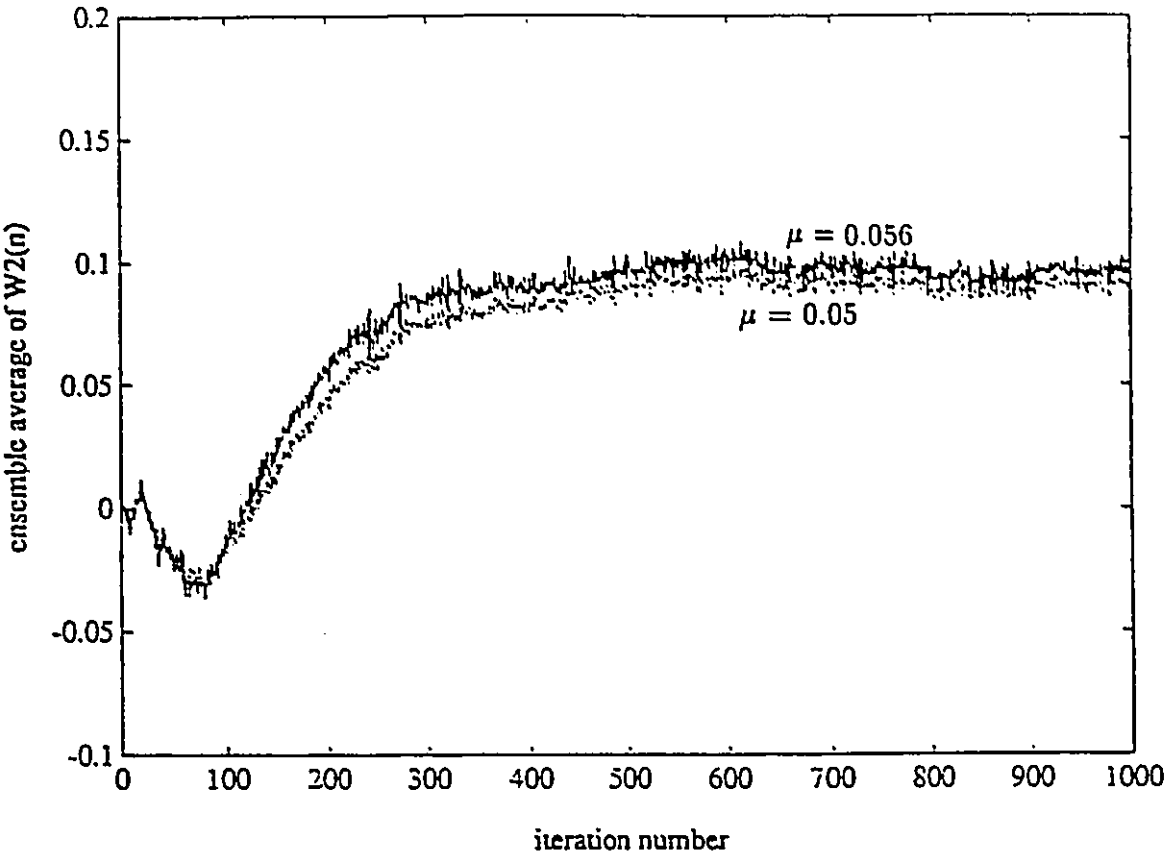

Fig. 6.4 Ensemble average of the second component of $\mathbf{W}(n)$ for the channel equalization example.

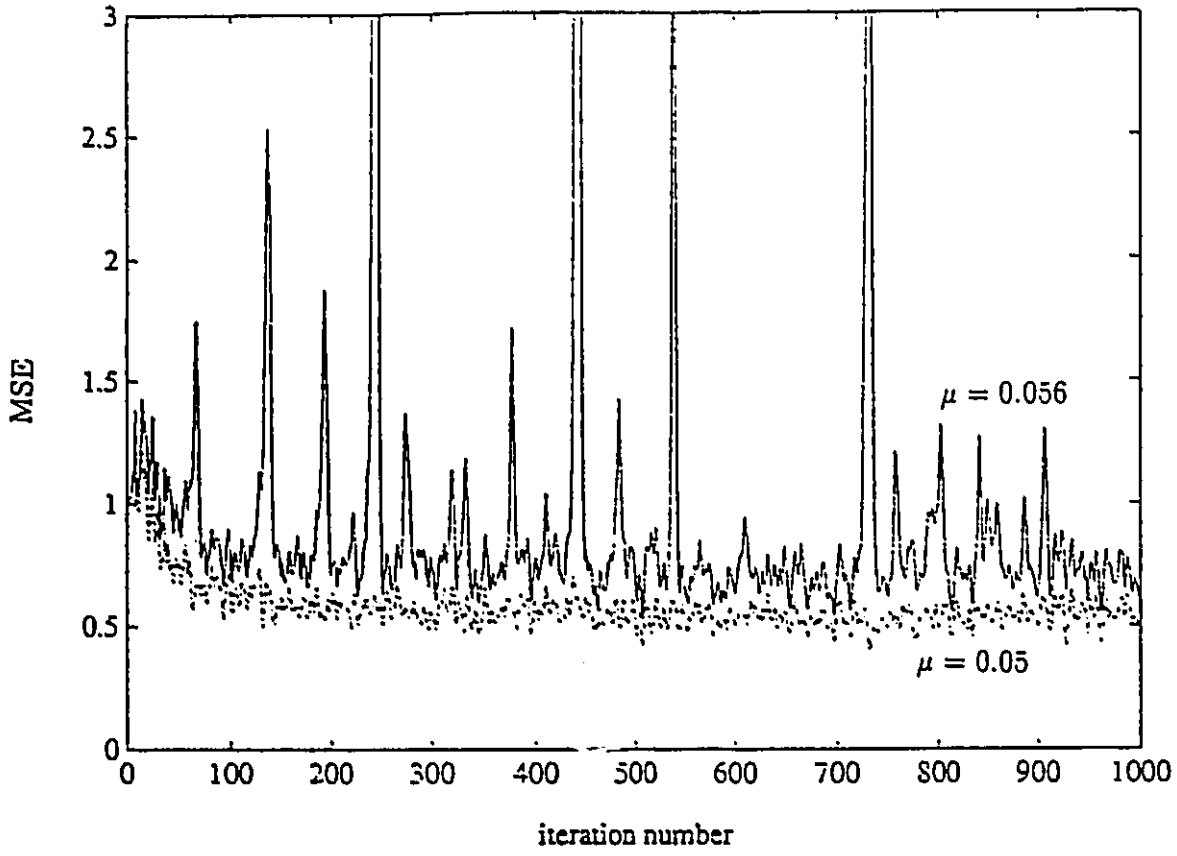

Fig. 6.5 MSE for the channel equalization exam

6.3 Conclusion

In this chapter, convergence properties of the mean-square error for the Leaky LMS algorithm have been studied. This was done when the signals involved are zero-mean, stationary Gaussian. An expression for the second moment of the weight vector, which is the key to MSE analysis, was obtained. Then, we presented a direct stability bound on the step size for MSE convergence. It has been illustrated that the upper bound on the step size for the Leaky LMS algorithm (that guarantees MSE convergence) exceeds that for the LMS algorithm by an amount that depends on the value of the leakage factor γ. We also presented exact and approximate expressions for the Leaky LMS algorithm steady sate excess MSE. The approximate expression provides a clear distinction between the error due to bias and the error due to fluctuations of the coefficients about the solution of the Leaky LMS algorithm. It should be noted that all results presented here for the Leaky LMS algorithm are identical to those of the LMS algorithm when $\gamma = 0$ [31]. The analytical results were shown to agree well with simulation results.

Chapter 7

A Modified Echo Canceler for Adaptive Hybrids with no Bursting

In a telephone system, the conversion between the two-wire subscriber line and the four-wire trunk is done inside the hybrid. The functional role of the hybrid is to couple all signal energy arriving from the far-end speaker into the two-wire line of the near-end speaker, and stop any leakage to the near-end transmission path. Due to impedance mismatch in the real hybrid, some energy leaks and is transferred via the near-end transmit line back to the far-end speaker as an echo. Such mismatches are usually unknown and may be time-varying due to aging, inaccurate component values, moisture, etc. Thus, traditionally adaptive filters have been used as echo cancelers.

The echo canceler tries to match the echo path impulse response thus generating an echo replica. The residual error resulting from subtracting this replica from the

sum of the near-end signal and the real echo, is used as a learning signal for the adjustment of the adaptive filter coefficients. Under double-talk conditions (both near-end and far-end speakers are talking) the learning signal become erroneous leading to coefficient divergence. Therefore, double-talk detectors are used in the echo canceler to freeze coefficients adaptation during double-talk periods to prevent echo canceler misbehavior.

Fig. 7.1 shows a system with a single adaptive hybrid. A robust performance of the echo canceler in Fig. 7.1 is obtained under typical conditions of persistently exciting input $x(n)$, i.e., the input signal should contain sufficient frequencies to allow the adaptive algorithm to correctly update all its coefficients [36]. Also, a reliable performance of the adaptive echo canceler depends considerably on the decorrelation of the input $x(n)$ from the near-end signal $v(n+1)$. Suppose that $v(n+1)$ is highly correlated with $x(n)$, the adaptive filter would be then driven by large error $r(n+1)$ that is highly correlated with its input thus attempts to cancel this large near-end signal. The inevitable result is a continuous growth in the filter coefficients (drifting) to produce an output equivalent to the near-end signal from a continuously decreasing input signal. The growth in filter coefficients eventually leads to the overall system becoming unstable and "bursting" with the error suddenly becoming very large in magnitude and rich in frequencies, as shown

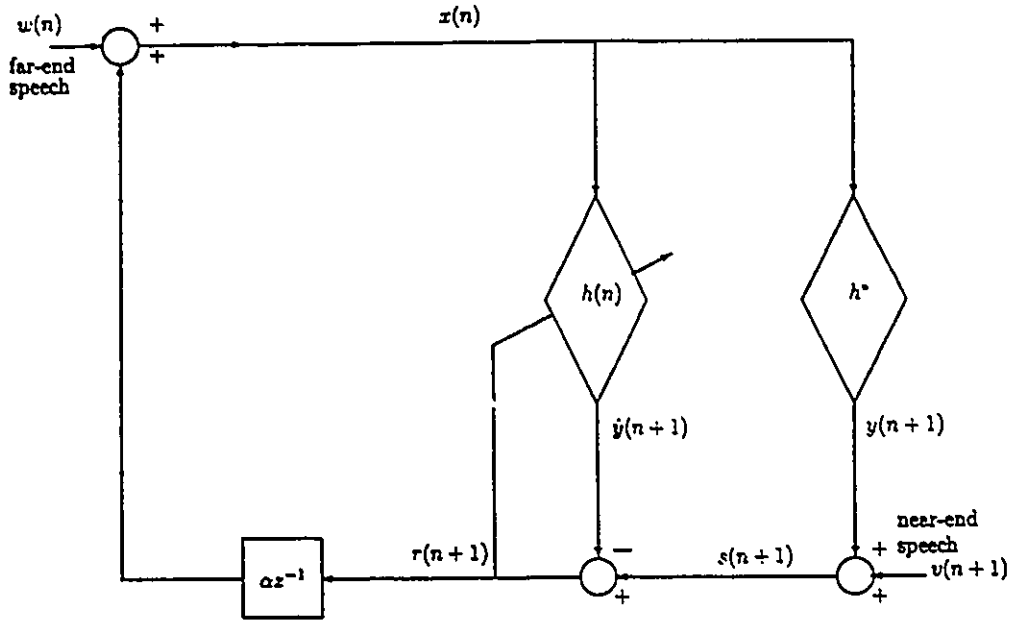

Fig. 7.1 A simple model of a single adaptive hybrid.

in Fig. 7.2. This "rich" signal stabilizes the system only so it can start drifting again towards instability if same input conditions continue. Practically, it is not likely for $w(n)$ and $v(n+1)$ to be correlated. However, correlation of $x(n)$ and $v(n+1)$ is more apparent when $w(n)$ is absent and $v(n+1)$ is not. This is largely attributed to the existence of the the feed-back path, where a large part of $x(n)$ will be an attenuated delayed version of $v(n+1)$. When $v(n+1)$ is a narrow-band signal, such correlation is substantial.

Several published works tried to propose possible ways to avert bursting. One attempt is to modify the adaptive LMS algorithm by the addition of a leakage term [36]. This approach is directed at preventing the growth of the adaptive filter coefficients to large values which eventually leads to bursting. For the case of a single parameter hybrid, an expression is derived in [36] for precise limits on the leakage factor for bursting avoidance. However, evaluation of this limit requires a priori knowledge of the disturbance level $v(n+1)$ and the hybrid coefficients. Additionally, it is not possible to extend the results to a more realistic situation of higher order hybrids. A large leakage factor can ensure the containment of parameter drift. However, this large factor is associated with an increase in MSE in normal operation and as such would not be acceptable.

In [24], a new test signal that approximately measures the correlation between

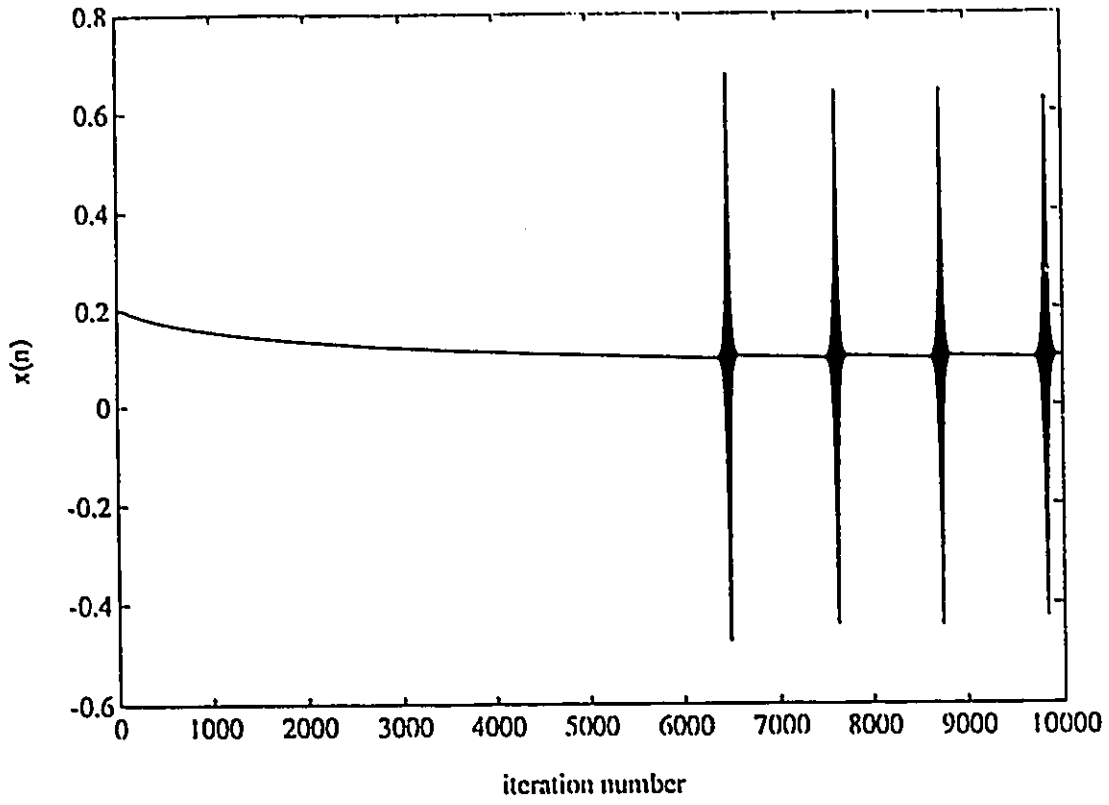

Fig. 7.2 The received signal $x(n)$ with a dc near-end signal $v(n+1)$ for the model in Fig. 7.1.

$x(n)$ and $v(n+1)$ is proposed for use in the double-talk detector. The test signal is directly related to the system pole $\alpha|h^* - h(n)|$, and accordingly parameter drift is halted if a predetermined threshold is surpassed and bursting can be avoided. The method was derived and successfully applied to a case of first order hybrid. Again, extension of the method to higher order cases is not obvious.

In [109], two approaches are proposed to ensure bursting elimination. Both are based on the properties of the crosscorrelation of $x(n)$ and $r(n+1)$ in a bursting environments. In the first approach an estimate of the crosscorrelation is obtained. For a pure near-end tone, the crosscorrelation was found to be flat and with a magnitude larger than 0.4. This unique property was used as a simple test to determine if conditions eventually leading to bursting exist. If so, adaptation is frozen. The major drawback of this approach is its substantial computational requirements which is approximately $8N$ multiplications, where N is the adaptive filter order. In the second approach, a simplified test for the flatness of the crosscorrelation is used. If the presence of a tone at the near-end is detected, then adaptation of the near-end echo canceler should be prohibited. The second approach requires significantly less complexity than the first one, but needs careful selection of its terms. All the above mentioned approaches assume that $v(n+1)$ is a pure tone signal for simplicity. However, a narrowband $v(n+1)$ could lead to the same bursting

phenomenon.

This work is aimed at eliminating the cause of bursting, i.e the strong crosscorrelation between the received signal at the near-end and the disturbance $v(n+1)$. We introduce here a new configuration for the adaptive echo canceler. Under bursting conditions, the new setup ensures the crosscorrelation between the input to the adaptive filter $x(n)$ and the desired signal (approximately $v(n+1)$) diminishes towards zero. As a result, the filter output is forced to converge to the neighborhood of zero and bursting will not happen. The approach is directly generalizable to practical hybrids and does not assume a priori knowledge of any specific parameters of the system or special characteristics of the near-end signal. Also, no threshold values are being used. Under normal conditions (when the far-end speaker is talking and the near-end speaker is silent) the proposed configuration maintains proper operation equivalent to the conventional echo canceler.

7.1 Bursting in a single adaptive hybrid

Though bursting can come about in higher order systems and under-modeled double adaptive hybrids, the single parameter model in Fig. 7.1 has been generally adopted in the published research studies on bursting [24, 36, 93]. This is mainly due to the analytical difficulties associated with dealing with the practical cases. Here, we examine a typical bursting situation when $v(n+1)$ is present and $w(n)$ is absent ($w(n) \approx 0$). The assumption of a simple unit delay for far-end hybrid

and no far-end echo canceler corresponds to the worst conditions of a maximum crosscorrelation between $x(n)$ and $v(n+1)$ [92]. The model in Fig. 7.1 can be viewed as feedback system with a reference signal $v(n+1)$ and an output $r(n+1)$. Defining the coefficient error as $\hat{h}(n) = h^* - h(n)$, then $r(n+1)$ can be written as

$$r(n+1) = \alpha\hat{h}(n)r(n) + v(n+1) \tag{7.1}$$

Therefore, as long as $\alpha|\hat{h}(n)| < 1$, the system in Fig. 7.1 is stable. In the echo canceler, coefficients are updated as

$$h(n+1) = h(n) + \mu x(n)r(n+1) \tag{7.2}$$

Noting that $x(n) = \alpha r(n)$, where α is a small positive number accounting for the attenuation in the transmission loop, then invoking Eq.(7.1) into (7.2) yields

$$\hat{h}(n+1) = (1 - \mu x^2(n))\hat{h}(n) - \mu x(n)v(n+1) \tag{7.3}$$

Under the conditions leading to bursting, $x(n)$ is too small compared with $v(n+1)$ while the correlation between $x(n)$ and $v(n+1)$ is high. This is evident by noting that [24]

$$\begin{aligned} x(n) &= \alpha r(n) = \alpha(v(n) + y(n) - \hat{y}(n)) \\ &\approx \alpha v(n) \end{aligned} \tag{7.4}$$

then,

$$\begin{aligned} E\{x(n)v(n+1)\} &\approx \alpha E\{v(n)v(n+1)\} \\ &\approx \alpha R_v(1) \end{aligned} \tag{7.5}$$

As a result the driving term in Eq.(7.3) will drag the instantaneous pole $\alpha\hat{h}(n)$ to cross the stability boundaries causing bursting to occur for highly correlated $v(n+1)$ as shown in Fig. 7.2. Thus, basically the adaptive algorithm tries to increase its gain for the given (low) input $x(n)$ to produce a large (fixed) output $v(n+1)$ resulting in an even lower input. This continues on causing the coefficient to drift and eventually the system to burst.

7.2 A proposed adaptive echo canceler with no bursting

Based on the above discussion, reducing the correlation between successive samples of the disturbance $v(n+1)$ would ensure the drifting phase does not start. In other words, if the near-end signal $v(n+1)$ is transformed from a narrow-band signal to a wide-band one, then the autocorrelation $R_v(1)$ is significantly reduced and the driving term in Eq.(7.3) is no longer dominant. This can be accomplished by inserting a "decorrelator" block as shown in Fig. 7.3. Using the notation q^{-1} for the unit delay operator [62], i.e, $q^{-1}v(n) = v(n-1)$, the output of the decorrelator $e_1(n+1)$ can be written as follows

$$\begin{aligned} e_1(n+1) &= s(n+1) - \sum_{i=1}^{M} a_i(n)s(n-i+1) \\ &= (1 - \sum_{i=1}^{M} a_i(n)q^{-i})s(n+1) \\ &= Q(q^{-1},n)s(n+1) \end{aligned} \tag{7.6}$$

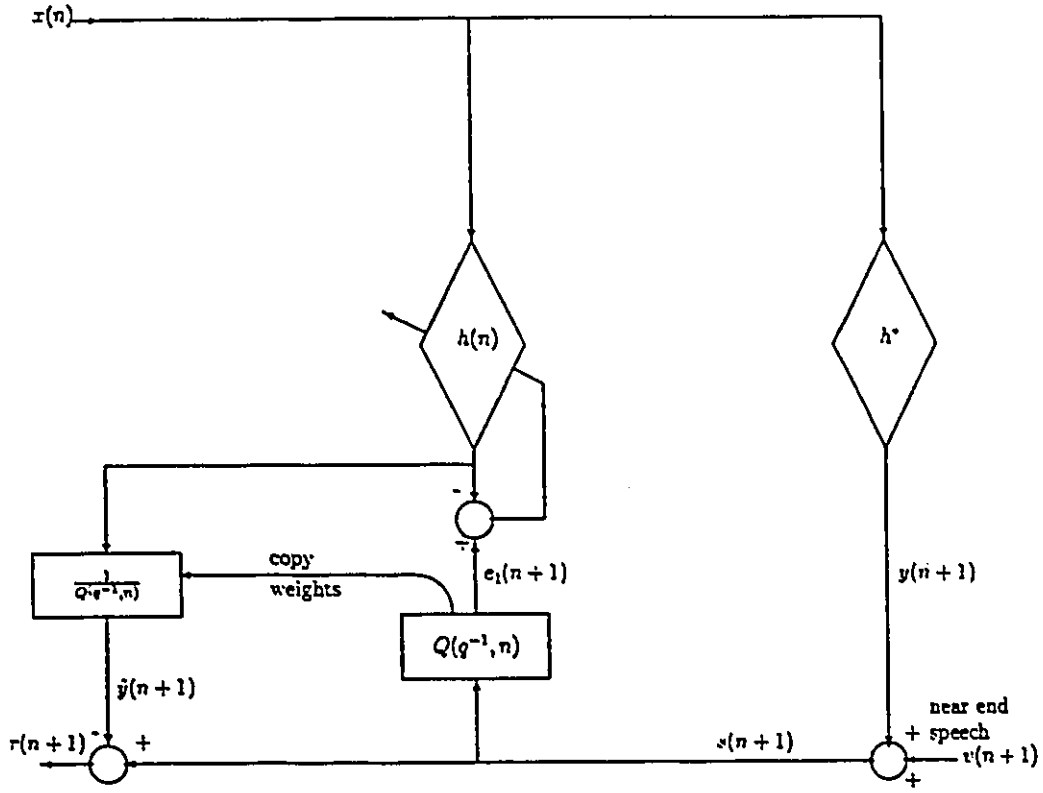

Fig. 7.3 Illustration of a single adaptive hybrid using the proposed echo canceler.

and the coefficients $\{a_1(n), a_2(n), \ldots, a_M(n)\}$ are adaptively adjusted to minimize the mean-square error $E\{e_1(n+1)^2\}$. We assume a typical bursting environment where $w(n)$ is absent and $v(n+1)$ is not, then

$$s(n+1) \approx v(n+1) \tag{7.7}$$

Assuming the adaptive linear predictor (decorrelator) has converged to the set of optimal coefficients that produce the minimum of $E\{e_1^2(n+1)\} \approx E\{(Q(q^{-1},n)v(n+1))^2\}$, then the linear predictor filter will reduce the correlation between the adjacent samples of $v(n+1)$, and if the filter is of sufficient order to whiten $v(n+1)$ then the output process $Q(q^{-1},n)s(n+1) \approx Q(q^{-1},n)v(n+1)$, which is used as a desired signal for the LMS adaptive echo canceler, will consist of a sequence of white samples. Therefore the input signal to the adaptive filter $x(n) \approx \alpha v(n)$ will be uncorrelated with the desired signal. In this case, the adaptive filter is guaranteed to converge to solution providing zero output [46]. This opens the path including the adaptive filter and resulting in the overall system having a single pole at αh^* which is fixed and guaranteed inside the unit circle since for real hybrids $|\alpha h^*| \ll 1$. Thus, bursting cannot happen in this setup. Moreover, by examining the SNR at the near-end in Fig. 7.1, which is given by [24]

$$SNR = \frac{E\{v^2(n+1)\}}{E\{(r(n+1)-v(n+1))^2\}} = \frac{E\{v^2(n+1)}{\hat{h}^2 E\{x^2(n)\}} \tag{7.8}$$

where we assumed that the DTD has turned off adaptation at $\hat{h}$, the convergence of the adaptive filter output to a value in the neighborhood of zero ensures higher SNR for the proposed setup in Fig. 7.3,

$$SNR \approx \frac{E\{v^2(n+1)}{h^{*2}E\{x^2(n)\}} \tag{7.9}$$

In order to compensate for the effect of $Q(q^{-1},n)$ which is deliberately inserted in the path of the adaptive filter desired signal, we incorporated the inverse of $Q(q^{-1},n)$ in the model of Fig. 7.3. More specifically, under normal conditions of persistently exciting $x(n)$ and small disturbance $v(n+1)$ the conventional adaptive echo canceler in Fig. 7.1 converges to h^*. Assuming that the linear predictor in Fig. 7.3 has converged, on the average, to the optimal filter $E\{Q(q^{-1},n)\} = Q^*(q^{-1})$, then the coefficient of the adaptive echo canceler filter will converge to those formed by the cascade of h^* and $Q^*(q^{-1})$. Therefore, the echo canceler filter is followed by the inverse of $Q^*(q^{-1})$, so that $\hat{y}(n+1)$ in Fig. 7.3 is a replica of the real echo $y(n+1)$. This added block is basically an IIR filter whose transfer function is the inverse of the transfer function of the original decorrelator $Q(q^{-1},n)$. Thus the coefficients of the decorrelator are basically copied to denominator of the IIR filter. This ensures that the operation of the proposed system in Fig. 7.3 is the same as of the basic setup under normal conditions.

7.3 Stability monitoring of the inverse of the decorrelation filter

Note that $\frac{1}{Q(q^{-1},n)}$ is an autoregressive filter and to retain a bounded $\hat{y}(n+1)$ the instantaneous zeros of $Q(q^{-1},n)$ should be constrained inside the unit circle. Thus, it is necessary to monitor the instantaneous roots of $Q(q^{-1},n)$ before copying its coefficients to the inverse filter to ensure stability of the resulting inverse filter. The on-line stability test of the linear predictor in Fig. 7.3 will be accompanied by an added computational complexity especially when the linear predictor filter is implemented in a direct form. Though there exist simple stability tests for second order filters, for higher orders, the current techniques are either computationally excessive or unrobust [97]. However, lattice filters have been proven as an alternative providing simple stability guarantees [46]. The lattice form is mathematically equivalent to the direct form, but it only requires the magnitude of all the reflection coefficients to be less than unity as a necessary and sufficient condition for the instantaneous roots of $Q(q^{-1},n)$ to be inside the unit circle. Moreover, in finite precision environment the lattice structure is much less sensitive to quantization noise and round-off errors effects than the direct form. This makes it a robust realization for implementing the autoregressive filter $\frac{1}{Q(q^{-1},n)}$. This is achieved at the expense of an increase in the complexity. However, in the proposed structure, only low order FIR filters are used making the price for lattice structures very

reasonable.

The gradient lattice algorithm is used to update the lattice coefficients. This algorithm has been used in diverse of applications and has proven its faster convergence compared to the standard LMS [1]. Let $k_p(n)$, $e_p^f(n)$ and $e_p^b(n)$ be the reflection coefficients, the forward prediction error and the backward prediction error, respectively, of the *pth* stage of the lattice filter at time instant n, where $1 \leq p \leq M$. The gradient lattice algorithm updates the reflection coefficient $k_p(n)$ in an attempt to minimize the sum of squared forward prediction error and backward prediction error. This leads to the updating relation [1]

$$k_p(n+1) = k_p(n) + \beta(e_p^f(n)e_{p-1}^b(n-1) + e_p^b(n)e_{p-1}^f(n)) \tag{7.10}$$

where $1 \leq p \leq M$ and β is the adaptation step size which has similar properties to that used in the weight recursion of the standard LMS algorithm. The forward and the backward prediction errors have the following order update equations

$$e_p^f(n) = e_{p-1}^f(n) - k_p(n)e_{p-1}^b(n-1) \tag{7.11}$$

$$e_p^b(n) = e_{p-1}^b(n-1) - k_p(n)e_{p-1}^f(n) \tag{7.12}$$

7.4 Complexity of the proposed algorithm

To asses the complexity of the proposed adaptive echo canceler, we observe that the gradient lattice filter requires $4M$ multiplications, $2M$ for updating the reflection coefficients and $2M$ in calculating the forward and backward prediction errors,

M being the order of the predictor. Also, the all-pole lattice filter requires $2M$ multiplications. Therefore, $6M$ more multiplications are needed over the conventional adaptive echo canceler. In the the application considered here, we believe that a linear predictor of order four can handle possible near-end narrow band signals. Even if the order of the linear predictor is not sufficient to completely whiten unexpected $v(n+1)$, reducing the correlation between the successive samples of $v(n+1)$ will severely delay the drift of $h(n)$ towards instability, and as a result, making it certain for any practical system that the DTD would have halted adaptation before bursting starts.

7.5 Analysis of the proposed echo canceler under normal conditions

In this section, we will will show that our proposed system in Fig. 7.4 minimizes the echo $E\{r^2(n)\}$ under normal operation by showing that the $r(n)$ is orthogonal to $\mathbf{X_N}(n)$, which is the condition for convergence of the basic setup in Fig. 7.1, where $\mathbf{X_N}(n) = [x(n), x(n-1), ..., x(n-N+1)]^T$. The adaptive LMS algorithm used in the echo canceler minimizes asymptotically $E\{e_2^2(n)\}$ where

$$e_2(n) = e_1(n) - c(n) \tag{7.13}$$

and $c(n) = \mathbf{h_N^T}(n)\mathbf{X_N}(n)$ as shown in Fig. 7.4. The quantity $e_1(n)$ is given by

$$e_1(n) = y(n) - \mathbf{A_M^T}(n)\mathbf{Y_M}(n) \tag{7.14}$$

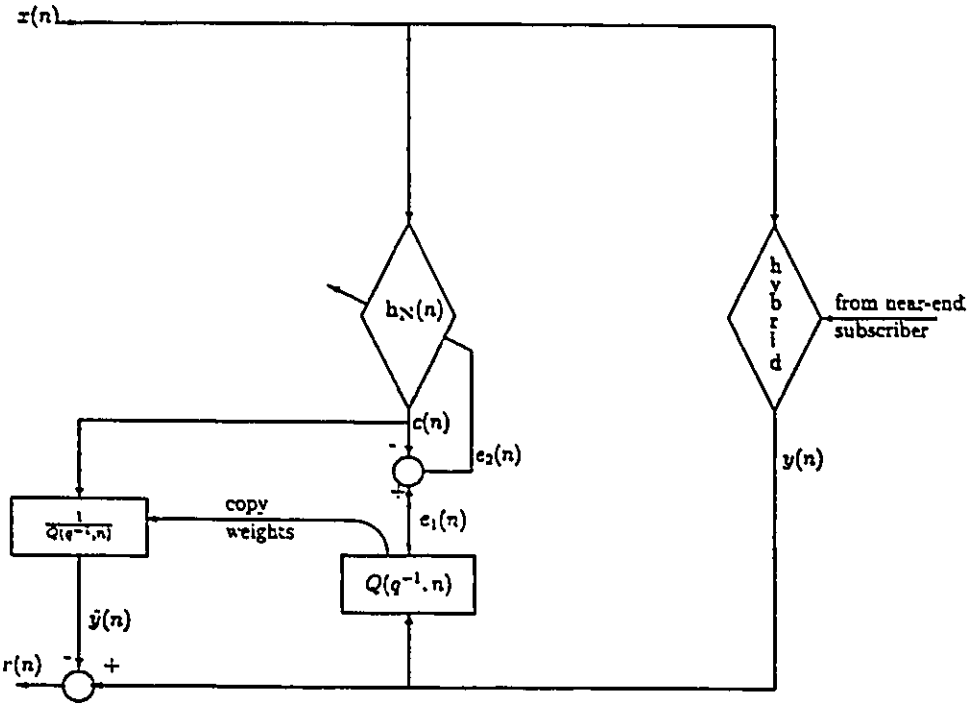

Fig . 7.4 General configuration of the proposed echo canceler.

where $\mathbf{Y_M}(n)$ is the M-dimensional vector of the form

$$\mathbf{Y_M}(n) = [y(n-1), y(n-2), \ldots, y(n-M)]^T \tag{7.15}$$

and $\mathbf{A_M}(n)$ is the coefficient vector of the linear predictor defined as

$$\mathbf{A_M}(n) = [a_1(n), a_2(n), \ldots, a_M(n)] \tag{7.16}$$

Thus,

$$e_2(n) = y(n) - \mathbf{A_M^T}(n)\mathbf{Y_M}(n) - \mathbf{h_N^T}(n)\mathbf{X_N}(n) \tag{7.17}$$

and $e_2^2(n)$ can be expressed as follows

$$\begin{aligned} e_2^2(n) \;=\; & y^2(n) - 2y(n)\mathbf{A_M^T}(n)\mathbf{Y_M}(n) + \mathbf{A_M^T}(n)\mathbf{Y_M}(n)\mathbf{Y_M^T}(n)\mathbf{A_M}(n) \\ & - 2y(n)\mathbf{h_N^T}(n)\mathbf{X_N}(n) + 2\mathbf{h_N^T}(n)\mathbf{X_N}(n)\mathbf{Y_M^T}(n)\mathbf{A_M}(n) \\ & + 2\mathbf{h_N^T}(n)\mathbf{X_N}(n)\mathbf{X_N^T}(n)\mathbf{h_N}(n) \end{aligned} \tag{7.18}$$

Assuming that the step size of the LMS is selected properly [113], then the LMS algorithm will solve asymptotically for

$$\frac{\partial}{\partial \mathbf{h_N}(n)} E\{e_2^2(n)\} = E\{\frac{\partial e_2^2(n)}{\partial \mathbf{h_N}(n)}\} = 0 \tag{7.19}$$

Differentiating Eq.(7.18) with respect to $\mathbf{h_N}(n)$

$$\frac{\partial e_2^2(n)}{\partial \mathbf{h_N}(n)} = \mathbf{X_N}(n)\mathbf{X_N^T}(n)\mathbf{h_N}(n) - \mathbf{X_N}(n)(y(n) - \mathbf{A_M^T}(n)\mathbf{Y_M}(n)\,) \tag{7.20}$$

Substituting, Eqs.(7.17), (7.20) into (7.19), we can show that

$$E\{\mathbf{X_N}(n)e_2(n)\} = 0 \tag{7.21}$$

Using the previous definition of the delay operator [62], i.e. $q^{-j}x(n) = x(n-j)$, in Eq.(7.17) we get

$$\begin{aligned} e_2(n) &= Q(q^{-1},n)y(n) - c(n) \\ &= Q(q^{-1},n)(y(n) - \frac{c(n)}{Q(q^{-1},n)}) \\ e_2(n) &= \mathcal{Q}(q^{-1},n)r(n) \end{aligned} \tag{7.22}$$

where we have assumed the stability of the filter $\frac{1}{Q(q^{-1},n)}$. Note that if the linear predictor converged to the optimal filter $E\{Q(q^{-1},n)\} = Q^*(q^{-1})$, then all roots of $Q^*(q^{-1})$ are constrained in the unit circle if and only if the autocorrelation matrix of the input signal to the linear predictor is positive definite [46]. This condition is naturally met when the far-end signal is a speech signal and thus the stability of the inverse filter is guaranteed at convergence. To show that the proposed system satisfies $E\{\mathbf{X_N}(n)r(n)\} = 0$ at convergence, we rewrite Eq.(7.22) as

$$\begin{aligned} r(n) &= \frac{e_2(n)}{Q(q^{-1},n)} \\ &= \left(\sum_{j=0}^{\infty} b_j(n)q^{-j}\right) e_2(n) \\ &= \sum_{j=0}^{\infty} b_j(n)e_2(n-j) \end{aligned} \tag{7.23}$$

Then,

$$E\{\mathbf{X}_N(n)r(n)\} = E\{\mathbf{X}_N(n)\sum_{j=0}^{\infty} b_j(n)e_2(n-j)\} \tag{7.24}$$

Before proceeding, we will assume the input signal is of zero-mean, and the coefficients $b_j(n)$, $0 \leq j < \infty$ of the filter $\frac{1}{Q(q^{-1},n)}$ are independent of $\mathbf{X}_N(n)$ and the $e_2(n-j)$ at convergence. Then we can write Eq.(7.24) as

$$E\{\mathbf{X}_N(n)\sum_{j=0}^{\infty} b_j(n)e_2(n-j)\} = \sum_{j=0}^{\infty} E\{b_j(n)\}E\{\mathbf{X}_N(n)e_2(n-j)\} \tag{7.25}$$

Note that, since the current data is independent of the previous error, then $E\{x(n-i)e_2(n-j)\} = E\{x(n-i)\}E\{e_2(n-j)\} = 0$ for $i > j$, where $0 \leq j < \infty$ and $0 \leq i \leq N-1$. Moreover, we deduce from Eq.(7.21) that $E\{x(n-i)e_2(n-j)\} = 0$ for $i \leq j$, where $0 \leq j < \infty$ and $0 \leq i \leq N-1$. As a result,

$$E\{\mathbf{X}_N(n)r(n)\} = \sum_{j=0}^{\infty} E\{b_j(n)\}E\{\mathbf{X}_N(n)e_2(n-j)\} = 0 \tag{7.26}$$

and the orthogonality principle is satisfied.

7.6 Examples

Figure 7.2 illustrates the result of applying a dc input at the far-end, i.e. $v(n+1) = 1$ in Fig. 7.1, with the parameter values, echo attenuation $\alpha = 0.2$, and $h^* = 0.1$ [92]. The LMS algorithm is used with a step size $\mu = 0.01$ and the filter coefficient was initialized at zero. The same previous parameter values are employed in the proposed echo canceler with a third order adaptive filter, a second

order lattice predictor $M = 2$, and adaptation constant $\beta = 0.01$. The output of the decorrelator and the received signal at the near-end $x(n)$ are depicted in Figs. 7.5, 7.6 respectively. As noted, simulation results confirmed previous discussion: the proposed echo canceler forced the filter output , Fig. 7.5, to converge to zero and accordingly bursting does not happen as seen from Fig. 7.6.

The previous example is repeated for a near-end $v(n+1) = sin(0.05(n+1))$ as used in [92]. The system in Fig. 7.1 exhibits bursting as shown in Fig. 7.7. It is evident from Fig. 7.8 that the proposed system maintains stability, and bursting does not happen.

Next, the operation of the new setup is verified in comparison with the conventional echo canceler for a practical case of real hybrid. A practical hybrid that has an effective impulse response of 20 taps is utilized [108] and α was set to 0.2. The conventional echo canceler has an FIR adaptive LMS filter of a dimension N=15, and a step size $\mu = 1 \times 10^{-5}$ is used. The proposed configuration is used with an adaptive filter of a dimension N=17, LMS step size $\mu = 1 \times 10^{-5}$, and a second order gradient adaptive lattice filter with $\beta = 1 \times 10^{-5}$. To compare the conventional echo canceler and the proposed one under normal condition when the far-end is talking and the near-end is silent, a white Gaussian signal is applied at the far-end part. Results are obtained by averaging over 100 independent runs. Figs. 7.9 and 7.10 plot the Echo Return Loss Enhancement (ERLE) of the conventional and the

proposed echo cancelers [76]. The ERLE is defined as

$$\text{ERLE} = 10 \log_{10}(E\{y^2(n)\}/E\{r^2(n)\}) \tag{7.27}$$

Comparing Figs. 7.9 and 7.10 illustrates that under normal situation the new modification has no impact on the echo canceler two fundamental requirements: convergence time and the level of returned echo. The new setup provides similar convergence rate and final steady state of the ERLE to those of the conventional echo canceler. This is expected since in practical applications β should have small values (similar to the one used in this example) to maintain the stability of the linear predictor and accordingly the fluctuation of the linear predictor coefficients around their optimal values are too small so that it has negligible effect on the echo canceler performance. When, the far-end is silent and a tone signal $v(n+1) = 100sin(0.05(n+1))$ is applied at the near-end. The misbehavior of the conventional echo canceler is evident from Fig. 7.11. On the other hand, Fig. 7.12 shows that the new configuration allows the passage of the near-end transmitted signal uncancelled and also exhibits stable operation.

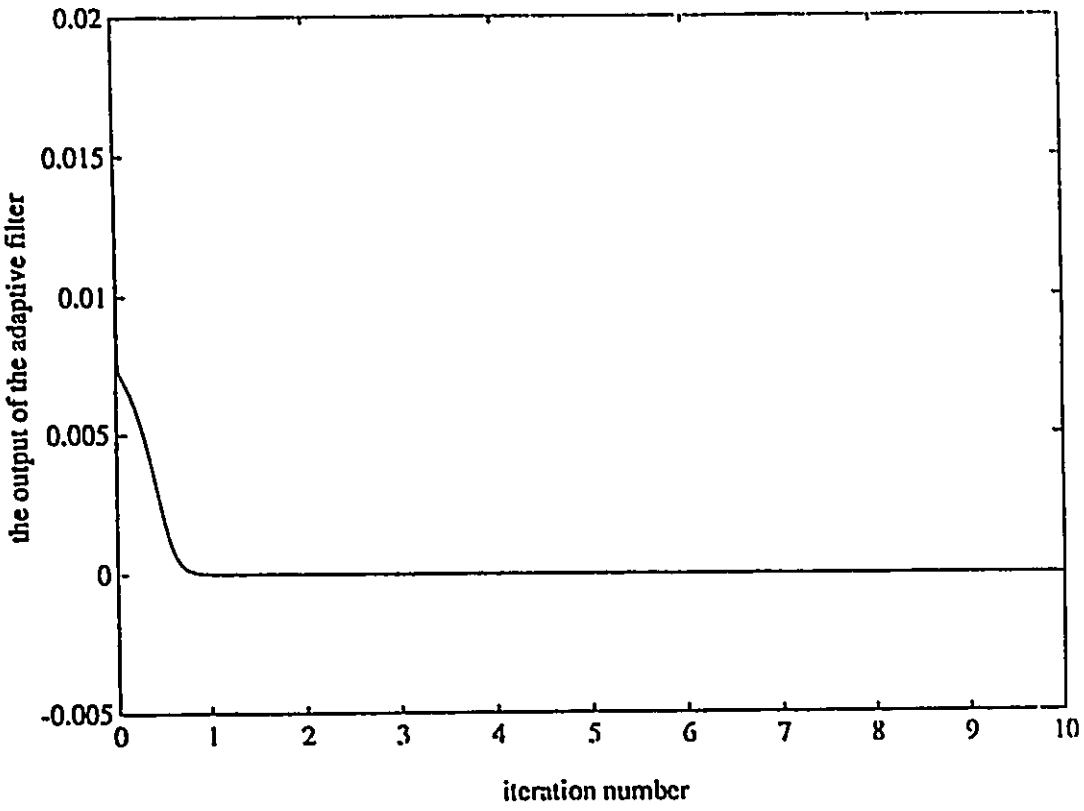

Fig. 7.5 The output of the adaptive filter with a dc near-end signal $v(n+1)$ for the proposed model in Fig. 7.3.

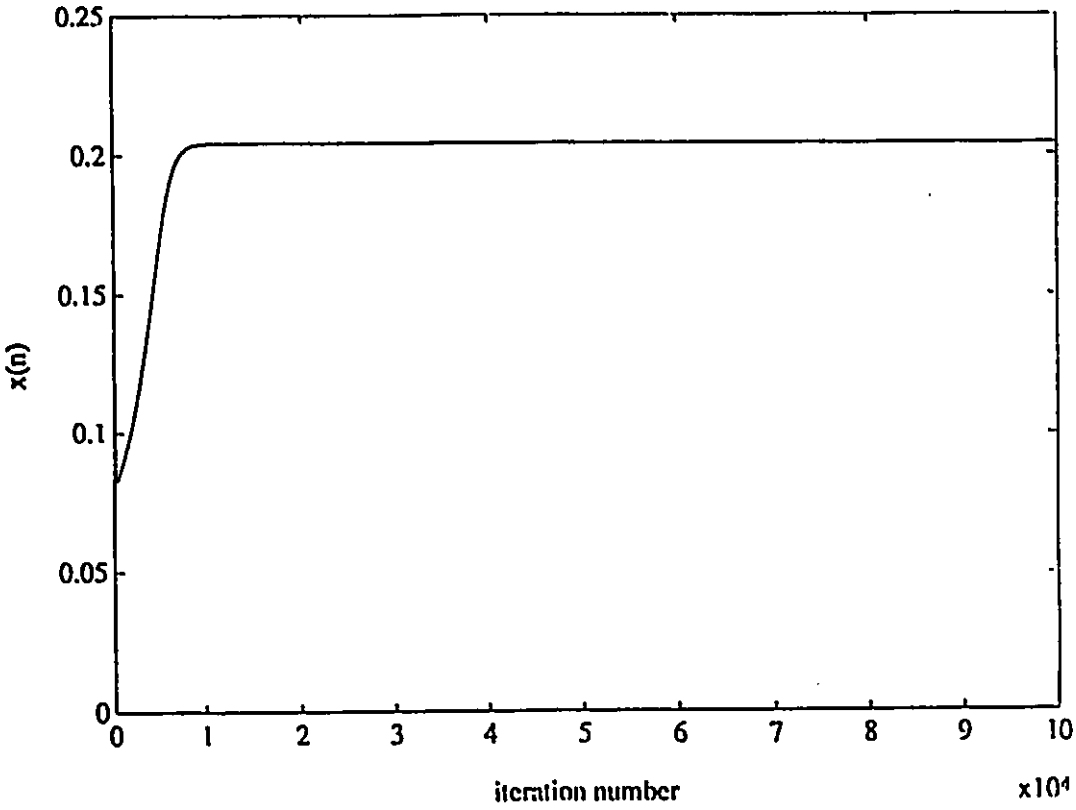

Fig. 7.6 The received signal $x(n)$ with a dc near-end signal $v(n+1)$ for the proposed model in Fig. 7.3.

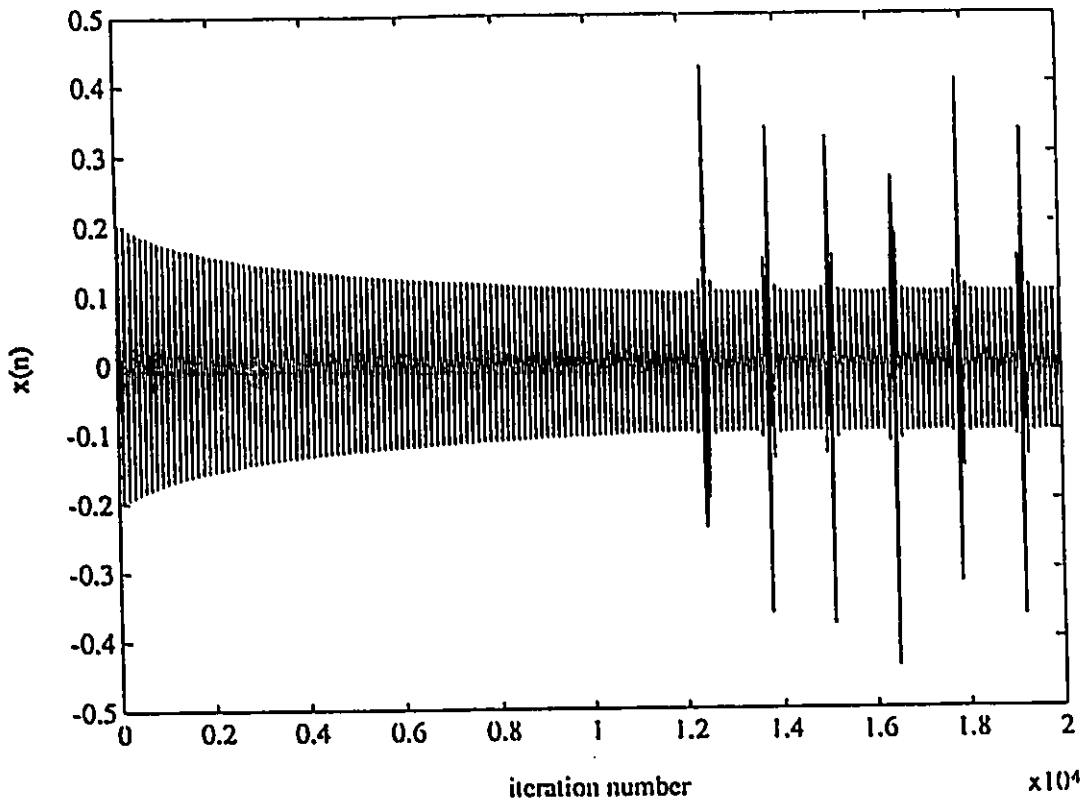

Fig. 7.7 The received signal $x(n)$ with a sinusoidal near-end signal $v(n+1)$ for the model in Fig. 7.1.

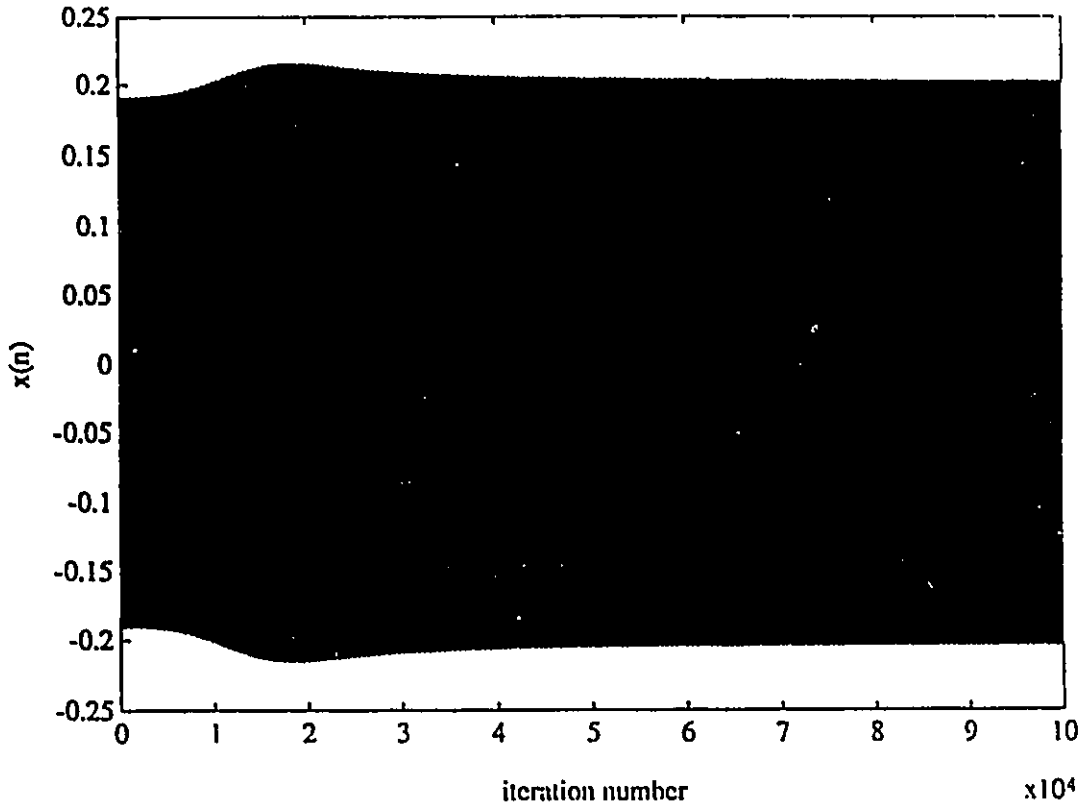

Fig. 7.8 The received signal $x(n)$ with a sinusoidal near-end signal $v(n+1)$ for the proposed model in Fig. 7.3.

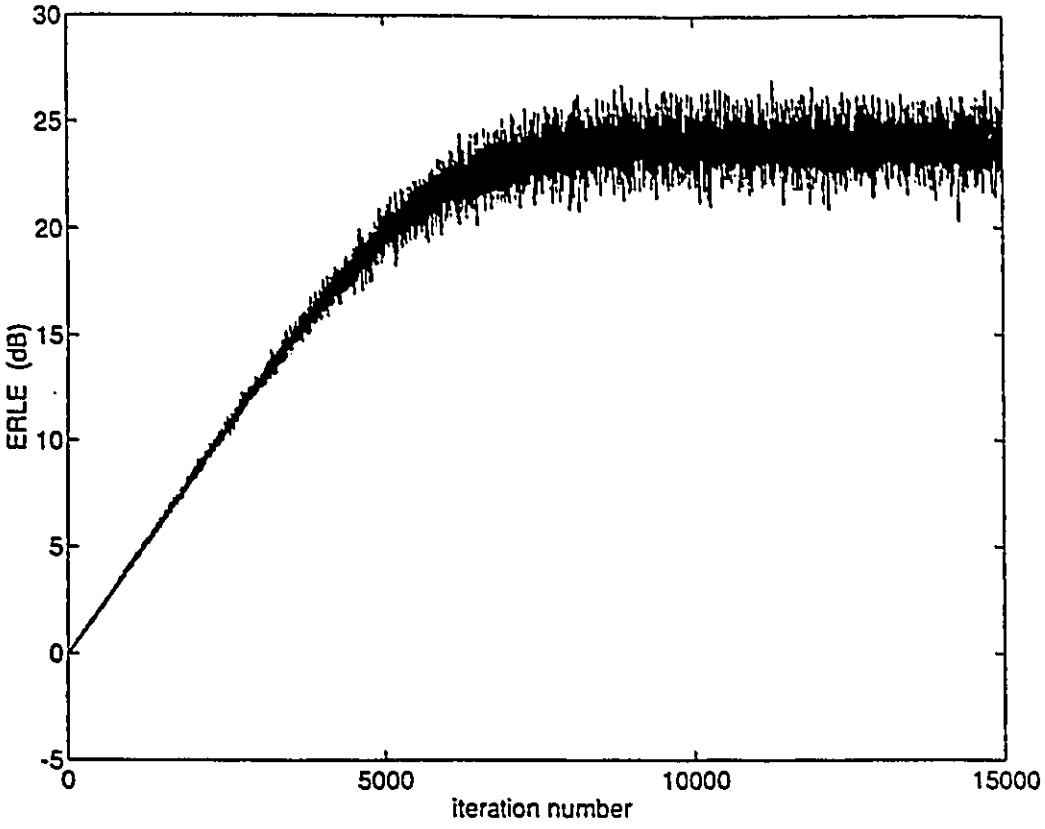

Fig. 7.9 Echo return loss enhancement with a white input signal at the far-end for the model in Fig. 7.1.

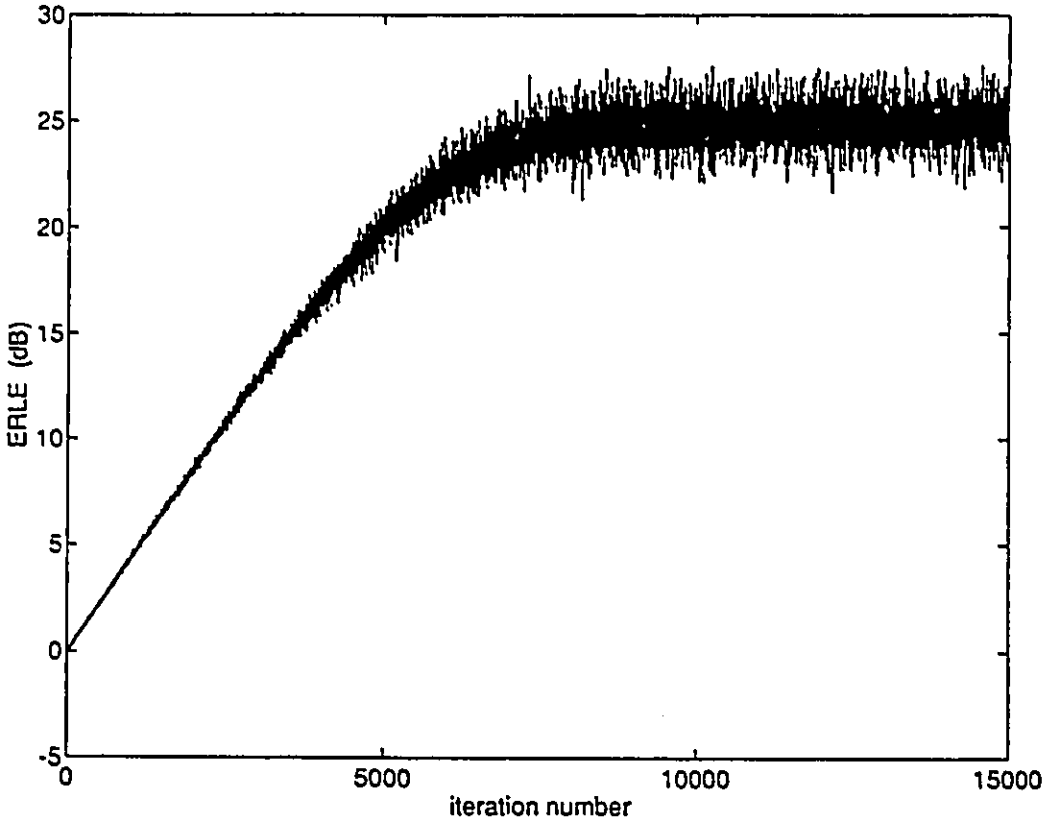

Fig. 7.10 Echo return loss enhancement with a white input signal at the far-end for the proposed model in Fig. 7.3.

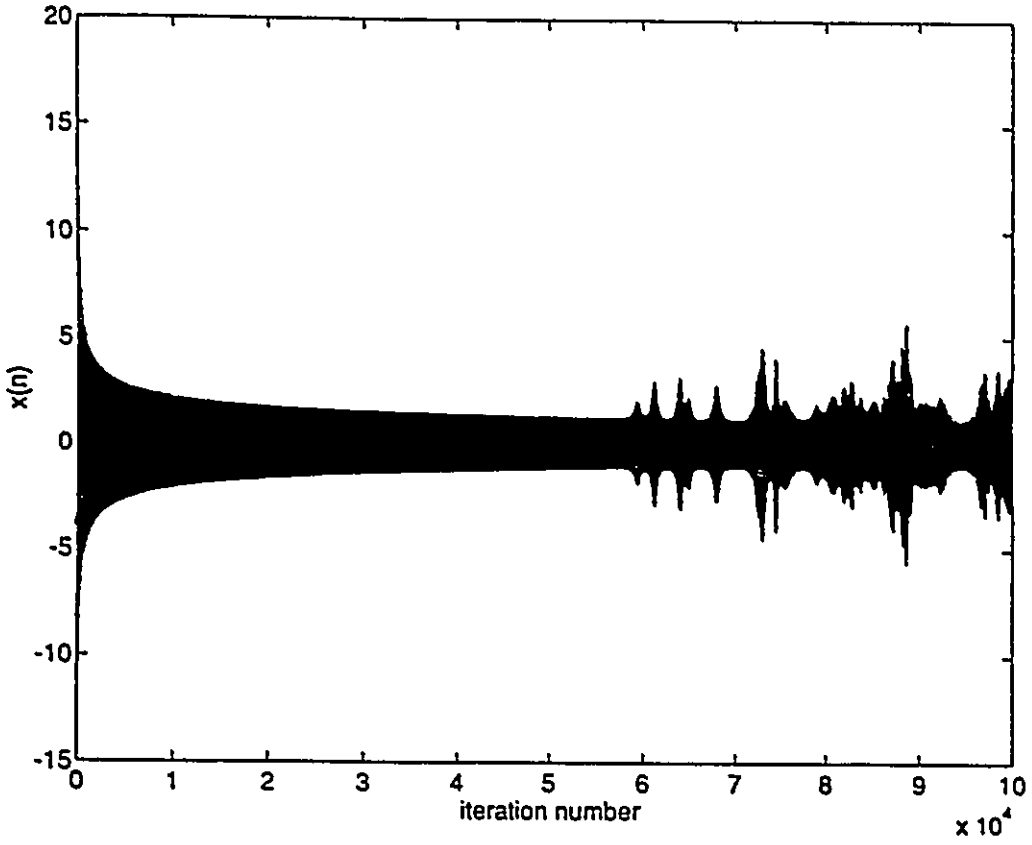

Fig. 7.11 The received signal $x(n)$ in the model of Fig. 7.1 with a sinusoidal near-end $v(n+1)$ for the real hybrid example.

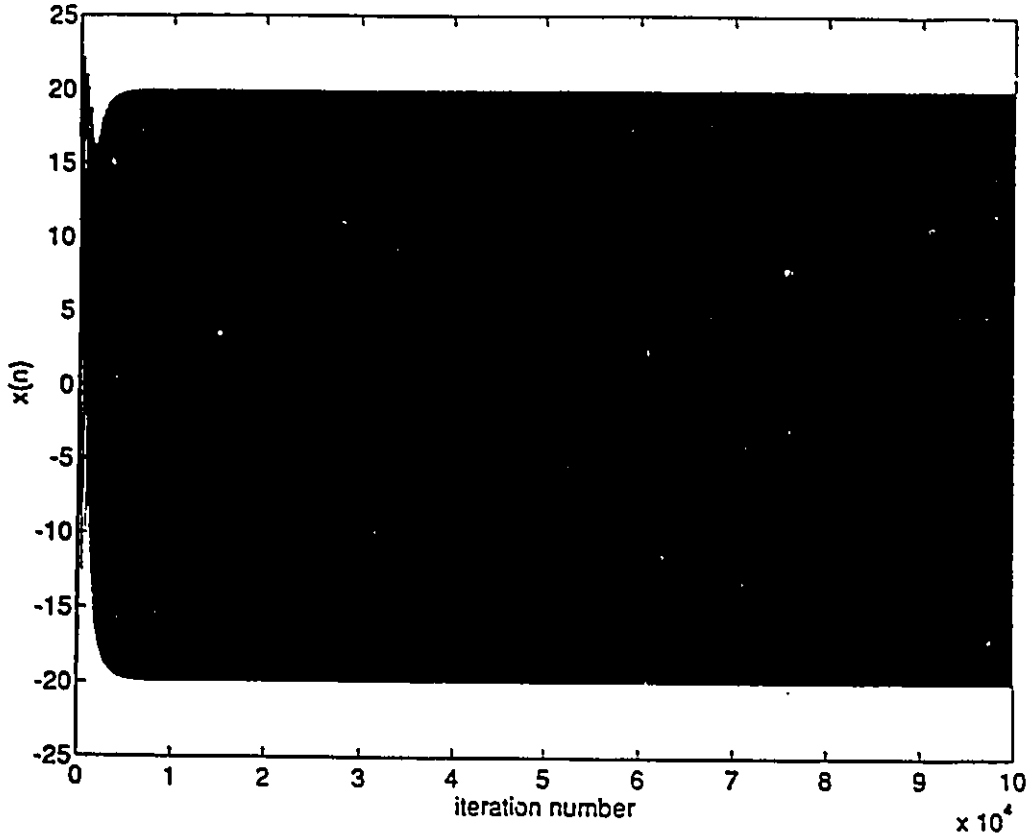

Fig. 7.12 The received signal $x(n)$ in the proposed model of Fig. 7.3 with a sinusoidal near-end $v(n+1)$ for the real hybrid example.

7.7 Conclusion

Bursting phenomenon in telephone networks with adaptive hybrids has been the focus of much attention in recent years. This chapter presented a remedial technique that basically relies on weakening the root cause of the bursting problem: the high crosscorrelation between the input to the adaptive echo canceler and the transmitted signal at the near-end. The conventional echo canceler was modified such that under bursting circumstances the crosscorrelation is substantially reduced and bursting is averted. No a priori knowledge of the system or preselection of thresholds were required. The proposed system is directly generalizable for practical hybrids. The system ensures normal operation is not affected. Implementation details of the proposed system were studied. Examples were provided to verify system performance in bursty as well as normal environments.

Chapter 8

Conclusion

In chapter 3, 4, and 5, we dealt with a common problem of the LMS algorithm; namely its slow convergence when it encounters nonideal conditions. Chapter 3 and 4 tackled this problem for FIR adaptive filters, where in chapter 5 the problem was considered for IIR adaptive filters.

The new variable step size algorithm presented in chapter 3 represents one common approach to provide fast convergence at initial stage of adaptation while ensuring small final misadjustment. In our proposed approach, the step size of the algorithm is adjusted according to the square of a time-averaging estimate of autocorrelation of $e(n)$ and $e(n-1)$. {Eq.(3.12)} showed that this measure can effectively sense the adaptation process while maintaining the immunity against independent noise. Consequently, the algorithm can perform efficiently in low signal-to-noise ratio environments and is not affected by changes in the level of the noise.

The analysis of the steady state performance of the proposed algorithm is also provided. Based on this analysis we showed how to select the algorithm parameters γ, α, and β to ensure MSE convergence {Eq.(3.33)} and to provide the desired misadjustment {Eq.(3.35)}. We also obtained expressions for the algorithm misadjustment in nonstationary environments {Eq.(3.35)} as well as stationary environments {Eq.(3.37)}. Moreover, an optimal value for γ was derived in a nonstationary environment that minimizes the level of misadjustment for given values of α, β, and level of nonstationarity. The performance of the algorithm was compared through simulations with the standard LMS [47], and other variable step size algorithms: the VSS algorithm [56], and the SGA-GAS algorithm [68]. Results show that the algorithm has a significant convergence rate improvement over those algorithms in stationary environment for the same excess MSE, and a comparable performance with the standard LMS in nonstationary cases.

Chapter 4 introduced a different approach to improve the LMS algorithm convergence rate by presenting a new LMS-type adaptive algorithm. The proposed algorithm utilized a new technique that combines both time and order update to search for the bottom of the MSE surface. The algorithm has been observed to have significant convergence rate improvement when the signal is correlated. This is attributed to the diversity of the steepness of the gradient vector components in such environments, where the movement of the weight vector component in the

steepest direction will be beneficial to the other components in the context of order updating. The algorithm has also been observed to efficiently operate in nonstationary environments and to be fairly robust to variations in signal power. Approximate theoretical analysis is provided to to determine the convergence speed and steady state performance of the algorithm when the input signal is zero-mean and white Gaussian. Results demonstrated that the algorithm can provide less misadjustment compared to the standard LMS algorithm for the same step size value, and is inherently convergent in the mean. Simulation results agreed well with theoretical results. Two versions of the algorithm were presented: the sequential order-updating (SQOU) algorithm, and the selective order-updating (SLOU) algorithm. Simulation results showed that the SLOU is favorable over the SQOU only when the input signal is correlated.

The above discussion centered on FIR filters. For IIR adaptive filters, the source of the inherent slow convergence of the LMS is more related to the recursive nature of the IIR filter. This results in nonquadratic, nonlinear error functions where gradient search methods are inefficient. To get around this problem with minimal additional complexity, a new adaptive scheme that couples the LMS and the LS methods was presented in chapter 5. The idea takes advantage of the bowl-shaped property of the error surface formed by the MA part of the adaptive IIR filter and uses the LMS algorithm to determine those coefficients. On the other hand,

the error surface formed by the AR part is highly nonlinear and, therefore, a LS approach is used to search the AR coefficients. The proposed algorithm in Table 5.1 requires $O(M^2)+O(N)$ multiplications, where M and N are the orders of the AR and the MA filters, respectively. The complexity of the algorithm is brought down to $O(M)+O(N)$ by taking advantage of the redundancy in the calculation of the gain vector. Simulation results showed that the algorithm converges fast while being fairly unaffected by signal statistics and the initial value of the weight vector. Moreover, since the LMS is relieved from updating the denominator coefficients, a large step size can be used for the LMS to speed up adaptation. The algorithm showed through simulations the ability to achieve global convergence for sufficient and some reduced order cases.

In chapter 6, we presented a theoretical analysis of the leaky LMS algorithm through studying the convergence properties of the mean square. This was under the assumptions of zero-mean stationary Gaussian input signals and the independence of the data vector from the weight vector. We derived a recursion characterizing the evolution of the weight vector covariance matrix, {Eq.(6.9)}. Then an expression for the second moment of the weight vector, which is the key to MSE analysis, was obtained. We presented direct stability bounds for the convergence of the MSE, {Eq.(6.40)}, {Eq.(6.42)}. Furthermore, we derived an exact expression for the algorithm steady state excess MSE {Eq.(6.54)}. The equation quantifies

the impact of all input parameters: input signal statistics, crosscorrelation between the input signal and the desired signal, step size value, and the leakage factor on the algorithm steady state performance. Additionally, it shows that the excess MSE of the Leaky LMS is due to two sources. The first is due to the noisy gradient estimate as is common for LMS algorithms. This is directly proportional to the noise level. The second is due to the leakage term. This is proportional to the crosscorrelation between the input and the desired signals (as opposed to the LMS contribution which is independent of crosscorrelation) and most importantly is independent of the disturbance noise. It should be noted that all results presented for the Leaky LMS are identical to those of the LMS algorithm when $\gamma = 0$ [31, 47]. The analytical results were shown to agree well with simulation results.

Finally, we considered one of the applications of the LMS algorithm in chapter 7. Problem of bursting in adaptive echo cancelers used in conjunction with hybrids on telephone lines commonly arises when operating under nonideal conditions. It was recognized that the presence of the adaptive algorithm in a feedback loop along with the strong crosscorrelation between the input signal and the near-end signal are the root causes of bursting in adaptive hybrids.

Current methods introduced to prevent bursting suffer several disadvantages. Some can be only applied to the case of first order hybrid and are not generalizable to practical cases of higher orders. They also require knowledge of the hybrid and

use some predetermined thresholds [24, 36]. Other practical methods [108, 109] restrict the nature of the near-end signal to be a pure tone and use predetermined thresholds.

We have proposed a modified echo canceler that aims at stopping bursting by weakening the driving force behind it: the crosscorrelation between the near-end signal and the received signal at the near-end. The proposed solution prevents the occurrence of bursting while ensuring normal operation is not affected. This was verified by theoretical analysis of the proposed setup under normal operation. The approach does not require knowledge of the hybrid parameters nor does it constrain the order of the hybrid or the adaptive echo canceler. Consequently, it is directly applicable for practical hybrids with a large number of unknown taps. The proposed setup adds $6M$ more multiplications over the conventional echo canceler, M being the order of the adaptive decorrelating filter. Simulation examples have verified the effectiveness of the proposed approach in bursty as well as normal environments. It was demonstrated that the setup retains performance equivalent to the conventional one when it operates under normal conditions.

8.1 Suggestions for further research

Through the course of this research, several issues and ideas that can be further investigated come up. Here, we summarize a few:

1. The least squares approach utilized to search the denominator coefficients of the IIR adaptive filter can also be applied to seek the numerator coefficients. However, this entails a heavy toll when considering the computational complexity. An alternative is to derive a fast version. Numerical stability is a serious issue in this case. The investigation of the possibility of applying some of the stabilizing method that rely on feeding back the numerical errors [6, 98] is needed. A comparison of this approach with the RML algorithm [62] and the IIR SER algorithm [113] can highlight the potential advantages and disadvantages of the proposed approach.

2. The LMS-LS algorithm discussed in chapter 4 and the one suggested above are formulated for direct form adaptive IIR filters. To solve the problem of stability monitoring, it is worthwhile to investigate applying the approach to the cascade form [97]. In this case, the filter comprises second order sections in which the stability triangle method can directly be utilized.

3. Of primary concern is the operation of a given algorithm in finite precision environment. This reveals the true performance of the algorithm when implemented using fixed point arithmetic. The analysis presented for the Leaky LMS assumed infinite precision operation. Therefore, a study of the performance of the Leaky LMS when implemented with finite precision can be of relevant importance.

4. Investigation of different approaches to overcome bursting in adaptive hybrids is still an open issue for further improvement.

Appendix A

In this Appendix, we derive an expression for the steady state excess MSE of the Leaky LMS algorithm resulting from the fluctuations of the algorithm weight vector $\mathbf{W}(n)$ about the steady state mean solution $\mathbf{W_L}$ of the algorithm. Defining $\mathbf{V_L}(n) = \mathbf{W}(n) - \mathbf{W_L}$, then Eq.(6.2) can be written as

$$\hat{\mathbf{V}}_{\mathbf{L}}(n+1) = [\mathbf{I} - \mu(\gamma\mathbf{I} + \hat{\mathbf{X}}(n)\hat{\mathbf{X}}^{\mathbf{T}}(n))]\hat{\mathbf{V}}_{\mathbf{L}}(n) - \mu\gamma\hat{\mathbf{W}}^{*} + \mu e^{*}(n)\hat{\mathbf{X}}(n)$$
$$+\mu(\gamma\mathbf{I} + \hat{\mathbf{X}}(n)\hat{\mathbf{X}}^{\mathbf{T}}(n))]\,\mathbf{L} \qquad \text{(A.1)}$$

where $\mathbf{W_L} = (\mathbf{R}+\gamma\mathbf{I})^{-1}\mathbf{P}$, $\mathbf{L} = \hat{\mathbf{W}}^{*} - \hat{\mathbf{W}}_{\mathbf{L}} = [l_1, l_2, \ldots, l_N]^T$, and $\hat{\mathbf{L}} = [l_1^2, l_2^2, \ldots, l_N^2]^T$ as defined in chapter 6. Note that from Eq.(A.1), as expected, $E\{\hat{\mathbf{V}}_{\mathbf{L}}(\infty)\} = \mathbf{0}$. We define $\mathbf{Z_{L1}}(n)$ as an $N \times 1$ whose components are the diagonal elements of $E\{\hat{\mathbf{V}}_{\mathbf{L}}(n)\hat{\mathbf{V}}_{\mathbf{L}}^{\mathbf{T}}(n)\}$. Using Eq.(A.1) and after some manipulations, it can be shown that

$$\mathbf{Z_{L1}}(\infty) \;=\; \mathbf{A}\mathbf{Z_{L1}}(\infty) + \mu^2\mathbf{\Gamma}\epsilon_{Lmin} + \mu^2\mathbf{\Lambda}^2\hat{\mathbf{L}} \qquad \text{(A.2)}$$

Following the same approach in chapter 6, it can be shown that the steady state MSE in terms of the newly defined coefficient error vector $\mathbf{Z_{L1}}(n)$ is given by

$$\epsilon(\infty) = \epsilon_{Lmin} + \mathbf{\Gamma^T Z_{L1}}(\infty) \tag{A.3}$$

Consequently, the increase in the MSE of the Leaky LMS algorithm resulting exclusively from its fluctuations about the minimum attainable MSE can be expressed as

$$\begin{aligned} \epsilon_{Lex}(\infty) &= \epsilon(\infty) - \epsilon_{Lmin} \\ &= \mathbf{\Gamma^T Z_{L1}}(\infty)) \end{aligned} \tag{A.4}$$

Using Eq.(A.2) in (A.4), we obtain the expression for $\epsilon_{Lex}(\infty)$ in Eq.(6.58).

Bibliography

[1] Alexander, S. T., *Adaptive Signal Processing: Theory and Applications.*, New York: Springer Verlag, 1986.

[2] Alexandar, S. T., and S. H. Ardalan, "Adaptive Telephony Echo Cancellation Using Fast Kalman Pole-Zero Modeling," *Int Conf. Commun.*, pp. 46.2.1-5, June 1985.

[3] Ardalan, S. H., "Pole/Zero Fast Kalman Echo Cancellation and Application to Actual Measured Echo," *Int Conf. Commun.*, pp. 46.3.1-5, June 1985.

[4] Ardalan, S. H., "Derivation ot Fast Pole-Zero (ARMA) Recursive Least Squares Algorithm Using Geometric Projections," *ICASSP'86*, Tokyo, Japan, pp. 2095-2098.

[5] Ardalan, S. H., and L. J. Feber, "A Fast ARMA Transversal RLS Filter Algorithm," *IEEE Trans. on Acoust., Speech, and Signal Processing*, vol. 36, no. 3, pp. 349-358, March 1988.

[6] Benalla, A., and A. Gilloire, "A New Method to Stabilize Fast RLS Algorithms Based on a First Order Model of the Propagation of Numerical Errors," *ICASSP'88*, pp. 1365-1368.

[7] Bellanger, M. C., *Adaptive Digital Filters and Signal Analysis*, Madison Avenue, New York: Marcel Dekker, 1987.

[8] Bershard, N. J., " Analysis of the Normalized LMS Algorithm with Gaussian Input," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no. 4, pp. 793-806, Aug. 1986.

[9] Bershard, N. J., and L. Z. Qu, "On the Probability Density Function of the LMS Adaptive Filter Weights," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 1, pp. 43-56, Jan. 1989.

[10] Boray, G. K., and M. D. Strinath, "Conjugate Gradient Techniques for Adaptive Filtering," *IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications*, vol. 39, no. 1, pp. 1-10, Jan. 1992.

[11] Boroujeny, B., "Variable Step Size LMS Algorithms: New Developments and Experiments," *Proc. ICASSP'93*, pp. 265-268.

[12] Caraynnis, G. et al., "A Fast Sequential Algorithm for Least-Squares Filtering and Prediction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1394-1402, Dec. 1983.

[13] Chao, J. et al., "A Fast Adaptive Filter Algorithm using Eigenvalue Reciprocal as Step Sizes," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 8, pp. 1343-1352, Aug. 1990.

[14] Chao, J., et al., "A New IIR Adaptive Echo Canceler: GIVE," *IEEE Journal on selected area in Commun.*, vol. 12, no. 9, pp. 1530-1539, Dec. 1994.

[15] Chen, J. et al., "A New Structure for Subband Acoustic Echo Canceler," *ICASSP'88*, pp. 2566-2569.

[16] Cioffi, J. M., "When Do I Use RLS Adaptive Filter?," *In Proc. 1985 Asilomar Conf. Circuits, Systems, Signal Processing*, Nov. 1985.

[17] Cioffi, J. M, "The Block-Processing FTF Adaptive Algorithm," *IEEE Trans. on Acoust., Speech, and Signal Processing*, vol. ASSP-34, no. 1, pp. 77-90, Feb. 1986.

[18] Cioffi, J. M., "Limited-Precision Effects in Adaptive Filtering," *IEEE Trans. Circuit and Systems*, vol. CAS-34, no. 7, pp. 821-833, July 1987.

[19] Cioffi, J. M., and T. Kailath, "An Efficient RLS Data-Driven Each Canceler for Fast Initialization of Full-Duplex Data Transmission," *IEEE Trans. on Commun.*, vol. COMM-33, no. 7, July 1985.

[20] Cioffi, J. M., and T. Kailath, "Fast Recursive-Least-Squares Transversal Filters for Adaptive Filtering, " *IEEE Trans. on Acoust., Speech, and Signal Processing*, vol. ASSP-32, no. 2, pp. 304-338, April 1984.

[21] Clark, G. A. et al., "Block Implementation of Adaptive Digital Filters," *IEEE Trans. Circuit and Systems*, vol. CAS-28, no. 6, pp. 584-592, June 1981.

[22] Clark, G. A. et al., "A Unified Approach Time- and Frequency-Domain Realization of FIR Adaptive Digital Filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, no. 5, pp. 1073-1083, Oct. 1983.

[23] Cowan, C. F. N., "Performance Comparison of Finite Linear Adaptive Filters," *Proc. Inst. Elec. Eng.*, vol. 139, pt. F, no. 3, pp. 211-216, June 1987.

[24] Ding, Z. et al., "Frequency-Dependent Bursting in Adaptive Echo Cancellation and its Preventing Using Double-Talk Detectors," *International Journal of Adaptive Control and Signal Processing*, pp. 219-236, 1990.

[25] Etter, D. M., and M. M. Masukama, "A Comparison of Algorithms for adaptive Estimations of Time Delay Between Sampled Signals," *ICASSP'81*, pp. 1253-1256.

[26] Evans, J. B., P. Xue, and B. Liu, "Analysis and Implementation of Variable Step Size Adaptive Algorithms," *IEEE Trans. on Signal Processing*, vol. 41, no. 8, pp. 2517-2535, Aug. 1993.

[27] Fang, H., and W. K. Jenkins, "An Investigation of an Adaptive IIR Echo Canceler: Advantages and Problems," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 36, no. 12, pp. 1819-1833, Dec. 1988.

[28] Feber, L. J. et al., "A Derivation of A General Order, Multichannel Fast Transversal, Recursive Least Squares Filter Algorithm," *Proc. IEEE Military Commun., Conf.*, pp. 6.1.1-6.1.5, Oct. 1986.

[29] Ferrara, E. R., "Fast Implementation of the LMS Adaptive Filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, no. 4 pp. 474-480, June 1980.

[30] Feuer, A., "A Performance Analysis of Block Least Mean Square Algorithm," *IEEE Trans. Circuit and Systems*, vol. CAS-32, no. 9, pp. 960-963, Sept. 1985.

[31] Feuer, A., and E. Weinstein, "Convergence Analysis of LMS Filters with Uncorrelated Gaussian Data," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-33, no. 1, pp. 222-230, Feb. 1985.

[32] Fisher, B., and N. J. Bershad, "The Complex LMS Adaptive Algorithm transient weight mean and convenience with applications to ALE," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 34-44, Feb. 1983.

[33] Foley, J. B., and F. M. Boland, "A Note on Convergence Analysis of LMS Adaptive Filters with Gaussian Data," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 36, no. 7, pp. 1087-1089, July 1988.

[34] Gilloire, A., and M. Vetterli, "Adaptive Filtering in Subbands with Critical Sampling: Analysis, Experiments, and Application to Acoustic Echo Cancellation," *IEEE Trans. Signal Processing*, vol. 40, no. 8, pp. 1862-1875, Aug. 1992.

[35] Gitlin, R. P. et al., "The Tap-Leakage Algorithm: An Algorithm for the Stable Operation of a Digitally implemented Fractionally Spaced Equalizer," *Bell Sys. Tech. Journal*, vol. 61, no. 8, pp. 1817-1839, Oct. 1982.

[36] Gonzal, J. R. et al., "The Dynamics of Bursting in Simple Adaptive Feedback Systems with Leakage," *IEEE Trans. on Circuit and Systems*, vol. 38, no. 5, pp. 475-488, May 1991.

[37] Goodch, R. P., and J. J. Shynk, "Wide-Band Adaptive Array Processing Using Pole-Zero Digital Filters," *IEEE Trans. on Antennas Propagation*, vol. AP-34, no.3, pp. 355-367, Mar. 1986.

[38] Goodwin, G. C. et al., "Stochastic Adaptive Control For Exponentially Convergent Time-Varying Systems," *SIAM J. Contr. Optimization,* vol. 24, pp. 589-603, July 1986.

[39] Goodwinad, G. C., and K. S. Sin, *Adaptive Filtering, prediction and control*, Englewood Cliffs, NJ: Prentice-Hall, 1985.

[40] Gray, R. M., "On the Asymptotic Eigenvalue Distribution of Toeplitz Matrices," *IEEE Trans. Inform. Theory*, vol. IT-18 no. 6, pp. 725-730., Nov. 1972.

[41] Griffiths, L. J., "A Simple Adaptive Algorithm for Real-Time Processing in Antenna Arrays," *Proc. IEEE,* vol. 57, no. 10, pp. 1696-1704, Oct. 1969.

[42] Gudvangen, S., and S. J. Flockton, "Comparison of Pole-Zero and All-Zero Modeling of Acoustic Transfer Functions," *Electronic Letters*, vol. 28, no. 21, pp. 1976-1978, Oct. 1992.

[43] Gudvangen, S., and S. J. Flockton, "Modeling of acoustic transfer functions for echo cancelers," *IEE Proc.-Vis. Image Signal Process.,*, vol. 142, no. 1, pp. 47-51, Feb. 1995.

[44] Haneda, Y., et al., "Common Acoustical Pole and Zero Modeling of Room Transfer Functions," *IEEE Trans. on Speech and Audio Processing*, vol. 2, no. 2, pp. 320-328, April 1994.

[45] Harris, R. W. et al., "The Variable Step (VS) Adaptive Filter," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-34, no. 2, pp. 309-316, April 1986.

[46] Haykin, S., *Adaptive Filter Theory*, Englewood Cliffs, NJ: Prentice-Hall, 1991.

[47] Horowitz, L. L., and K. D. Senne, "Performance Advantage of Complex LMS For Controlling Narrow-Band Adaptive Arrays," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. Assp-29, pp. 722-736, June 1981.

[48] Jagerman, D. L., "Nonstationary Blocking in Telephone Traffic," *Bell System Tech. system Journal*, vol. 54, no. 3, pp. 625-661, March 1975.

[49] Johnson, C. R. Jr., "Adaptive IIR Filtering: Current Results and Open Issues," *IEEE Trans. Inform. Theory*, vol. IT-30 no. 2, pp. 237-250, Mar. 1984.

[50] Johnson, C. R. Jr. , "On the Interaction of Adaptive Filtering, Identification, and Control," *IEEE Magazine*, pp. 22-37, Mar. 1995.

[51] Johnson, C. R. Jr., and M. G. Larimore, "Comments on and Additions on "An Adaptive Recursive LMS Filter," *Proc. IEEE*, vol. 65, no. 9, pp. 1399-1402, sept. 1977.

[52] Kellermann, W., "Analysis and Design of Multirate Systems for Cancellation of Acoustical Echoes," *Proc. ICASSP'88*, New York. pp. 2570-2573.

[53] Khasawneh, M. A., and W. E. Alexander, "System Identification Using the Fast LMS-Sine Algorithm," *Proc. IEEE ISCAS'89* , pp. 1736-1739.

[54] Khasawneh, M. A., and K. A. Mayyas, "A Newly Derived Variable Degree Variable Step Size LMS Algorithm," *Accepted for publication in the International Journal of Electronics,* Jan. 1995.

[55] Kno, S. M., and J. Chen, "Multiple Microphone Acoustic Echo Cancellation System with the Partial Adaptive Process," *Digital Signal Processing*, vol. 3, pp. 54-63, 1993.

[56] Kwong, R. H., and E. W. Johnston, "A Variable Step Size LMS Algorithm," *IEEE Trans. Signal Processing*, vol. 40, no. 7, pp. 1633-1642 , July 1992.

[57] Laichi, F., *Performance Study of the Leaky Least Mean Square Adaptive Algorithm with Delayed Adjustments*, M. A. Sc. Thesis, University of Ottawa, Ottawa, Canada, 1993.

[58] Lee, J. C., and C. K. Un, "A Reduced Structure of the Frequency-Domain Block LMS Adaptive Digital Filters," *Proc. IEEE*, vol. 72, no. 12, pp. 1816-1818, Dec. 1984.

[59] Lee, D. T. L. et al., "Recursive Least-Squares Ladder Estimation Algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 627-641, June 1981.

[60] Ljung, L. et al., "Fast Calculation of Gain Matrices for Recursive Estimation Schemes," *Int. J. Contr.*, vol. 27, pp. 1-19, Jan. 1978.

[61] Ljung, S., and L. Ljung, "Error Propagation of Recursive Least-Squares Adaptation Algorithms," *Automatica*, vol. 21, no. 2, pp. 157-167, 1985.

[62] Ljung, L., and T. Soderstrom, *Theory and Practice of Recursive Identification*, MIT Press, 1983.

[63] Lin, D. W., "On Digital Implementation of the Fast Kalman Algorithms," *IEEE Trans. on Acoust. Speech, and Signal Processing*, vol. ASSP-32, no. 5, pp. 998-1005, Oct. 1984.

[64] Makhoul, J., "Stable and Efficient Lattice Methods for Linear Predictions," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, no. 5, pp. 423-428, Oct. 1977.

[65] Makino, S. et al., "Exponentially Weighted Step Size NLMS Adaptive Filter Based on the Statistics of a Room Impulse Response," *IEEE Trans. on Speech and Audio Processing*, vol. 1, no. 1, pp. 101-108, Jan. 1993.

[66] Mansour, D., and A. H. Gray Jr., " Unconstrained Frequency-Domain Adaptive Filter," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, no. 5 pp. 726-734, Oct. 1982.

[67] Mathews, V. J., and S. H. Cho, "Improved Convergence Analysis of Stochastic Gradient Adaptive Filters Using the Sign Algorithm," *IEEE Trans. Acoust., Speech and Signal Processing*, vol. ASSP-35, no. 4, April 1987.

[68] Mathews, V. J., and Z. Xie, "A Stochastic Gradient Adaptive Filter with Gradient Adaptive Step Size," *IEEE Trans. Signal Processing*, vol. 41, no. 6, pp. 2075-2087 , June 1993.

[69] Marshall, D. F., and W. K. Jenkins, "A Fast Quasi-Newton Adaptive Filtering Algorithm," *IEEE Trans. Signal Processing*, vol. 40, no. 7, pp. 1653-1662, July 1992.

[70] Mayyas, K. A., and M. A. Khasawneh, "Application of the LMS-Order Updating in System Identification and LPC of Speech," *ECCTD'93*, pp. 1027-1032.

[71] Mayyas, K. A., and M. A. Khasawneh, "An LMS-Based Algorithm with Better Convergence and Tracking Capabilities for Correlated Signals," *ECCTD'93*, pp. 103-107.

[72] Mayyas, K., and T. Aboulnasr, "A Robust Variable Step Size LMS-Type Algorithm: Analysis and Simulations," *ICASSP'95*, pp. 1408-1411.

[73] Mayyas, K., and T. Aboulnasr, "Leaky LMS: A Detailed Analysis," *ISCAS'95*, pp. 1255-1258.

[74] Mboup, M. et al, "LMS Coupled Adaptive Prediction and System Identification: A Statistical Model and Transient Mean Analysis", *IEEE Trans. Signal Processing*, vol. 42, no. 10, pp. 2607-2615, Oct. 1994.

[75] Mboup, M., and M. Bonnet, "IIR Filtering for Acoustic Echo Cancellation," *Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 203-206, Nov. 1991.

[76] Messerschmitt, D. et. al., "Digital Voice Echo Canceler with a TMS32020".

[77] Mikhael, W. B. et al, "Adaptive Filters with Individual Adaptation of Parameters," *IEEE Trans. Circuit and Systems*, vol. CAS-33, no. 7, pp. 677-685, July 1986.

[78] Mitra, S. K. et al., "Structural Subband Implementation of Adaptive Filters," *ICASSP'90.*

[79] Muller, K. H., "A New Digital Echo Canceler for Two Wire Full-Duplex Data Transmission," *IEEE Trans. on Commun.*, vol. COMM-24, Sep. 1976.

[80] Nagumo, J., and A. Noda, "A Learning Method for System Identification ," *IEEE Trans. on Automatic Control*, vol. AC-12, no. 3, pp. 280-289, June 1967.

[81] Narayan, S. S. et al., "Transform Domain LMS algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, no. 3 pp. 609-615, June 1983.

[82] Park, S. W., "Speech Compression Using ARMA Model and Wavelet Transform," *ICASSP'94*, pp. 209-212.

[83] Pentino, M., "Frequency Domain Adaptive Correlator," *In Proc. 11th Asilomar Conf. Circuits, Systems, Computers*, Pacific Grove, CA, pp. 267-271, Nov. 1977.

[84] Petillon, T. et al., "A Fast Newton Transversal Filter: An Efficient Scheme for Acoustic Echo Cancellation in Mobile Radio," *IEEE Trans. Signal Processing*,vol. 42, no. 3, pp. 509-518, March 1994.

[85] Petraglia ,M. R., and S. K. Mitra, " Performance Analysis of Adaptive Filter Structures Based on Subband Decomposition," *ISCAS'93*, pp. 60-63.

[86] Proakis, J. G., "Channel Identification for High Speed Digital Communications," *IEEE Trans., Automat. Contr.*, vol. AC-19, pp. 918-922, Dec. 1974.

[87] Qureshi, S., "Adaptive Equalization," *IEEE Commun. Mag.*, pp. 459-467, Mar. 1982.

[88] Roy, S., and J.J Shynk, "Analysis of Momentum LMS Algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 11 pp. 2087-2098, Dec. 1990.

[89] Ryan, J. G., *Subband Adaptive Filters*, M. Eng. Thesis, Carleton University, Ottawa, Canada, Mar. 1992.

[90] Sayed, A. H., and T. Kailath, "A State-Space Approach to Adaptive RLS Filtering," *IEEE Magazine*, pp. 18-60, July 1994.

[91] Sethares, W. A. et al., " Parameter Drift in LMS Adaptive Filters," *IEEE Trans. Acoust., Signal Processing*,vol. ASSP-34, no. 4, pp. 868-876, Aug. 1986.

[92] Sethares, W. A. et al., "Analysis of Bursting in Telephone Loops with Adaptive Hybrids," *ICASSP'88.*

[93] Sethares, W. A. et al., "Bursting in Adaptive Hybrids," *IEEE Trans. on Communications*, vol. 37, no. 8, pp. 791-799, Aug. 1989.

[94] Shaffer, S., and C. S. Williams, "Comparison of the LMS, αLMS, and Data Reusing LMS Algorithms," *In Proc. Seventeenth Asilomar Conf. Circuits, Systems, Computers*, pp. 260-264, Nov. 1983.

[95] Shin, V. K., and J. Lee, "A Study of the Fast Convergence Algorithm for the LMS Adaptive Design," *Proc. of KIEE*, vol. 22, no. 6, pp. 76-82, 1985.

[96] Shynk, J. J., "Frequency-Domain and Multirate Adaptive Filtering," *IEEE Magazine*, pp. 14-35, Jan. 1992.

[97] Shynk, J. J., "Adaptive IIR Filtering," *IEEE ASSP Magazine*, pp. 4-21, April 1989.

[98] Slock, D. T. M., and T. Kailath, "Numerically Stable Fast Transversal Filters for Recursive Least Squares Adaptive Filtering," *IEEE Trans. Signal Processing*, vol. 39, no. 1, pp. 92-114, Jan. 1991.

[99] Sommen, P. L. W., et al., "Convergence Analysis of a Frequency-Domain Adaptive Filter with Exponential Power Averaging nd Generalized Window Function," *IEEE Trans. Circuit and Systems*, vol. CAS-34, no. 7, pp. 788-798, July 1987.

[100] Soo, J., and K. K. Pang, "A Multistep Size (MSS) Frequency Domain Adaptive Filter," *IEEE Trans. Signal Processing*,vol. 39, no. 1, pp. 115-121, Jan. 1991.

[101] Stearns, S. D., "Error Surfaces of Recursive adaptive Filters," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-29, no. 3, pp. 763-766, June 1981.

[102] Special Issue On Adaptive Arrays, *IEEE Trans. on Antennas Propagation*, vol. AP-24, Sept. 1976.

[103] Steiglitz, K., "On the Simultaneous Estimation of Poles and Zeros in Speech Analysis," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. ASSP-25, no. 3, pp. 229-234, June 1977.

[104] Stonick, V. L., "Time-Varying Performance Surfaces for Adaptive IIR Filters: Geometric Properties and Implications for For Filter Stability," *IEEE Trans. on. Signal Processing*, vol. 43, no. 1, pp. 29-42, Jan. 1995.

[105] Wallace, R. B., and R. A. Goubran, "Noise Cancellation Using Parallel Adaptive Filters," *IEEE Trans. on Circuits and Systems-II: Analog and Digital Processing*, vol. 39, no. 4, pp. 239-243, April 1992.

[106] Wallace, R. B., *Parallel Adaptive Noise Cancellation Filter Structures*, M. Eng. Thesis, Carleton University, Ottawa, Canada, Jan. 1991.

[107] Walzman, T., and M. Schwartz, "Automatic Equalization Using the Discrete Frequency Domain," *IEEE Trans. Inform. Theory*, vol. IT-19, no. 1, pp. 59-68., Jan. 1973.

[108] Wang, L., "Practical Adaptive Hybrids with no Bursting," MSc. Thesis, University of Ottawa, 1993.

[109] Wang, L., and T. Aboulnasr, " Practical Adaptive Hybrids with No Bursting," *ICASSP'93*, Minneeapolis, Minnoesta, April 27-30, 1993.

[110] Widrow, B. et al. , "Adaptive Noise Canceling: principle and applications," *Proc. IEEE,* vol. 63, pp. 1692-1716, Dec. 1975.

[111] Widrow, B. et al., "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, pp. 1151-1162, Aug. 1976.

[112] Widrow, B., and . M. McCool, "A Comparison of Adaptive Algorithms Based on the Method of Steepest Descent and Random Search," *IEEE Trans. on Antennas Propagation*, vol. AP-24, no.5, pp. 615-637, 1976.

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

[114] Yuan, H., *Dynamic Behavior of Acoustic Echo Cancellation*, M. Eng. Thesis, Carleton University, Ottawa, Canada, Jan. 1994.