



Pilkington Library

Author/Filing Title ZERGUINE, A.

Accession/Copy No. 040129673

Vol. No. Class Mark

*date due
for return:-
7 MAY 1998
No renewal*

LOAN COPY

0401296733





Algorithms and Structures for Long Adaptive Echo Cancellers

by

Azzedine Zerguine, B. Sc., M. Sc.


A Doctoral Thesis

Submitted in partial fulfillment of the requirements
for the Award of Doctor of Philosophy
of the Loughborough University

April 1996

Supervisor: Professor C. F. N. Cowan
Department of Electronic and Electrical Engineering

©Azzedine Zerguine 1996

 Loughborough University Pitt Rivers Library
Date Jan 97
Class
Acc No 040129673

q 6447558

Abstract

The main theme of this thesis is adaptive echo cancellation. Two novel independent approaches are proposed for the design of long echo cancellers with improved performance.

In the first approach, we present a novel structure for bulk delay estimation in long echo cancellers which considerably reduces the amount of excess error. The miscalculation of the delay between the near-end and the far-end sections is one of the main causes of this excess error. Two analyses, based on the Least Mean Squares (LMS) algorithm, are presented where certain shapes for the transitions between the end of the near-end section and the beginning of the far-end one are considered. Transient and steady-state behaviours, and convergence conditions for the proposed algorithm are studied. Comparisons between the algorithms developed for each transition are presented, and the simulation results agree well with the theoretical derivations.

In the second approach, a generalised performance index is proposed for the design of the echo canceller. The proposed algorithm consists of simultaneously applying the LMS algorithm to the near-end section and the Least Mean Fourth (LMF) algorithm to the far-end section of the echo canceller. This combination results in a substantial improvement of the performance of the proposed scheme over both the LMS and other algorithms proposed for comparison. In this approach, the proposed algorithm will be henceforth called the Least Mean Mixed-Norm (LMMN) algorithm.

The advantages of the LMMN algorithm over previously reported ones are two folds: it leads to a faster convergence and results in a smaller misadjustment error.

Finally, the convergence properties of the LMMN algorithm are derived and the simulation results confirm the superior performance of this proposed algorithm over other well known algorithms.

Acknowledgements

I would like to express my deep appreciation to my thesis advisor, Professor Colin Cowan, for his patient guidance and his generous support for this research.

I would also like to express my gratefulness to Dr. Maamar Bettayeb, my local supervisor, for his valuable suggestions and his helpful remarks, as well as Dr. Lahouari Cheded for his comments.

Special thanks are to Dr. Abdulaziz Al-Harhi, Dean, College of Sciences at KFUPM, Dr. Mohammed Garwan, Chairman of the Physics Department at KFUPM, and to numerous people during the preparation of this thesis.

Finally, I would like to thank my parents and all the members of my family whose encouragements made it possible. Also, the support from my wife and my children Dalila, Asma, Abdelhakim, and Imane is unforgettable.

Contents

1	Introduction	1
1.1	Introduction	1
1.2	Organisation of thesis	4
2	Adaptive Filtering: Algorithms and Structures	6
2.1	Introduction	6
2.2	Applications of adaptive filters	7
2.2.1	System identification	8
2.2.2	Inverse modelling	8
2.2.3	Prediction	8
2.2.4	Noise cancellation	10
2.3	Adaptive filters	10
2.3.1	Cost functions	11
2.3.2	Structures	14
2.3.3	The FIR adaptive filter	14
2.4	Adaptive Algorithms	17
2.4.1	The LMS algorithm	18
2.4.2	The LMF algorithm	20
2.4.3	The RLS algorithm	21
2.5	Summary	23

3	Echo Cancellation Techniques	26
3.1	Introduction	26
3.2	Echo cancellers	30
3.2.1	Echo cancellers in speech communication	30
3.2.2	Echo cancellers in data transmission	33
3.2.3	Differences between speech and data echo cancellers	38
3.3	Filter structures	39
3.4	Adaptive mechanisms	41
3.5	Problems encountered in echo cancellation	43
3.6	Summary	44
4	Adaptive Echo Cancellation Using Statistical Shaping	45
4.1	Introduction	45
4.2	Algorithm development	48
4.2.1	The optimal solution	53
4.2.2	The updating scheme	53
4.3	Convergence behaviour of the algorithm when using the sharp tran- sition	54
4.3.1	Convergence in the mean	54
4.3.2	Convergence in the mean square	57
4.3.3	The steady MSE	59
4.3.4	The misadjustment factor	60
4.3.5	The time constant of the algorithm	60
4.4	Convergence behaviour of the algorithm when using the smooth transition	60
4.4.1	Convergence in the mean	63
4.4.2	Convergence in the mean square	66
4.4.3	The steady state MSE	68
4.4.4	The misadjustment factor	69

4.4.5	The time constant of the algorithm	69
4.5	Excess mean square error	69
4.6	Computational complexity	72
4.7	Simulation results	73
4.8	Summary	75
5	Adaptive Echo Cancellation Using Least Mean Mixed-Norm Al-	
	gorithm	86
5.1	Introduction	86
5.2	Derivation of the algorithm	90
5.2.1	The proposed performance functions	90
5.2.2	Analysis of the error surfaces	93
5.2.3	The optimal solution	94
5.2.4	The updating scheme	95
5.3	Convergence behaviour of the LMMN algorithm	97
5.3.1	Convergence in the mean	97
5.3.2	Convergence in the mean square	101
5.3.3	Time constants of the algorithm	108
5.3.4	The misadjustment factor	108
5.4	Summary	110
6	Performance of the Least Mean Mixed-Norm Adaptive Algo-	
	rithm	111
6.1	Introduction	111
6.2	Convergence behaviour of algorithm I	113
6.2.1	The performance function	115
6.2.2	The updating scheme	115
6.2.3	Convergence in the mean	115
6.2.4	Time constants of the algorithm	116
6.2.5	The misadjustment factor	117

6.3	Convergence behaviour of algorithm II	117
6.3.1	The performance function	117
6.3.2	The updating scheme	118
6.3.3	Convergence in the mean	119
6.3.4	Time constants of the algorithm	119
6.3.5	The misadjustment factor	119
6.4	Convergence behaviour of algorithm III	120
6.4.1	The performance function	120
6.4.2	The updating scheme	120
6.4.3	Convergence in the mean	122
6.4.4	Time constants of the algorithm	122
6.4.5	The misadjustment factor	122
6.5	Computational complexity	123
6.6	Simulation results	126
6.6.1	Performance of the LMMN algorithm	127
6.6.2	The effect of the noise variance	138
6.6.3	The effect of the noise distribution	142
6.6.4	The rescue condition	146
6.7	Summary	146
7	Conclusion	148
7.1	Achievements of the work	148
7.2	Summary of main contributions	150
7.3	Suggestions for further work	150
	Bibliography	152
A	Publications	163

List of Tables

4.1	Comparison of the algorithm for sharp and smooth transitions.	70
4.2	Computational complexity of the algorithms with different transitions.	74
5.1	The mathematical description of the LMMN algorithm.	96
5.2	The main parameters of the LMMN algorithm.	109
6.1	The mathematical description of algorithm I.	116
6.2	The mathematical description of algorithm II.	118
6.3	The mathematical description of algorithm III.	121
6.4	Main parameters for the LMS, the LMF, and the LMFS algorithms.	124
6.5	Computational complexity of the LMS, the LMF, the LMFS, and the LMMN algorithms.	125

List of Figures

2.1	Direct system modelling configuration of an adaptive filter.	9
2.2	Inverse system modelling configuration of an adaptive filter.	9
2.3	Configuration of an adaptive filter as a predictor.	9
2.4	Configuration of an adaptive filter as a noise canceller.	11
2.5	General form of an adaptive filter.	12
2.6	Structure of a linear transversal FIR filter.	15
2.7	Structure of an IIR filter.	15
2.8	Tree structure for adaptive algorithm development.	24
3.1	Long-distance telephone connection.	27
3.2	Hybrid circuit.	28
3.3	Simplified echo suppressor logic.	29
3.4	Structure of a voice telephone echo canceller.	31
3.5	Echo cancellers at station locations for data transmission.	34
3.6	Echo canceller with bulk delay.	35
3.7	Voice-type echo canceller.	36
3.8	Data-type echo canceller.	36
4.1	Structure of an echo canceller with bulk delay.	47
4.2	FIR filter.	49
4.3	Probability density function of the tap coefficients belonging to the bulk delay.	51
4.4	Sharp and smooth transitions.	62

4.5	Impulse response of the channel.	77
4.6	Learning curves when using the sharp transition and no error in the bulk delay.	78
4.7	Learning curves when using the smooth transition and 4 coefficients in each of the near-end and the far-end sections of the canceller have been missed.	79
4.8	MSE versus the number of missed coefficients in the near-end section of the canceller when using the sharp transition.	80
4.9	MSE versus the number of missed coefficients in the far-end section of the canceller when using the sharp transition.	81
4.10	MSE versus the number of missed coefficients in the near-end and the far-end sections of the canceller when using the sharp transition.	82
4.11	MSE versus the number of missed coefficients in the near-end section of the canceller when using the smooth transition.	83
4.12	MSE versus the number of missed coefficients in the far-end section of the canceller when using the smooth transition.	84
4.13	MSE versus the number of missed coefficients in the near-end and the far-end sections of the canceller when using the smooth transition.	85
5.1	Echo canceller with new updating scheme.	89
5.2	Block diagram of adaptive system identification.	92
6.1	System considered for the adaptive system identification.	114
6.2	Learning curves for the four algorithms for channel 1.	128
6.3	Learning curves for the four algorithms for channel 2.	129
6.4	Learning curves for the four algorithms for channel 3.	130
6.5	Theoretical and experimental learning curves of the LMMN algorithm for channel 1.	132
6.6	Theoretical and experimental learning curves of the LMMN algorithm for channel 2.	133

6.7	Theoretical and experimental learning curves of the LMMN algorithm for channel 3.	134
6.8	Learning curves for the LMMN and the LMS algorithms with the same steady state value used for channel 1.	135
6.9	Learning curves for the LMMN and the LMS algorithms with the same steady state value used for channel 2.	136
6.10	Learning curves for the LMMN and the LMS algorithms with the same steady state value used for channel 3.	137
6.11	Effect of noise variance on the convergence behaviour of the LMMN algorithm for channel 1.	139
6.12	Effect of noise variance on the convergence behaviour of the LMMN algorithm for channel 2.	140
6.13	Effect of noise variance on the convergence behaviour of the LMMN algorithm for channel 3.	141
6.14	Effect of noise distribution on the convergence behaviour of the LMMN algorithm for channel 1.	143
6.15	Effect of noise distribution on the convergence behaviour of the LMMN algorithm for channel 2.	144
6.16	Effect of noise distribution on the convergence behaviour of the LMMN algorithm for channel 3.	145

Abbreviations

BLMS	Block Least Mean Squares
DPCM	Differential Pulse Code Modulation
DS	Delay Section
DSP	Digital Signal Processing
FDM	Frequency Division Multiplexing
FFT	Fast Fourier Transform
FE	Far-End
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
LLMS	Leakage Least Mean Squares
LMF	Least Mean Fourth
LMFS	Least Mean Fourth-Squares
LMMN	Least Mean Mixed-Norm
LMS	Least Mean Squares
LPC	Linear Predictive Coding
LS	Least Squares
MFE	Mean Fourth Error
MMSE	Minimum Mean Square Error
MSE	Mean Square Error
NE	Near-End
NLMS	Normalised Least Mean Squares
RLS	Recursive Least Squares
SG	Stochastic Gradient
SLMS	Sign Least Mean Squares
VLSI	Very Large Scale Integration
i.i.d	independent and identically distributed
pdf	probability density function

Principal Symbols

$d(n)$	desired response
$e(n)$	error between the desired response and the output
$H(z)$	transfer function of the unknown system
$J(n)$	cost function
J_{min}	minimum mean square error
M	misadjustment factor
N	filter length
$y(n)$	output of the adaptive filter
$w(n)$	additive noise
$E[w^2(n)]$	variance of the noise
$E[\]$	statistical expectation
$S_x(f)$	power spectrum of the input signal
$C(n)$	coefficient vector of the adaptive filter
C_{opt}	optimum filter coefficients
$\hat{C}(n)$	coefficient error vector
H	coefficient vector of the unknown system
I	identity matrix
P	crosscorrelation vector
Q	orthonormal matrix
R	autocorrelation matrix
$X(n)$	vector of the last N samples
Λ	diagonal matrix of eigenvalues
$tr(\)$	trace of a matrix
κ_w	kurtosis of the noise
μ	step size
λ_i	i th eigenvalue

λ_{max}	largest eigenvalue
λ_{min}	smallest eigenvalue
$\nabla J(n)$	gradient of the error norm $J(n)$
$\hat{\nabla} J(n)$	instantaneous estimate of the gradient
$\rho(n)$	normalised weight error norm
σ_x^2	power of the input signal
σ_e^2	power of the error
τ_i	<i>i</i> th time constant
τ_{max}	largest time constant

Chapter 1

Introduction

1.1 Introduction

Applications of digital signal processing (DSP) are currently abundant in the field of communications. They are expected to grow even more with the advent of digital communications and the availability of inexpensive DSP chips.

The birth of the least mean squares (LMS) [1] algorithm gave a greater prominence to the adaptive filtering field. Since then, the increasing interest in this algorithm led to the development of now-well established algorithms related to the least squares (LS) [2] technique. The main reasons for this success of the LMS algorithm are its simplicity, its easy implementation, and its numerical robustness.

Adaptive filtering is used to solve a variety of transmission problems. Cancelling echoes [3] in voice and data communications and providing channel equalisation [4] are only a few examples. In the former application, adaptive echo cancellers are constantly invading telephone systems and have almost replaced echo suppressors which were originally designed to suppress echoes in speech. Also, their use for data echo cancellation gained a lot of attention since they were capable of providing high data rates between the transmitter and receiver, an area where the use of multiplexing generally proved to be an inefficient solution.

However, a serious difficulty, unique to data echo cancellation, is the very large delay exhibited by the far echo returning from the far end of the circuit. This would entail implementing echo cancellers with a large number of tap coefficients if, for example, the far echo is delayed by some hundreds of milliseconds.

A solution to this problem is the use of a bulk delay [5] which separates the echo canceller into two separate adaptive transversal filters, one for each end of the echo canceller. Unfortunately this delay is not accurately estimated, which leads to the adaptive algorithm not converging to the exact solution, and ultimately ending up with an uncancelled echo.

To overcome this problem, transitions between the sections of the canceller, whose shapes are controlled by certain probability density functions (pdf) are proposed in this work. An enhancement in performance is obtained through the use of this technique. The trivial solution to this problem, of course, would be achieved by adding a certain number of coefficients in these regions. This, however, would result in adding extra computations.

The new proposed scheme [6] is based on the LMS algorithm where a probabilistic approach is used to analyse the modified LMS algorithm. Two different transitions between the near-end and the far-end sections of the echo canceller are used in this study. These consist of a sharp transition and a smooth one. The latter transition is chosen according to the uniform probability density function.

Also, the performance of an adaptive filter is mainly governed by the algorithm used for updating its filter coefficients. Since adaptive filters primarily depend on the choice of the cost function used in the minimisation process, one can expect their respective performances to be different for different criteria.

For example, the recursive least-squares (RLS) algorithm [7] is designed to minimise the sum of the squares of the errors (the difference between the desired and the estimated values). However, algorithms based on the least mean square error consist of minimising the square of the error, and those based on the mean fourth power of the error minimise the fourth power of the error. It is the latter crite-

tion that gives the algorithm its name of the least mean fourth (LMF) algorithm [8]. Both the LMS and the LMF algorithms belong to the family of stochastic gradient-based algorithms [7].

As can be seen from the above mentioned algorithms, each cost function plays a key role in the performance of its corresponding algorithm. The structure used in implementing these algorithms is of importance as well.

Since the LMS algorithm is only optimal if the input signal is white, its convergence will be the fastest for this case only. However, if the input signal is not white, as is likely to be the case in real life applications, the convergence behaviour and speed of the LMS algorithm would differ from those of the ideal case. Newly developed algorithms [9], [10] based on a controlled combination of the LMS and the LMF algorithms are proposed, resulting in faster convergence rates and lower misadjustment errors. These are designed to minimise a single combined cost function.

In this work, a new algorithm applied to long echo cancellers is proposed. It is different from that of [9] and [10] in the sense that we will have two distinct cost functions to be minimised, one for each section of the echo canceller. These consist of minimising the MSE in the near-end section of the echo canceller and the mean fourth-error (MFE) in its far-end section. This algorithm leads to a smaller minimum mean square error (MMSE) and a faster convergence compared to the standard algorithm, i.e., the LMS algorithm.

Another possible configuration would involve exchanging the two cost functions, that is applying the MFE to the near-end section and the MSE to the far-end section. This will also be investigated, and it will be shown that, unfortunately, this choice will result in a poor performance.

This work then seeks to address the problem of improving the performance of stochastic gradient (SG) methods, while maintaining the relative simplicity of their implementation. Methods of increasing the convergence rate and simplifying the implementation of the adaptive filter used for echo cancellation, with little cost

in steady-state error, will be the focus of our work. The algorithmic modifications for each proposed scheme will be investigated.

It should be noted that the computational requirements for the two proposed algorithms, as explained later, are not very demanding and hence are not prohibitive as far as the implementation of these algorithms is concerned.

1.2 Organisation of thesis

This thesis investigates the performance of various algorithms for stochastic gradient adaptive filters applied to long data echo cancellers. As mentioned before, the continuing need for adaptive filters in a wide variety of applications (especially communications, signal processing and control) is attracting much research into ways of improving their performance. The remainder of this thesis consists of six chapters.

Chapter 2, concentrates on the commonly-used adaptive algorithms and their various structures. One of their main applications is in echo cancellation [11].

The investigation of echo cancellation techniques is treated in Chapter 3. Differences and similarities between echo cancellers in speech and data are reviewed. Some of the problems encountered in data echo cancellers are discussed and the suggested solutions are treated in Chapters 4 and 5.

Chapter 4 deals with a possible solution to the bulk delay problem. The bulk delay is inserted between the near-end and the far-end sections of the echo canceller. If the performance of the standard algorithm, i.e., the LMS algorithm, is to be enhanced its structure then needs to be modified. To this end, two analyses are presented based on the use of transitional shaping across the bulk delay area. Sharp and smooth transitions are used for this purpose. An enhancement in performance is obtained when the latter transition is used. The choice of the bulk delay coefficients is done according to the uniform probability density function. A lower mean square error is obtained in this case.

The use of two different error norms in the same filter can also improve both the convergence speed and the steady state MSE. Our application of the LMS and the LMF algorithms to the near-end and the far-end sections of the echo canceller, respectively, results in a novel algorithm that will be henceforth called the least mean mixed-norm (LMMN) algorithm. A thorough analysis of this algorithm is reported in Chapter 5, where its convergence behaviour is also studied.

The comparison between the LMMN algorithm and other algorithms reported in the literature is carried out in Chapter 6. Specifically, these algorithms are the LMS, the LMF, and the least mean fourth-square (LMFS) algorithms. The latter algorithm, i.e., the LMFS, is similar to our LMMN in that it is based on the minimisation of two cost functions. The LMFS, however, minimises the MFE in the near-end section of the echo canceller and the MSE in its far-end section.

Finally, Chapter 7 summarises the main results obtained in this work and highlights some key problems which are yet to be investigated in the area of adaptive echo cancellation.

Chapter 2

Adaptive Filtering: Algorithms and Structures

2.1 Introduction

Adaptive systems are playing a vital role in the development of modern telecommunications. Also, adaptive systems proved to be extremely effective in achieving high efficiency, high quality and high reliability of around-the-world ubiquitous telecommunication services.

The role of adaptive systems is wide spread covering almost all aspects of telecommunication engineering, but perhaps most notable in the following context [12] of ensuring high-quality signal transmission over unknown and time varying channels.

Interest in adaptive filters continues to grow as they begin to find practical real-time applications in areas such as echo cancellation [11], channel equalisation [13], noise cancellation [14]-[15] and many other adaptive signal processing applications. This is due mainly to the recent advances in the very large-scale integration (VLSI) technology.

The key to successful adaptive signal processing is understanding the funda-

mental properties of adaptive algorithms. These properties are stability, speed of convergence, misadjustment errors, robustness to both additive noise and signal conditioning (spectral colouration), numerical complexity, and round-off error analysis of adaptive algorithms. However, some of these properties are often in direct conflict with each other, since consistent fast converging algorithms tend to be in general more complex and numerically sensitive. Also, the performance of any algorithm with respect to any of these criteria is entirely dependent on the choice of the adaptation update function, that is the cost function used in the minimisation process. A compromise must be than reached among these conflicting factors to come up with the appropriate algorithm for the concerned application.

After presenting, in Section 2.2, the common adaptive system configurations using adaptive filters, Section 2.3 will deal with a more explicit development of adaptive filters. Performance evaluation of the resulting algorithms using the properties of the finite impulse response (FIR) adaptive filter are also mentioned.

Finally, Section 2.4 reviews the theory of adaptive algorithms used with these filters, including the least mean squares (LMS) algorithm. Also, the shortcomings of the LMS algorithm are discussed.

2.2 Applications of adaptive filters

Adaptive filtering has been successfully applied in such diverse fields as communications, radar, sonar, and biomedical engineering. Although these applications are indeed quite different in nature, nevertheless, they have one basic common feature: an input signal and a desired response are used to compute the error, which is in turn used to control the values of a set of adjustable filter coefficients. However, the main difference among the various applications of adaptive filtering arises in the manner in which the desired response is extracted.

In this context, we may classify an adaptive filter into one of the four following categories:

2.2.1 System identification

In this first application, depicted in Fig. 2.1, the adaptive filter is used to provide a linear model that represents the best fit to the unknown system. Both the adaptive filter and the unknown system are driven by the same input. The error estimate is used to update the filter coefficients of the adaptive filter. After convergence, the adaptive filter output will approximate the output of the unknown system in an optimum sense. Provided that the order of the adaptive filter matches that of the unknown system and the input, $x(n)$, is broad band (flat spectrum) this will be achieved by convergence of adaptive filter coefficients to the same values as the unknown system.

The major practical use of this structure in telecommunications is for echo cancellation [11], [16]-[17]. Typically, the input signal $x(n)$ will be either speech or data.

2.2.2 Inverse modelling

In this second class of applications, the function of the adaptive filter is to provide an inverse model that represents the best fit to the unknown system. Thus, at convergence, the adaptive filter transfer function approximates the inverse of the transfer function of the unknown system. As can be seen from Fig. 2.2, the desired response is a delayed version of the input signal.

The primary use of inverse system modelling is for reducing the effects of intersymbol interference (ISI) in digital receivers. This is achieved through the use of equalisation [13], [18] techniques.

2.2.3 Prediction

In this structure, the function of the adaptive filter is to provide the best prediction of the present value of the input signal from its previous values. The configuration shown in Fig. 2.3 is used for this purpose, where the desired signal, $d(n)$, is the

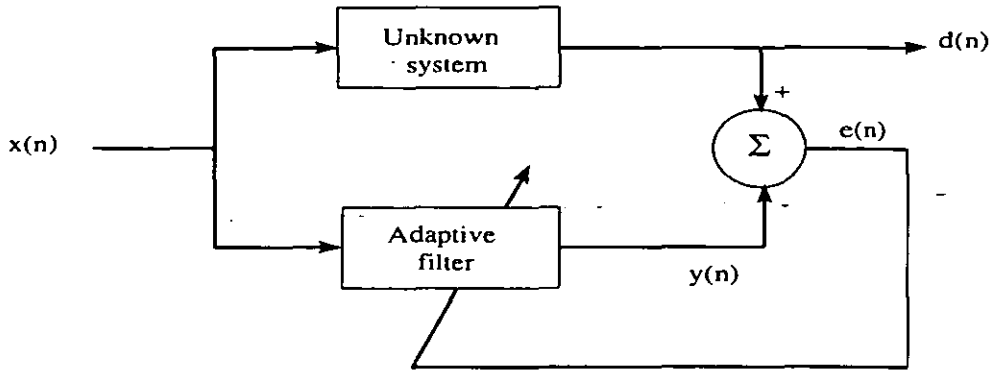


Figure 2.1: Direct system modelling configuration of an adaptive filter.

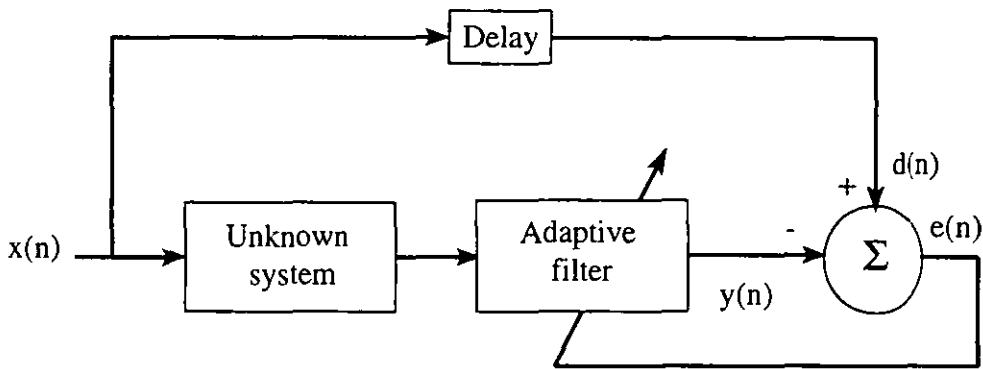


Figure 2.2: Inverse system modelling configuration of an adaptive filter.

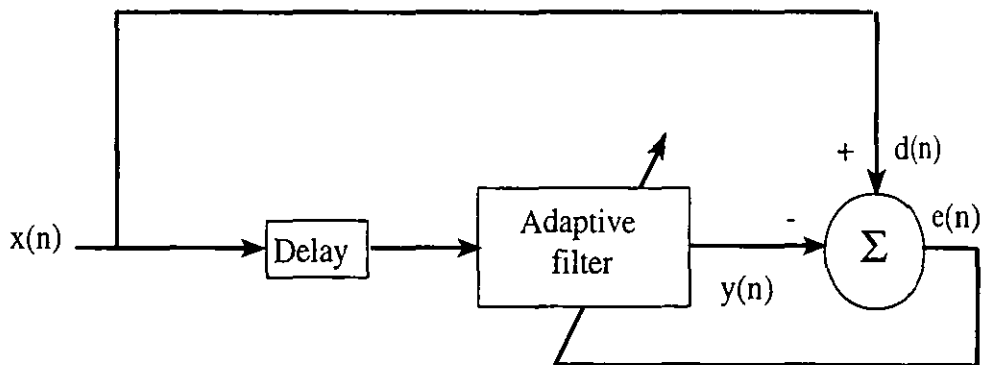


Figure 2.3: Configuration of an adaptive filter as a predictor.

instantaneous value and the input to the adaptive filter is a delayed version of the same signal.

This application is widely used in linear predictive coding (LPC) of speech [19]-[20] and in adaptive differential pulse-code modulation (DPCM) [21]. Another approach to prediction is given in [22].

2.2.4 Noise cancellation

In this final class of applications, the adaptive filter is used to cancel unknown interference contained in a primary signal, as Fig. 2.4 depicts it. The primary signal serves as the desired response of the adaptive filter. This type of application is used in adaptive noise cancellation [14]-[15], and adaptive beamforming or adaptive array processing [23].

The principle operation of the adaptive filter in all the four cases is mainly the same, and for the purposes of further development only the case of system identification will be considered. Also, interest in this configuration is related to the type of application we are dealing with in this thesis, namely echo cancellation.

2.3 Adaptive filters

Adaptive filters are an important part of signal processing. They are generally defined as filters whose characteristics can be modified to achieve desired objectives and accomplish this modification or adaptation automatically without user intervention. Due to the uncertainty about the input signal characteristics, the designer then uses an adaptive filter which can learn the signal characteristics when first turned on and can later track changes in these characteristics. Adaptive algorithms are responsible for the learning process.

A large number of algorithms for adaptive filters have been proposed. Indeed, adaptive filtering is an example of an optimisation problem and optimisation techniques form an important part of mathematics [24]-[26]. The additional constraint

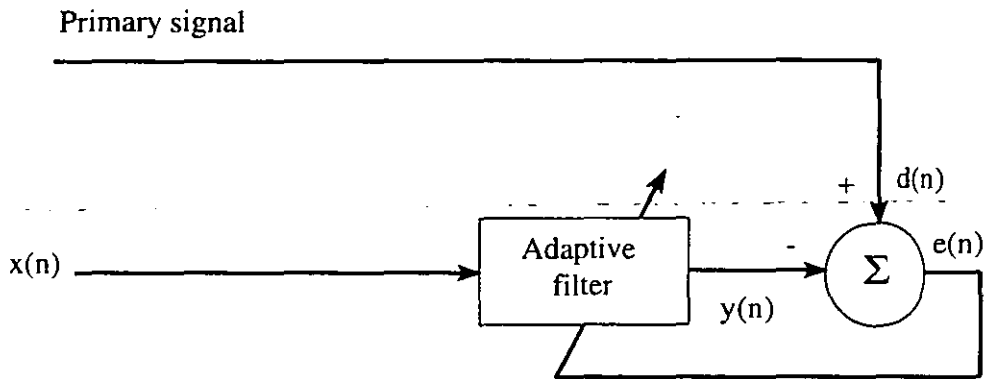


Figure 2.4: Configuration of an adaptive filter as a noise canceller.

in adaptive filtering is that many of the applications require this optimisation to be performed in real time and so the complexity of the computations must be kept to a minimum.

In what is remaining of this section, different cost functions for adaptive filters are defined with some of the possible structures used in their implementation, and the expression for the optimum FIR filter in the mean square error (MSE) sense is given in terms of autocorrelation and crosscorrelation functions [27].

2.3.1 Cost functions

Before proceeding to discuss any adaptive algorithm, it is necessary to discuss the performance measure (cost function) which is used in adaptive filtering. The adaptive filter has the general form shown in Fig. 2.5, where the FIR filter of order N is considered here. The filter output $y(n)$ is given by

$$\begin{aligned} y(n) &= \sum_{i=0}^{N-1} C_i(n)x(n-i) \\ &= \mathbf{C}^T(n)\mathbf{X}(n), \end{aligned} \quad (2.1)$$

where $\mathbf{X}(n)$ and $\mathbf{C}(n)$ are, respectively, the vector of the last N samples from the time series $x(n)$ and the filter coefficients at sample n , defined as follows:

$$\mathbf{X}^T(n) = [x(n), x(n-1), \dots, x(n-N+1)]. \quad (2.2)$$

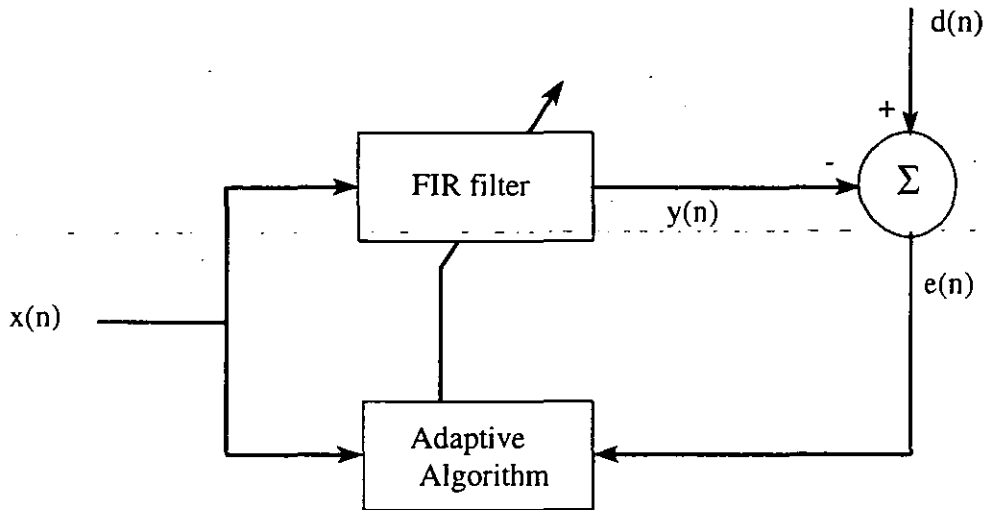


Figure 2.5: General form of an adaptive filter.

and

$$\mathbf{C}^T(n) = [C_0(n), C_1(n), \dots, C_{N-1}(n)], \quad (2.3)$$

where T denotes transpose.

In general, adaptive techniques have been classified under two main categories. In one category, the cost function to be optimised in a running sum of squared errors is given by:

$$J(n) = \sum_{j=0}^n e^2(j), \quad (2.4)$$

where the error $e(n)$ is defined to be the difference value between the desired response $d(n)$ and the output of the adaptive filter $y(n)$, that is,

$$e(n) = d(n) - y(n). \quad (2.5)$$

The approach, defined by (2.4), is based on the method of least squares [2], [28]-[29], which contains the whole class of recursive least squares (RLS) algorithms [7], [16], [30]-[32].

In the other category, the cost function to be optimised is a statistical measure of the squared error, known as the mean squared-error (MSE) [33]. This cost

function is given by

$$J(n) = E[e^2(n)], \quad (2.6)$$

where $E[\]$ denotes statistical expectation. This category contains the whole class of gradient algorithms, which includes the least mean-squared (LMS) algorithm [1], [7], [16].

The two procedures described above for deriving adaptive algorithms differ in some respect on how their respective cost function is chosen. The theory for the Wiener filter is based on statistical concepts, while it is based on the use of time averages for the method of least squares. Also, least squares techniques and stochastic techniques have a number of differences in the way that they perform [34]. Among these differences are on the one hand the much longer time taken for a stochastic gradient algorithm to converge close to the optimum solution and on the other hand the much higher computational complexity in least squares algorithms. Nevertheless, the less computational complexity in the stochastic gradient methods make them much more attractive than their least squares counterparts.

Recently, other minimisation criteria have emerged, in which adaptive structures are derived from minimisation of a class of functions of the form [8]:

$$J^k(n) = E[e^{2k}(n)]; \quad k \geq 1, \quad (2.7)$$

where k is an integer constant. It is seen from (2.7) that when $k = 1$, the usual MSE criterion is obtained, while the mean fourth-error (MFE) results when $k = 2$.

The cost functions, (2.4) and (2.6), are both convex with a unique minimum point. Accordingly, their use yields a unique solution for the coefficient vector of the FIR filter. Also, the minimisation function, (2.7), is a convex function, and therefore has no local minima. Hence, the use of a gradient based adaptation scheme for the convergence to the minimum can be applied.

Finally, before stating the possible linear structures used in implementing adaptive filters, it is worth mentioning the properties of the cost functions. All the functions presented in this section and others not mentioned in this work should

be positive and monotonically increasing [35] for their corresponding algorithms to perform correctly.

2.3.2 Structures

A number of different linear structures for adaptive systems have been proposed, which may be subdivided into finite and infinite impulse structures. For the finite impulse response [36]-[38], the transfer function is realised by zeros only, as all the poles of the filter are located at the origin. In the case of the infinite impulse response (IIR) [16], [39] filter, however, both poles and zeros are used to realise the transfer function. Examples of the FIR filter are the linear transversal filter depicted in Fig. 2.6, and the lattice filter [40]-[42]. The structure of the IIR filter is shown in Fig. 2.7. However, difficulties associated with developing adaptive techniques for IIR filter are considerable, because the filter is not unconditionally stable, as it has both poles and zeros in its transfer function. The danger is that the adaptive algorithm will choose a set of coefficients which may place poles outside the unit circle in the z -plane and so provoke an unstable response. These difficulties, hence, make the IIR structure less attractive than the well established FIR one.

The work of this thesis is, therefore, concerned with the linear transversal filter structure and the emphasis is on developing highly efficient algorithms for this well understood and often used structure.

2.3.3 The FIR adaptive filter

Assuming that the input sequence $\{x(n)\}$ and the desired sequence $\{d(n)\}$ are in wide sense stationary, the mean-square-error function, equation (2.6), can be more conveniently expressed in terms of the input autocorrelation matrix, \mathbf{R} , and the crosscorrelation vector, \mathbf{P} , between the desired response and the input components,

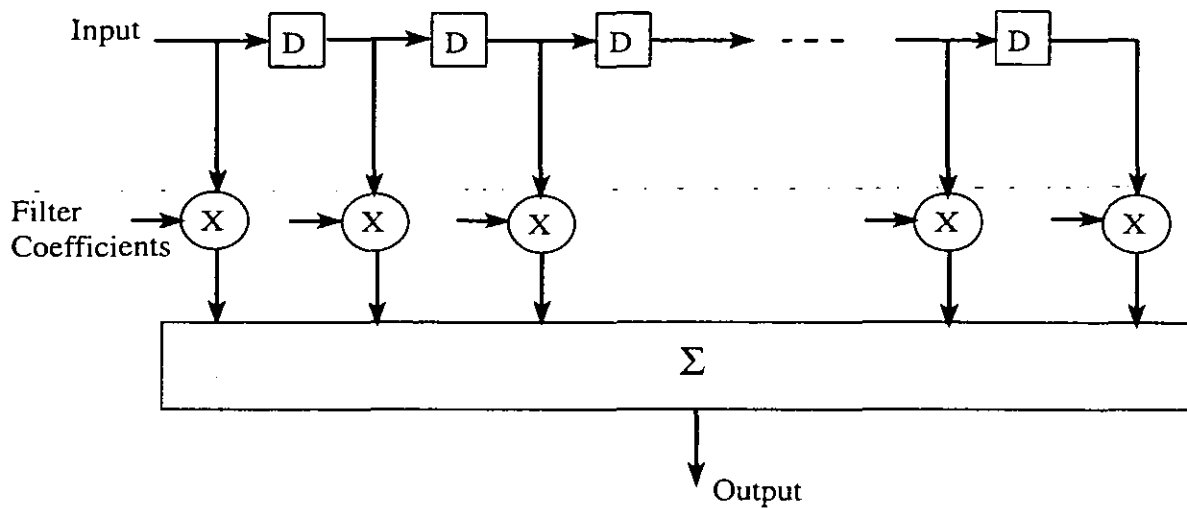


Figure 2.6: Structure of a linear transversal FIR filter.

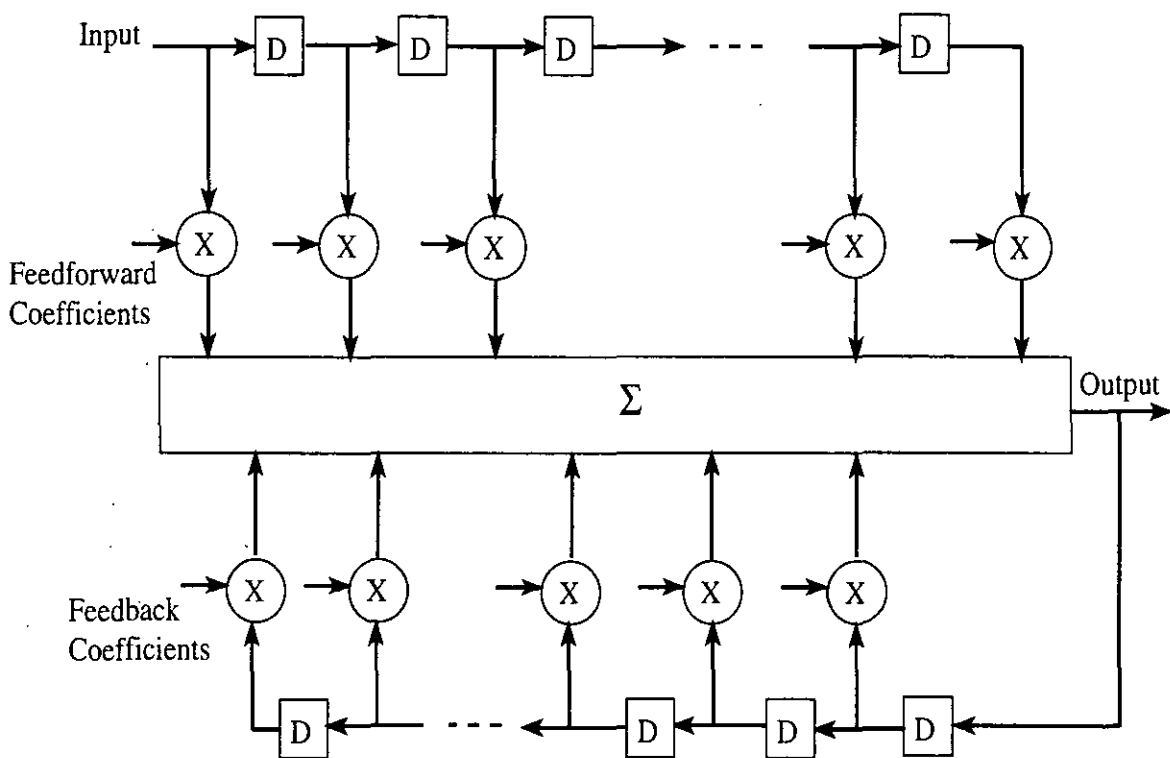


Figure 2.7: Structure of an IIR filter.

as follows:

$$J(n) = E[d^2(n)] - 2\mathbf{C}^T(n)\mathbf{P} + \mathbf{C}^T(n)\mathbf{R}\mathbf{C}(n), \quad (2.8)$$

where

$$\mathbf{R} = E[\mathbf{X}(n)\mathbf{X}^T(n)], \quad (2.9)$$

and

$$\mathbf{P} = E[\mathbf{X}(n)d(n)]. \quad (2.10)$$

It is clear from expression (2.8) that the MSE is precisely a quadratic function of the components of the tap coefficients. Thus, the shape associated with this MSE is hyperboloid.

In general, for the linear transversal structure, the surface will be quadratic, when the MSE is used, with a single global minimum. The goal of an adaptive algorithm is to set the filter coefficients so as to obtain an operating point at this minimum, where the filter gives optimum performance.

The point at the bottom of the performance surface corresponds to the optimal tap coefficients, \mathbf{C}_{opt} , or minimum MSE. The gradient method is used to cause the tap coefficients vector to seek the minimum of the performance surface. It is defined as

$$\begin{aligned} \nabla E[e^2(n)] &= \frac{\partial J(n)}{\partial \mathbf{C}(n)} \\ &= \left[\frac{\partial J(n)}{\partial C_0(n)}, \frac{\partial J(n)}{\partial C_1(n)}, \dots, \frac{\partial J(n)}{\partial C_{N-1}(n)} \right]^T \\ &= -2E[e(n)\mathbf{X}(n)] \\ &= 2\mathbf{R}\mathbf{C}(n) - 2\mathbf{P}. \end{aligned} \quad (2.11)$$

To obtain the minimum MSE, the tap-coefficients vector $\mathbf{C}(n)$ is set to its optimal value, \mathbf{C}_{opt} , where the gradient is zero, that is,

$$\nabla E[e^2(n)] = \mathbf{R}\mathbf{C}_{opt} - \mathbf{P} = \mathbf{0}. \quad (2.12)$$

Under this condition, the optimum value is given by:

$$\mathbf{C}_{opt} = \mathbf{R}^{-1}\mathbf{P}, \quad (2.13)$$

where this is obtained under the assumption that the autocorrelation matrix \mathbf{R} of the input signal is positive definite and hence non singular. Properties of the the autocorrelation matrix \mathbf{R} of the input signal can be found in [7]. The minimum MSE , J_{min} , is hence obtained by substitution of (2.13) in (2.8), that is,

$$J_{min} = E[d^2(n)] - \mathbf{C}_{opt}^T \mathbf{P}. \quad (2.14)$$

The solution for \mathbf{C}_{opt} involves inverting the input autocorrelation matrix \mathbf{R} , hence, requiring precise knowledge of the second order statistics of the data, i.e., the autocorrelation matrix and the crosscorrelation vector. Unfortunately, it is the data sequences rather than their second order statistics that are available in practice. Alternatively, an iterative procedure may be used to determine \mathbf{C}_{opt} . This is the function of an adaptive FIR filter algorithm which has to find the optimum filter from available data rather than from the second statistics of the data [43]. Thus, an adaptive FIR filter can be defined as an algorithm which operates on the sequences $\{x(n)\}$ and $\{d(n)\}$ to form a time-varying impulse response $\mathbf{C}(n)$ which converges in the mean to \mathbf{C}_{opt} as the number of iterations becomes very large, that is:

$$\lim_{n \rightarrow \infty} E[\mathbf{C}(n)] = \mathbf{C}_{opt}. \quad (2.15)$$

2.4 Adaptive Algorithms

In the previous section it was shown that the optimum tap coefficient vector for the adaptive FIR filter could be defined by the statistical properties of the input and desired signals. This implies that if these properties were known then the optimum tap coefficients could be obtained directly. However, it is unlikely to have an accurate measurement, they may be varying with time and the matrix inversion would require considerable amount of computations, specifically if there were a large number of coefficients. Practical adaptation algorithms usually involve iterative techniques. The following gives the two most widely used adaptive

algorithms suitable for practical real time applications. These are the least mean-squares (LMS) algorithm and the recursive least-squares (RLS) algorithm. The least mean fourth (LMF) algorithm is also highlighted.

2.4.1 The LMS algorithm

Probably the simplest iterative procedure is the method of steepest descent defined according to the following relation [7]

$$\mathbf{C}(n+1) = \mathbf{C}(n) - \frac{1}{2}\mu\nabla E[e^2(n)], \quad (2.16)$$

where μ is a positive number chosen small enough to insure convergence of the iterative procedure.

Given that the gradient vector, $\nabla E[e^2(n)]$, depends on both the input autocorrelation matrix, \mathbf{R} , and the vector \mathbf{P} of cross correlations, this makes the steepest descent difficult for determining the optimum tap coefficients. Instead, estimates of the gradient vector may be used. That is, the LMS algorithm for recursively adjusting the tap coefficient of the adaptive filter is expressed in the form

$$\mathbf{C}(n+1) = \mathbf{C}(n) + \mu e(n)\mathbf{X}(n). \quad (2.17)$$

The convergence behaviour of the LMS algorithm given in equation (2.17) is governed by the step size parameter μ . For a larger value of μ , the convergence becomes faster, but it results in a larger residual error and is more prone to instability. Consequently, the tap coefficients will converge to their optimum values if μ satisfies the inequality [1], [44]

$$0 < \mu < \frac{2}{\lambda_{max}}, \quad (2.18)$$

where λ_{max} is the largest eigenvalue of \mathbf{R} . The convergence condition, (2.18), can be derived in the following manner. Subtracting \mathbf{C}_{opt} from both sides of (2.17) and then taking the expected value of the result, gives

$$E[\hat{\mathbf{C}}(n+1)] = [\mathbf{I} - \mu\mathbf{R}]E[\hat{\mathbf{C}}(n)], \quad (2.19)$$

where this is obtained under the assumption that the vectors $\mathbf{X}(n)$ and the coefficient error vector $\hat{\mathbf{C}}(n)$, defined as

$$\hat{\mathbf{C}}(n) = \mathbf{C}(n) - \mathbf{C}_{opt}, \quad (2.20)$$

are independent [7], [45].

Equation (2.19) reveals that the algorithm will converge to the optimal value if all the eigenvalues of the matrix $(\mathbf{I} - \mu\mathbf{R})$ are less than unity, that is

$$|1 - \mu\lambda_i| < 1, \quad i = 0, 1, \dots, N-1 \quad (2.21)$$

where it is assumed that the autocorrelation matrix, \mathbf{R} , is positive definite with eigenvalues, λ_i , hence, it can be factorised as

$$\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T \quad (2.22)$$

where $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues

$$\mathbf{\Lambda} = \text{diag}[\lambda_0, \lambda_1, \dots, \lambda_{N-1}] \quad (2.23)$$

and \mathbf{Q} is the orthonormal matrix whose i th column is the eigenvector of \mathbf{R} associated with the i th eigenvalue. The convergence of the algorithm is then obtained, which is basically (2.18), and the time constant, τ_i , associated with the eigenvalue, λ_i , can be derived to give the approximated value:

$$\tau_i = \frac{1}{\mu\lambda_i}, \quad i = 0, 1, \dots, N-1. \quad (2.24)$$

Hence the longest time constant, τ_{max} , is associated with the smallest eigenvalue, λ_{min} , of the autocorrelation matrix \mathbf{R} , that is

$$\tau_{max} = \frac{1}{\mu\lambda_{min}}. \quad (2.25)$$

Equations (2.18) and (2.25) can be combined to give the following result in terms of the eigenvalue spread (condition number), $\frac{\lambda_{max}}{\lambda_{min}}$,

$$\tau_{max} > \frac{\lambda_{max}}{2\lambda_{min}}. \quad (2.26)$$

Hence, from the point of view of convergence speed, the ideal value of the condition number is unity; the larger the value, the slower will be the convergence of the LMS algorithm.

It is therefore concluded that the optimum signal for fastest convergence of the LMS algorithm is white noise, and that any form of colouring in the signal will increase the convergence time.

Other developed adaptive schemes as well, all of which are LMS variants, e.g., the sign LMS [16], the normalised LMS (NLMS) [46], the leaky LMS [47], the variable step (VS) size algorithm [48], the block LMS (BLMS) [49], and many others, have been studied to enhance more the performance of the LMS algorithm for the desired application.

The LMS algorithm can be regarded as being obtained from the general expression, (2.7), when the value $k = 1$. However, the least mean-fourth (LMF) algorithm [8], which is another modification to the general expression, is obtained when $k = 2$. This algorithm is presented next.

2.4.2 The LMF algorithm

The performance function of the LMF algorithm is based on the minimisation of the mean of the fourth power of the error, that is, setting $k = 2$ in (2.7) results the cost function for this algorithm:

$$J(n) = E[e^4(n)]. \quad (2.27)$$

The algorithm for recursively adjusting the tap coefficients, $\mathbf{C}(n)$, is expressed in the following form

$$\mathbf{C}(n+1) = \mathbf{C}(n) + 2\mu e^3(n)\mathbf{X}(n). \quad (2.28)$$

The sufficient condition for convergence in the mean for this algorithm can be shown to be given by [8]:

$$0 < \mu < \frac{2}{6\lambda_{max}E[w^2(n)]}, \quad (2.29)$$

where $E[w^2(n)]$ is the variance of the additive noise $w(n)$, Fig. 5.2 depicts this clearly, and λ_{max} is the largest eigenvalue of \mathbf{R} .

Again, let the autocorrelation matrix \mathbf{R} be positive definite with eigenvalues, λ_i . Hence, the different time constants for the weights are defined by:

$$\tau_i = \frac{1}{6\mu E[w^2(n)]\lambda_i}, \quad i = 0, 1, \dots, N - 1. \quad (2.30)$$

As for the LMS algorithm, the largest time constant for this algorithm can be bounded by:

$$\tau_{max} > \frac{\lambda_{max}}{2\lambda_{min}}, \quad (2.31)$$

with the same conclusion obtained for the LMS algorithm; fast convergence only happens if the input signal is white noise. More details on this algorithm are given in Chapters 5 and 6.

2.4.3 The RLS algorithm

The LMS algorithm is widely used due to its comparatively easy implementation, lower order of complexity (only N operations (additions and multiplications) are required per update), and its well-established characteristics. However, the convergence is slow for highly correlated signals. The RLS algorithm, as it is discussed next, however, does not exhibit this dependence behaviour.

The RLS algorithm determines the coefficients that minimise the squared error summed over time [16], [50], i.e.,

$$J(n) = \sum_{j=0}^n e^2(j). \quad (2.32)$$

Due to the fact that the values of the filter coefficients, that minimise the above cost function, are functions of all past inputs, the associated adapted algorithm will have an infinite memory. A more convenient way, to limit this infinite memory problem, is to introduce a weighting function, $\gamma(n)$, in the cost function so that recent data are given more weight than past data. The resulting new cost function,

that will replace that of equation (2.32), is defined as

$$J(n) = \sum_{j=0}^n e^2(j)\gamma(n-j), \quad (2.33)$$

with weighting function $\gamma(n)$ taken as an example, as follows:

$$\gamma(n) = (1 - \beta)^n, \quad 0 < \beta < 1. \quad (2.34)$$

The tap coefficients are adapted to minimise $J(n)$. Taking the derivative of $J(n)$ with respect to $C_i(n)$, $i = 0, 1, \dots, N - 1$, and setting it equal to zero, i.e.,

$$\frac{\partial J(n)}{\partial \mathbf{C}(n)} = \mathbf{0}, \quad (2.35)$$

the following vector of the adaptive filter is obtained [28]

$$\mathbf{C}(n+1) = \mathbf{C}(n) + \beta \mathbf{R}^{-1}(n) \mathbf{X}(n) e(n), \quad (2.36)$$

where $\mathbf{R}(n)$, the autocorrelation matrix of the input signal vector $\mathbf{X}(n)$, is found recursively

$$\mathbf{R}(n) = (1 - \beta) \mathbf{R}(n-1) + \beta \mathbf{X}(n) \mathbf{X}^T(n). \quad (2.37)$$

Comparing (2.36) with (2.17) we see that the simple scalar loop gain in LMS has been replaced with β times the inverse of $\mathbf{R}(n)$. The normalisation with $\mathbf{R}^{-1}(n)$ offers more than the simple power normalisation in LMS; it normalises the adaptation in each eigenvector direction by the signal power in that direction. Thus the convergence becomes independent of both the signal type and power [51].

The solution to the above equations, (2.36) and (2.37), would require a large number of computations per update. This algorithm requires approximately $2.5N^2 + 4N$ multiplications and additions per update [16], N being the number of tap coefficients, significantly greater than the order of N for the LMS algorithm. Thus, as N increases, the number of operations increases in proportion to the square of the filter order. Hence, obtaining the optimum coefficient value involves computation of the inverse of the autocorrelation matrix and results in complex implementation. However, the advantage of this algorithm is fast convergence irrespective of the correlation characteristics of the input signal.

In contrast to the good feature of fast convergence observed with the RLS based algorithms, their computational complexities are still not attractive as those of the LMS algorithm. Moreover, instability problems are still annoying these algorithms. Also, it is interesting to note that if the input signal is white noise, the autocorrelation of the input signal may be evaluated as such $\mathbf{R} = \sigma_x^2 \mathbf{I}$, and if the step size μ in (2.17) is chosen to be $\frac{1}{N\sigma_x^2}$ such that $\beta = \frac{1}{N}$ in (2.36), the convergence properties of both the LMS and the RLS algorithms become approximately identical. Hence, in the work taken in this research, only the LMS and the LMF algorithms are considered in the development of new procedures used in the design of long echo cancellers.

Ultimately, Fig. 2.8 shows a tree structure of the families of adaptive algorithms which have been suggested for all the cost functions presented in this work.

2.5 Summary

This chapter concentrated on basic ideas for which both adaptive filters and adaptive algorithms are made up. The issue of adaptive filtering is still and will remain a very active field of research for some considerable time. This is mainly due to the advances in the computing facilities that were not previously available and to the need for such algorithms.

The wide spread use of the least-demanding computing algorithm, i.e., the LMS algorithm, is with no doubt due to its both simplicity and relative performance. The RLS algorithm, for example, gives very fast convergence to the algorithm at the expense of very heavy computational loads, irrespective of the input signal statistics. However, things change when the input signal is white noise, and the convergence properties of the LMS algorithm, under certain circumstances, become comparable to or the same as those of the RLS algorithm.

Both of these algorithms, and in general all algorithms, operate under different minimisation functions, which are the main reason for their different performances.

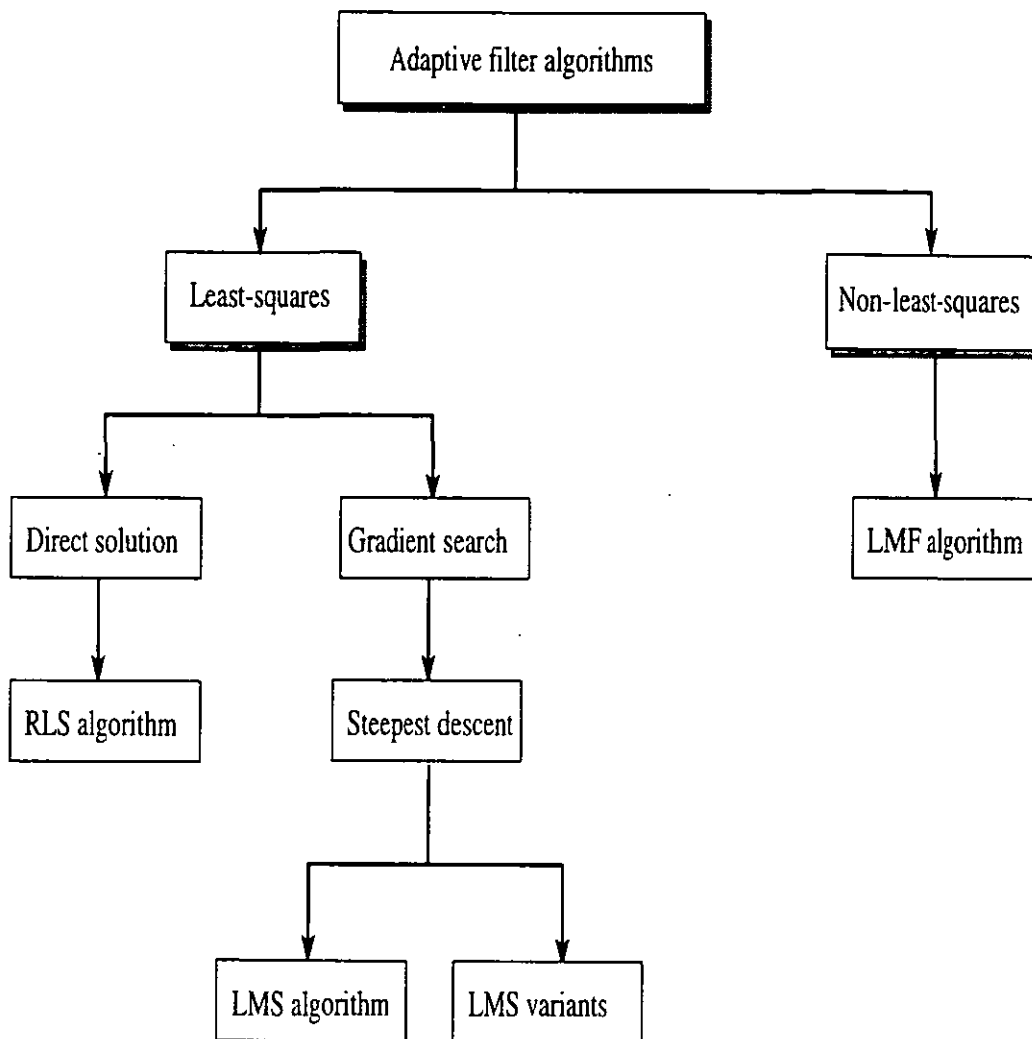


Figure 2.8: Tree structure for adaptive algorithm development.

While the next chapter treats echo cancellers in general, particular attention is made to some of the problems encountered when implementing long echo cancellers. In this work we propose new algorithms based on either a modification or a combination of the existing cost functions for stochastic gradient algorithms and that are implemented with the linear transversal filter, the FIR one, to enhance the performance of long echo cancellers in terms of their speed of convergence and steady state value. These proposed algorithms are treated in Chapters 4 and 5.

Chapter 3

Echo Cancellation Techniques

3.1 Introduction

Echoes are generated whenever part of a speech is reflected back to the source by the floor, walls, and other neighbouring objects. An echo is noticeable (or audible) only if the time delay between it and the speech exceeds a few tens of milliseconds [17].

As the result of impedance mismatches in telephone circuits, echoes are also generated. They arise in various situations in telecommunications networks and impair communication quality. Long-delay echoes are irritating, whereas shorter ones, called sidetones, are actually desirable and are intentionally inserted in telecommunications networks to make the telephone circuit seem alive [5].

Echoes with long delay are observed only on long-distance connections. To clearly understand the echo phenomenon, one is referred to Fig. 3.1 which illustrates a typical long distance telephone connection. It represents two two-wire segments at the ends, called the customer's loop, which connects a customer to the central office, and a four-wire carrier section (which might include satellite links). Two-wire circuits are bidirectional whereas four-wire ones are made of two distinct channels, one for each direction.

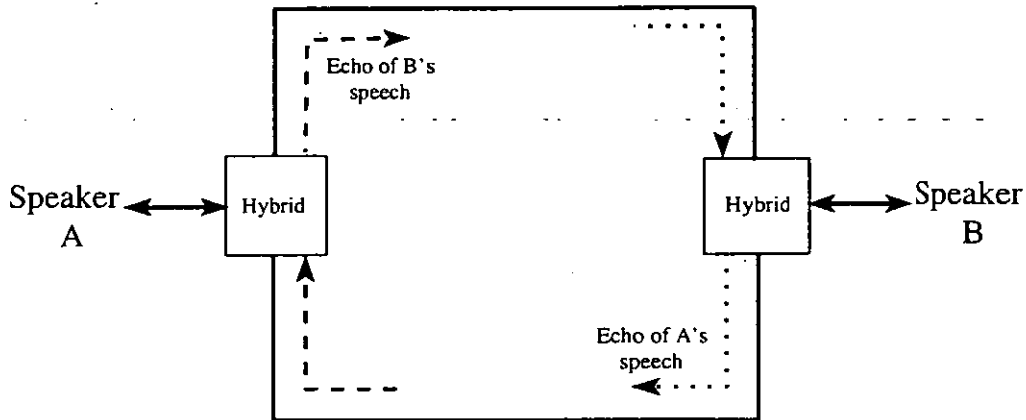


Figure 3.1: Long-distance telephone connection.

Echoes in the received signal are caused by impedance mismatches in the hybrid couplers which interface the two-wire and four-wire circuits. The hybrid is a device which provides a large loss around this loop without affecting any loss in the two talker speech paths. Figure 3.2 shows the details of such a device which is required at each end of a communication link as shown in Fig. 3.1. A hybrid circuit is basically a bridge circuit where the balancing impedance is (ideally) to be kept equal to the impedance of the two-wire circuit at all frequencies, hence removing any impedance mismatch, and consequently any echo from the network. Since many different local loops have access to a given hybrid coupler, it is unlikely that a single fixed balancing impedance will always be satisfactory, and hence it should come as no surprise that a considerable amount of “leakage” and reflected signal energy may be generated by a hybrid coupler.

The received echo generally consists of two components in either speech [3] or data [52] transmission applications. In speech communication, two components are observable: the “talker echo” which results in the talker hearing a delayed version of his or her own speech; and the “listener echo” where the listener hears a delayed version of the talker’s speech. As for data transmission, the two components are

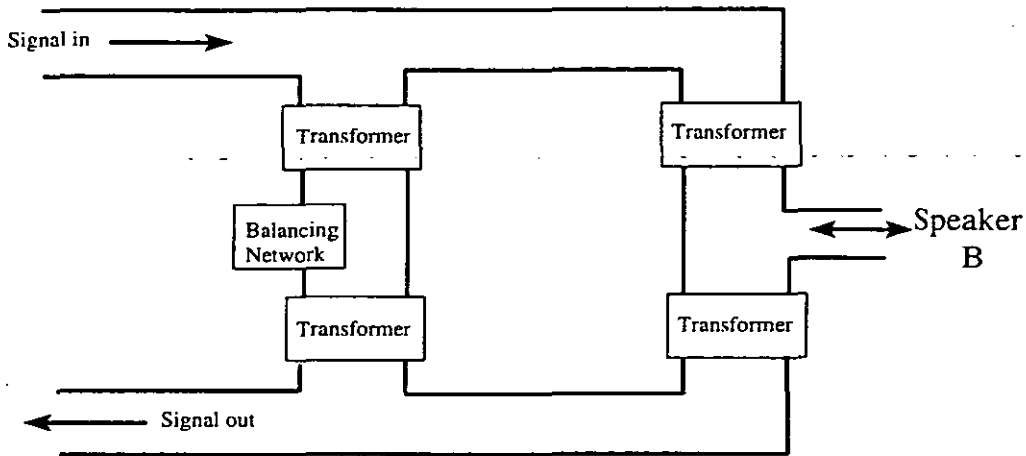


Figure 3.2: Hybrid circuit.

the near-end echo characterised by a short time delay (usually less than 20 *ms*), and the far-end echo that has a much lower level, but a longer time delay, that is a function of the transmission distance and varies from a few milliseconds to 600 *ms* for a satellite link.

Both the talker and the listener can suffer from echo problems. The talker echo can disturb the normal pattern of speech production, while the listener echo reduces intelligibility [53]. The severity of these effects is strongly dependent on reflection amplitude and the delay between the original signal and its echo. The echo is not audible if the delay is less than 20 *ms*; however, conversation is disturbed if the delay is more than about 100 *ms* [53].

To counteract this echo phenomenon, schemes must be developed to either completely eliminate it (the ideal requirement) or at least substantially reduce its adverse effect so as to achieve a transmission of good quality.

One of the early methods used in speech communication to combat echoes was the use of a device called an echo suppressor [54], whose simplified version is shown in Fig. 3.3. It conceals echo by detecting when the distant customer is speaking and the near customer is silent. The signal from the near customer is then heavily

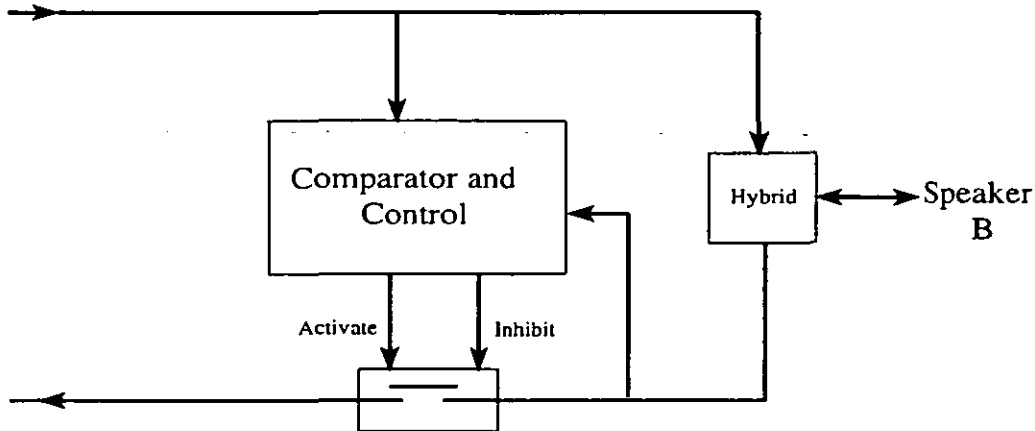


Figure 3.3: Simplified echo suppressor logic.

attenuated, so preventing echoes from being heard. This alternate-simplex mode of operation is enhanced by detecting when both customers are talking (double talk) and inserting a small loss in the send path [55]. Satisfactory operation depends on accurate speech detection to avoid clipping and utterance blocking [53].

Another technique that can be used for the same purpose is the use of an adaptive balancing hybrid [56], [57]. Ultimately, the most widely used of these schemes is echo cancellation [11] which is a rich body of algorithms derived from the powerful theory of adaptive signal processing and filtering [1], [7], [16]. Adaptive echo cancellers offer somewhat better call quality than suppressors on long delay circuits because they have greater signal transparency and fewer speech clipping effects [53].

Also, for high-speed full-duplex data transmission, it is necessary to use the entire bandwidth continuously, rather than the frequency or time-division techniques employed at lower data rates. This achieved through the use of echo cancellation techniques.

After discussing echo cancellation in both speech and data transmission in Section 3.2, the filter structures suitable for echo cancellation are treated in Section

3.3. Adaptive algorithms for coefficients control will be covered in Section 3.4, while Section 3.5 will present some of the problems encountered in echo cancellation.

3.2 Echo cancellers

Echoes may be generated due to reflections from objects surrounding an acoustic source and, as such, are called acoustic echoes. They can also be generated in a telephone network and are, in this case, called telephone echoes. Echo cancellers for cancelling such echoes are called acoustic cancellers and telephone echo cancellers, respectively. Our discussion will mainly focus on telephone echo cancellers which are the prime concern of this work. However, topics discussed here can also be applied to other echo cancellers with some modifications

Echo cancellation is a suitable area for the application of adaptive filtering. An adaptive echo canceller estimates the responses of an underlying echo-generating system in real time, in the face of unknown and time-varying echo path characteristics [58], generates a synthesised echo based on the estimate, and cancels the echo by subtracting the synthesised echo from the received signal.

In our analysis, it will be assumed that the echo path is linear, and therefore completely specified by its impulse response. Some older work in speech cancellers and some more recent progress in the theory of data cancellers have both extended echo cancellation techniques to nonlinear echo generation phenomena [59]-[61].

Echo cancellers are designed for both speech and data transmission. Since the treatments of both of these cancellers are quite different from one another, they will then be investigated separately.

3.2.1 Echo cancellers in speech communication

As previously mentioned, the listener echo and the talker echo are the main components of the overall echo in speech communication. These are shown in Fig. 3.1

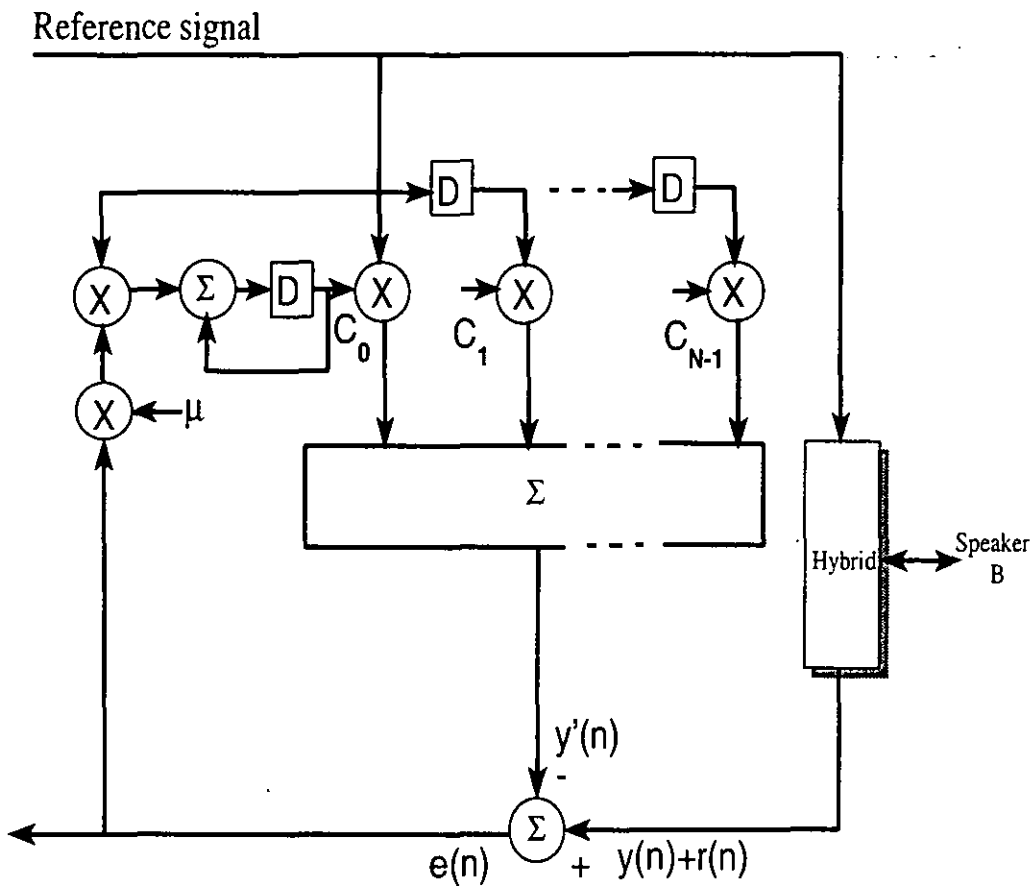


Figure 3.4: Structure of a voice telephone echo canceller.

where the adaptive canceller is placed in the four-wire path near the origin of the echo.

A block diagram of the adaptive canceller is shown in Fig. 3.4. The synthetic echo, $y'(n)$, is generated by passing the reference signal (i.e., speaker A's speech) through the adaptive filter that ideally matches the impulse response of the echo path. Thus, the transversal filter generates an estimate $y'(n)$ of the echo, given by

$$y'(n) = \sum_{i=0}^{N-1} C_i(n)x(n-i), \tag{3.1}$$

where $\{C_i(n)\}$ is the estimated echo-path impulse response sample, $x(i)$ is the input sample to the i th-tap delay, and N is the number of tap coefficients typically of order 256 [17]. While passing through the hybrid, the speech from speaker A results in the echo signal $y(n)$. This echo, together with the speech from speaker B, $r(n)$, constitutes the desired response for the adaptive filter. The canceller error signal is obtained as follows:

$$\begin{aligned}\zeta(n) &= y(n) - y'(n) + r(n) \\ &= e(n) + r(n).\end{aligned}\tag{3.2}$$

The error signal, $e(n)$, is used to control the adjustments to the adaptive filter coefficients, according to some specified adaptive schemes, in order to continuously improve the echo estimate $y'(n)$.

Ideally, the system should eventually converge to the condition $\zeta(n) = r(n)$. The effect of this ideal condition on the echo cancellation is naturally of some concern. Convergence of echo to zero, however, is not an adequate criterion of performance for a system of this sort [17], because this is possible only if $y(n)$ is exactly representable as the output of a fixed-tap filter. A better performance criterion would be the convergence of the filter's impulse response to the response of the echo path.

In any case, it is easy to see that the echo canceller can effectively eliminate the echo under double-talk conditions, which is not the case in echo suppressors. During double talk, the signal at the output port of the hybrid contains both the far echo and the speech signal from the talker at the other end of the circuit. After the synthesised echo is subtracted from the hybrid output signal, the remaining signal is basically the speech signal.

During double-talk situation, the adaptation scheme must either stop or the updating step size is reduced. Otherwise the coefficients rapidly diverge from their optimum values. Appropriate control of adaptation is therefore vital for satisfactory performance.

It is noteworthy that a voice echo canceller cancels the echo for the remote telephone user. The advantage of such an arrangement is that, because the canceller is close to the source of the echo, the delay between the echo and the reference signal is short. Therefore, the delay line of the echo canceller can also be short.

3.2.2 Echo cancellers in data transmission

The same echoes encountered in speech transmission would be observed on the telephone network if an attempt is made to transmit data signals through this network.

Echoes arise only in full-duplex data transmission, where the data signals are transmitted in both directions of the network simultaneously. These echoes result from the interference between the data signals transmitted in one direction and those flowing in the opposite direction. However, the echo phenomenon is not present in half-duplex data transmission, where transmission exists only in one direction, since there is no receiver at the transmitting end to be affected by the echo.

Because the telephone network typically provides only a two-wire connection to each customer premise due to the high cost of copper and other related matters, full-duplex data transmission over a common media has arisen in two important applications [3].

The first application is digital transmission over the subscriber loop, in which the basic voice service as well as data services are provided over the two-wire subscriber loop. This is an important characteristic of the well-known integrated services digital network (ISDN).

The second application is in voiceband data transmission where the basic customer interface to the network is often the same two-wire subscriber loop.

These two applications have many differences depending on the type of problem to be overcome. For the digital subscriber loop, the bandwidth of the medium,

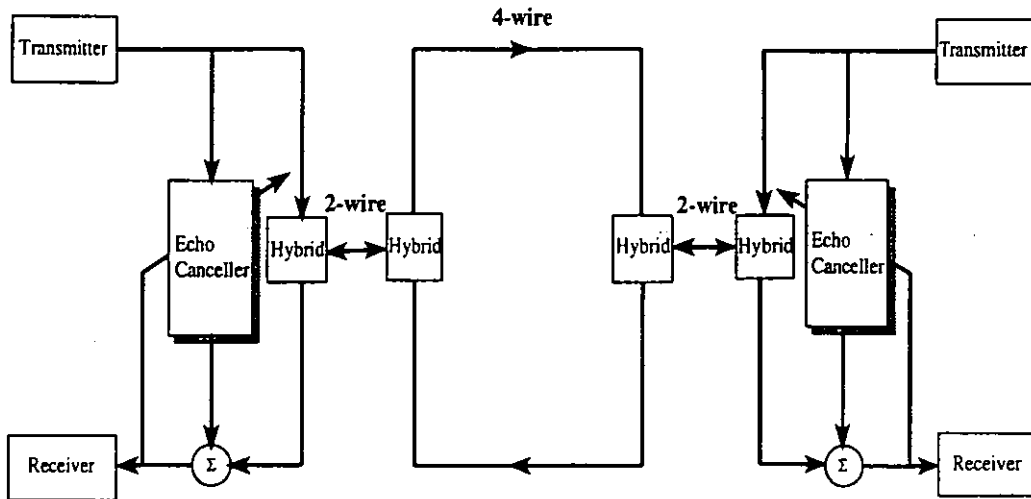


Figure 3.5: Echo cancellers at station locations for data transmission.

consisting of a pair of cables, is relatively wide and baseband transmission [62] is used. The voiceband data transmission, while requiring a lower speed than the subscriber loop transmission, uses passband transmission [63], i.e., modulates a carrier with the data stream, which makes the bandwidth relatively limited.

Isolation of the two directions of transmission in full-duplex data transmission is obtained only if echo cancellers are used at both end of the data sets [63]. Figure 3.5 illustrates the placement of the echo canceller for full-duplex data transmission. The transmit and receive directions are separated, at each data set, by the use of a hybrid, which will provide a virtual four-wire connection during communication. As in the case of voice transmission, this procedure would synthesise a replica of the echo which is subtracted from the received signal to give an error signal. This error is then used in the update of the taps of the adaptive echo canceller. The adaptation issue will be addressed in Section 3.4.

Two-wire full duplex digital transmission can be realised by frequency-division multiplexing (FDM). However, the lowest transmission rate and hence the longest range is obtained by the use of adaptive echo canceller. An echo canceller will also provide the best bandwidth efficiency and consequently minimum interference with

other services in the telephone network [64].

Special problems are associated with echo cancellation for full-duplex data transmission. Among them is the very large delay which can be exhibited by the distant echo. This requires implementing echo cancellers with at least 4000 taps if the distant echo is delayed by 600 *ms* or more when a satellite link is included in the four-wire circuit [17].

The solution for this problem is to provide a bulk between the near-end and far-end sections of the echo canceller, as illustrated in Fig. 3.6. This approach will considerably reduce the number of coefficients of the filter.

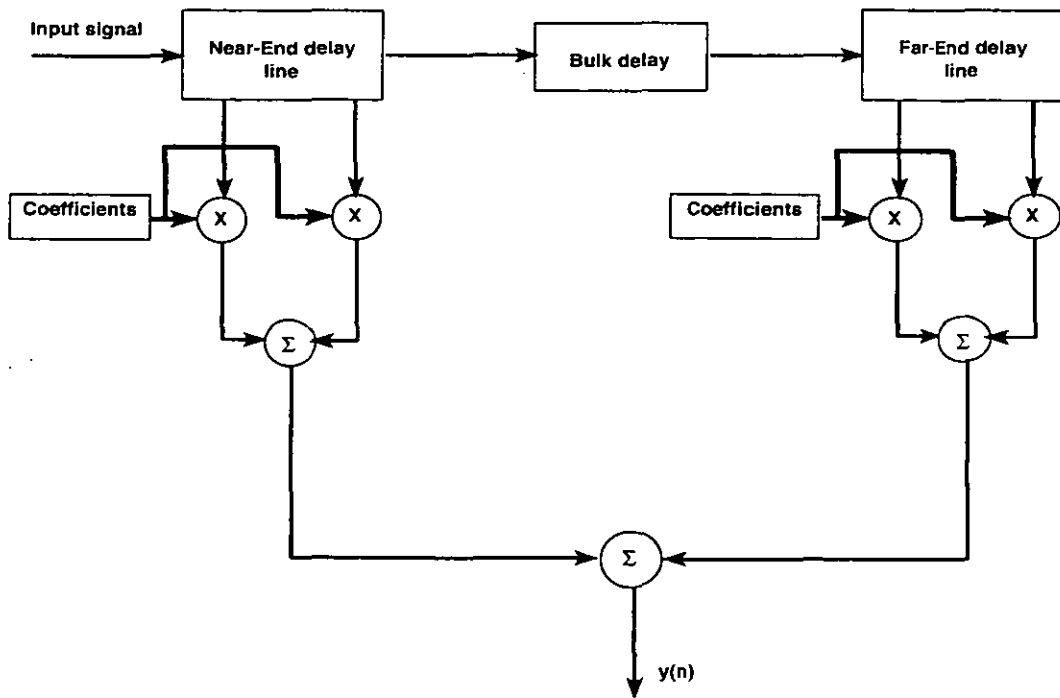


Figure 3.6: Echo canceller with bulk delay.

Echo cancellers in data transmission are classified as either voice-type [65] or data-type (data-driven) cancellers [66]. In the first configuration, the echo canceller gets its input from the output of the transmitter whereas in the second one the data symbols are the input to the echo canceller. Figures 3.7 and 3.8 show both of these configurations, respectively. In both of these realisations, the subtraction

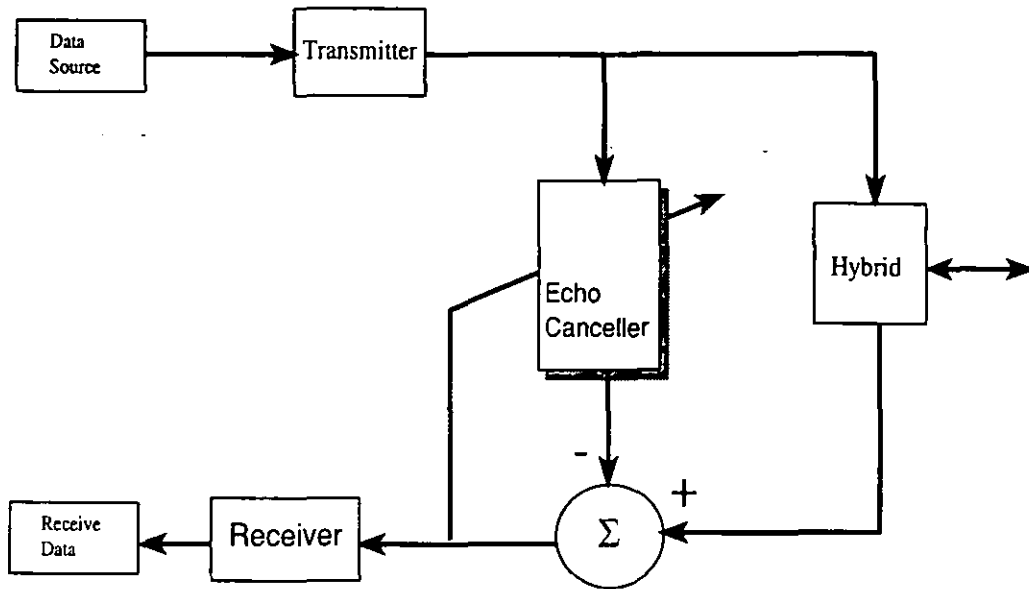


Figure 3.7: Voice-type echo canceller.

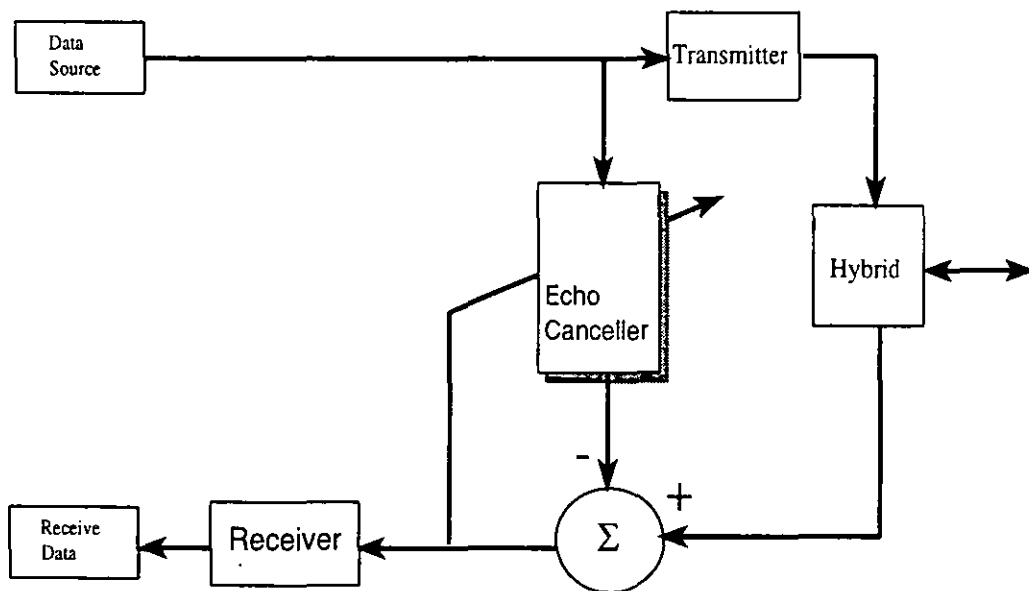


Figure 3.8: Data-type echo canceller.

of the echo estimate can be implemented in three different ways [62].

The main advantage of data-driven echo cancellers [66] over the voice-type ones [65] is that the tap signals can be binary, thus reducing multiplications to additions and subtractions. Moreover, transmission data rates can be as high as 4800 bits/s in the former canceller whereas only up to 2000 bits/s in the latter one. Also, data-driven echo cancellers are always stable, whereas voice-type cancellers can become unstable [67].

Mueller's configuration [66], which is a symbol-rate baseband canceller, requires synchronisation of received and transmitted sequences. Weinstein's proposal [63] eliminates this synchronisation requirement by achieving the necessary cancellation in the passband at a sampling rate which is greater than the Nyquist rate, and it is for this reason that they are called Nyquist cancellers.

Another realisation that lends itself to an easy implementation is the look-up table approach [64], which is a special case of the distributed arithmetic adaptive filter [16]. A property of this adaptive filter realisation is that the look-up table places no constraint on the type of echo path response, other than its time duration. Therefore, any nonlinearity in the transmitter, line interface, or echo-canceller digital to analogue (D/A) converter is also modeled. The analysis of this memory-based approach, when it is jointly adapted with a linear transversal echo canceller, is given in [61].

In all of the previously-mentioned configurations, there are some fundamental parameters that control the complexity of the echo canceller. Among them are the number of echo canceller taps, the maximum echo signal level, the minimum received signal level, and the required signal-to-uncanceled echo ratio [68].

Although echo cancellers designed either for speech transmission or data transmission would ultimately achieve the purpose they are designed for, namely echo cancellation, there are significant differences to be briefly described in the following section.

3.2.3 Differences between speech and data echo cancellers

As previously mentioned, there are implementational differences that are worth mentioning in connection with speech and data echo cancellers. The first difference to point out is that in data transmission, an echo canceller is placed at each data set whereas in speech transmission an echo canceller is part of the telephone network. The second difference is that, in data transmission, the transversal filter is a combination of two adjustable transversal filters separated by a bulk delay whereas only one transversal filter is required for speech transmission. Finally, there are other significant issues related to synchronisation, timing recovery, and equalisation which interact with data echo cancellation and which do not exist in speech cancellation [69].

Also, with the usual long wordlengths required for the tap values, a speech echo canceller has to perform a large number of complicated multiplications and store long wordlength values. Even in their simplest form speech echo cancellers are more difficult to implement than their data-driven counterparts [68].

Unlike data-echo cancellers where during initialisation, some training data are sent, speech echo cancellers cannot be trained perfectly for various reasons, one of which is that some residual echo will often remain. Further reduction of the residual echo is achieved by using techniques such as centre clipping [5]. A detailed explanation of the operation of these techniques can be found in [70].

The echo rejection requirements for voice echo cancellers are not as high as those for the data echo cancellers. This would have required the former cancellers to accurately estimate the echo path, as it is done with the latter ones.

The design of an echo canceller requires a knowledge of the echo delay and duration with reference to the position of the echo canceller in the network [68]. Speech echo responses can have delays as long as 20 *ms* and very oscillatory tails with significant energy up to 30 *ms* beyond the start of the echo. This is in contrast to the echoes experienced in speech-band data transmission, where the

band limitation of the data signal causes the echo tails to die away more quickly.

3.3 Filter structures

Various filter structures of practical importance can, in general, be modelled by the following rational transfer function

$$H(z) = \frac{A(z)}{B(z)}, \quad (3.3)$$

which can be realised in three possible ways, depending upon the forms of $A(z)$ and $B(z)$. These three different realisations are based on either the all-zero transfer function ($B(z) = \text{constant}$), or the all-pole one ($A(z) = \text{constant}$), or a combination of both, i.e., the pole-zero transfer function. Whereas the first realisation is nonrecursive, the second and the third are recursive ones. Hence, in practice, only these two types of realisations will be implemented.

In telecommunications, almost all current applications use linear finite impulse response (FIR) architectures, where the filter output is a weighted sum of present and past inputs. The transfer function for such architectures are of the all-zero type. However, real world transfer functions have both poles and zeros, and therefore realised in recursive forms. Nonetheless, FIR architectures are popular for several reasons. They are inherently stable as no feedback is involved and are more easily analyzed than their recursive counterparts, which can go unstable because of their reliance of feedback processing.

Other configurations can be used to implement the structure of an echo canceller. The choice of one of these configurations is dictated by many parameters, such as the amount of computations involved, the stability of the structure itself, and the convergence rate. These structures have one thing in common which is the estimation of the echo path characteristics.

The major drawbacks of FIR structures is that as the echo duration becomes longer, the number of taps increases proportionally and the convergence speed

decreases. To overcome this, other adaptive structures were investigated to reduce complexity and improve performance, such as the infinite impulse response (IIR) structure [71], the lattice structure [3], and the frequency-domain structure [51], [72].

The estimation of the transfer function of the echo path by an IIR involves a combination of poles and zeros and is very suitable in implementing an echo canceller when the number of taps required by an FIR structure is very large. However, stability tests are required to keep the poles inside the unit circle and this will make the convergence speed very slow. Although the potential of IIR filters seems to be great, much more study is needed before they become practical [12].

Another promising approach is to convert signals from the time domain to the frequency domain using the discrete Fourier transform (DFT) [36] and carry out the required echo cancellation in the frequency domain [72]. In this domain, convolution for a block of time-domain signals simply becomes coefficient multiplication, substantially reducing the structure complexity. In the frequency-domain structure, echo cancellation is carried out in every frequency bin.

- The convergence rate for an FIR filter depends on whether the signals are correlated (coloured) or uncorrelated (white), in which case, it is faster. This can be a serious problem for FIR filters with voice, and hence coloured, inputs, particularly when a large number of taps is required.

To decorrelate the input signal, so that its final replica is formed from a sum of uncorrelated signals, the lattice filter is used so that rapid convergence is obtained. The weighted sum of signals obtained at each stage of the lattice gives the echo replica. The weights are adapted as for the FIR filter.

3.4 Adaptive mechanisms

The two most widely used algorithms, already mentioned in Section 2.4, for adaptive filters are the least mean-squares (LMS) and the recursive least-squares (RLS) algorithms. These are used to estimate the echo path in an echo canceller structure [50], and out of the two, the LMS algorithm is the most widely used.

For reasons of practical usefulness and interest to our work, in this section, emphasis will be placed only on the LMS algorithm. For more clarification, the adaptation rule of the LMS algorithm is reproduced here,

$$\mathbf{C}(n+1) = \mathbf{C}(n) + \mu e(n)\mathbf{X}(n). \quad (3.4)$$

Although the simplicity and easy implementation of the LMS algorithm make it the predominant learning method used in echo cancellers, the undesirable dependence of its convergence rate on input statistics has prompted much research into alternatives that converge less dependently on such statistics.

To see the effect of the input statistics on the convergence of the echo canceller, let us consider the following. The error signal, that is the residual or uncanceled echo, is given by the following form:

$$\begin{aligned} e(n) &= y(n) - y'(n) \\ &= \sum_{i=0}^{N-1} [h_i - C_i(n)]x(n-i), \end{aligned} \quad (3.5)$$

where $\mathbf{H} = [h_0, h_1, \dots, h_{N-1}]^T$ and $\mathbf{C}(n) = [C_0(n), C_1(n), \dots, C_{N-1}(n)]^T$ are the coefficient vectors of the echo path and the echo canceller, respectively. The input signal is denoted by $\mathbf{X}(n)$ which is assumed to be a zero-mean, independent and identically distributed sequence.

Our objective is to minimise σ_e^2 , the power of the error, which is given by:

$$\begin{aligned} \sigma_e^2 &= E[e^2(n)] \\ &= E[y^2(n)] - 2\mathbf{C}^T(n)\mathbf{P} + \mathbf{C}^T(n)\mathbf{R}\mathbf{C}(n) \end{aligned} \quad (3.6)$$

where \mathbf{R} is the autocorrelation matrix of the input, and \mathbf{P} is the crosscorrelation vector between the desired response and the input components.

The optimal coefficients, \mathbf{C}_{opt} , are found by setting the first derivative of σ_e^2 with respect to each coefficient to zero. This gives:

$$\mathbf{C}_{opt} = \mathbf{R}^{-1}\mathbf{P}. \quad (3.7)$$

If it is assumed that the input signal is white, then $\mathbf{R} = \sigma_x^2\mathbf{I}$ and $\mathbf{P} = \sigma_x^2\mathbf{H}$ where σ_x^2 is the power of the input signal and \mathbf{I} is the identity matrix. The optimal solution, in (3.7), is obtained in its simplest form:

$$\mathbf{C}_{opt} = \mathbf{H}, \quad (3.8)$$

that is, the optimal solution is the impulse response of the echo path.

However, in cases where the input signal is not white, which is the case in real life applications, the power spectrum $S_x(f)$ of the input signal will not be flat, hence \mathbf{C}_{opt} will therefore not be equal to \mathbf{H} which affects the residual echo. To analyse this effect, let us write the error power as:

$$\begin{aligned} \sigma_e^2 &= E\left\{\left[\sum_{i=0}^{N-1} (h_i - C_i(n))x(n-i)\right]^2\right\} \\ &= \hat{\mathbf{C}}^T(n)\mathbf{R}, \end{aligned} \quad (3.9)$$

where $\hat{\mathbf{C}}(n)$ is the coefficient error vector. Using some algebraic manipulations, the above equation is rewritten in the following form:

$$\sigma_e^2 = \frac{1}{f_s} \int_{-\frac{f_s}{2}}^{+\frac{f_s}{2}} |\hat{C}(f)|^2 S_x(f) df, \quad (3.10)$$

where $\hat{C}(f)$ is the Fourier transform of $\hat{\mathbf{C}}(n)$. Clearly if $S_x(f)$ is very small or zero at any frequency or range of frequencies, there will then be no control on the frequency response error at that particular frequency or range of frequencies. To avoid this situation from happening, randomising the input signal will cure the problem temporarily.

Randomising the input signal is like performing a whitening process on it. For adaptive data echo cancellers, their input signal is a sequence of data symbols which are randomised by a scrambler before transmission. As a result of this process, randomisation improves the convergence of the echo canceller [73], since, as it has been shown, the adaptive filter has the best convergence when the input signal is white.

The choice of the step size μ , in equation (3.4), is also of some concern. This has always been a trade off between fast convergence and large mean square error (MSE) at the output of the echo canceller since a smaller μ will slow down the convergence rate at the expense of small steady-state MSE.

A common technique used in designing echo cancellers is to use different step sizes at different stages of the training. That is, at the initialization process, μ_{opt} is used to achieve the fastest convergence. When the echo canceller is close to convergence, the algorithm switches to a smaller μ to reduce the steady-state MSE.

Finally, the performance of the LMS algorithm depends not only on the step size, μ , but also on the filter length, N , the signal to noise ratio, and nonlinearities in the echo path.

3.5 Problems encountered in echo cancellation

There are several problems encountered in the design of an echo canceller. Some of these problems are frequency offset, finite wordlength effects, fast-initialization process or start up, combination of echo cancellation and equalisation, and the exact estimation of the delay between the near-end and far-end sections in data echo cancellers.

In each of these problems, further research is needed to improve the performance of the echo cancellers.

The work addressed in this dissertation looks at ways of improving the effect

of the wrong estimation of the bulk delay. This consists of applying transitional shaping between the near-end and the far-end sections of the data echo canceller. An enhancement in performance is achieved through the use of a modified version to the LMS algorithm.

Also, the application of the LMS and the LMF algorithms to the near-end and the far-end sections, respectively, can improve the convergence rate, that is the start up, and the steady-state value as well. This is another modification to the standard LMS algorithm which is usually applied to both sections of the echo canceller.

Details of these two approaches are investigated in later chapters.

3.6 Summary

Echo cancellers have become a reality in the telephone system. This is primarily due to the advance in the very large-scale integration (VLSI) technology. The first fully integrated single-chip adaptive echo canceller chip was developed in 1978 [74]. Echo cancellers for telephone circuits and data transmissions have now reached a low-cost realisation by using VLSI technology.

Today, there exists a rich body of technical literature on different aspects of echo cancellation. This chapter covers limited but fundamental material on this topic.

Chapter 4

Adaptive Echo Cancellation Using Statistical Shaping

4.1 Introduction

Echo cancellation [3], [17] is a major application of adaptive noise cancelling techniques [14]-[15] which are of importance in telecommunications. The primary objective of an echo canceller is to counteract a common but undesirable phenomenon in the telephone network, namely echo. The purpose of an echo canceller is, then, to compensate for this distortion by synthesising and subtracting a replica of the echo from the returned signal.

The theoretical basis for echo cancellers is in the field of adaptive filtering [1], [7], [16]. This field has been extensively studied for the past few decades, and practical adaptive echo cancellation was well conceived as well [11]. Adaptive digital filtering is required to obtain a good echo replica in the face of unknown and time-varying echo path characteristics [58].

Echoes are only significant in full-duplex data transmission, where the data signals are transmitted in both directions of the network simultaneously. These echoes result from impedance mismatches in the hybrid couplers which interface

the two-wire and four-wire circuits.

A serious difficulty which is unique to echo cancellation at station locations is the very large delay which can be exhibited by the distant echo, which returns from the far end of the circuit. This would in return require implementing echo cancellers with at least 4000 taps if the distant echo is delayed by 600 ms or more, for instance when a satellite link is included in the four-wire circuit [17].

This problem is dealt with by splitting the transversal filter into two adjustable transversal filters separated by a bulk delay [5] as shown in Fig. 4.1. The bulk delay is estimated at the beginning of the operation, but unfortunately not precisely. Thus, the adaptive algorithm does not converge to the exact solution, resulting in an excess error (uncancelled echo).

Many ways can be used to overcome this problem. In a situation where a rational function is a good approximation of the echo path transfer function, a recursive filter might be an alternative. Unfortunately, the stability of the recursive filter is usually a critical problem [16], [75]-[77]. In addition, the error surface may be multi modal, making conventional gradient adaptation algorithms problematic [77].

Another way would be by adding a certain number of coefficients in the end of the near-end section and at the beginning of the far-end section. This, however will end up adding additional computations to the algorithm. A compromise to the previous mentioned methods is the use of transitions instead, where the coefficients are chosen according to a certain probability density function (pdf). The advantages of this new approach is less mean square error at the output of the echo canceller, and less misadjustment.

Another point worth mentioning is that this is new proposed algorithm will not have an increase in complexity over the other algorithm. An enhancement in performance is obtained over the one that would be represented by all coefficients present in the transition bands.

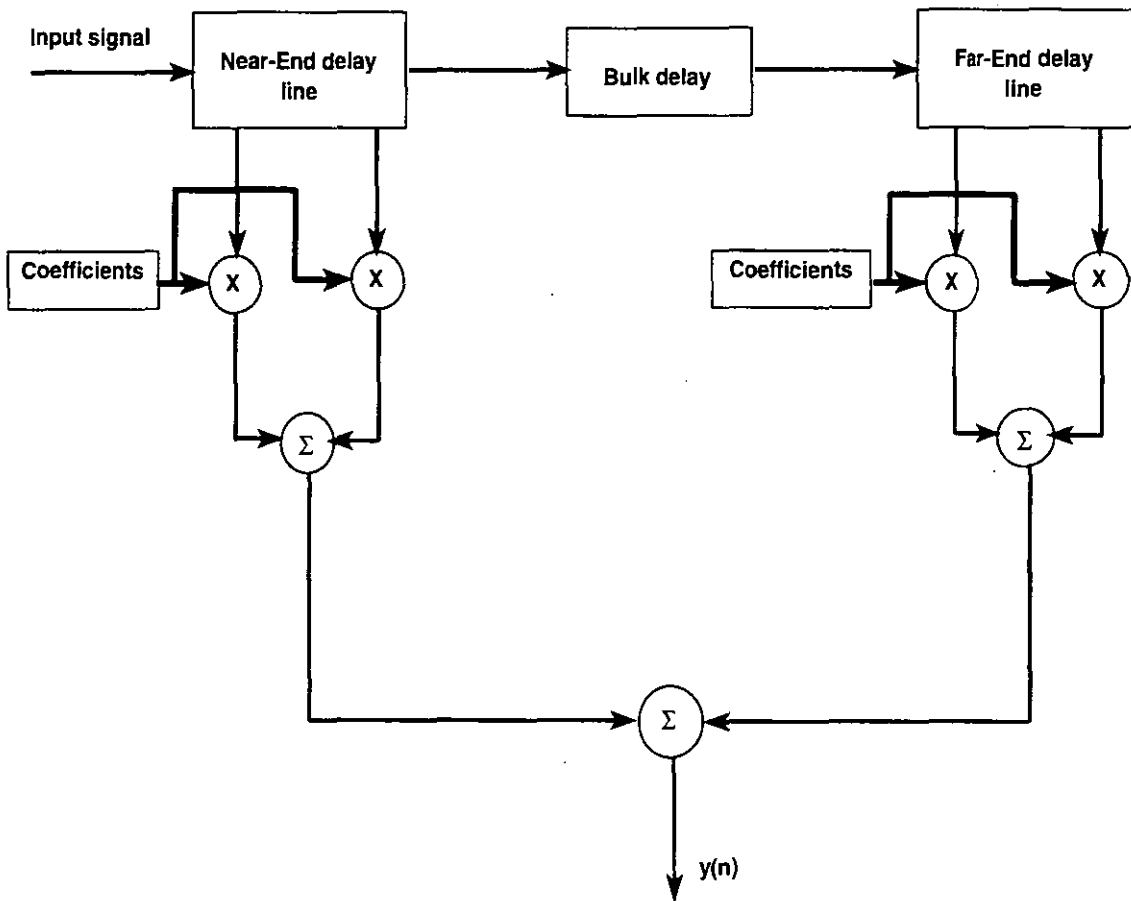


Figure 4.1: Structure of an echo canceller with bulk delay.

The new proposed study is based on the the least mean-squares (LMS) algorithm where a probabilistic approach is used to analyze the modified LMS algorithm. Two different transition bands are used in this study. These are a sharp transition and a smooth transition. The latter one is chosen according to the uniform pdf.

This analysis presents a complete derivation of the algorithm where it identifies the largest eigenvalue in either the sharp transition or the smooth transition. It generalises previous works on echo cancellation [3], [17], as well. In the analysis, it will be assumed that the echo path is linear, and therefore completely specified by its impulse response.

During the derivations which follow, the convergence in the mean and in the mean square are examined and the time constant of the algorithm and bounds on the convergence factor are obtained for each transition. The excess error due to the use of the first approach, namely the one based on the sharp transition, is also evaluated.

4.2 Algorithm development

The most widely used adaptive algorithm for adaptive filters is the LMS algorithm [1]. In this algorithm, the tap coefficients are adapted to minimise a certain cost function. For simplicity, this algorithm will be described for the adaptive FIR filter in Fig. 4.2.

As can be seen from Fig. 4.1, that a bulk delay is inserted between the near-end section and the far-end section of the canceller to reduce the number of taps considerably. One could interpret this situation as if the coefficients are distributed with a certain probability density function (pdf). It is clear that one can say that those belonging to the delay section (DS) have probability one of being zero, and probability zero when outside DS . That is

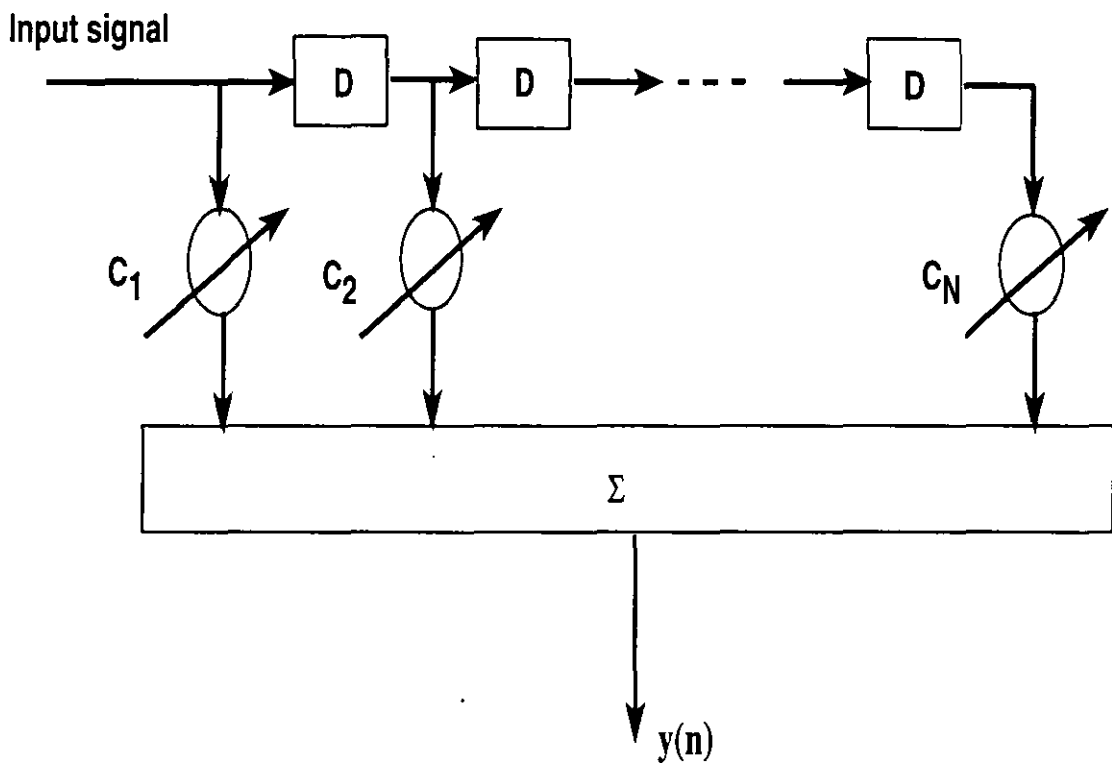


Figure 4.2: FIR filter.

$$P(C_i(n) = 0) = 1, \quad i \in DS, \quad (4.1)$$

and

$$P(C_i(n) \neq 0) = 0, \quad i \in DS. \quad (4.2)$$

This is clearly a uniform pdf, which is depicted in Fig. 4.3. Then, one can think of the output of the canceller at the n th iteration instant being equal to:

$$y(n) = \mathbf{C}^T(n) \mathbf{W} \mathbf{X}(n), \quad (4.3)$$

where the filter coefficients are given by:

$$\mathbf{C}(n) = [C_1(n), C_2(n), \dots, C_N(n)]^T, \quad (4.4)$$

and T denotes the vector transposition operation. The reference input samples to the adaptive FIR filter are given in vector form by:

$$\mathbf{X}(n) = [x_n, x_{n-1}, \dots, x_{n-N+1}]^T, \quad (4.5)$$

and

$$\mathbf{W} = \text{diag}(1, \dots, 1, 0, \dots, 0, 1, \dots, 1) \quad (4.6)$$

is an $N \times N$ diagonal matrix present in the computation of the output to select the coefficient identified as present or absent. The number of 1's in the upper and the lower part of this matrix corresponds to the number of coefficients not equal to zero. That is

$$P(C_i(n) = 0) = 0, \quad i \notin DS, \quad (4.7)$$

and

$$P(C_i(n) \neq 0) = 1, \quad i \notin DS. \quad (4.8)$$

The coefficients vector can be written in the following format

$$\mathbf{C}(n) = \begin{pmatrix} \mathbf{C}^u(n) \\ \mathbf{0} \\ \mathbf{C}^l(n) \end{pmatrix}, \quad (4.9)$$

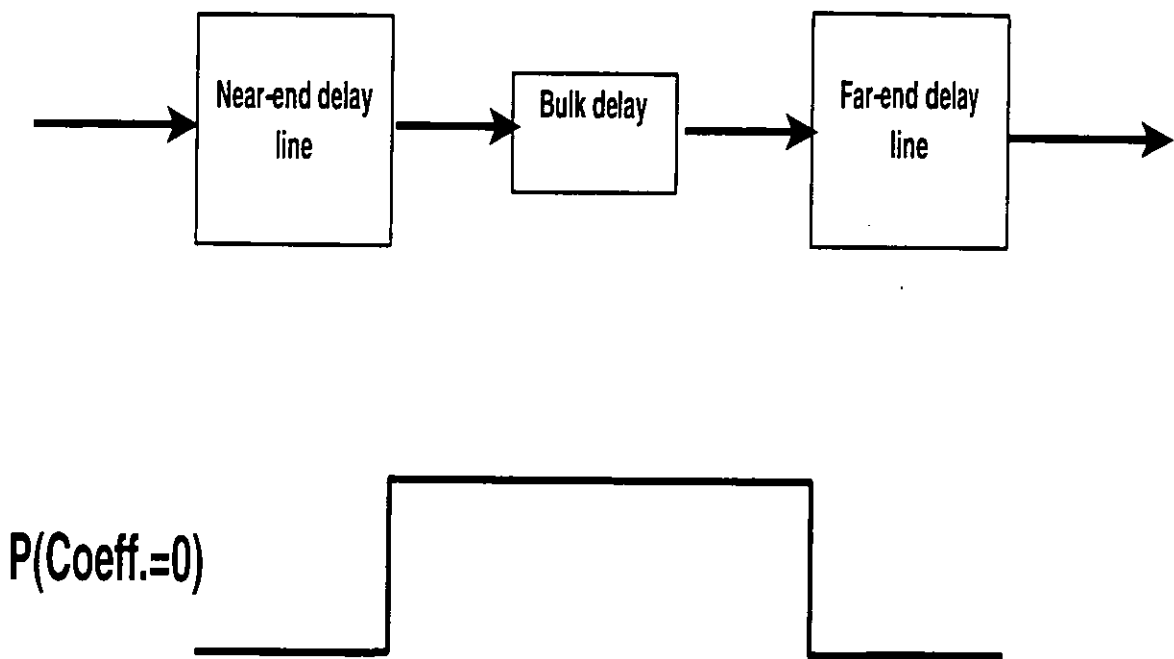


Figure 4.3: Probability density function of the tap coefficients belonging to the bulk delay.

where $\mathbf{C}^u(n)$ and $\mathbf{C}^l(n)$ designate the portions of the coefficients belonging to the near-end and the far-end sections of the canceller, respectively. The delay section is the part of the echo canceller where the coefficients are zeros, and that is why it is indicated in equation (4.9) by a zero-vector. Therefore, the total active number of coefficients in the calculation of the output of the canceller is $(N - D)$, where N is the total number of coefficients and D is the number of coefficients being zero.

In most practical instances the adaptive process is oriented towards minimising the mean-square value (MSE), or average power of the error signal. The performance index for the MSE criterion, denoted $J(n)$, is defined as

$$J(n) = E[e^2(n)], \quad (4.10)$$

where $e(n)$ is the error between the output and the desired value of the canceller given by

$$e(n) = y(n) - d(n). \quad (4.11)$$

The mean-square-error function can be more conveniently expressed in terms of the input autocorrelation matrix

$$\mathbf{R} = E[\mathbf{X}(n)\mathbf{X}^T(n)], \quad (4.12)$$

and the crosscorrelation vector

$$\mathbf{P} = E[\mathbf{X}(n)d(n)], \quad (4.13)$$

between the desired response and the input components, as follows:

$$J(n) = E[d^2(n)] - 2\mathbf{C}^T(n)\mathbf{W}\mathbf{P} + \mathbf{C}^T(n)\mathbf{W}\mathbf{R}\mathbf{W}\mathbf{C}(n). \quad (4.14)$$

It is clear from this expression that the MSE is precisely a quadratic function of the components of the tap coefficients. Thus, the shape associated with this MSE is hyperparaboloid, and then the algorithm used will seek the bottom of the error surface to get to the optimal value.

4.2.1 The optimal solution

The point at the bottom of the performance surface corresponds to the optimal tap coefficients, \mathbf{C}_{opt} , or minimum MSE. The gradient method is used to cause the tap coefficients vector to seek the minimum of the performance surface. The gradient of the performance criterion is defined as

$$\nabla J(n) = \frac{\partial J(n)}{\partial \mathbf{C}(n)}. \quad (4.15)$$

To obtain the minimum MSE, the tap-coefficients vector $\mathbf{C}(n)$ is set to its optimal value, \mathbf{C}_{opt} , where the gradient is set to zero, that is,

$$\mathbf{C}_{opt} = [\mathbf{WRW}]^{-1}\mathbf{WP}. \quad (4.16)$$

The minimum MSE associated with this optimal value is found to be

$$J_{min} = E[d^2(n)] - \mathbf{C}_{opt}^T \mathbf{WP}. \quad (4.17)$$

4.2.2 The updating scheme

Due to its simplicity and ease of implementation, the LMS algorithm [1], which is used recursively to update the tap coefficients of the adaptive filter, is the most widely used algorithm. The adaptation number, called the step size, used in this algorithm must be chosen small enough to ensure convergence of the iterative procedure.

The update relation for the the LMS algorithm is derived from the following steepest descent type weight update equation:

$$\mathbf{C}(n+1) = \mathbf{C}(n) - \frac{1}{2}\mu \hat{\nabla} J(n), \quad (4.18)$$

where μ is the adaptation gain or step size, and $\hat{\nabla} J(n)$ is the instantaneous estimate of the gradient of the error norm $J(n)$ evaluated at the current value of the weight vector $\mathbf{C}(n)$.

Differentiation of equation (4.10) with respect to $\mathbf{C}(n)$ yields the recursion equation used in the update of the proposed algorithm:

$$\mathbf{C}(n+1) = \mathbf{C}(n) - \mu e(n) \mathbf{W} \mathbf{X}(n), \quad (4.19)$$

where the step size parameter μ is used in the iterative process to control the convergence of the algorithm, and \mathbf{W} is the matrix defined by (4.6). The proposed algorithm defined by (4.19) will update only those coefficients defined by (4.7) and (4.8), that is, only those present in the computation of the output.

The choice of the step size parameter, and the convergence behaviour of the algorithm are investigated in the next section, where the sharp transition is used. A similar study for the algorithm, where the smooth transition is used instead, will be presented in Section 4.4.

4.3 Convergence behaviour of the algorithm when using the sharp transition

This section will deal with the convergence of the above stated algorithm when the sharp transition is used. The convergence in the mean and the convergence in the mean square are both analyzed and for each, bounds on the step size parameter are obtained. The convergence time constant of the algorithm is also obtained. The convergence of the algorithm in the sense of the mean is first studied.

4.3.1 Convergence in the mean

In this section, the convergence of the algorithm in the mean is presented. The value of the error defined in (4.11) and the output $y(n)$ in (4.3) are substituted in equation (4.19) to get the following:

$$\mathbf{C}(n+1) = [\mathbf{I} - \mu \mathbf{W} \mathbf{X}(n) \mathbf{X}^T(n) \mathbf{W}] \mathbf{C}(n) + \mu \mathbf{W} \mathbf{X}(n) d(n). \quad (4.20)$$

Define now

$$\hat{\mathbf{C}}(n) = \mathbf{C}(n) - \mathbf{C}_{opt} \quad (4.21)$$

to be the coefficient error vector at the n th iteration.

Subtracting \mathbf{C}_{opt} from both sides of (4.20) and making use of (4.21), equation (4.20) becomes

$$\begin{aligned} \hat{\mathbf{C}}(n+1) = & [\mathbf{I} - \mu \mathbf{W} \mathbf{X}(n) \mathbf{X}^T(n) \mathbf{W}] \hat{\mathbf{C}}(n) \\ & + \mu \mathbf{W} \mathbf{X}(n) d(n) - \mu \mathbf{W} \mathbf{X}(n) \mathbf{X}^T(n) \mathbf{W} \mathbf{C}_{opt}. \end{aligned} \quad (4.22)$$

Under the fundamental assumption [7], $\mathbf{X}(n)$ and $\hat{\mathbf{C}}(n)$ are independent, so that taking the expectation of both sides of the above equation, and using (4.12) and (4.13), equation (4.22) becomes

$$E[\hat{\mathbf{C}}(n+1)] = [\mathbf{I} - \mu \mathbf{W} \mathbf{R} \mathbf{W}] E[\hat{\mathbf{C}}(n)] + \mu \mathbf{W} \mathbf{P} - \mu \mathbf{W} \mathbf{R} \mathbf{W} \mathbf{C}_{opt}. \quad (4.23)$$

The value of \mathbf{C}_{opt} defined in (4.16) is substituted in (4.23) to get the following result

$$E[\hat{\mathbf{C}}(n+1)] = [\mathbf{I} - \mu \mathbf{W} \mathbf{R} \mathbf{W}] E[\hat{\mathbf{C}}(n)]. \quad (4.24)$$

Equation (4.24) reveals that the algorithm will converge to the optimal value if all the eigenvalues of the matrix $(\mathbf{I} - \mu \mathbf{W} \mathbf{R} \mathbf{W})$ are less than unity.

To be able to evaluate the mean value in (4.24), all that is needed is to compute the value of $\mathbf{W} \mathbf{R} \mathbf{W}$. Since \mathbf{W} is a diagonal matrix, it can be written in the following format

$$\mathbf{W} = \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \quad (4.25)$$

where \mathbf{I} is the identity matrix whose length depends upon the length of the corresponding part of the filter (i.e., the near-end and the far-end of the canceller).

The same partition can be done for the autocorrelation matrix \mathbf{R} ,

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \mathbf{R}_{13} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \mathbf{R}_{23} \\ \mathbf{R}_{31} & \mathbf{R}_{32} & \mathbf{R}_{33} \end{pmatrix} \quad (4.26)$$

where each \mathbf{R}_{ij} ($1 \leq i \leq 3, 1 \leq j \leq 3$) has dimensions equal to the corresponding entry of the matrix \mathbf{W} in (4.25).

The product \mathbf{WRW} is found to be equal to the following

$$\mathbf{WRW} = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{0} & \mathbf{R}_{13} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{R}_{31} & \mathbf{0} & \mathbf{R}_{33} \end{pmatrix}. \quad (4.27)$$

If the canceller is designed in such a way to make the number of coefficients in the near-end and the far-end parts of the canceller equal, then \mathbf{R}_{13}^T would be equal to \mathbf{R}_{31} and equation (4.27) is transformed to the following

$$\mathbf{WRW} = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{0} & \mathbf{R}_{13} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{R}_{13}^T & \mathbf{0} & \mathbf{R}_{33} \end{pmatrix}. \quad (4.28)$$

Since the canceller is mainly composed of three components, the near-end and the far-end sections, and the bulk delay between these two elements, one then can use (4.9) so that (4.24) can be expressed as

$$E \begin{pmatrix} \hat{\mathbf{C}}^u(n+1) \\ \mathbf{0} \\ \hat{\mathbf{C}}^l(n+1) \end{pmatrix} = \begin{pmatrix} \mathbf{I} - \mu\mathbf{R}_{11} & \mathbf{0} & -\mu\mathbf{R}_{13} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ -\mu\mathbf{R}_{13}^T & \mathbf{0} & \mathbf{I} - \mu\mathbf{R}_{33} \end{pmatrix} E \begin{pmatrix} \hat{\mathbf{C}}^u(n) \\ \mathbf{0} \\ \hat{\mathbf{C}}^l(n) \end{pmatrix}. \quad (4.29)$$

From the above solution, we see that we have two equations to deal with, which are

$$E[\hat{\mathbf{C}}^u(n+1)] = [\mathbf{I} - \mu\mathbf{R}_{11}]E[\hat{\mathbf{C}}^u(n)] - \mu\mathbf{R}_{13}E[\hat{\mathbf{C}}^l(n)], \quad (4.30)$$

and

$$E[\hat{\mathbf{C}}^l(n+1)] = [\mathbf{I} - \mu\mathbf{R}_{33}]E[\hat{\mathbf{C}}^l(n)] - \mu\mathbf{R}_{13}^TE[\hat{\mathbf{C}}^u(n)]. \quad (4.31)$$

Because of this coupling between these two equations, one would have to find a compromise for μ , the step size, so that convergence will take place.

Equations (4.30) and (4.31) can be set up in a matrix form as follow

$$E \begin{pmatrix} \hat{\mathbf{C}}^u(n+1) \\ \hat{\mathbf{C}}^l(n+1) \end{pmatrix} = \left(\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} - \mu \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{13} \\ \mathbf{R}_{13}^T & \mathbf{R}_{33} \end{bmatrix} \right) E \begin{pmatrix} \hat{\mathbf{C}}^u(n) \\ \hat{\mathbf{C}}^l(n) \end{pmatrix}, \quad (4.32)$$

where it is clear that the step size, μ , depends on the largest eigenvalue of the following matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{13} \\ \mathbf{R}_{13}^T & \mathbf{R}_{33} \end{bmatrix}. \quad (4.33)$$

From here we can tell that at steady state the following will happen, provided the right step size has been chosen,

$$\hat{\mathbf{C}}(n) \rightarrow \mathbf{0}, \quad (4.34)$$

that is

$$\mathbf{C}(n) = \mathbf{C}_{opt}. \quad (4.35)$$

From this point we can argue that $\hat{\mathbf{C}}^u(n)$ and $\hat{\mathbf{C}}^l(n)$ approach zero, then we have

$$0 < \mu < \frac{2}{\lambda_{max}}, \quad (4.36)$$

where λ_{max} is the largest eigenvalue of \mathbf{A} .

As a special case (a mathematical interest), when the matrix \mathbf{R}_{13} is equal to $\mathbf{0}$, that is, the near-end and the far-end sections are uncorrelated, the step size reduces to the following

$$0 < \mu < \frac{2}{\max[\lambda_{max11}, \lambda_{max33}]}, \quad (4.37)$$

where λ_{max11} and λ_{max33} are the largest eigenvalues of \mathbf{R}_{11} and \mathbf{R}_{33} , respectively .

4.3.2 Convergence in the mean square

The convergence behaviour of the algorithm in the mean square is studied next.

Define the following measure

$$\theta_n = E[\|\hat{\mathbf{C}}(n)\|^2], \quad (4.38)$$

where $\|\cdot\|$ denotes the Euclidean norm, then

$$E[\|\hat{\mathbf{C}}(n+1)\|^2] = E[\hat{\mathbf{C}}^T(n+1)\hat{\mathbf{C}}(n+1)]. \quad (4.39)$$

It follows from (4.19) that

$$\theta_{n+1} = \theta_n - 2\mu E[e(n)\mathbf{X}^T(n)\mathbf{W}\hat{\mathbf{C}}(n)] + \mu^2 E[e^2(n)\mathbf{X}^T(n)\mathbf{W}\mathbf{X}(n)]. \quad (4.40)$$

Under the often-made assumption [78]-[79] that the input signal sequence $\{x_i\}$ is an independent and identically distributed Gaussian random sequence with zero mean and variance σ_x^2 , the derivation of (4.40) is obtained as follow. Consider the following expression

$$E[e(n)\mathbf{X}^T(n)\hat{\mathbf{C}}(n)] = J(n) - J_{min} \quad (4.41)$$

evaluated in [80]. Then, taking this result into account one can derive an expression for the second term of equation (4.40), given by

$$E[e(n)\mathbf{X}^T(n)\mathbf{W}\hat{\mathbf{C}}(n)] = J(n) - J_{min}. \quad (4.42)$$

The third term of the right-hand side of equation (4.40) is readily evaluated, given that

$$\mathbf{X}^T(n)\mathbf{W}\mathbf{X}(n) = \|\mathbf{X}(n)\|^2 - \sum_{i \in DS} x_i^2, \quad (4.43)$$

and from previously

$$E[\|e(n)\|^2] = J(n),$$

then we can write the following

$$\begin{aligned} E[e^2(n)\mathbf{X}^T(n)\mathbf{W}\mathbf{X}(n)] &= E[\|e(n)\|^2\|\mathbf{X}(n)\|^2] - E[\|e(n)\|^2 \sum_{i \in DS} x_i^2] \\ &= NJ(n)\sigma_x^2 - DJ(n)\sigma_x^2 \\ &= (N - D)J(n)\sigma_x^2. \end{aligned} \quad (4.44)$$

Equation (4.40) becomes, when (4.44) is inserted,

$$\theta_{n+1} = \theta_n - 2\mu E[e(n)\mathbf{X}^T(n)\mathbf{W}\hat{\mathbf{C}}(n)] + \mu^2(N-D)J(n)\sigma_x^2. \quad (4.45)$$

Substituting equation (4.42) into (4.45), this will result in the following expression:

$$\theta_{n+1} = \theta_n - 2\mu[J(n) - J_{min}] + \mu^2(N-D)J(n)\sigma_x^2. \quad (4.46)$$

At convergence when θ_{n+1} reaches θ_n , the value of the step-size is found to be

$$\mu = \frac{2(1 - \frac{J_{min}}{J(n)})}{(N-D)\sigma_x^2}. \quad (4.47)$$

When $[J(n) - J_{min}]$ is approximated by $\sigma_x^2\theta_n$ [81], equation (4.46) becomes

$$\theta_{n+1} = [1 - 2\mu\sigma_x^2 + \mu^2(N-D)\sigma_x^4]\theta_n + \mu^2 J_{min}(N-D)\sigma_x^2 \quad (4.48)$$

The stability of the algorithm is guaranteed if:

$$|1 - 2\mu\sigma_x^2 + \mu^2(N-D)\sigma_x^4| < 1, \quad (4.49)$$

which is satisfied if :

$$0 < \mu < \frac{2}{(N-D)\sigma_x^2}. \quad (4.50)$$

The step size for fastest convergence is obtained by minimising the left hand side of (4.49) with respect to μ and setting the result to zero. It is found to be

$$\mu_{opt} = \frac{1}{(N-D)\sigma_x^2}. \quad (4.51)$$

4.3.3 The steady MSE

When condition (4.50) is satisfied, the output MSE converges to a steady-state value [81] which is obtained by letting $n \rightarrow \infty$ in (4.46). It is given by

$$J(\infty) = \frac{J_{min}}{1 - \frac{1}{2}\mu(N-D)\sigma_x^2}. \quad (4.52)$$

For small value of μ we have

$$J(\infty) \cong J_{min} + \frac{1}{2}\mu(N-D)\sigma_x^2 J_{min}. \quad (4.53)$$

Substituting (4.51) into (4.53) we find that the steady-state MSE associated with the optimum step-size is:

$$J(\infty) \cong \frac{3}{2} J_{min}. \quad (4.54)$$

with J_{min} defined by (4.17).

4.3.4 The misadjustment factor

The average excess MSE is $[J(n) - J_{min}]$. The ratio of this to J_{min} can be regarded as a misadjustment factor M . It is given by

$$M = \frac{\mu(N - D)\sigma_x^2}{2 - \mu(N - D)\sigma_x^2}. \quad (4.55)$$

The misadjustment in the case when the optimal step size μ_{opt} is employed can be obtained by substituting (4.51) into (4.55); the result is simply

$$M_{\mu=\mu_{opt}} = 1. \quad (4.56)$$

4.3.5 The time constant of the algorithm

It can be seen from equation (4.48) that convergence is exponential and the time-constant is:

$$\tau = -\frac{1}{\ln\{1 - 2\mu\sigma_x^2 + \mu^2(N - D)\sigma_x^4\}}. \quad (4.57)$$

When $\mu = \mu_{opt}$, the minimum time constant is

$$\tau_{min} = -\frac{1}{\ln\{1 - \frac{1}{(N-D)}\}}. \quad (4.58)$$

4.4 Convergence behaviour of the algorithm when using the smooth transition

So far, only the case where the order of the system to be identified is less than that of the adaptive filter is considered. No problem occurs when the order of the adaptive filter is larger than that of the system to be identified. In practice, however,

they usually do not coincide. To overcome this difficulty, the algorithm developed so far is studied again for the case when smooth transitions are introduced into equations (4.1) and (4.2) in the hope that convergence to the exact echo's impulse response will take place.

The idea of introducing smooth transitions rather than sharp ones in the development of the proposed algorithm is due to the fact that often the calculation of the delay between the near-end canceller and the far-end canceller is not correct, hence one or more coefficients in the part of the impulse response identified as zero are in fact non-zero. In this case, excess error is introduced, as will be shown later in the derivations, and thus the algorithm does not converge to the optimal value.

The types of possible transitions between the zero and nonzero parts of the impulse response could be sharp or smooth. Figure 4.4 depicts these transitions. Other transitions with different shapes could be used as well. This type of technique is identical to the one used in designing low-pass FIR filters when the sampling method [36] is used. Adding more samples in the transition band, which exists between the passband and the stopband, enhance the performance of the filter much better than if no samples are present in this band.

Moreover, the coefficients in the transition bands are distributed according to a certain pdf, that is, the probability of a coefficient to be present in the transition band is equal to its cumulative density function. For the coefficients at the end of the near-end canceller, these are defined as follows

$$0 \leq f(i) \leq 1, \quad i = 1, \dots, n_1, \quad (4.59)$$

whereas for those at the beginning of the far-end canceller, they are defined in the following way

$$0 \leq g(i) \leq 1, \quad i = 1, \dots, n_2. \quad (4.60)$$

The functions specified in (4.59) and (4.60) characterise the near-end and far-end transition bands, respectively. The numbers n_1 and n_2 are the numbers of coefficients in their respective transition bands.

The output of the canceller at the n th instant defined by (4.3) is also used, here, with the modification made to \mathbf{W} as defined by

$$\mathbf{W} = \text{diag}(1, \dots, 1, f(1), \dots, f(n_1), 0, \dots, 0, g(1), \dots, g(n_2), 1, \dots, 1), \quad (4.61)$$

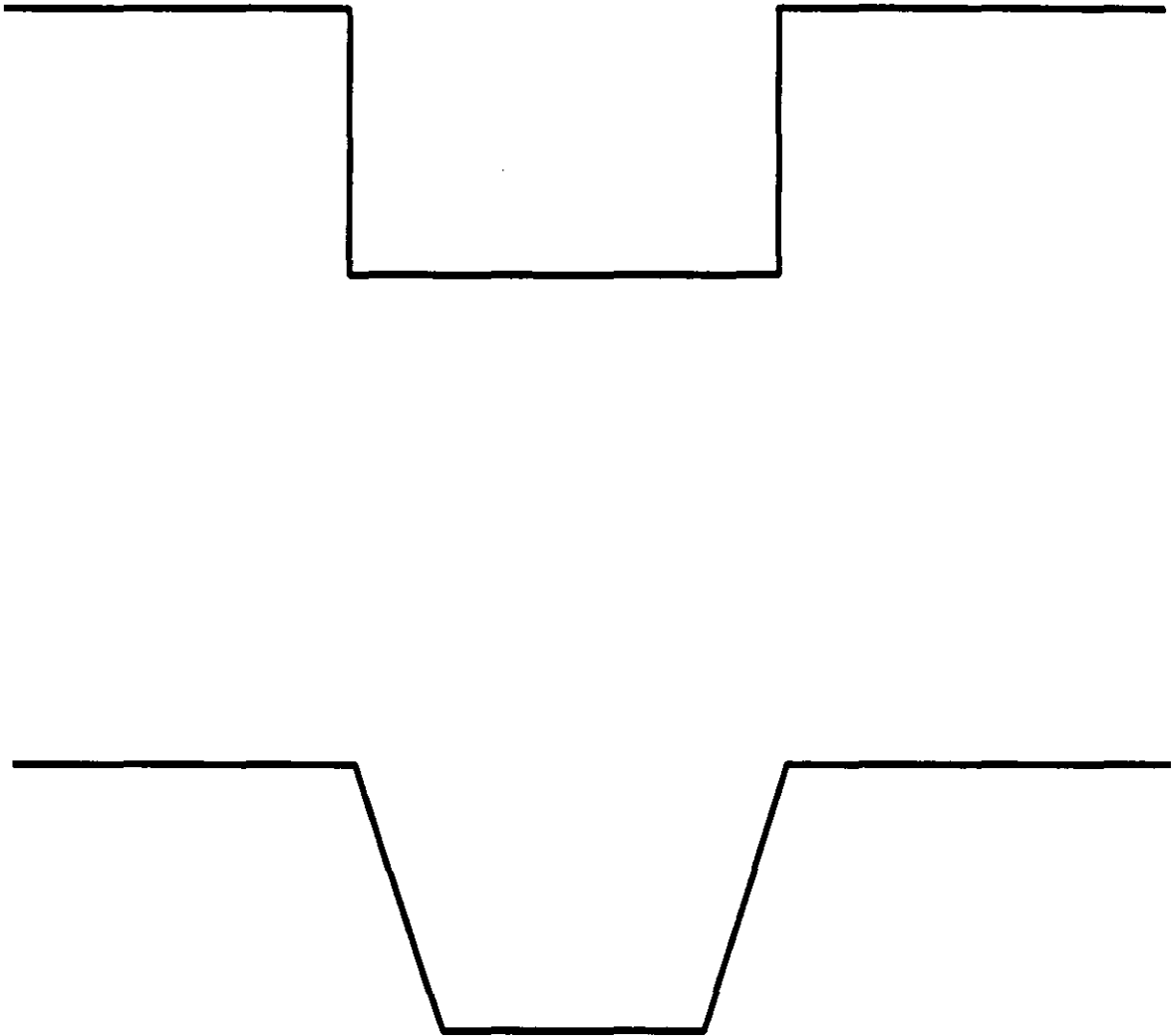


Figure 4.4: Sharp and smooth transitions.

and the the coefficient vector can be written in the following format

$$\mathbf{C}(n) = \begin{pmatrix} \mathbf{C}^u(n) \\ \mathbf{C}'^u(n) \\ \mathbf{0} \\ \mathbf{C}'^l(n) \\ \mathbf{C}^l(n) \end{pmatrix}, \quad (4.62)$$

where $\mathbf{C}^u(n)$ and $\mathbf{C}^l(n)$ designate the portions of the coefficients belonging to the near-end and the far-end sections of the canceller, respectively. The *DS* is the section where the coefficients are zeros, and this is why it is indicated in equation (4.62) by a zero-vector, whereas $\mathbf{C}'^u(n)$ and $\mathbf{C}'^l(n)$ are the coefficients in the near-end and far-end transition bands, respectively.

4.4.1 Convergence in the mean

Similarly as has been done in the section of the convergence of the algorithm in the mean as presented with the sharp transition, the following result is obtained

$$E[\hat{\mathbf{C}}(n+1)] = [\mathbf{I} - \mu \mathbf{WRW}]E[\hat{\mathbf{C}}(n)]. \quad (4.63)$$

Equation (4.63) reveals that the algorithm will converge to the optimal value if all the eigenvalues of the matrix $(\mathbf{I} - \mu \mathbf{WRW})$ are less than unity.

To be able to evaluate the mean value in (4.63), all one needs is to compute the value of \mathbf{WRW} . Since \mathbf{W} is a diagonal matrix, it can be written in the following form

$$\mathbf{W} = \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \quad (4.64)$$

where \mathbf{I} is the identity matrix whose length depends upon the length of the corresponding part of the filter (i.e., the near-end and the far-end of the canceller), and

\mathbf{F} and \mathbf{G} are both $n_1 \times n_1$ and $n_2 \times n_2$ diagonal matrices, respectively, defined by:

$$\mathbf{F} = \text{diag}(f(1), f(2), \dots, f(n_1)), \quad (4.65)$$

and

$$\mathbf{G} = \text{diag}(g(1), g(2), \dots, g(n_2)). \quad (4.66)$$

The elements of the diagonal matrix \mathbf{F} are decreasing as i increases from 1 to n_1 , whereas the elements of \mathbf{G} are increasing as i increases from 1 to n_2 .

The same partition can be done for the autocorrelation matrix \mathbf{R} ,

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \mathbf{R}_{13} & \mathbf{R}_{14} & \mathbf{R}_{15} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \mathbf{R}_{23} & \mathbf{R}_{24} & \mathbf{R}_{25} \\ \mathbf{R}_{31} & \mathbf{R}_{32} & \mathbf{R}_{33} & \mathbf{R}_{34} & \mathbf{R}_{35} \\ \mathbf{R}_{41} & \mathbf{R}_{42} & \mathbf{R}_{43} & \mathbf{R}_{44} & \mathbf{R}_{45} \\ \mathbf{R}_{51} & \mathbf{R}_{52} & \mathbf{R}_{53} & \mathbf{R}_{54} & \mathbf{R}_{55} \end{pmatrix} \quad (4.67)$$

where each \mathbf{R}_{ij} ($1 \leq i \leq 5, 1 \leq j \leq 5$) has dimensions equal to the corresponding entry of the matrix \mathbf{W} in (4.64).

The product \mathbf{WRW} is then found to be equal to the following

$$\mathbf{WRW} = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12}\mathbf{F} & 0 & \mathbf{R}_{14}\mathbf{G} & \mathbf{R}_{15} \\ \mathbf{F}\mathbf{R}_{21} & \mathbf{F}\mathbf{R}_{22}\mathbf{F} & 0 & \mathbf{F}\mathbf{R}_{24}\mathbf{G} & \mathbf{F}\mathbf{R}_{25} \\ 0 & 0 & 0 & 0 & 0 \\ \mathbf{G}\mathbf{R}_{41} & \mathbf{G}\mathbf{R}_{42}\mathbf{F} & 0 & \mathbf{G}\mathbf{R}_{44}\mathbf{G} & \mathbf{G}\mathbf{R}_{45} \\ \mathbf{R}_{51} & \mathbf{R}_{52}\mathbf{F} & 0 & \mathbf{R}_{54}\mathbf{G} & \mathbf{R}_{55} \end{pmatrix}. \quad (4.68)$$

Since the canceller is mainly composed of three components, the near-end and the far-end sections, and the bulk delay between these two sections, one then can

use (4.62) in (4.63) to get :

$$E \begin{pmatrix} \hat{C}^u(n+1) \\ \hat{C}'^u(n+1) \\ 0 \\ \hat{C}'^l(n+1) \\ \hat{C}^l(n+1) \end{pmatrix} = \left(I - \mu \mathbf{WRW} \right) E \begin{pmatrix} \hat{C}^u(n) \\ \hat{C}'^u(n) \\ 0 \\ \hat{C}'^l(n) \\ \hat{C}^l(n) \end{pmatrix}. \quad (4.69)$$

Equation (4.69) can be further simplified if expressed in a vector form as in the following :

$$E \begin{pmatrix} \hat{C}^u(n+1) \\ \hat{C}'^u(n+1) \\ \hat{C}'^l(n+1) \\ \hat{C}^l(n+1) \end{pmatrix} = \left(I - \mu \mathbf{A} \right) E \begin{pmatrix} \hat{C}^u(n) \\ \hat{C}'^u(n) \\ \hat{C}'^l(n) \\ \hat{C}^l(n) \end{pmatrix}, \quad (4.70)$$

with

$$\mathbf{A} = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12}\mathbf{F} & \mathbf{R}_{14}\mathbf{G} & \mathbf{R}_{15} \\ \mathbf{FR}_{21} & \mathbf{FR}_{22}\mathbf{F} & \mathbf{FR}_{14}\mathbf{G} & \mathbf{FR}_{25} \\ \mathbf{GR}_{41} & \mathbf{GR}_{42}\mathbf{F} & \mathbf{GR}_{44}\mathbf{G} & \mathbf{GR}_{45} \\ \mathbf{R}_{51} & \mathbf{R}_{52}\mathbf{F} & \mathbf{R}_{54}\mathbf{G} & \mathbf{R}_{55} \end{pmatrix} \quad (4.71)$$

where it is clear that the step size, μ , depends on the largest eigenvalue of \mathbf{A} .

It therefore follows that, at steady state, the following will be obtained, provided the right step size has been chosen,

$$\hat{C}(n) \rightarrow 0, \quad (4.72)$$

or equivalently

$$\mathbf{C}(n) = \mathbf{C}_{opt}, \quad (4.73)$$

where \mathbf{C}_{opt} is still defined by $\mathbf{C}_{opt} = [\mathbf{WRW}]^{-1}\mathbf{WP}$, and \mathbf{W} is given by (4.61).

This happens if the step size is chosen to be:

$$0 < \mu < \frac{2}{\lambda_{max}}, \quad (4.74)$$

where λ_{max} is the largest eigenvalue of \mathbf{A} .

If we now consider a practically-motivated assumption that the near-end and the far-end sections are uncorrelated, (4.71) reduces to the following

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 \end{pmatrix}, \quad (4.75)$$

with

$$\mathbf{A}_1 = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12}\mathbf{F} \\ \mathbf{F}\mathbf{R}_{21} & \mathbf{F}\mathbf{R}_{22}\mathbf{F} \end{pmatrix}, \quad (4.76)$$

and

$$\mathbf{A}_2 = \begin{pmatrix} \mathbf{G}\mathbf{R}_{44}\mathbf{G} & \mathbf{G}\mathbf{R}_{45} \\ \mathbf{R}_{54}\mathbf{G} & \mathbf{R}_{55} \end{pmatrix}, \quad (4.77)$$

and the corresponding step size is found to be

$$0 < \mu < \frac{2}{\max[\lambda_{max1}, \lambda_{max2}]}, \quad (4.78)$$

where λ_{max1} and λ_{max2} are the largest eigenvalues of \mathbf{A}_1 and \mathbf{A}_2 , respectively.

4.4.2 Convergence in the mean square

Here, too, for the convergence in mean square of the algorithm, it can be shown, following the derivation of the algorithm in the mean square section with the sharp transition, that

$$\theta_{n+1} = \theta_n - 2\mu E[e(n)\mathbf{X}^T(n)\mathbf{W}\hat{\mathbf{C}}(n)] + \mu^2 E[e^2(n)\mathbf{X}^T(n)\mathbf{W}\mathbf{X}(n)]. \quad (4.79)$$

Under the often-made assumption [78]-[79] that the input signal sequence $\{\mathbf{x}_i\}$ is an independent and identically distributed Gaussian random sequence with zero mean and variance σ_x^2 , the derivation of (4.79) is obtained as explained in the following. Consider the following expression

$$E[e(n)\mathbf{X}^T(n)\hat{\mathbf{C}}(n)] = J(n) - J_{min} \quad (4.80)$$

evaluated in [80]. Then, taking this result into account one can obtain an expression for the second term of equation (4.79), given by

$$E[e(n)\mathbf{X}^T(n)\mathbf{W}\hat{\mathbf{C}}(n)] = J(n) - J_{min}. \quad (4.81)$$

The third term of the right-hand side of equation (4.79) is readily evaluated. Given that

$$\mathbf{X}^T(n)\mathbf{W}\mathbf{X}(n) = \|\mathbf{X}(n)\|^2 - \sum_{i \in DS} x_i^2 + \sum_{i=1}^{n_1} (f(i) - 1)x_i^2 + \sum_{i=1}^{n_2} (g(i) - 1)x_i^2, \quad (4.82)$$

we can then write the following

$$\begin{aligned} E[e^2(n)\mathbf{X}^T(n)\mathbf{W}\mathbf{X}(n)] &= E \left[\|e(n)\|^2 \left[\|\mathbf{X}(n)\|^2 - \sum_{i \in DS} x_i^2 + \sum_{i=1}^{n_1} (f(i) - 1)x_i^2 \right. \right. \\ &\quad \left. \left. + \sum_{i=1}^{n_2} (g(i) - 1)x_i^2 \right] \right] \\ &= (N - D)J(n)\sigma_x^2 \\ &\quad + \left[\sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2 \right] J(n)\sigma_x^2. \end{aligned} \quad (4.83)$$

Inserting (4.83) in (4.79) yields :

$$\begin{aligned} \theta_{n+1} &= \theta_n - 2\mu E[e(n)\mathbf{X}^T(n)\mathbf{W}\hat{\mathbf{C}}(n)] \\ &\quad + \mu^2 [N - D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2] J(n)\sigma_x^2. \end{aligned} \quad (4.84)$$

Given the expression of (4.81) and using it in (4.84), the following is obtained:

$$\begin{aligned} \theta_{n+1} &= \theta_n - 2\mu [J(n) - J_{min}] \\ &\quad + \mu^2 [N - D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2] J(n)\sigma_x^2. \end{aligned} \quad (4.85)$$

In the limit when θ_{n+1} reaches θ_n , the value of the step size is found to be

$$\mu = \frac{2(1 - \frac{J_{min}}{J(n)})}{[N - D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2]\sigma_x^2}. \quad (4.86)$$

When $[J(n) - J_{min}]$ is approximated by $\sigma_x^2\theta_n$ [81], (4.85) becomes

$$\begin{aligned} \theta_{n+1} &= \{1 - 2\mu\sigma_x^2 + \mu^2 [N - D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2]\sigma_x^4\}\theta_n \\ &\quad + \mu^2 J_{min}(N - D)\sigma_x^2. \end{aligned} \quad (4.87)$$

For the expression in (4.87) to converge, it is required that

$$\left| 1 - 2\mu\sigma_x^2 + \mu^2[N - D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2]\sigma_x^4 \right| < 1. \quad (4.88)$$

Hence, for convergence, the step size μ has to satisfy :

$$0 < \mu < \frac{2}{[N - D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2]\sigma_x^2}. \quad (4.89)$$

The step size which provides the fastest speed of convergence to the corresponding steady-state MSE is obtained by setting the derivative of the expression in the left hand side of (4.88) to zero, that is,

$$\mu_{opt} = \frac{1}{[N - D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2]\sigma_x^2}, \quad (4.90)$$

or one-half the maximum step size.

4.4.3 The steady state MSE

When condition (4.89) is satisfied, the output MSE converges to a steady-state value obtained by letting $n \rightarrow \infty$ in (4.85), given by :

$$J(\infty) = \frac{J_{min}}{1 - \frac{1}{2}\mu[N - D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2]\sigma_x^2}. \quad (4.91)$$

For small values of μ , we have the following approximation :

$$J(\infty) \cong J_{min} \left\{ 1 + \frac{1}{2}\mu[N - D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2]\sigma_x^2 \right\}. \quad (4.92)$$

Substituting (4.90) into (4.92) we find that the steady-state MSE associated with the optimum step-size is given by:

$$J(\infty) \cong \frac{3}{2}J_{min}, \quad (4.93)$$

with $J_{min} = E[d^2(n)] - \mathbf{C}_{opt}^T \mathbf{W} \mathbf{P}$, and \mathbf{W} defined by (4.61).

4.4.4 The misadjustment factor

The average excess MSE is $(J(n) - J_{min})$. The ratio of this error to the J_{min} can be regarded as a misadjustment factor M . It is given by

$$M = \frac{\mu[N - D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2]\sigma_x^2}{2 - \mu[N - D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2]\sigma_x^2}. \quad (4.94)$$

In the case when the optimum size μ_{opt} is employed the misadjustment can be obtained by substituting (4.90) into (4.94). This results in the worst misadjustment value of :

$$M_{\mu=\mu_{opt}} = 1. \quad (4.95)$$

4.4.5 The time constant of the algorithm

It can be seen from equation (4.87) that the convergence is exponential and the time-constant is:

$$\tau = \frac{1}{\ln\{1 - 2\mu\sigma_x^2 + \mu^2[N - D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2]\sigma_x^4\}}. \quad (4.96)$$

When $\mu = \mu_{opt}$, the minimum time constant is

$$\tau_{min} = \frac{1}{\ln\{1 - \frac{1}{[N - D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2]}\}}. \quad (4.97)$$

Table 4.1 summarises and compares all the results found when using the sharp transition as compared to those obtained when using the smooth one.

4.5 Excess mean square error

Before we see that excess error has been introduced to the algorithm, let us first see that, in fact, there is no potential for bias.

Define $\bar{e}(n)$ to be the following value

$$\begin{aligned} \bar{e}(n) &= e(n) - e_{opt} \\ &= [\mathbf{C}(n) - \mathbf{C}_{opt}]^T \mathbf{W} \mathbf{X}(n). \end{aligned} \quad (4.98)$$

Table 4.1: Comparison of the algorithm for sharp and smooth transitions.

	Sharp transition	Smooth transition
μ_{max}	$\frac{2}{(N-D)\sigma_x^2}$	$\frac{2}{[N-D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2]\sigma_x^2}$
μ_{opt}	$\frac{1}{(N-D)\sigma_x^2}$	$\frac{1}{[N-D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2]\sigma_x^2}$
τ	$\frac{-1}{\ln\{1 - 2\mu\sigma_x^2 + \mu^2(N-D)\sigma_x^4\}}$	$\frac{-1}{\ln\{1 - 2\mu\sigma_x^2 + \mu^2[N-D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2]\sigma_x^4\}}$
τ_{min}	$\frac{-1}{\ln\{1 - \frac{1}{(N-D)}\}}$	$\frac{-1}{\ln\{1 - \frac{1}{[N-D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2]}\}}$
M	$\frac{\mu(N-D)\sigma_x^2}{2 - \mu(N-D)\sigma_x^2}$	$\frac{\mu[N-D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2]\sigma_x^2}{2 - \mu[N-D + \sum_{i=1}^{n_1} f(i) + \sum_{i=1}^{n_2} g(i) - n_1 - n_2]\sigma_x^2}$
$M_{\mu=\mu_{opt}}$	1	1

Taking the expectation of both sides of the above equation, then

$$E[\tilde{e}(n)] = E \left[[C(n) - C_{opt}]^T \right] W E[X(n)]; \quad (4.99)$$

since $\{x_i\}$ is an i.i.d Gaussian random sequence with zero mean as mentioned earlier, the result is simply

$$E[\tilde{e}(n)] = 0. \quad (4.100)$$

This therefore indicates clearly that there is no potential for bias. In this case, there should therefore be additional excess error introduced. To check the validity of this statement let us do the following.

Subscripts “*u*” and “*s*” will be used in this part of the derivation to distinguish between the sharp transition and the smooth transition, respectively.

Recall that the minimum MSE associated with the optimum value, $C_{opt,s}$, when using the smooth transition is found to be

$$J_{min,s} = E[d^2(n)] - C_{opt,s}^T W_s P. \quad (4.101)$$

Similarly, the corresponding value derived for the sharp transition is given by

$$J_{min,u} = E[d^2(n)] - C_{opt,u}^T W_u P. \quad (4.102)$$

Subtracting equation (4.101) from equation (4.102), leads to :

$$J_{min,u} = J_{min,s} + [C_{opt,s}^T W_s - C_{opt,u}^T W_u] P. \quad (4.103)$$

Since W_s can be written in terms of W_u , equation (4.103) then becomes

$$\begin{aligned} J_{min,u} &= J_{min,s} + [C_{opt,s} - C_{opt,u}]^T W_u P \\ &\quad + \sum_{i=1}^{n_1} f(i) C_{i,opt,s} P_i + \sum_{i=1}^{n_2} g(i) C_{i,opt,s} P_i. \end{aligned} \quad (4.104)$$

This result shows that excess error has been introduced to the algorithm when using the sharp transition defined in section 4.3. The amount of excess error is clearly seen through the second, the third, and the fourth terms on the right hand side of (4.104).

Ultimately it has been shown that the algorithm when using the smooth transition over the sharp one will end up with less excess mean square error; hence good echo cancellation is realised with the algorithm using the smooth transition. Simulation results support this statement, as will be shown in the simulation results section.

4.6 Computational complexity

A question which might be asked at this stage is : which transition outperforms the other one in terms of computational effort? The order of computations can be explained as follows. Suppose that n_1 and n_2 coefficients are missing in the near-end and the far-end sections of the canceller, respectively. When the sharp transition is used to compute the output of the canceller and no errors are made in the bulk delay, $(N - D - 1)$ additions and $(N - D)$ multiplications are needed for this purpose. Furthermore $(N - D - n_1 - n_2 - 1)$ additions and $(N - D - n_1 - n_2)$ multiplications are needed to compute the output of the canceller when the sharp transition is used with errors in the bulk delay defined by the number of coefficients n_1 and n_2 . Now, if the smooth transition is used, the number of additions (*adds*) needed is:

$$(N - D - n_1 - n_2 - 1) < \text{adds} < (N - D - 1), \quad (4.105)$$

and the number of multiplications (*mults*) is:

$$(N - D - n_1 - n_2) < \text{mults} < (N - D). \quad (4.106)$$

This is due to the fact that since the coefficients in the smooth transition, which have been selected using the uniform pdf, are either present or not present, not all of them contribute in the computation of the output, and hence lead to the number of additions and multiplications bounded by (4.105) and (4.106), respectively.

Ultimately, a compromise between computational complexity and performance is observed when using either of the transitions.

Table 4.2 summarises the computational complexity of the algorithms with different transitions.

4.7 Simulation results

In the simulations presented in this section, the signaling is binary ($x_i = \pm 1$) and the arbitrarily chosen sampled impulse response is shown in Fig. 4.5. The additive noise was simulated as a uniformly random process with zero mean, and the signal to noise ratio is 30 dB. The optimal step-size parameters obtained from (4.51) and (4.90) are used in all the simulations, and the learning curves obtained are the average of 200 runs.

The coefficients missed in the near-end section of the echo canceller start from the end of the impulse response and go to the left, while those missed in the far-end start from the beginning of the impulse response and go to the right. We will continue with to this notation during the remainder of this work.

Figure 4.6 shows good agreement between the theoretical and the experimental learning curves when a sharp transition is used and no error is made in the bulk delay. Figure 4.7, however, shows the closeness of the learning characteristics of the simulation and the theoretical results when four coefficients in each of the near-end and the far-end sections of the canceller have been missed and using the smooth transition. The uniform pdf is given to the smooth transition.

Figure 4.8 through Fig. 4.13 depict the amount of mean square error obtained for the two types of transitions, against the number of missed coefficients in the transition bands of the echo canceller. The sharp transition has been used in Fig. 4.8 through Fig. 4.10, while in Fig. 4.11 until Fig. 4.13 the smooth transition was used. The coefficients in the transition bands were distributed uniformly, that is using the uniform pdf, when using the smooth transition.

To validate the theoretical derivations, in each of these figures, the experimental result is plotted against its theoretical counterpart.

Table 4.2: Computational complexity of the algorithms with different transitions.

	Number of additions	Number of multiplications
The sharp transition with no errors	$N - D - 1$	$N - D$
The sharp transition with errors	$N - D - n_1 - n_2$	$N - D - n_1 - n_2$
The smooth transition	$< (N - D - 1)$ and $> (N - D - n_1 - n_2 - 1)$	$< (N - D)$ and $> (N - D - n_1 - n_2)$

Figure 4.8 and Fig. 4.11 correspond to the near-end of the canceller alone, while Fig. 4.9 and Fig. 4.12 correspond for the far-end of the canceller only. Similarly, Fig. 4.10 and Fig. 4.13 are the result of the near-end and the far-end combined together. It can be seen from the figures plotted for the smooth transition that the number of missed coefficients start with 1, while they start with 0 when the sharp transition is used. The reason behind it that they start with 0 when the sharp transition is used means that no coefficient was missed. But, this is not the case for the smooth transition, since it is a matter of whether a coefficient is present or not.

One observes that in all the figures presented the simulation results are in quite close agreement with the theoretical predictions.

Now if we compare the figures obtained when using the sharp transition to those obtained when using the smooth transition, we can draw the conclusion that the amount of mean square error is much less than when using the smooth transition. That is, the value obtained when using the smooth transition gives less mean square error to the algorithm. This clearly supports the theoretical derivations.

4.8 Summary

A probabilistic study was suggested for the study of the newly proposed algorithm. It was shown that coefficients belonging to the delay section and those not belonging to this section were assigned a probability value depending on their presence or absence in these sections.

The study was carried out, in this chapter, evaluated the performance of the proposed algorithm. The performance of the algorithm has been studied for two transitions, a sharp transition and a smooth one. For both of these transitions, steady-state and transient behaviour were investigated, as well as a comparison between the algorithms developed for each transition.

Also, an analytical expression has been also given for the amount of excess mean square error when using the sharp transition over the smooth transition.

Finally, simulation results are found to agree well with the analysis, and the algorithm presented reasonable complexity as far as the computational burden is concerned.

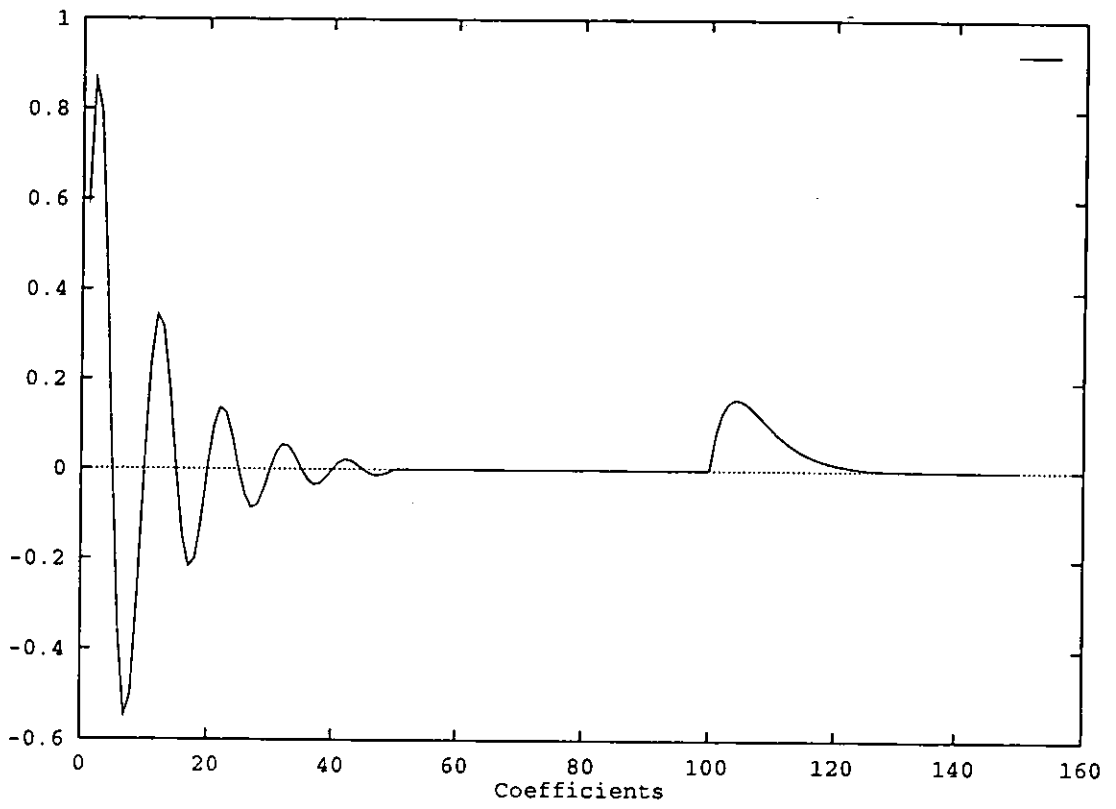


Figure 4.5: Impulse response of the channel.

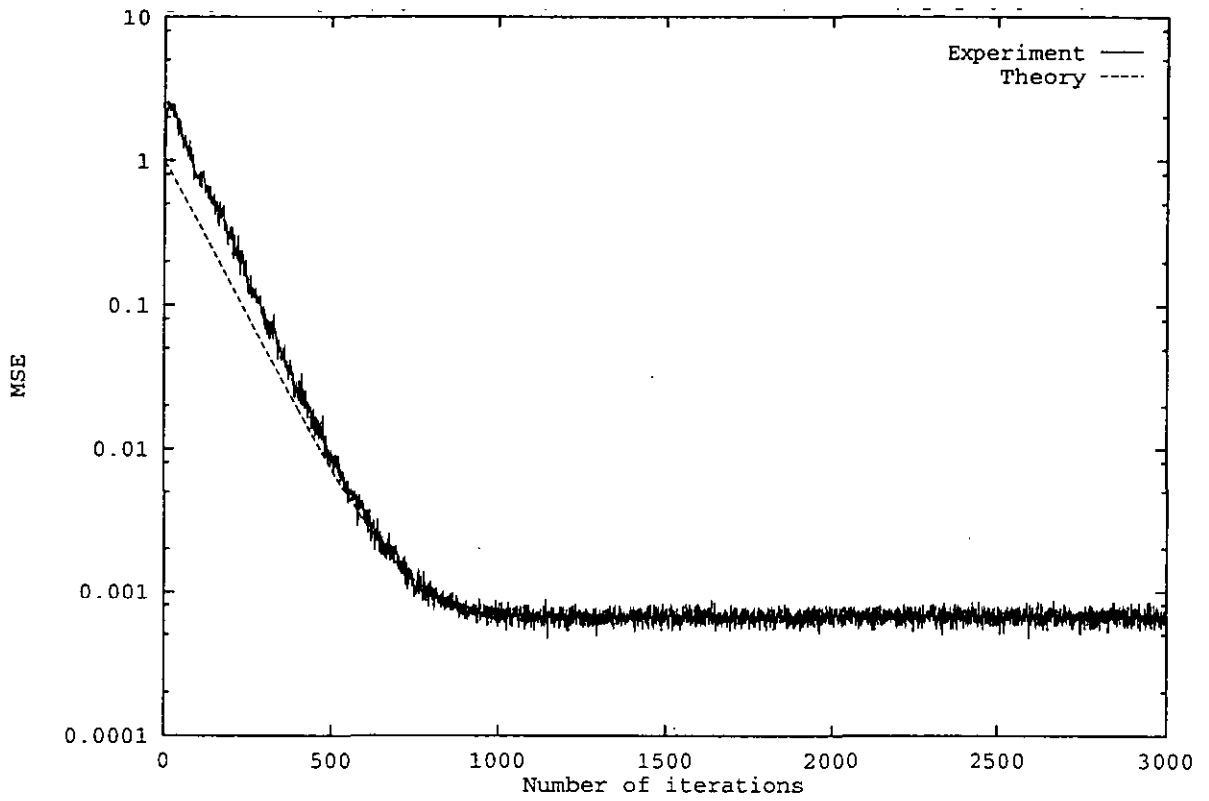


Figure 4.6: Learning curves when using the sharp transition and no error in the bulk delay.

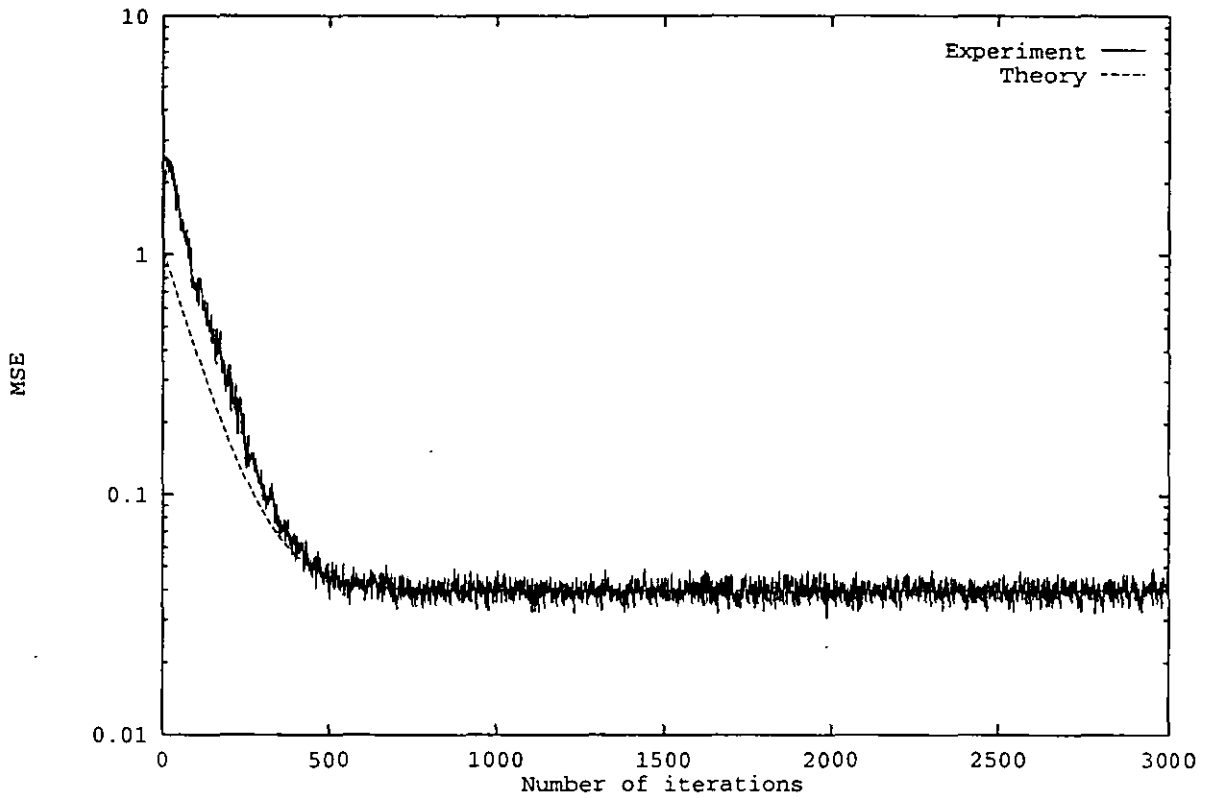


Figure 4.7: Learning curves when using the smooth transition and 4 coefficients in each of the near-end and the far-end sections of the canceller have been missed.

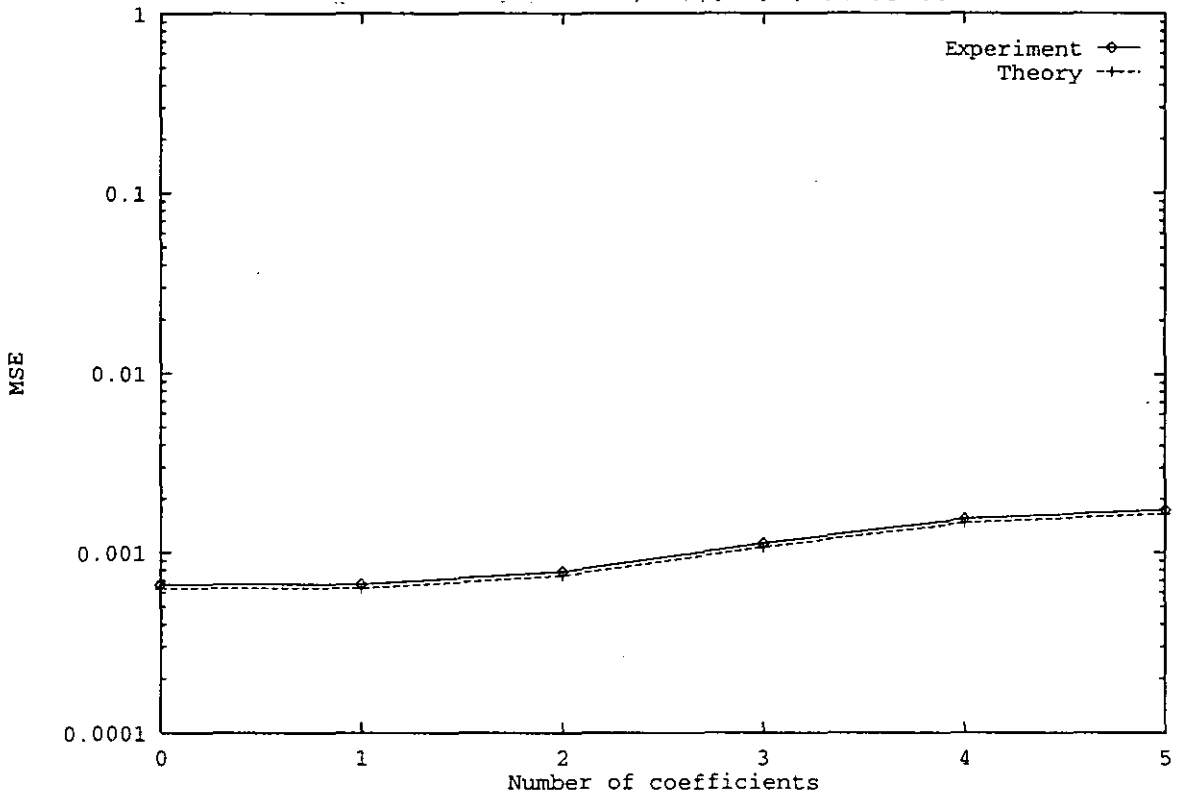


Figure 4.8: MSE versus the number of missed coefficients in the near-end section of the canceller when using the sharp transition.

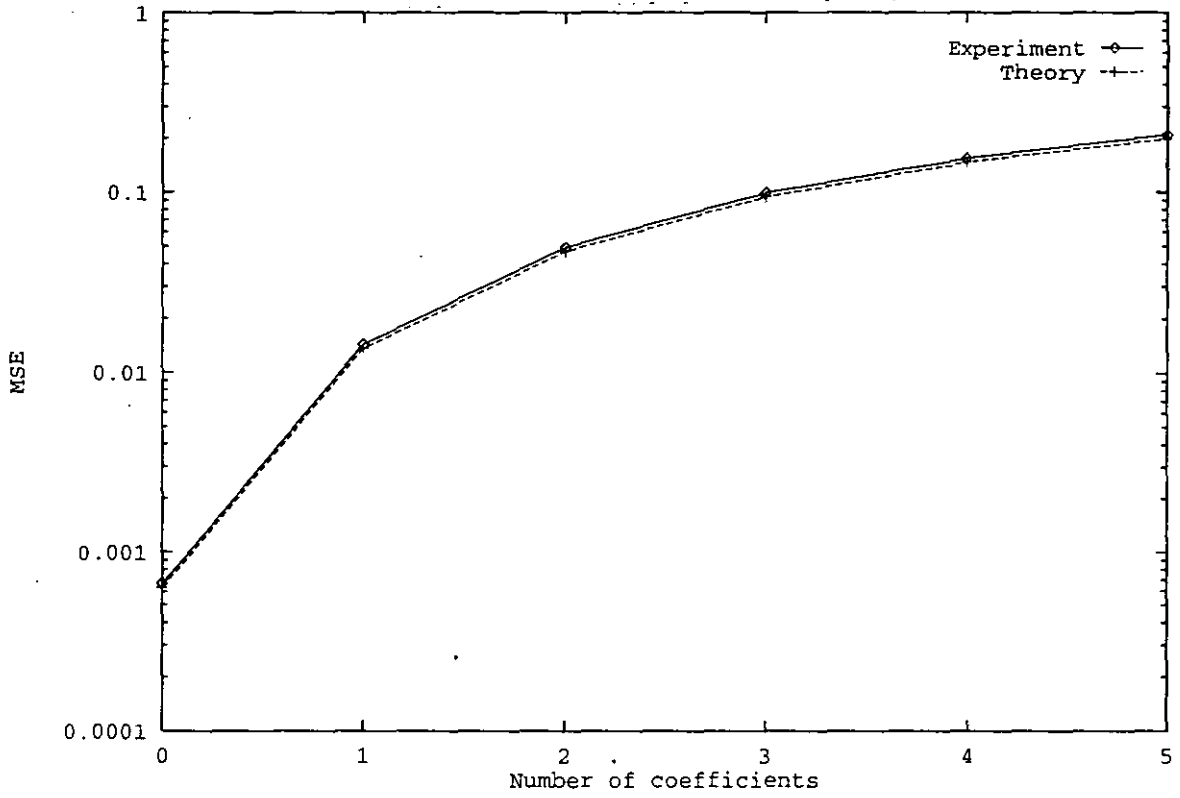


Figure 4.9: MSE versus the number of missed coefficients in the far-end section of the canceller when using the sharp transition.

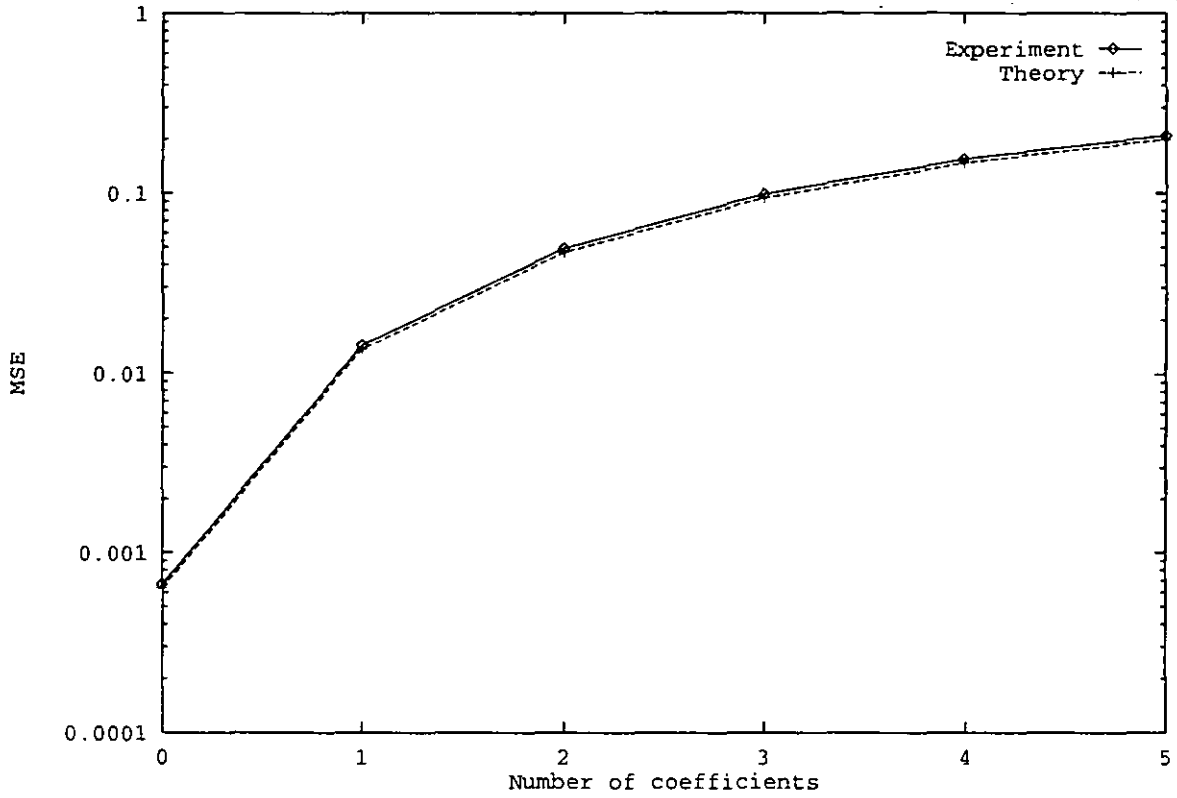


Figure 4.10: MSE versus the number of missed coefficients in the near-end and the far-end sections of the canceller when using the sharp transition.

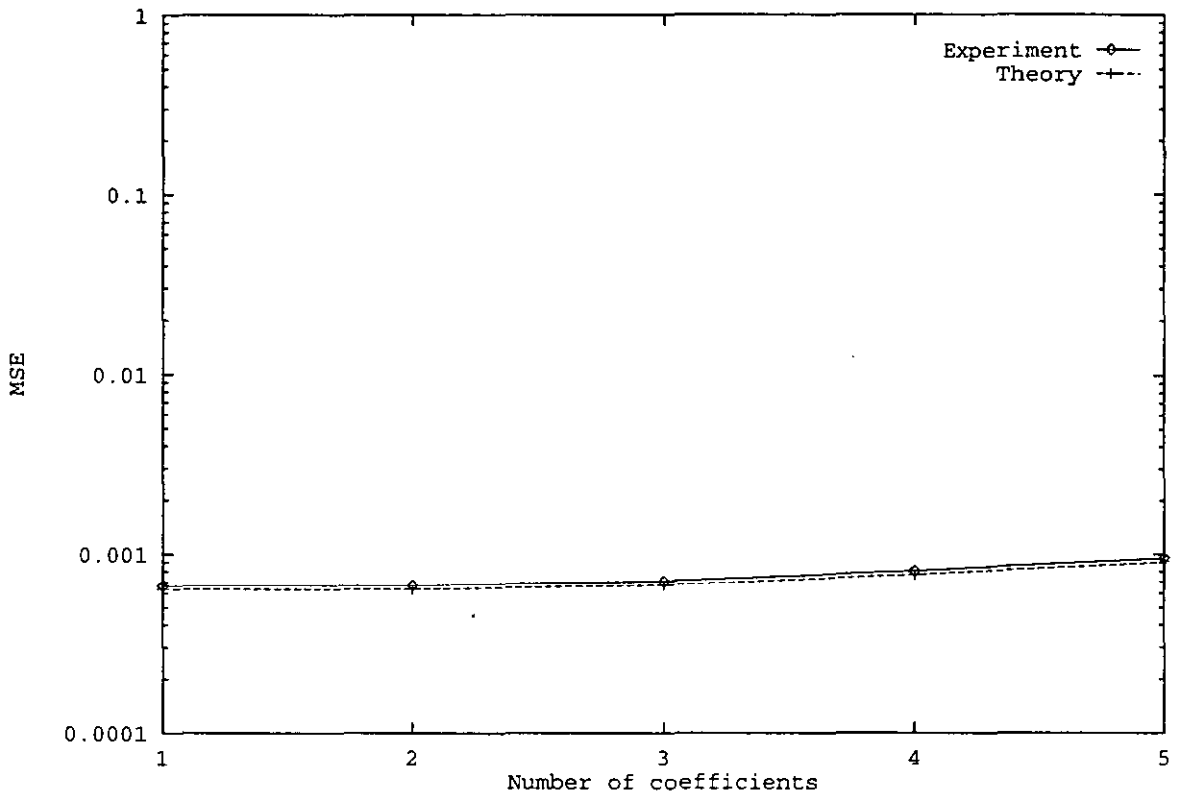


Figure 4.11: MSE versus the number of missed coefficients in the near-end section of the canceller when using the smooth transition.

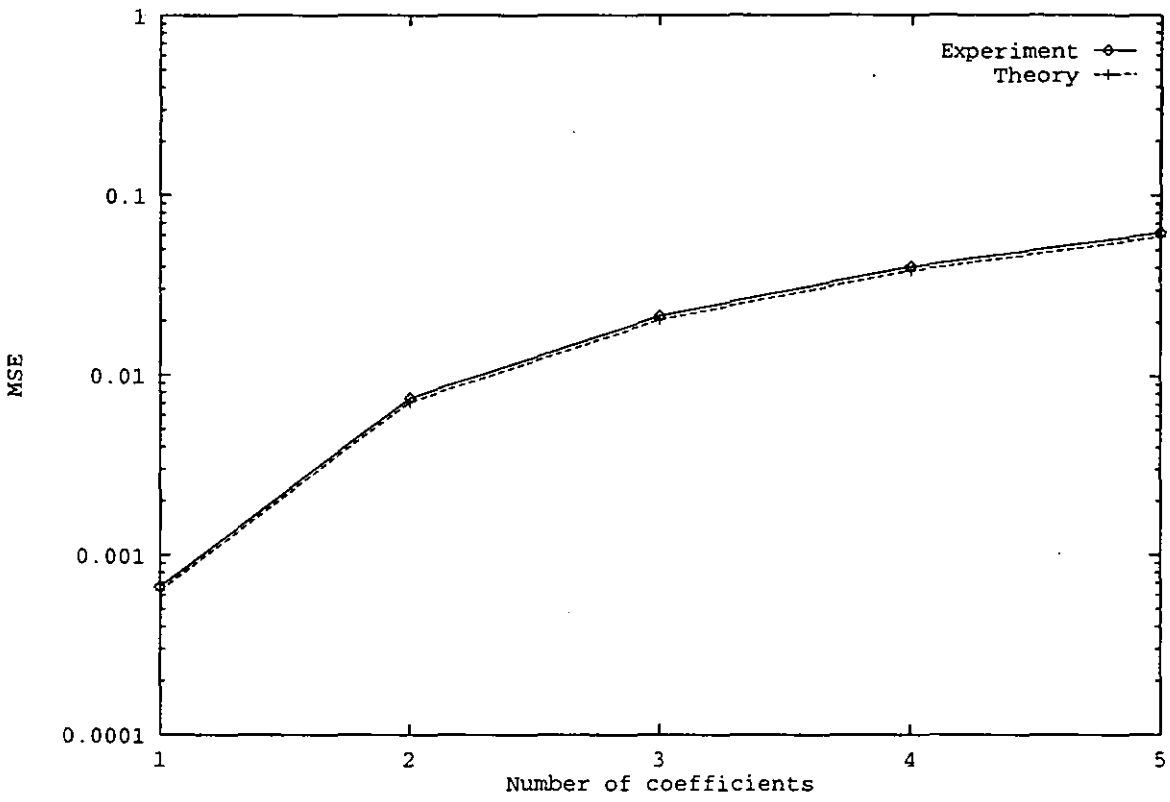


Figure 4.12: MSE versus the number of missed coefficients in the far-end section of the canceller when using the smooth transition.

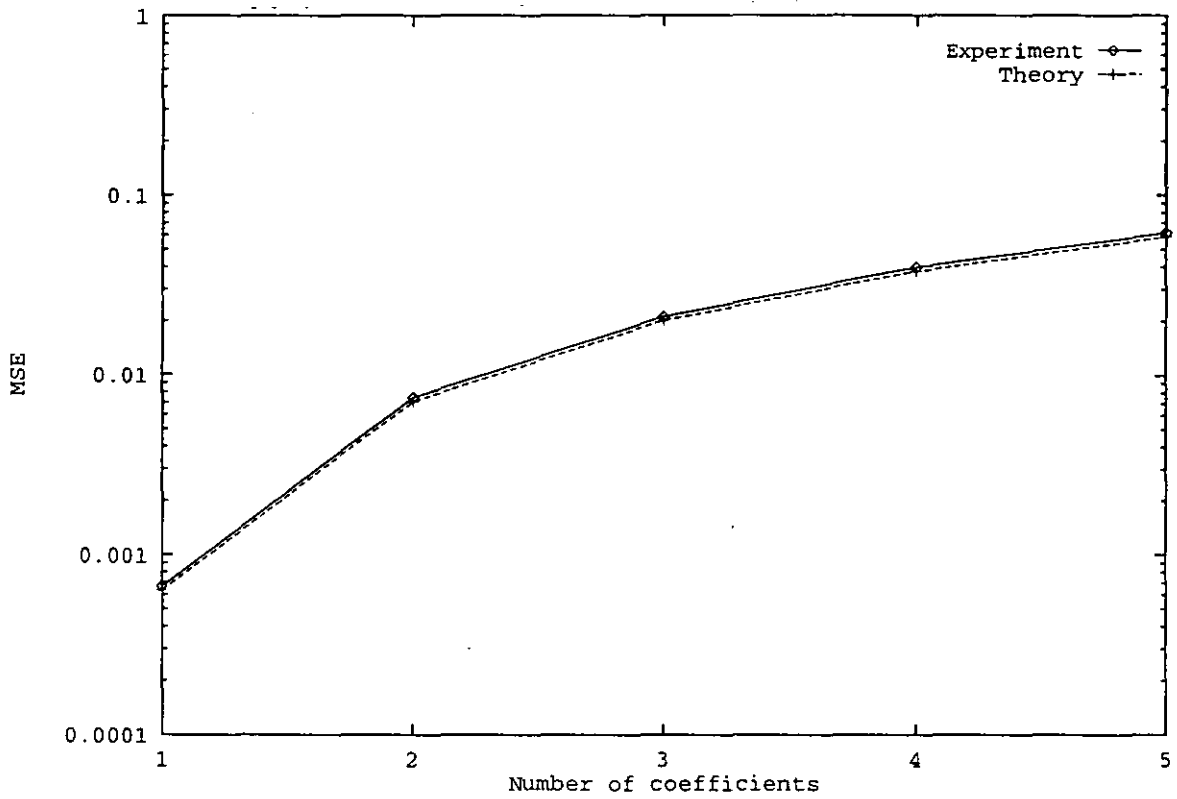


Figure 4.13: MSE versus the number of missed coefficients in the near-end and the far-end sections of the canceller when using the smooth transition.

Chapter 5

Adaptive Echo Cancellation Using Least Mean Mixed-Norm Algorithm

5.1 Introduction

The performance of an adaptive filter depends mainly on the algorithm used for updating the filter weights. Since adaptive filters depend mainly on the choice of the cost function used in the minimisation of the process, one can expect their respective performances to be different.

Based on the choice of the minimisation function, these adaptive schemes can then be either simple or cumbersome, and stable or unstable. The recursive least-squares (RLS) algorithm [7] is designed to minimise the sum of squares of the output. This, however, will end up requiring infinite memory since the values of the filter coefficients are functions of all past inputs. To overcome this burden, a forgetting factor must be introduced into the algorithm, so that recent data are given greater importance than the old data. This modification will end up with an exponentially weighted sum of squares at the output having a finite memory and the corresponding algorithm will be named the exponentially weighted RLS [15].

The RLS algorithm is closely related to Kalman filter theory. The derivation of the Kalman filter [82]-[83] is based on modelling a linear dynamical system by forming a pair of equations, namely a state equation describing the motion of the

system, and a measurement equation describing the observation process. Even though system-based Kalman filters converge faster than their RLS counterpart [84], they still require heavy computations. A solution to this problem is to use fast RLS algorithms [7].

Fast RLS algorithms offer the improved convergence properties of least-squares algorithms by exploiting the so-called shifting property that is encountered in most sequential estimation problems. Depending on the type of filter structure employed, three different classes of fast algorithms may be identified. The first class is the transversal filter-based fast algorithms with two sub-classes, the fast Kalman algorithm [30], and the fast transversal filters (FTF) algorithm [31]-[32]. The second class is the lattice predictor-based fast RLS algorithms [85]-[88], and finally the third class is the QR decomposition-based least-squares algorithms [89]-[90].

In all the above mentioned algorithms, the effect of the statistics of the signal on the convergence of the algorithm is not observed. However, algorithms based on the least mean squares (LMS) algorithm [1] are sensitive to the eigenvalue spread of the input signal. But, contrary to other algorithms, their complexity is much lower which is why they are widely used. The LMS algorithm consists of minimising the square of the error. In [8] the least mean fourth (LMF) algorithm was suggested where it arose as a special case of the more general family of steepest descent algorithms [7] with $2k$ error norms, k being a positive integer.

Being structurally simple and computationally efficient, the LMS algorithm has been widely used in system identification [91], equalisation [13], noise cancellation [14], and echo cancellation [17]. Its performance is now well understood, by means of independence theory [1], [45].

Since echo cancellers [17] are part of these adaptive filters, adaptive algorithms are used to update their tap weights, and depending on the choice of both the cost function and the structure used, their respective performance will differ from one choice to another one.

It is known that the LMS algorithm is optimal only if the input signal is white, and consequently its convergence time will be the shortest in this case. Recently developed algorithms [9], [10] based on a combination of the LMS and the LMF algorithms are proposed. These are designed to minimise one single controlled-mixed cost function. They result in fast convergence rates and small misadjustment errors.

In this work, a new algorithm applied to long echo cancellers with two sections, the near-end and the far-end sections, is proposed. This is depicted in Fig. 5.1. It is different from that of [9] and [10] in the sense that it will minimise two distinct cost functions, i.e., one for each section of the echo canceller. This consists of minimising the MSE in the near-end section of the echo canceller and the MFE in its far-end section, and it will be henceforth called the least mean mixed-norm (LMMN) algorithm. As can be seen from Fig. 5.1, the proposed scheme takes advantage of the structure of the echo canceller.

The LMMN algorithm leads to a lower minimum mean square error, hence results in less misadjustment, and a faster convergence compared to the one obtained by the standard algorithm, i.e., the LMS algorithm when applied to both sections of the echo canceller [3], [16], [17], [92].

However, as it will be later discussed in the derivations, the LMMN algorithm relies on the variance of the noise as it is the case for the LMF algorithm. It happens that if this value is not the actual one, the algorithm would deviate from reaching the optimum solution. Since this value, in general, is not known a priori, suggests then, that it should be estimated close to its range. This parameter and related ones will be investigated in detail in the next chapter.

In the next section the algorithm is derived. Section 5.3 deals with the convergence of the algorithm, while the simulations performed to support the theoretical derivations are presented in the next chapter along with the algorithms used for comparison.

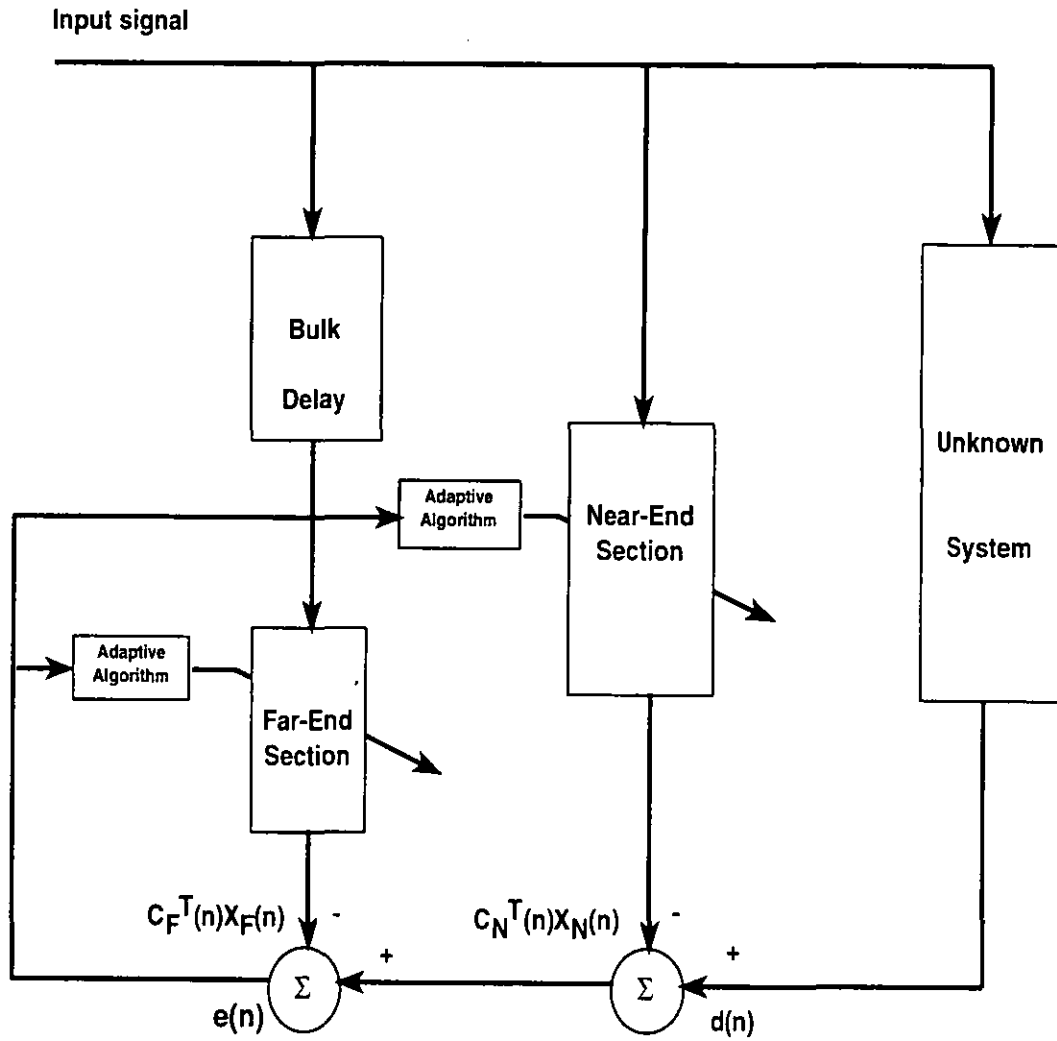


Figure 5.1: Echo canceller with new updating scheme.

5.2 Derivation of the algorithm

Echoes are generated in telephone circuits as the result of impedance mismatches in these circuits. They arise in various situations in telecommunications networks and impair communication quality. To counteract this echo phenomenon, tools must be developed to either completely eliminate it, the ideal requirement, or at least substantially reduce its adverse effects so as to achieve transmission of good quality. Echo cancellation techniques [3] are used for this purpose. The purpose of an echo canceller is, then, to compensate for this distortion caused by echoes, by synthesising and subtracting a replica of it from the returned signal.

the LMMN algorithm will be used to update the tap coefficients of the echo canceller using the new proposed cost functions. The algorithm will be described for the adaptive finite impulse response (FIR) filter.

Since we have two sections, the near-end and the far-end sections, it is convenient to define a vector notation for the filter coefficients

$$\mathbf{C}(n) = [\mathbf{C}_N(n), \mathbf{C}_F(n)]^T, \quad (5.1)$$

where T denotes the vector transposition operation and subscripts N and F represent the near-end section and the far-end section of the canceller, respectively. Identically to (5.1), the reference input samples to the adaptive FIR filter is given by:

$$\mathbf{X}(n) = [\mathbf{X}_N(n), \mathbf{X}_F(n)]^T. \quad (5.2)$$

The output of the canceller at the n th instant is given by:

$$\begin{aligned} y(n) &= \mathbf{C}_N^T(n)\mathbf{X}_N(n) + \mathbf{C}_F^T(n)\mathbf{X}_F(n) \\ &= \mathbf{C}^T(n)\mathbf{X}(n). \end{aligned} \quad (5.3)$$

5.2.1 The proposed performance functions

In the course of the analysis, the following assumptions are made for the performance analysis:

- The noise $w(n)$ is independent of the input signal $\mathbf{X}(n)$, both of zero mean, and \mathbf{x} ; has variance σ_x^2 .

- The input process $\mathbf{X}(n)$ is an independent, identically distributed Gaussian random variable. This is a fairly restrictive assumption and under this assumption the autocorrelation matrix $\mathbf{R} = E[\mathbf{X}(n)\mathbf{X}^T(n)]$ becomes $\mathbf{R} = \sigma_x^2\mathbf{I}$.

- The weight error vector, as will be defined later, is independent of the input $\mathbf{X}(n)$; this is a consequence of the fundamental assumption [7].

In most applications in data transmission the second mentioned assumption above will be valid, since mostly the data signals that are used in the canceller are statically independent [78].

In most practical instances, the adaptive process is oriented towards minimising the mean-square value (MSE). In this work the proposed performance criteria, are defined, respectively, for the near-end section and the far-end section as:

$$J_N(n) = E[e^2(n)], \quad (5.4)$$

$$J_F(n) = E[e^4(n)], \quad (5.5)$$

$e(n)$ is the error given by:

$$e(n) = d(n) + w(n) - y(n), \quad (5.6)$$

where $d(n)$ is the desired value, $y(n)$ is the output of the adaptive system, and $w(n)$ is the additive noise, Fig. 5.2 depicts this clearly.

It is worth noticing that when the absolute value of the error is less than one, the fourth power and its gradient are less than those of the square. However, the situation reverses when the absolute value of the error is greater than one. One would expect that the LMF algorithm can quickly become unstable and diverge. To overcome this situation, a rescue condition must be used to prohibit the algorithm from doing so.

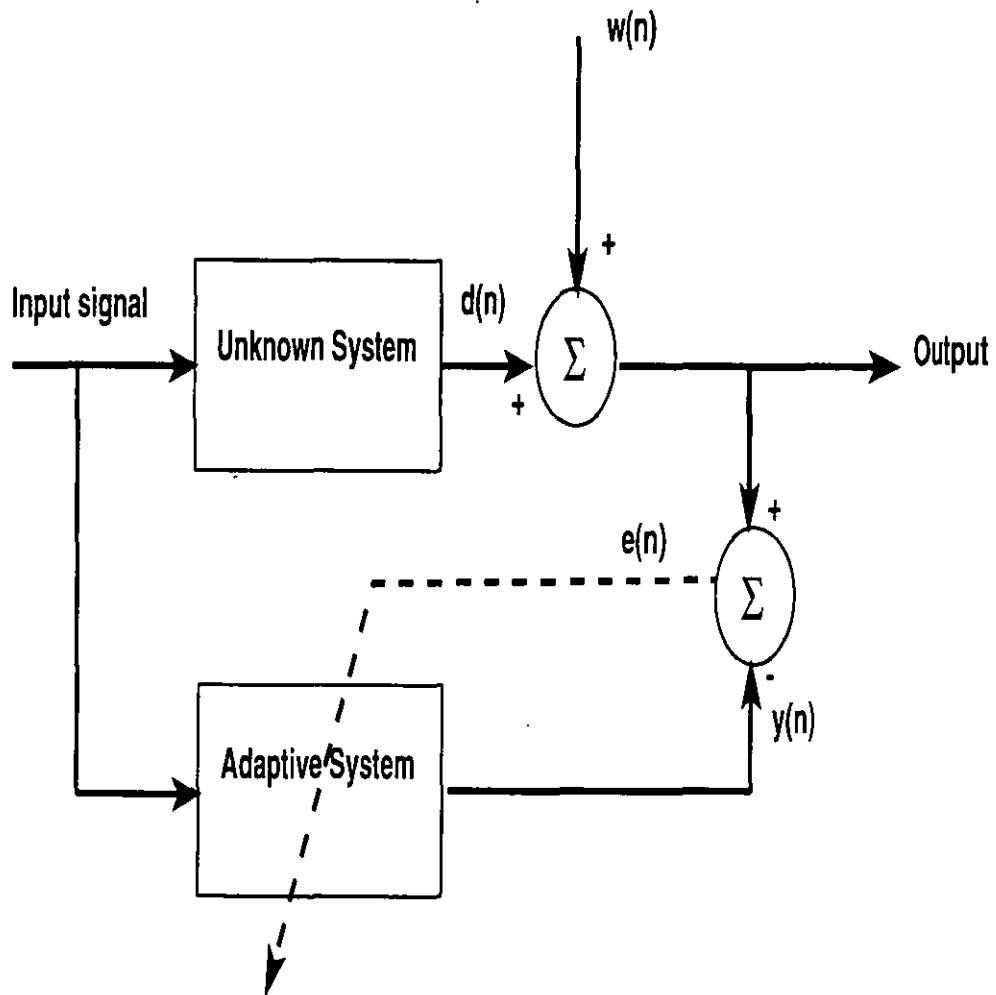


Figure 5.2: Block diagram of adaptive system identification.

5.2.2 Analysis of the error surfaces

The error functions can be more conveniently expressed in terms of the input autocorrelation matrix of the near-end

$$\mathbf{R}_N = E[\mathbf{X}_N(n)\mathbf{X}_N^T(n)], \quad (5.7)$$

the input autocorrelation matrix of the far-end

$$\mathbf{R}_F = E[\mathbf{X}_F(n)\mathbf{X}_F^T(n)], \quad (5.8)$$

the crosscorrelation vector of the near-end

$$\mathbf{P}_N = E[\mathbf{X}_N(n)d(n)], \quad (5.9)$$

and the crosscorrelation vector of the far-end

$$\mathbf{P}_F = E[\mathbf{X}_F(n)d(n)], \quad (5.10)$$

as follows:

$$\begin{aligned} J_N(n) = & E[d^2(n)] + E[w^2(n)] - 2\mathbf{C}_N^T(n)\mathbf{P}_N - 2\mathbf{C}_F^T(n)\mathbf{P}_F \\ & + \mathbf{C}_N^T(n)\mathbf{R}_N\mathbf{C}_N(n) + \mathbf{C}_F^T(n)\mathbf{R}_F\mathbf{C}_F(n), \end{aligned} \quad (5.11)$$

which can be written in the following format:

$$J_N(n) = E[d^2(n)] + E[w^2(n)] - 2\mathbf{C}^T(n)\mathbf{P} + \mathbf{C}^T(n)\mathbf{R}\mathbf{C}(n), \quad (5.12)$$

where

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_N \\ \mathbf{P}_F \end{bmatrix}, \quad (5.13)$$

and

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_F \end{bmatrix}. \quad (5.14)$$

It is clear from (5.12) that the MSE is precisely a quadratic function of the components of the tap coefficients, and the shape associated with it is hyperparaboloid. The adaptive process will be continuously adjusting the tap coefficients, seeking the bottom of this hyperparaboloid.

It can be shown as well that the error-function for the far-end section, taking into considerations the assumptions stated above, can be set into the following

$$\begin{aligned}
 J_F(n) &= E[d^4(n)] + E[w^4(n)] \\
 &\quad - 4\mathbf{C}^T(n)E[\mathbf{X}(n)d^3(n)] \\
 &\quad + 6\{E[d^2(n)] - 2\mathbf{C}^T(n)\mathbf{P} + \mathbf{C}^T(n)\mathbf{R}\mathbf{C}(n)\}E[w^2(n)] \\
 &\quad + 6\mathbf{C}^T(n)E[\mathbf{X}(n)\mathbf{X}^T(n)d^2(n)]\mathbf{C}(n) \\
 &\quad - 4\mathbf{C}^T(n)E[\mathbf{X}(n)\mathbf{C}^T(n)\mathbf{X}(n)\mathbf{X}^T(n)d(n)]\mathbf{C}(n) \\
 &\quad + \mathbf{C}^T(n)E[\mathbf{X}(n)\mathbf{X}^T\mathbf{C}(n)\mathbf{C}^T(n)\mathbf{X}(n)\mathbf{X}^T(n)]\mathbf{C}(n). \quad (5.15)
 \end{aligned}$$

The above expression shows that indeed the error-to-the-power-four will have a global minimum since the latter one is a convex function.

As in the near-end section, the adaptive process will continuously seek the bottom of the error-function of the far-end section.

5.2.3 The optimal solution

The point at the bottom of the performance surface corresponds to the optimal tap coefficients, \mathbf{C}_{opt} , or minimum MSE. The gradient method is used to cause the tap coefficients vector to seek the minimum of the performance surface. The gradient of the performance criterion for the near-end section is defined as

$$\begin{aligned}
 \nabla J_N(n) &= \frac{\partial J_N(n)}{\partial \mathbf{C}_N(n)} \\
 &= -2\mathbf{P}_N + 2\mathbf{R}_N\mathbf{C}_N(n). \quad (5.16)
 \end{aligned}$$

To obtain the minimum MSE, the tap-coefficients vector $\mathbf{C}(n)$ is set to its optimal value, \mathbf{C}_{opt} , where the gradient is set to zero, that is, for the near-end

section it is found to be:

$$\mathbf{C}_{Nopt} = [\mathbf{R}_N]^{-1} \mathbf{P}_N, \quad (5.17)$$

and for the far-end section it is obtained by setting the following:

$$\nabla J_F(n) = \frac{\partial J_F(n)}{\partial \mathbf{C}_F(n)} \quad (5.18)$$

to zero and solving for \mathbf{C}_{Fopt} . Since $J_F(n)$ is function of \mathbf{C}_{Fopt} in the fourth power, this will result in a gradient in a power of three of \mathbf{C}_{Fopt} , hence will make it cumbersome to solve (5.18) and find an explicit value for \mathbf{C}_{Fopt} . It will be taken as stated above that since $J_F(n)$ is a convex function will have then a global minimum, thus the updating scheme will continuously search for this value until (5.18) is approximately zero.

The minimum achievable MSE associated with these optimum values is found for the near-end section to be:

$$\begin{aligned} J_{Nmin} = & E[d^2(n)] + E[w^2(n)] - \mathbf{C}_{Nopt}^T \mathbf{P}_N \\ & - 2\mathbf{C}_F^T(n) \mathbf{P}_F + \mathbf{C}_F^T(n) \mathbf{R}_F \mathbf{C}_F(n). \end{aligned} \quad (5.19)$$

The far-end section value, J_{Fmin} , can be obtained as well when \mathbf{C}_{Fopt} found from $\nabla J_F(n) = \mathbf{0}$ is substituted for $\mathbf{C}_F(n)$ in (5.15).

5.2.4 The updating scheme

Based on this motivation, the LMMN algorithm for recursively adjusting the tap coefficients of the NE canceller, $\mathbf{C}_N(n)$, and those of the FE canceller, $\mathbf{C}_F(n)$, is expressed in the following form

$$\mathbf{C}_N(n+1) = \mathbf{C}_N(n) + \mu_1 e(n) \mathbf{X}_N(n), \quad (5.20)$$

$$\mathbf{C}_F(n+1) = \mathbf{C}_F(n) + 2\mu_2 e^3(n) \mathbf{X}_F(n). \quad (5.21)$$

These expressions are obtained by differentiating the instantaneous gradient vectors $\hat{\nabla} J_N(n)$ and $\hat{\nabla} J_F(n)$ with respect to $\mathbf{C}_N(n)$ and $\mathbf{C}_F(n)$, respectively, and

replacing them in the recursive relation which uses the steepest descent method [7]. The convergence properties of this algorithm are controlled by the step sizes μ_1 and μ_2 . The step sizes are chosen small enough to ensure convergence of the iterative procedure and produce less misadjustment error.

The choice of μ_1 and μ_2 , and the convergence behaviour of the new adaptive algorithm are investigated in the next section.

Finally, Table 5.1 illustrates the mathematical description of the LMMN algorithm.

Table 5.1: The mathematical description of the LMMN algorithm.

Adaptive filter	$y(n) = \mathbf{C}_N^T(n)\mathbf{X}_N(n) + \mathbf{C}_F^T(n)\mathbf{X}_F(n)$
Error equation	$e(n) = d(n) + w(n) - y(n)$
Adaptive Control Equations	$\mathbf{C}_N(n+1) = \mathbf{C}_N(n) + \mu_1 e(n)\mathbf{X}_N(n)$ $\mathbf{C}_F(n+1) = \mathbf{C}_F(n) + 2\mu_2 e^3(n)\mathbf{X}_F(n)$
Initialization	$\mathbf{C}_N(0) = \mathbf{0}$ $\mathbf{C}_F(0) = \mathbf{0}$

5.3 Convergence behaviour of the LMMN algorithm

This section deals with the convergence of the above stated algorithm. The convergence in the mean and the convergence in the mean square are analyzed and bounds on the step size parameter are obtained for the near-end and the far-end sections. Expressions for the time constants of the algorithm and the misadjustment factor are also obtained. The convergence of the algorithm in the sense of the mean is first studied.

5.3.1 Convergence in the mean

Examining the mean behaviour of (5.20) under the above stated independence assumption, the value of the error defined in (5.6) and the output $y(n)$ in (5.3) are substituted in equation (5.20) to get the following:

$$\begin{aligned} \mathbf{C}_N(n+1) = & [\mathbf{I} - \mu_1 \mathbf{X}_N(n) \mathbf{X}_N^T(n)] \mathbf{C}_N(n) + \mu_1 \mathbf{X}_N(n) d(n) \\ & - \mu_1 \mathbf{X}_N(n) \mathbf{X}_F^T(n) \mathbf{C}_F(n) + \mu_1 \mathbf{X}_N(n) w(n). \end{aligned} \quad (5.22)$$

Now define

$$\hat{\mathbf{C}}_N(n) = \mathbf{C}_N(n) - \mathbf{C}_{N_{opt}} \quad (5.23)$$

to be the weight error vector at the n th iteration.

Subtracting $\mathbf{C}_{N_{opt}}$ from both sides of (5.22) and making use of (5.23), equation (5.22) becomes

$$\begin{aligned} \hat{\mathbf{C}}_N(n+1) = & [\mathbf{I} - \mu_1 \mathbf{X}_N(n) \mathbf{X}_N^T(n)] \hat{\mathbf{C}}_N(n) + \mu_1 \mathbf{X}_N(n) d(n) \\ & - \mu_1 \mathbf{X}_N(n) \mathbf{X}_F^T(n) \mathbf{C}_{N_{opt}} - \mu_1 \mathbf{X}_N(n) \mathbf{X}_F^T(n) \mathbf{C}_F(n) \\ & + \mu_1 \mathbf{X}_N(n) w(n). \end{aligned} \quad (5.24)$$

Taking the expectation of both sides of the above equation and using the fact that $\mathbf{X}_N(n)$ and $\mathbf{X}_F(n)$ are uncorrelated (even though the presence of correlation in

$\mathbf{X}(n)$ considerably complicates the analysis, it is known that for other algorithms the influence of correlation in the input on the convergence is not very large [44]), using the assumption that the input signal and the noise are independent (both of zero mean), and taking advantage of the independence assumption [7], it can be shown that equation (5.24) becomes

$$E[\hat{\mathbf{C}}_N(n+1)] = [\mathbf{I} - \mu_1 \mathbf{R}_N]E[\hat{\mathbf{C}}_N(n)] + \mu_1 \mathbf{P}_N - \mu_1 \mathbf{R}_N \mathbf{C}_{Nopt}. \quad (5.25)$$

When the value of \mathbf{C}_{Nopt} defined in (5.17) is substituted in (5.25), the following result is obtained

$$E[\hat{\mathbf{C}}_N(n+1)] = [\mathbf{I} - \mu_1 \mathbf{R}_N]E[\hat{\mathbf{C}}_N(n)]. \quad (5.26)$$

Equation (5.26) reveals that the algorithm will converge to the optimal value if all the eigenvalues of the matrix $(\mathbf{I} - \mu_1 \mathbf{R}_N)$ are less than unity.

It therefore follows that, at steady state, the following will be obtained, provided the right step size has been chosen,

$$\hat{\mathbf{C}}_N(n) \rightarrow \mathbf{0}, \quad (5.27)$$

or equivalently

$$\mathbf{C}_N(n) = \mathbf{C}_{Nopt}, \quad (5.28)$$

where this happens if the step size is chosen to be:

$$0 < \mu_1 < \frac{2}{\lambda_{Nmax}}, \quad (5.29)$$

where λ_{Nmax} is the largest eigenvalue of \mathbf{R}_N .

Similarly, as was done for the near-end section, equation (5.21) for the far-end section will look like

$$\begin{aligned}
\hat{\mathbf{C}}_F(n+1) = & \hat{\mathbf{C}}_F(n) + 2\mu_2 \mathbf{X}_F(n) [w^3(n) - 3w^2(n) \mathbf{X}_F^T(n) \hat{\mathbf{C}}_F(n) \\
& - 3w^2(n) \mathbf{X}_N^T(n) \hat{\mathbf{C}}_N(n) + 3w(n) \mathbf{X}_F^T(n) \hat{\mathbf{C}}_F(n) \hat{\mathbf{C}}_F^T(n) \mathbf{X}_F(n) \\
& + 3w(n) \mathbf{X}_N^T(n) \hat{\mathbf{C}}_N(n) \hat{\mathbf{C}}_N^T(n) \mathbf{X}_N(n) \\
& + 6w(n) \mathbf{X}_F^T(n) \hat{\mathbf{C}}_F(n) \hat{\mathbf{C}}_N^T(n) \mathbf{X}_N(n) \\
& - 3\mathbf{X}_N^T(n) \hat{\mathbf{C}}_N(n) \mathbf{X}_F^T(n) \hat{\mathbf{C}}_F(n) \hat{\mathbf{C}}_F^T(n) \mathbf{X}_F(n) \\
& - 3\mathbf{X}_F^T(n) \hat{\mathbf{C}}_F(n) \mathbf{X}_N^T(n) \hat{\mathbf{C}}_N(n) \hat{\mathbf{C}}_N^T(n) \mathbf{X}_N(n) \\
& - \mathbf{X}_F^T(n) \hat{\mathbf{C}}_F(n) \mathbf{X}_F^T(n) \hat{\mathbf{C}}_F(n) \hat{\mathbf{C}}_F^T(n) \mathbf{X}_F(n) \\
& - \mathbf{X}_N^T(n) \hat{\mathbf{C}}_N(n) \mathbf{X}_N^T(n) \hat{\mathbf{C}}_N(n) \hat{\mathbf{C}}_N^T(n) \mathbf{X}_N(n)], \tag{5.30}
\end{aligned}$$

where $\hat{\mathbf{C}}_F(n)$ is the weight error vector of the far-end section, defined to be:

$$\hat{\mathbf{C}}_F(n) = \mathbf{C}_F(n) - \mathbf{C}_{Fopt}. \tag{5.31}$$

The analysis of equation (5.30) will be limited to the relatively simple case of small deviations from the Wiener solution when the weight error vectors $\hat{\mathbf{C}}_N(n)$ and $\hat{\mathbf{C}}_F(n)$ are close to zero [8]. Hence the impact of terms which include high powers of $\hat{\mathbf{C}}_N(n)$ and $\hat{\mathbf{C}}_F(n)$ can be neglected in this equation. Thus, when this assumption is considered, equation (5.30) appears as follows:

$$\begin{aligned}
\hat{\mathbf{C}}_F(n+1) = & [\mathbf{I} - 6\mu_2 w^2(n) \mathbf{X}_F(n) \mathbf{X}_F^T(n)] \hat{\mathbf{C}}_F(n) \\
& - 6\mu_2 w^2(n) \mathbf{X}_F(n) \mathbf{X}_N^T(n) \hat{\mathbf{C}}_N(n) + 2\mu_2 w^3(n) \mathbf{X}_F(n) \\
& + 12\mu_2 w(n) \mathbf{X}_F(n) \mathbf{X}_F^T(n) \hat{\mathbf{C}}_F(n) \hat{\mathbf{C}}_N^T(n) \mathbf{X}_N(n). \tag{5.32}
\end{aligned}$$

Taking the expectation value of the above equation, under the assumption that the measurement noise is independent with the input signal, both of zero mean, and also the error weight vectors and the input signal are independent of each other, results into the following:

$$E[\hat{\mathbf{C}}_F(n+1)] = [\mathbf{I} - 6\mu_2 E[w^2(n)] \mathbf{R}_F] E[\hat{\mathbf{C}}_F(n)], \tag{5.33}$$

where $E[w^2(n)]$ is the measurement noise power. In addition to the assumption used to derive (5.33), it was also assumed that for small values of μ_2 , $w(n)$ is independent of $\hat{\mathbf{C}}_F(n)$. Other works [93]-[94] when minimising the mean square error none of the above assumptions are used.

It can be shown, using the same procedure of [8], that the adaptation process will cause the convergence $E[\hat{\mathbf{C}}_F(n)] \rightarrow \mathbf{0}$, that is, the algorithm defined by (5.21) will provide an unbiased estimate of the optimal solution.

Equation (5.33) states that the algorithm will converge to the optimal value if all the eigenvalues of $[\mathbf{I} - 6\mu_2 E[w^2(n)]\mathbf{R}_F]$ are less than unity. This happens if the step size μ_2 is chosen to be in the following range:

$$0 < \mu_2 < \frac{2}{6\lambda_{Fmax} E[w^2(n)]}, \quad (5.34)$$

where λ_{Fmax} is the largest eigenvalues of \mathbf{R}_F .

Convergence of the mean is contingent, of course, on compliance with conditions (5.29) and (5.34). In practice these conditions might be difficult to check. However, we can bound the largest eigenvalue of the input positive definite autocorrelation matrix by its trace (tr), that is,

$$\lambda_{Nmax} < tr(\mathbf{R}_N) = N_1\sigma_x^2, \quad (5.35)$$

and

$$\lambda_{Fmax} < tr(\mathbf{R}_F) = N_2\sigma_x^2, \quad (5.36)$$

and therefore sufficient conditions for convergence in the mean of the new adaptive algorithm are:

$$0 < \mu_1 < \frac{2}{N_1\sigma_x^2}, \quad (5.37)$$

and

$$0 < \mu_2 < \frac{2}{6N_2\sigma_x^2 E[w^2(n)]}, \quad (5.38)$$

where N_1 and N_2 are the lengths of the NE the FE cancellers, respectively.

As can be seen from equations (5.37) and (5.38) that convergence in the mean of the LMMN algorithm takes place if both of these sufficient conditions are satisfied.

If, however, these conditions are not fulfilled, the algorithm, then, will not converge to a steady state value, probably it will diverge.

It is interesting to note the dependence of μ_2 on the variance of the noise, that is, $E[w^2(n)]$. If the value of $E[w^2(n)]$ is the actual one, the algorithm will converge to the optimal value; however, if $E[w^2(n)]$ is smaller than the actual value the algorithm will not converge and instability might happen. Note that if $E[w^2(n)]$ is larger than the actual one, the algorithm will converge to a steady state value, since this value keeps μ_2 in the range given by (5.38). The effect of $E[w^2(n)]$ on the convergence of the algorithm will be treated in the simulations.

5.3.2 Convergence in the mean square

The general conditions stated in the above derivations are cumbersome to apply, so we wish to present a simplified version of the above conditions on the assumption that, in addition to statistical independence of $\{\mathbf{X}(n)\}$, the elements x_i are independent and identically distributed (i.i.d). The simplification due to i.i.d assumption further enables us to establish optimum rate of convergence and optimum adjustment gain.

To examine the convergence behaviour of the mean square of the weight error vector and thus determine the misadjustment of the algorithm, we define the following measure

$$\gamma_n = E[\|\hat{\mathbf{C}}_N(n)\|^2], \quad (5.39)$$

where $\|\cdot\|$ denotes the Euclidean norm, then

$$E[\|\hat{\mathbf{C}}_N(n+1)\|^2] = E[\hat{\mathbf{C}}_N^T(n+1)\hat{\mathbf{C}}_N(n+1)]. \quad (5.40)$$

Using this result for the near-end section of the canceller, it follows from (5.20) that

$$\begin{aligned} \gamma_{n+1} &= \gamma_n + 2\mu_1 E[e(n)\mathbf{X}_N^T(n)\hat{\mathbf{C}}_N(n)] \\ &\quad + \mu_1^2 E[e^2(n)\mathbf{X}_N^T(n)\mathbf{X}_N(n)]. \end{aligned} \quad (5.41)$$

Under the often-made assumption [78], [79] that the input signal sequence $\{x_i\}$ is an independent and identically distributed Gaussian random sequence with zero mean and variance σ_x^2 , and independent of the noise, the derivation of (5.41) is obtained as follows.

To solve equation (5.41), let's see the second term of the right hand side of this equation. Substituting for the error and using the above assumptions, the following is obtained:

$$\begin{aligned}
E[e(n)\mathbf{X}_N^T(n)\hat{\mathbf{C}}_N(n)] &= E[e(n)\mathbf{X}_N^T(n)\mathbf{C}_N(n)] - E[e(n)\mathbf{X}_N^T(n)\mathbf{C}_{Nopt}] \\
&= E[(d(n) + w(n) - \mathbf{C}_N^T(n)\mathbf{X}_N(n) - \mathbf{C}_F^T(n)\mathbf{X}_F(n)) \\
&\quad * \mathbf{X}_N^T(n)\mathbf{C}_N(n)] - E[(d(n) + w(n) - \mathbf{C}_N^T(n)\mathbf{X}_N(n) \\
&\quad - \mathbf{C}_F^T(n)\mathbf{X}_F(n))\mathbf{X}_N^T(n)\mathbf{C}_{Nopt}] \\
&= 2\mathbf{C}_N^T(n)\mathbf{P}_N - \mathbf{C}_N^T(n)\mathbf{R}_N\mathbf{C}_N(n) - \mathbf{C}_{Nopt}^T\mathbf{P}_N \\
&= J_{Nmin} - J_N(n)
\end{aligned} \tag{5.42}$$

The third term of the right-hand side of equation (5.41) is readily evaluated. Given that

$$\mathbf{X}_N^T(n)\mathbf{X}_N(n) = \|\mathbf{X}_N(n)\|^2, \tag{5.43}$$

and (5.4) we can then write the following

$$\begin{aligned}
E[e^2(n)\mathbf{X}_N^T(n)\mathbf{X}_N(n)] &= E[\|e(n)\|^2\|\mathbf{X}_N(n)\|^2] \\
&= N_1\sigma_x^2J_N(n).
\end{aligned} \tag{5.44}$$

Inserting (5.42) and (5.44) in (5.41) yields :

$$\begin{aligned}
\gamma_{n+1} &= \gamma_n + 2\mu_1\{J_{Nmin} - J_N(n)\} \\
&\quad + \mu_1^2N_1\sigma_x^2J_N(n).
\end{aligned} \tag{5.45}$$

In the limit when γ_{n+1} reaches γ_n , the value of the step-size is found to be

$$\mu_1 = 2\frac{J_N(n) - J_{Nmin}}{N_1\sigma_x^2J_N(n)}. \tag{5.46}$$

Since $e(n)$ can be expressed as:

$$e(n) = e_{opt} - \hat{\mathbf{C}}_N^T(n)\mathbf{X}_N(n) - \hat{\mathbf{C}}_F^T(n)\mathbf{X}_F(n) \quad (5.47)$$

and by making use of the assumption that e_{opt} , $\mathbf{X}_N(n)$, and $\mathbf{X}_F(n)$ are mutually independent [14], [95], it can be shown that

$$J_N(n) - J_{Nmin} = \sigma_x^2 \gamma_n + \sigma_x^2 \theta_n. \quad (5.48)$$

This leads to equation (5.45), after some manipulations, to becoming:

$$\begin{aligned} \gamma_{n+1} = & [1 - 2\mu_1\sigma_x^2 + \mu_1^2 N_1 \sigma_x^4] \gamma_n + [-2\mu_1\sigma_x^2 + \mu_1^2 N_1 \sigma_x^4] \theta_n \\ & + \mu_1^2 N_1 \sigma_x^2 J_{Nmin}, \end{aligned} \quad (5.49)$$

where θ_n is defined as

$$\theta_n = E[\|\hat{\mathbf{C}}_F(n)\|^2]. \quad (5.50)$$

It is observed from equation (5.49) that the FE section is coupled to the NE section, this is shown in the term containing θ_n of the right hand side of this equation. This does not stop the algorithm to converge to the optimal solution if the right step size is used so that the term pertaining to θ_n will vanish completely. This will happen of course as will be seen next when the derivations show values of μ_1 that must be chosen for the convergence to take place.

Hence, the above expression, i.e., (5.49), to converge, it is required that

$$|1 - 2\mu_1\sigma_x^2 + \mu_1^2 N_1 \sigma_x^4| < 1, \quad (5.51)$$

the term containing θ_n in the right hand side of (5.49) must vanish as well, that is,

$$-2\mu_1\sigma_x^2 + \mu_1^2 N_1 \sigma_x^4 = 0, \quad (5.52)$$

and assuming that μ_1 is chosen small [1], [14], [95] such that the last term in (5.49) vanishes.

The above condition imposed on the factor of θ_n will be satisfied if μ_1 is chosen to satisfy (5.52). Accordingly, for convergence to take place in the mean-square sense, the step size μ_1 has to satisfy :

$$0 < \mu_1 \leq \frac{2}{N_1 \sigma_x^2}, \quad (5.53)$$

a safe choice for μ_1 is therefore any value in the interval

$$0 < \mu_1 < \frac{2}{N_1 \sigma_x^2}. \quad (5.54)$$

As can be seen from (5.54), the factor of θ_n in (5.49) will be almost always zero if μ_1 is chosen in this range.

Proposition 1 *If $E[x^2]$ is chosen much larger than $\frac{4}{N_1} J_{Nmin}$, then the last term of (5.49) will always vanish.*

Proof: Let

$$\mu_1 = \frac{2}{N_1 E[x^2]}, \quad (5.55)$$

the last term in (5.49), i.e.,

$$\begin{aligned} \mu_1^2 N_1 E[x^2] J_{Nmin} &= \frac{4 J_{Nmin}}{N_1 E[x^2]} \\ &= 0, \end{aligned} \quad (5.56)$$

since $E[x^2]$ is much larger than $\frac{4}{N_1} J_{Nmin}$. ■

In practice, this is always a valid assumption, since the energy of the signal is much larger than the optimal value of the error.

Finally, the step size that provides the fastest speed of convergence to the algorithm is obtained by setting the derivative of the expression in (5.51) to zero, that is,

$$\mu_{1opt} = \frac{1}{N_1 \sigma_x^2}, \quad (5.57)$$

or one-half the maximum step size.

Before ending up this treatment of the NE section, it is worth noticing that the bound (5.54) imposes a much narrower upper bound than (5.29) on μ_1 , since these are found for the convergence in the mean square and in the mean, respectively.

Now, let us come to the treatment of the far-end section after we have seen the convergence in the mean square of the near-end section. The derivation is based on the fundamental assumption that $\hat{\mathbf{C}}_N(n)$, $\hat{\mathbf{C}}_F(n)$, $\mathbf{X}_N(n)$, $\mathbf{X}_F(n)$, and $w(n)$ are independent random variables. It follows, then, from (5.21) that

$$\begin{aligned} \hat{\mathbf{C}}_F^T(n+1)\hat{\mathbf{C}}_F(n+1) &= \hat{\mathbf{C}}_F^T(n)\hat{\mathbf{C}}_F(n) + 4\mu_2 e^3(n)\mathbf{X}_F^T(n)\hat{\mathbf{C}}_F(n) \\ &\quad + 4\mu_2^2 e^6(n)\mathbf{X}_F^T(n)\mathbf{X}_F(n). \end{aligned} \quad (5.58)$$

The above equation will look like the following when the value of the error is inserted in it:

$$\begin{aligned} \hat{\mathbf{C}}_F^T(n+1)\hat{\mathbf{C}}_F(n+1) &= \hat{\mathbf{C}}_F^T(n)\hat{\mathbf{C}}_F(n) + 4\mu_2 [w(n) - \hat{\mathbf{C}}_N^T(n)\mathbf{X}_N(n) \\ &\quad - \hat{\mathbf{C}}_F^T(n)\mathbf{X}_F(n)]^3 \mathbf{X}_F^T(n)\hat{\mathbf{C}}_F(n) + 4\mu_2^2 [w(n) - \hat{\mathbf{C}}_N^T(n) \\ &\quad * \mathbf{X}_N(n) - \hat{\mathbf{C}}_F^T(n)\mathbf{X}_F(n)]^6 \mathbf{X}_F^T(n)\mathbf{X}_F(n). \end{aligned} \quad (5.59)$$

The expectations of both sides of the above equation are taken under the assumption that $\hat{\mathbf{C}}_F(n)$ is given, then obtain conditional expectation of both sides of (5.59). Since $w(n)$ was assumed to be independent of $\hat{\mathbf{C}}_N(n)$, $\hat{\mathbf{C}}_F(n)$, $\mathbf{X}_N(n)$, $\mathbf{X}_F(n)$, and to have zero odd moments, all the terms on the right-hand side of (5.59) which include odd power of $w(n)$ will vanish under expectation.

The conditional expectation of (5.59) obtained in the vicinity of the optimal solution, i.e., all high powers of $\hat{\mathbf{C}}_F(n)$ can be neglected, to find

$$\begin{aligned} E[\hat{\mathbf{C}}_F^T(n+1)\hat{\mathbf{C}}_F(n+1)/\hat{\mathbf{C}}_F(n)] &= \hat{\mathbf{C}}_F^T(n)\hat{\mathbf{C}}_F(n) - 12\mu_2 E [w^2(n)\hat{\mathbf{C}}_F^T(n)\mathbf{X}_F(n) \\ &\quad * \mathbf{X}_F^T(n)\hat{\mathbf{C}}_F(n)/\hat{\mathbf{C}}_F(n)] + 4\mu_2^2 E [\{w^6(n) \\ &\quad + 15w^4(n)\hat{\mathbf{C}}_F^T(n)\mathbf{X}_F(n)\mathbf{X}_F^T(n)\hat{\mathbf{C}}_F(n)\} \\ &\quad * \mathbf{X}_F^T(n)\mathbf{X}_F(n)/\hat{\mathbf{C}}_F(n)]. \end{aligned} \quad (5.60)$$

To be able to evaluate the above expression, one additional approximation is used, which is:

$$\mathbf{X}_F^T(n)\mathbf{X}_F(n) = N_2\sigma_x^2. \quad (5.61)$$

This is a legitimate approximation since in general N_2 is a large number. Hence, equation (5.60), after using this assumption, appears in the following form:

$$\begin{aligned} E[\hat{\mathbf{C}}_F^T(n+1)\hat{\mathbf{C}}_F(n+1)/\hat{\mathbf{C}}_F(n)] &= \hat{\mathbf{C}}_F^T(n)\mathbf{A}\hat{\mathbf{C}}_F(n) \\ &+ 4\mu_2^2 E[w^6(n)]E[\mathbf{X}_F^T(n)\mathbf{X}_F(n)], \end{aligned} \quad (5.62)$$

where \mathbf{A} is defined by:

$$\mathbf{A} = \mathbf{I} - \{12\mu_2 E[w^2(n)] + 60\mu_2^2 N_2 \sigma_x^2 E[w^4(n)]\} E[\mathbf{X}_F(n)\mathbf{X}_F^T(n)]. \quad (5.63)$$

When the autocorrelation matrix \mathbf{R}_F is substituted for $E[\mathbf{X}_F(n)\mathbf{X}_F^T(n)]$ in the above equation, the following is obtained:

$$\mathbf{A} = \mathbf{I} - 12\mu_2 \{E[w^2(n)] - 5\mu_2 N_2 \sigma_x^2 E[w^4(n)]\} \mathbf{R}_F. \quad (5.64)$$

As can be seen from (5.62), the algorithm converges in the mean square sense if and only if the eigenvalues of the matrix \mathbf{A} are less than one in magnitude.

The kurtosis for the zero mean variable noise, $w(n)$, is defined as:

$$\kappa_w \triangleq \frac{E[w^4(n)]}{E^2[w^2(n)]}, \quad (5.65)$$

and will be used in the following derivation.

Proposition 2 *The algorithm defined by (5.21) will converge in the mean-square sense if μ_2 is chosen to be in the following range:*

$$0 < \mu_2 < \frac{E[w^2(n)]}{5N_2\sigma_x^2 E[w^4(n)]} \quad (5.66)$$

and $\kappa_w > \frac{3}{10}$, where κ_w is the kurtosis of the noise.

Proof: First, since the autocorrelation matrix \mathbf{R}_F is assumed to be positive definite, all the eigenvalues of the matrix \mathbf{A} will have absolute values smaller than one, that is

$$1 - 12\mu_2 \{E[w^2(n)] - 5\mu_2 N_2 \sigma_x^2 E[w^4(n)]\} \lambda_{Fi} < 1, \forall i = 1, 2, \dots, N_2, \quad (5.67)$$

λ_{Fi} is the i th eigenvalue belonging to \mathbf{R}_F . Inequality (5.67) is satisfied if and only if μ_2 satisfies (5.66).

Second, the following inequality

$$1 - 12\mu_2 \{E[w^2(n)] - 5\mu_2 N_2 \sigma_x^2 E[w^4(n)]\} \lambda_{Fmax} > -1, \quad (5.68)$$

where λ_{Fmax} is the largest eigenvalue in the autocorrelation matrix \mathbf{R}_F , is valid if the discriminant of the following inequality is less than zero,

$$2 - 12\mu_2 \{E[w^2(n)] - 5\mu_2 N_2 \sigma_x^2 E[w^4(n)]\} \lambda_{Fmax} > 0, \quad (5.69)$$

so that any value of μ_2 will verify (5.69), and this is true only if the kurtosis of the noise is greater than $\frac{3}{10}$, which is always true for any pdf, since its minimal value, $\kappa_w = 1$, is obtained for the binary symmetric distribution and its maximal value, $\kappa_w = \infty$, is obtained for the Cauchy distribution [95].

In this last derivation the approximation $\lambda_{Fmax} = N_2 \sigma_x^2$ was used. ■

Stability condition (5.66) turned out to be quite robust and to provide a better approximation of the stable range of the step size μ_2 than (5.34).

Ultimately, the optimal step size that gives the fastest convergence in the mean-square sense for the algorithm is found to be:

$$\mu_{2opt} = \frac{E[w^2(n)]}{10N_2\sigma_x^2 E[w^4(n)]}. \quad (5.70)$$

This is found by setting the derivative of the expression on the left hand side of the inequality (5.67) to zero.

Table 5.2: The main parameters of the LMMN algorithm.

	Near-End Section	Far-End Section
Cost function	$J_N(n) = E[e^2(n)]$	$J_F(n) = E[e^4(n)]$
μ_{max}	$\frac{2}{\lambda_{N_{max}}(\mathbf{R}_N)}$	$\frac{2}{6E[w^2(n)]\lambda_{F_{max}}(\mathbf{R}_F)}$
μ_{max}	$\frac{2}{N_1\sigma_x^2}$	$\frac{E[w^2(n)]}{5N_2\sigma_x^2 E[w^4(n)]}$
μ_{opt}	$\frac{1}{N_1\sigma_x^2}$	$\frac{E[w^2(n)]}{10N_2\sigma_x^2 E[w^4(n)]}$
τ_i	$\frac{1}{\mu_1 \lambda_{N_i}(\mathbf{R}_N)}$	$\frac{1}{6\mu_2 E[w^2(n)]\lambda_{F_i}(\mathbf{R}_F)}$

The first term of the right hand side equation of the above equation is given by:

$$N_1 E[\hat{C}_{N_i}^2(n)] E[x^2(n)] = \mu_1 N_1 J_{N_{min}} E[x^2(n)], \quad (5.76)$$

since the algorithm minimises $E[e^2(n)]$ for the near end section of the echo canceller; whereas in the far end section of the canceller, the algorithm is minimising $E[e^4(n)]$, then the second term of (5.75) is evaluated using the result in [8]

$$N_2 E[\hat{C}_{F_i}^2(n)] E[x^2(n)] = \frac{2\mu_2 N_2 E[w^6(n)] E[x^2(n)]}{3E[w^2(n)]}. \quad (5.77)$$

Substituting (5.76) and (5.77) into (5.75), the misadjustment factor is found to be:

$$M = \frac{\mu_1 N_1 J_{N_{min}} E[x^2(n)]}{E[w^2(n)]} + \frac{2\mu_2 N_2 E[w^6(n)] E[x^2(n)]}{3\{E[w^2(n)]\}^2}. \quad (5.78)$$

Expressing the misadjustment factor using (5.71) and (5.72) yields:

$$M = \frac{J_{N_{min}}}{E[w^2(n)]} \sum_{i=1}^{N_1} \frac{1}{\tau_{N_i}} + \frac{E[w^6(n)]}{9\{E[w^2(n)]\}^3} \sum_{i=1}^{N_2} \frac{1}{\tau_{F_i}}. \quad (5.79)$$

It is clear from the above expression that the misadjustment factor is a function of the moments of the noise, and hence it is expected that the performance of the algorithm depends on these parameters.

5.4 Summary

A new adaptive scheme for echo cancellation has been introduced. It consists of minimising both the mean square and the mean fourth criteria over the near-end and the far-end sections of the echo canceller, respectively. The derivations consist of obtaining bounds on the step sizes for the recursion relations used to update the tap coefficients in the near-end and the far-end sections. Other parameters of useful importance are also obtained. It is also shown that the step size for the far-end section happened to be sensitive to the power of the noise. These factors are going to be investigated when the new updating scheme is going to be evaluated to other algorithms in the next chapter.

Chapter 6

Performance of the Least Mean Mixed-Norm Adaptive Algorithm

6.1 Introduction

This chapter addresses the issue of comparing the performance of the algorithm developed in the last chapter with that of the three other algorithms, denoted by algorithm I, II, and III. Algorithm I is the well-known LMS algorithm based on the minimisation of the MSE, i.e., $J_1(n) = E[e^2(n)]$. Algorithm II is based on the minimisation of the mean fourth error (MFE), i.e., $J_2(n) = E[e^4(n)]$. As for algorithm III, it is based on the minimisation of two functions, a MFE for the near-end section of the canceller, i.e., $J_{N3}(n) = E[e^4(n)]$, and a MSE for its far-end section, i.e., $J_{F3}(n) = E[e^2(n)]$. The latter algorithm will be called the least mean fourth-square (LMFS) algorithm. In a wide variety of applications, the LMS algorithm has become a benchmark against which all algorithms are compared. This is due to both its simplicity and relative performance.

Note that algorithm III is another proposition we are suggesting. Thus, the comparison is in fact against two existing algorithms, i.e., the LMS and the LMF algorithms and two proposed ones, which are the LMMN and the LMFS algo-

rithms.

In our comparison of our own algorithm with the three algorithms, we will concentrate on important parameters, such as the convergence performance, the speed of convergence, and the effect of the variance and distribution of the noise on their behaviour.

It is well known, as will be shown later, that the performance of the LMS algorithm is insensitive to the variance of the noise, but not to its distribution [8]. Also, the LMS algorithm only outperforms other related algorithms when the distribution of the noise is Gaussian [8].

As mentioned in the previous chapter, the performance of the LMMN algorithm depends on the variance of the noise. To overcome this difficulty, it will be shown that in general the algorithm converges to the optimum solution whenever this value is equal or greater than the actual value. Values closer to the actual value of the noise variance give satisfactory results, in the sense that the algorithm converges to a steady state value closer to that obtained by the actual value of the variance of the noise. The simulation results performed on different channels confirm these facts, that is, the dependence of the algorithm upon the variance of the noise.

Also the distribution of the noise is another factor that can affect the convergence of the algorithm. In general, it is found that regardless of whether the distribution of the noise is uniform or Gaussian, the difference between their corresponding steady state values is only minor.

Another point worth mentioning is the stability of algorithms containing terms of the form $E[e^4(n)]$. It is clear that if the absolute value of the error is less than one, the fourth power of the error is less than that of its square, however, the situation reverses if the absolute value of the error is greater than one. Unless a rescue condition is incorporated in this type of algorithm, it can quickly become unstable. In general a very small step size is employed to avoid instability.

In the next three sections, the algorithms used for comparison with the LMMN

one are presented and no derivations are included. Section 6.5 analyses in detail the computational complexity of the four algorithms, while Section 6.6 contains the results of their simulation.

6.2 Convergence behaviour of algorithm I

The following notation is used throughout the analysis of algorithm I, II and III.

The error $e(n)$ between the desired value, $d(n)$, and the output of the canceller is given by:

$$e(n) = d(n) - y(n) + w(n), \quad (6.1)$$

where $y(n)$ is the output of the unknown system and $w(n)$ is the additive noise, as depicted in Fig. 6.1.

The autocorrelation matrix of the input signal is defined in the following manner:

$$\mathbf{R} = E[\mathbf{X}(n)\mathbf{X}^T(n)], \quad (6.2)$$

which can be put in the following form

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_N & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_F \end{bmatrix}, \quad (6.3)$$

where \mathbf{R}_N and \mathbf{R}_F are the autocorrelation matrices of the near-end and the far-end sections of the echo canceller, respectively. Both of \mathbf{R}_N and \mathbf{R}_F are assumed to be positive definite matrices [7].

For the subsequent analysis, it will be assumed that the input $\mathbf{X}(n)$ is independent of the noise $w(n)$, and that both are symmetrically distributed around zero. This implies that all odd moments of $\mathbf{X}(n)$ and $w(n)$ are zero [96].

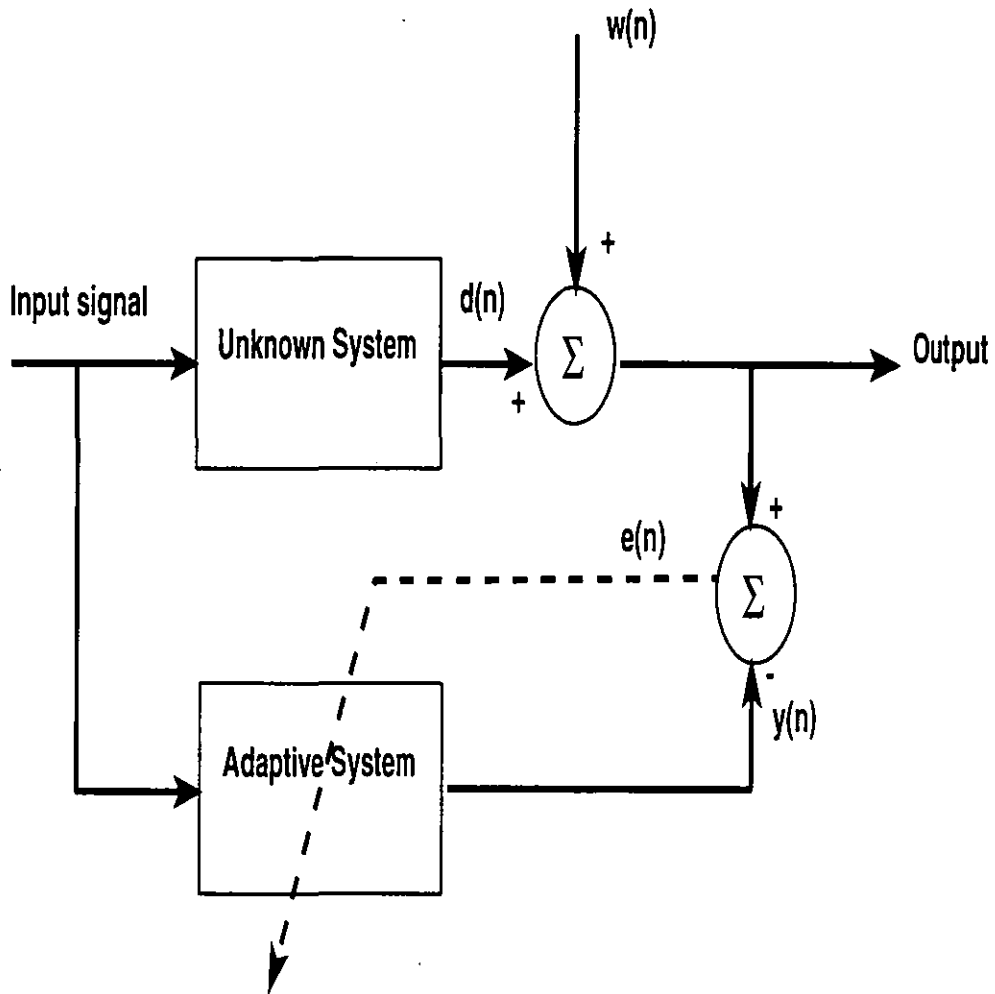


Figure 6.1: System considered for the adaptive system identification.

6.2.1 The performance function

When minimising the mean square error (MSE) at the output of the echo canceller, the corresponding cost function is used:

$$J_1(n) = E[e^2(n)]. \quad (6.4)$$

6.2.2 The updating scheme

The equation for recursively adjusting the tap coefficients, $\mathbf{C}(n)$, of the canceller known as the LMS algorithm [1] is given by:

$$\mathbf{C}(n+1) = \mathbf{C}(n) + \mu e(n)\mathbf{X}(n). \quad (6.5)$$

In all three algorithms, the derivation of the updating scheme is obtained by differentiating the instantaneous gradient vector with respect to their corresponding tap coefficients and replacing them in the recursive relation which uses the steepest descent method [7].

The convergence of all three algorithms is controlled by the choice of their respective step size, as shown in the next section for algorithm I.

The mathematical description of algorithm I is illustrated in Table 6.1, where all the pertinent parameters to the algorithm are detailed. As indicated in this table, for the initialization of the algorithm, it is customary to set all the initial values of the weights of the filter equal to zero.

6.2.3 Convergence in the mean

The sufficient condition for convergence in the mean of the adaptive algorithm can be shown to be:

$$0 < \mu < \frac{2}{(N_1 + N_2)\sigma_x^2}, \quad (6.6)$$

where N_1 and N_2 are the lengths of the near-end and the far-end cancellers, respectively, and σ_x^2 is the power of the input signal.

Table 6.1: The mathematical description of algorithm I.

Adaptive filter	$y(n) = \mathbf{C}^T(n)\mathbf{X}(n)$
Error equation	$e(n) = d(n) + w(n) - y(n)$
Update Equation	$\mathbf{C}(n+1) = \mathbf{C}(n) + \mu e(n)\mathbf{X}(n)$
Initialization	$\mathbf{C}(0) = \mathbf{0}$

From (6.6) it can be seen that the adaptation constant does not depend upon the variance of the noise. Also, the adaptive algorithm will converge to the optimum solution as the number of iterations approaches infinity and μ is set within the range given by (6.6).

6.2.4 Time constants of the algorithm

Let the autocorrelation matrix \mathbf{R} defined by (6.3) be positive definite. Accordingly, there will be different time constants for the weights, given by the following expression:

$$\tau_i = \frac{1}{\mu\lambda_i}, \quad i = 1, 2, \dots, N_1 + N_2, \quad (6.7)$$

where λ_i is the i th eigenvalue of \mathbf{R} . They will be all equal if all the eigenvalues of \mathbf{R} are all equal as well. When the input process is an independent, identically distributed Gaussian random process, then all time constants are equal as all

eigenvalues are equal in this case.

6.2.5 The misadjustment factor

In view of the fact that, in steady-state, the weight error vectors are uncorrelated [1], [95], the misadjustment factor M can be expressed by [14]:

$$M = \mu \cdot \text{tr}(\mathbf{R}). \quad (6.8)$$

The relationship between the step size and the misadjustment is clearly observed in the above expression. Since speed of convergence and misadjustment lead to conflicting requirements on the step size a compromise must then be reached. In general, to ensure convergence of the iterative procedure and produce less misadjustment error a small step size is chosen.

Finally, substituting (6.7) in (6.8) yields:

$$M = \sum_{i=1}^{N_1+N_2} \frac{1}{\tau_i}. \quad (6.9)$$

6.3 Convergence behaviour of algorithm II

Algorithm II is probably best known in the literature as the least mean-fourth (LMF) algorithm [8]. It is briefly described in subsequent sections.

6.3.1 The performance function

The performance function for this algorithm is based on the minimisation of the mean of the fourth power of the error, and is expressed by:

$$J_2(n) = E[e^4(n)]. \quad (6.10)$$

6.3.2 The updating scheme

The algorithm for recursively adjusting the tap coefficients, $\mathbf{C}(n)$, is expressed in the following form

$$\mathbf{C}(n+1) = \mathbf{C}(n) + 2\mu e^3(n)\mathbf{X}(n). \quad (6.11)$$

This algorithm is expected to be more complicated than the LMS algorithm, (6.5), because of the higher power of $e(n)$ involved in the computation of (6.11). This algorithm is also expected to be very sensitive to the value of the error, especially if the absolute value of the latter is greater than one. Differences in their computational burdens will be clarified in Section 6.5.

Finally, Table 6.2 summarises the mathematical description of algorithm II.

Table 6.2: The mathematical description of algorithm II.

Adaptive filter	$y(n) = \mathbf{C}^T(n)\mathbf{X}(n)$
Error equation	$e(n) = d(n) + w(n) - y(n)$
Update Equation	$\mathbf{C}(n+1) = \mathbf{C}(n) + 2\mu e^3(n)\mathbf{X}(n)$
Initialization	$\mathbf{C}(0) = \mathbf{0}$

6.3.3 Convergence in the mean

The sufficient condition for convergence in the mean of algorithm II can be easily derived along lines similar to those used for the LMMN algorithm. Specifically, it can be shown that:

$$0 < \mu < \frac{2}{6(N_1 + N_2)\sigma_x^2 E[w^2(n)]}, \quad (6.12)$$

where $E[w^2(n)]$ is the variance of the noise. This value affects the convergence behaviour of the algorithm if it is not equal to the actual value. But, in general if the value of $E[w^2(n)]$ is greater than the actual value it will enable μ to be in the range given by (6.12), only values less than the actual value will make μ greater than the upper bound of (6.12), which is not desired.

6.3.4 Time constants of the algorithm

Again, let the autocorrelation matrix \mathbf{R} be positive definite. Hence, there will be different time constants for the weights, given by the following expression

$$\tau_i = \frac{1}{6\mu E[w^2(n)]\lambda_i}, \quad i = 1, 2, \dots, N_1 + N_2, \quad (6.13)$$

where λ_i is the i th eigenvalue of \mathbf{R} . As in the LMS algorithm, the time constants of this algorithm will be equal only if the input signal is an independent, identically distributed Gaussian process.

6.3.5 The misadjustment factor

In view of the fact that, in steady-state, the weight error vectors are uncorrelated, the misadjustment factor M can in this case, be shown to be expressed by:

$$M = \frac{2\mu(N_1 + N_2)E[w^6(n)]E[x^2(n)]}{3\{E[w^2(n)]\}^2}. \quad (6.14)$$

Combining (6.13) and (6.14) yields:

$$M = \frac{E[w^6(n)]}{9\{E[w^2(n)]\}^3} \sum_{i=1}^{N_1+N_2} \frac{1}{\tau_i}. \quad (6.15)$$

Both of the above expressions indicate that the algorithm is dependent upon the statistics of the noise, that is on the distribution of the noise. Unlike for the LMS algorithm, the performance of this algorithm will therefore be affected by the noise statistics.

6.4 Convergence behaviour of algorithm III

This algorithm is based on two different cost functions, namely one for the near-end and one for the far-end sections of the canceller. The assignment of the cost functions is done in an opposite way to that of the LMMN algorithm.

6.4.1 The performance function

The cost function used for the near-end section is given by:

$$J_{N3}(n) = E[e^4(n)], \quad (6.16)$$

and for the far-end section by:

$$J_{F3}(n) = E[e^2(n)]. \quad (6.17)$$

6.4.2 The updating scheme

Based on this motivation, the algorithm for recursively adjusting the tap coefficients, $\mathbf{C}_N(n)$, of the NE section of the canceller and those of its FE section, i.e., $\mathbf{C}_F(n)$, is expressed in the following form:

$$\mathbf{C}_N(n+1) = \mathbf{C}_N(n) + 2\mu_1 e^3(n) \mathbf{X}_N(n), \quad (6.18)$$

$$\mathbf{C}_F(n+1) = \mathbf{C}_F(n) + \mu_2 e(n) \mathbf{X}_F(n), \quad (6.19)$$

where μ_1 and μ_2 are the step sizes for the near-end and the far-end sections, respectively.

Table 6.3 lists the key equations in the mathematical description of algorithm III.

Table 6.3: The mathematical description of algorithm III.

Adaptive filter	$y(n) = \mathbf{C}_N^T(n)\mathbf{X}_N(n) + \mathbf{C}_F^T(n)\mathbf{X}_F(n)$
Error equation	$e(n) = d(n) + w(n) - y(n)$
Update Equations	$\mathbf{C}_N(n+1) = \mathbf{C}_N(n) + 2\mu_1 e^3(n)\mathbf{X}_N(n)$ $\mathbf{C}_F(n+1) = \mathbf{C}_F(n) + \mu_2 e(n)\mathbf{X}_F(n)$
Initialization	$\mathbf{C}_N(0) = \mathbf{0}$ $\mathbf{C}_F(0) = \mathbf{0}$

6.4.3 Convergence in the mean

The sufficient conditions for convergence in the mean of algorithm III are given for the near-end and the far-end sections, respectively, by:

$$0 < \mu_1 < \frac{2}{6N_1\sigma_z^2 E[w^2(n)]}, \quad (6.20)$$

and

$$0 < \mu_2 < \frac{2}{N_2\sigma_x^2}. \quad (6.21)$$

Here, too it can be seen that instability might occur if the value of the noise variance, $E[w^2(n)]$, is smaller than the actual value.

6.4.4 Time constants of the algorithm

Assume that the weight vectors for the near-end and the far-end sections are close to their optimal values, and \mathbf{R} be positive definite. Accordingly, there will be different time constants for the weights, given by the following expressions for the near-end and the far-end sections, respectively,

$$\tau_{N_i} = \frac{1}{6\mu_1 E[w^2(n)] \lambda_{N_i}}, \quad i = 1, 2, \dots, N_1, \quad (6.22)$$

and

$$\tau_{F_i} = \frac{1}{\mu_2 \lambda_{F_i}}, \quad i = 1, 2, \dots, N_2, \quad (6.23)$$

where λ_{N_i} and λ_{F_i} are the i th eigenvalues of \mathbf{R}_N and \mathbf{R}_F , respectively.

6.4.5 The misadjustment factor

The misadjustment factor M can be derived in exactly the same way as was done for the LMMN algorithm in the previous chapter. This gives the following result:

$$M = \frac{2\mu_1 N_1 E[w^6(n)] E[x^2(n)]}{3\{E[w^2(n)]\}^2} + \frac{\mu_2 N_2 J_{F_{min}} E[x^2(n)]}{E[w^2(n)]}. \quad (6.24)$$

Expressing the misadjustment factor using (6.22) and (6.23) yields:

$$M = \frac{E\{w^6(n)\}}{9\{E\{w^2(n)\}\}^3} \sum_{i=1}^{N_1} \frac{1}{\tau_{Ni}} + \frac{J_{Fmin}}{E\{w^2(n)\}} \sum_{i=1}^{N_2} \frac{1}{\tau_{Fi}}. \quad (6.25)$$

Here, too, the misadjustment factor is a function of the moments of the noise, and hence its performance will depend on the statistics of the noise.

Table 6.4 summarises all the important parameters for the three algorithms. Note that some of the parameters listed in this table were not derived in this chapter. The first step size μ_{max} is obtained for the convergence of the algorithm in the mean, whereas the second step is obtained for the convergence in the mean-square. The step size that gives the fastest possible convergence is labeled μ_{opt} .

6.5 Computational complexity

If N_1 and N_2 are the number of coefficients used in the near-end and the far-end sections of the echo canceller, respectively, then the total number of operations (additions and multiplications), required to update each algorithm, is shown in Table 6.5.

As it can be observed from this table, all the algorithms require the same number of additions. However, the number of multiplications differ from one algorithm to another, but in general they are close to each other. The LMMN algorithm requires only three extra multiplications per update over the LMS algorithm (Algorithm I). This is an acceptable price to pay for the LMMN algorithm if its performance is going to be superior to that of the LMS algorithm.

This number of three extra multiplications per update can be reduced to only three multiplications for the whole update. From equation (6.5) it is observed that the LMS algorithm uses the $\mu e(n)$ in updating its coefficients. This term is calculated once and used by all the coefficients. However, the LMMN algorithm uses the term $2\mu e^3(n)$, equation (5.21), to update the coefficients in the far-end section. Hence, it is evaluated only once for the update of all the coefficients in

Table 6.4: Main parameters for the LMS, the LMF, and the LMFS algorithms.

	Algorithm I (LMS)	Algorithm II (LMF)	Algorithm III (LMFS)	
			Near-End Section	Far-End Section
Cost function	$J_1(n) = E[e^2(n)]$	$J_2(n) = E[e^4(n)]$	$J_{N3}(n) = E[e^4(n)]$	$J_{F3}(n) = E[e^2(n)]$
μ_{max}	$\frac{2}{\lambda_{max}(R)}$	$\frac{2}{6E[w^2(n)]\lambda_{max}(R)}$	$\frac{2}{6E[w^2(n)]\lambda_{Nmax}(R_N)}$	$\frac{2}{\lambda_{Fmax}(R_F)}$
μ_{max}	$\frac{2}{(N_1+N_2)\sigma_e^2}$	$\frac{E[w^2(n)]}{5(N_1+N_2)\sigma_e^2 E[w^4(n)]}$	$\frac{E[w^2(n)]}{5N_1\sigma_e^2 E[w^4(n)]}$	$\frac{2}{N_2\sigma_e^2}$
μ_{opt}	$\frac{1}{(N_1+N_2)\sigma_e^2}$	$\frac{E[w^2(n)]}{10(N_1+N_2)\sigma_e^2 E[w^4(n)]}$	$\frac{E[w^2(n)]}{10N_1\sigma_e^2 E[w^4(n)]}$	$\frac{1}{N_2\sigma_e^2}$
τ_i	$\frac{1}{\mu\lambda_i(R)}$	$\frac{1}{6\mu E[w^2(n)]\lambda_i(R)}$	$\frac{1}{6\mu_1 E[w^2(n)]\lambda_{Ni}(R_N)}$	$\frac{1}{\mu_2\lambda_{Fi}(R_F)}$

the far-end section. Consequently, only 3 multiplications for the whole update are added to the computational load of the LMMN algorithm as compared to that of the LMS algorithm.

Specially designed processors can reduce this computational burden further, and hence the total execution time.

Table 6.5: Computational complexity of the LMS, the LMF, the LMFS, and the LMMN algorithms.

Algorithm	Additions	Multiplications
LMS algorithm	$N_1 + N_2$	$2N_1 + 2N_2$
LMF algorithm	$N_1 + N_2$	$5N_1 + 5N_2$
LMFS algorithm	$N_1 + N_2$	$5N_1 + 2N_2$
LMMN algorithm	$N_1 + N_2$	$2N_1 + 5N_2$

6.6 Simulation results

The basic digital transmission system considered for the adaptive system identification is shown in Fig. 6.1. The input signal is binary ($x_i = \pm 1$) and of unit

power with a broadband power spectral density, such that a unique solution will exist. The additive noise was simulated by a uniformly distributed white random process with zero mean and a variance of -30 dB.

In comparing the performance of the LMMN algorithm to other three algorithms, three different channels are considered. These are adequately modelled by the following transfer function:

$$H(z) = \frac{1}{1 - az^{-1}}, \quad |a| < 1. \quad (6.26)$$

This gives an exponentially decaying response $\mathbf{H}^T = [h_0, h_1, \dots, h_{N-1}]$. Channels 1, 2 and 3 are characterized by $a = 0.1, 0.8,$ and -0.4 in (6.26), respectively. These channels have been carefully chosen to demonstrate the performance of the LMMN algorithm. All of the three channels have one zero at the origin and one pole at "a". The first channel, with a pole at $a = 0.1$, will have a fast decaying impulse response, in the sense that the far-end section will not be as effective as the near-end one. The second channel is different from the first channel in that its impulse response takes a long time to vanish. Hence, its near-end and far-end sections will be equally important. The main difference between these two channels is therefore in the effectiveness or lack of it far-end section. Finally, channel 3 has an impulse response that decays in an oscillatory fashion, [60], hence having a far-end section with an equally oscillating effectiveness. They are all of practical interest since they are good models for real channels. All of these channels were chosen to have a unit gain for a white noise input, i.e., $\sum_{i=0}^{N-1} h_i^2 = 1$. This is convenient for the computation of both the variance of the additive noise and the step size of each algorithm. Note that all the three channels are modelled as having a near-end section, a far-end section, and a bulk delay inserted between them.

The unknown system was chosen to have a near-end section, a far-end section, and a bulk delay section. The number of coefficients in the near-end and the far-end sections was equal to 50. The bulk delay section was chosen to have a span time equal to that of the near-end and the far-end sections and with no arithmetic

operations allowed in it. The total number of active coefficients in the unknown system is then equal to 100.

The performance measure most appropriate to echo cancellation problem considered here is the normalized weight error norm:

$$\rho(n) = \frac{\|(\mathbf{C}(n) - \mathbf{C}_{opt})^T(\mathbf{C}(n) - \mathbf{C}_{opt})\|}{\|\mathbf{C}_{opt}^T \mathbf{C}_{opt}\|}$$

where $\mathbf{C}(n)$ is the impulse response of the adaptive system at iteration n , and \mathbf{C}_{opt} is the impulse response of the unknown system. This performance measure was estimated in all the simulations by averaging over an ensemble of 50 runs of the adaptive system. The adaptive filter was initialized with a zero impulse response vector, in all runs. The number of coefficients used in the adaptive system was set equal to that of the unknown system.

The step sizes for the LMS algorithm, the LMF algorithm, the LMFS algorithm, and the LMMN algorithm were chosen so that the respective time constants of these algorithms approximately correspond to 100 samples. These step sizes are found from equations (6.7), (6.13), (6.22), (6.23), (5.71), and (5.72), respectively.

6.6.1 Performance of the LMMN algorithm

One of the ways algorithms are typically compared is by examining their convergence rates. The convergence performance of all four algorithms is illustrated in Figures 6.2, 6.3, and 6.4, for channels 1, 2, and 3, respectively. The LMMN algorithm clearly outperforms the other three. The second best in performance is the LMS algorithm. The LMF and the LMFS algorithms resulted in the worse performance. However, the LMFS algorithm is relatively better than the LMF algorithm. It is not surprising, if the LMF algorithm is malfunctioning in this kind of application, it is expected that the LMFS algorithm would behave the same way, since it is a derivative of the LMF algorithm. The cost function of the LMFS algorithm for the far-end section will not do any good if the total algorithm for the near-end section is not functioning properly.

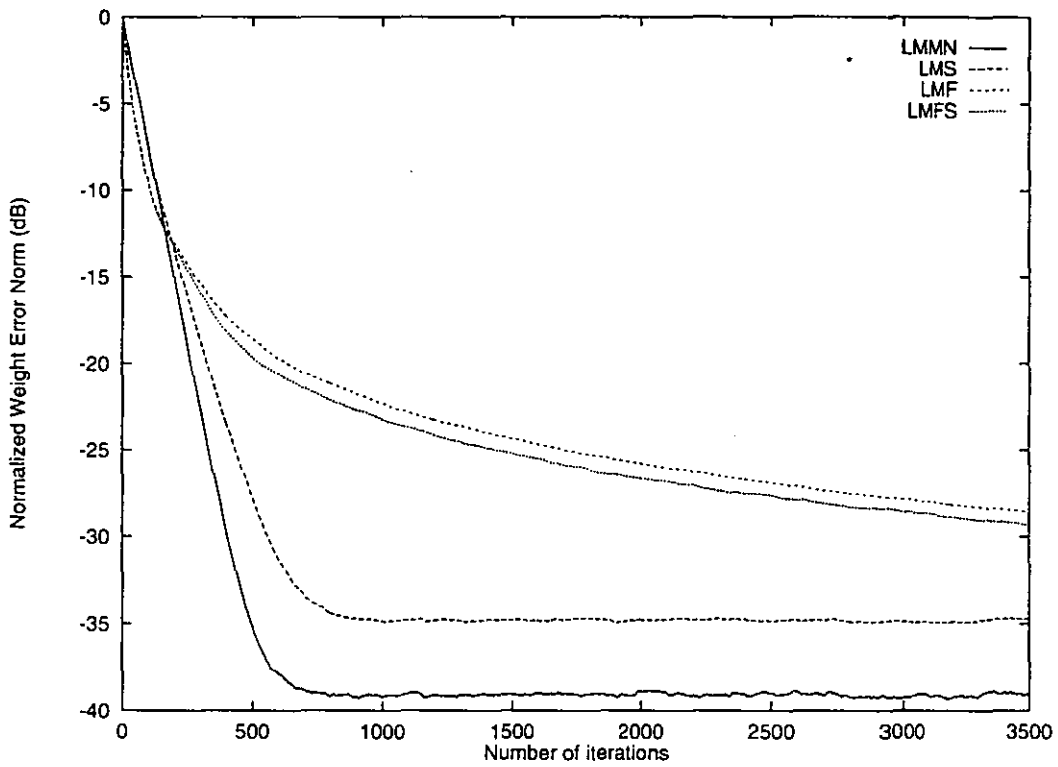


Figure 6.2: Learning curves for the four algorithms for channel 1.

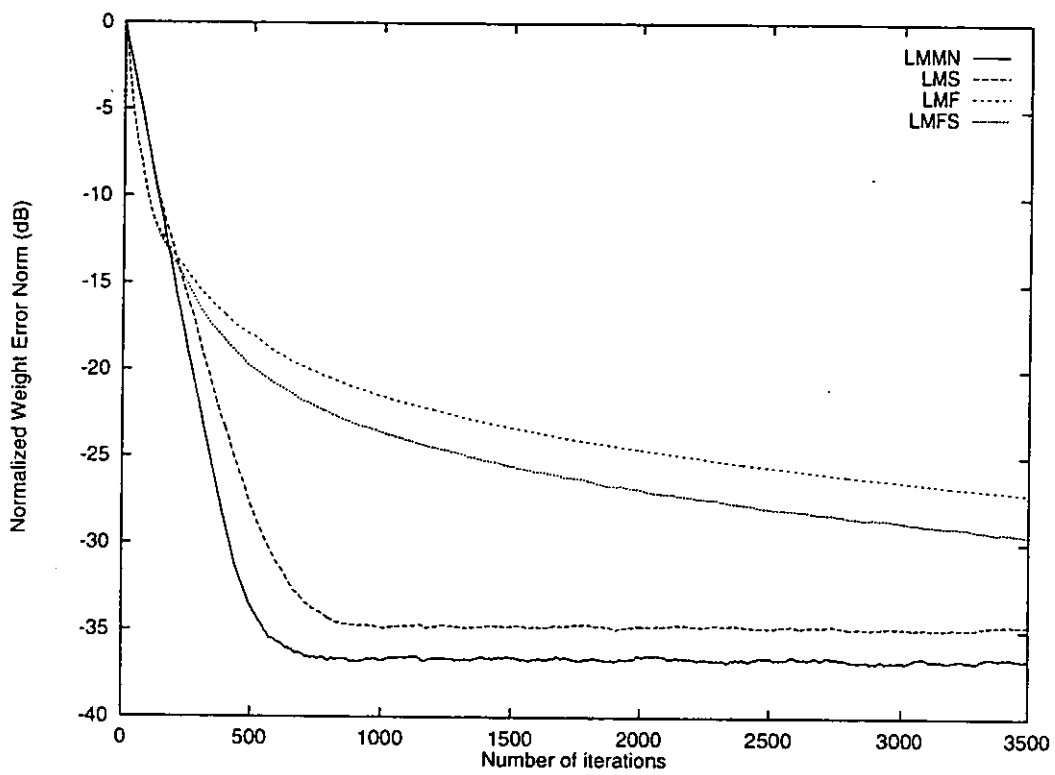


Figure 6.3: Learning curves for the four algorithms for channel 2.

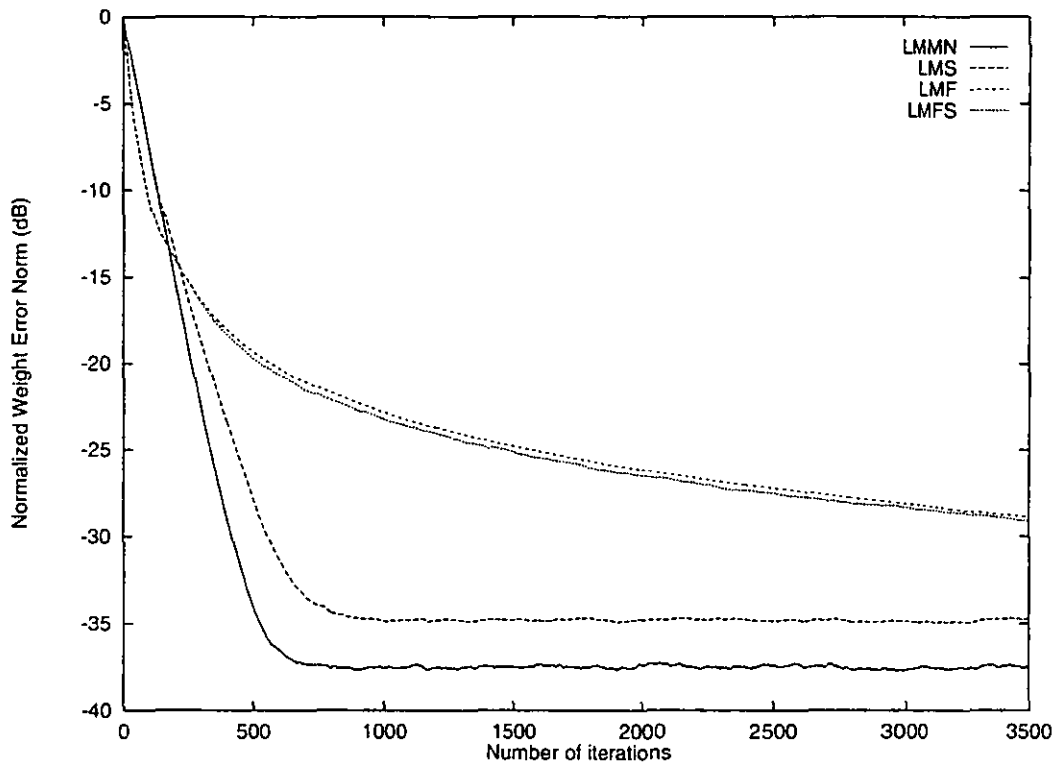


Figure 6.4: Learning curves for the four algorithms for channel 3.

With reference to both the LMMN algorithm and the LMS algorithm, it is important to note that the former algorithm is a derivative of the latter algorithm as far as the near-end section is concerned. The far-end section removes more of the noise in the weights, ending up with a better performance than the LMS algorithm. Minimising the square of the error in the far-end section does not result in much improvement over minimising its fourth power, when the near-end section is based on the minimisation of the square of the error.

The new updating algorithm (i.e., the LMMN algorithm) provides, on the average, about 2.6 dB less weight noise than the LMS algorithm does. The reduction in noise level in the weights was reached when minimising the fourth power of the error was applied to the far-end section, and minimising the square of the error in the near-end section. The usual way is to minimise the square of the error on both sections of the canceller, as is done in the LMS algorithm.

Also, the close agreement between theory and experiment is depicted in Figures 6.5, 6.6, and 6.7, for channels 1, 2, and 3, respectively. The theoretical curve is obtained as follows: equations (5.71) and (5.72) define the time constants of the algorithm. Since the input signal is a binary sequence drawn from a white random process generator, its eigenvalues are all equal [35]. Consequently, all the time constants of the weights in the near-end section are equal and so are those of the far-end section. Since the convergence of each coefficient decays exponentially with its corresponding time constant, the average of all these decays results in the theoretical curve.

The LMMN algorithm converges faster than the LMS algorithm in two respects: when the convergence rates are given the same time constant, and when both algorithms converge to the same steady state value. This is depicted in Figures 6.8, 6.9, and 6.10, for channels 1, 2, and 3, respectively. As indicated by these curves, the convergence time for the LMS algorithm is twice that of the LMMN algorithm.

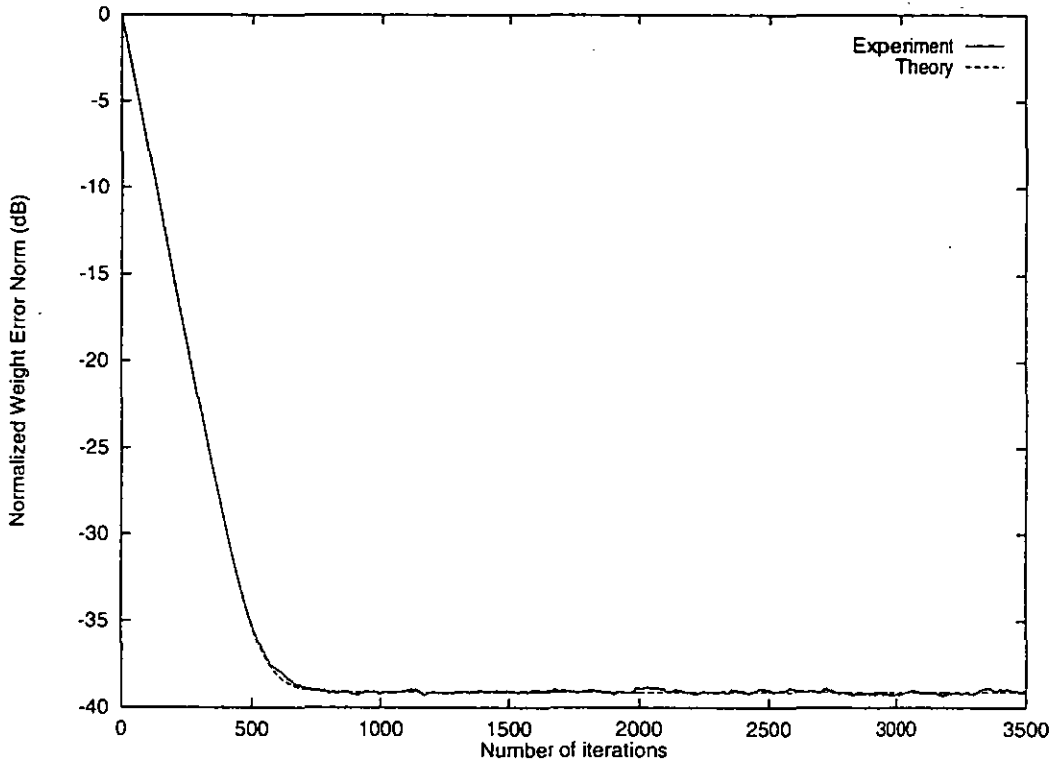


Figure 6.5: Theoretical and experimental learning curves of the LMMN algorithm for channel 1.

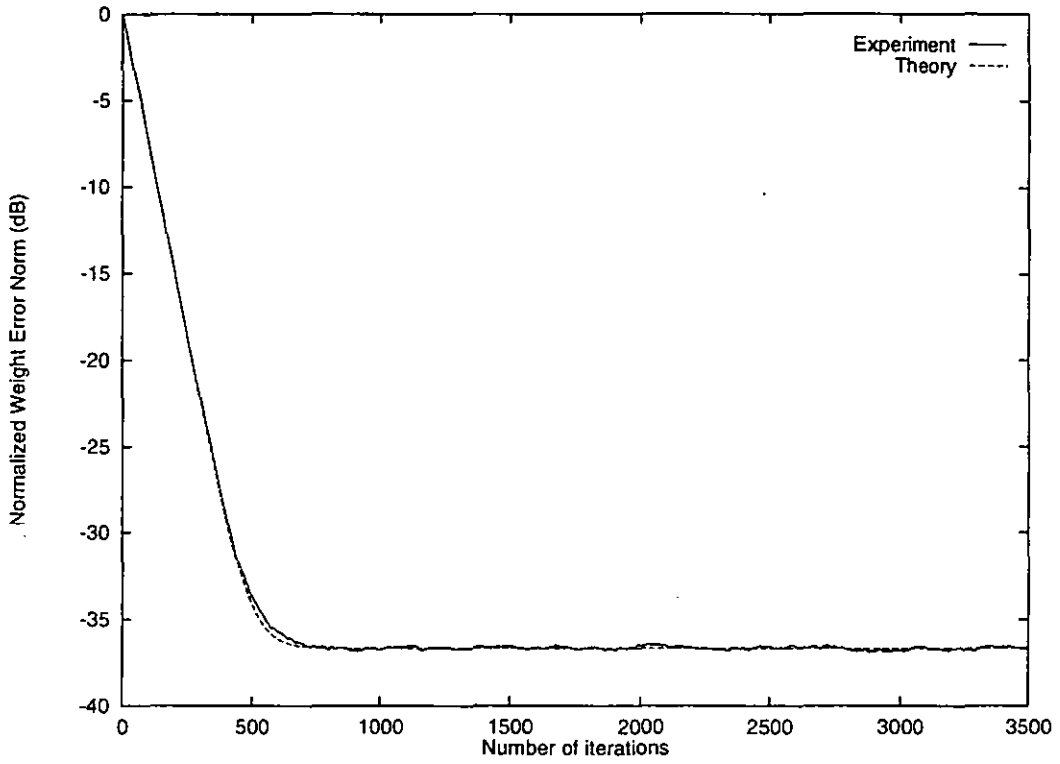


Figure 6.6: Theoretical and experimental learning curves of the LMMN algorithm for channel 2.

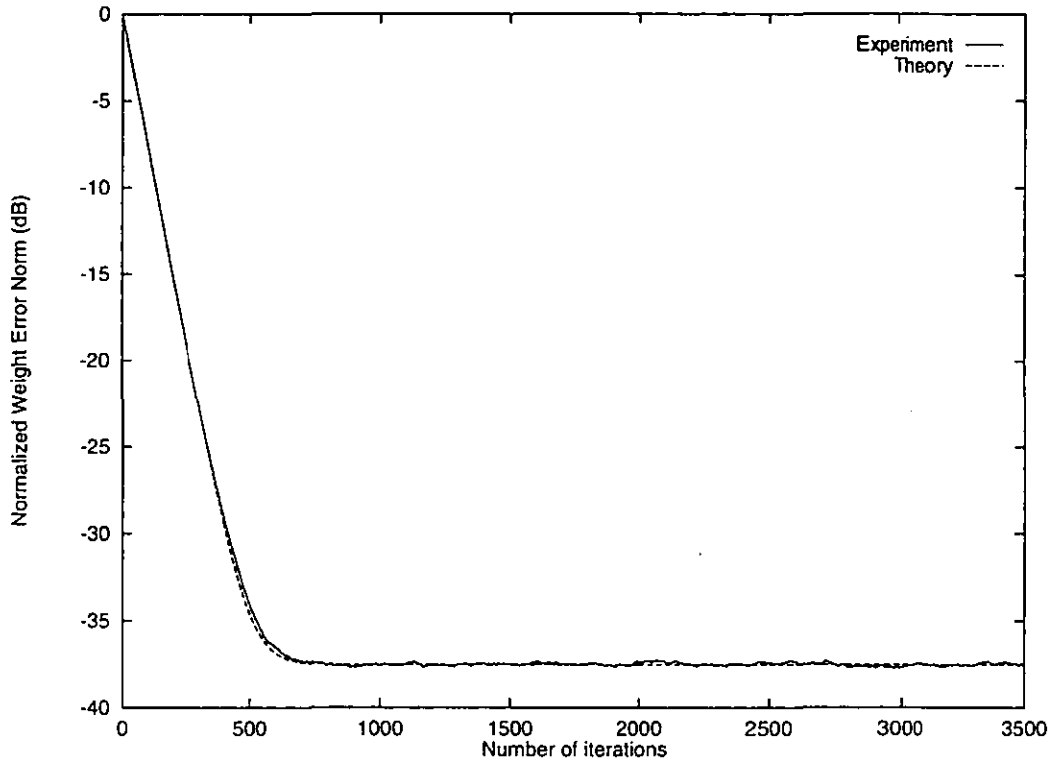


Figure 6.7: Theoretical and experimental learning curves of the LMMN algorithm for channel 3.

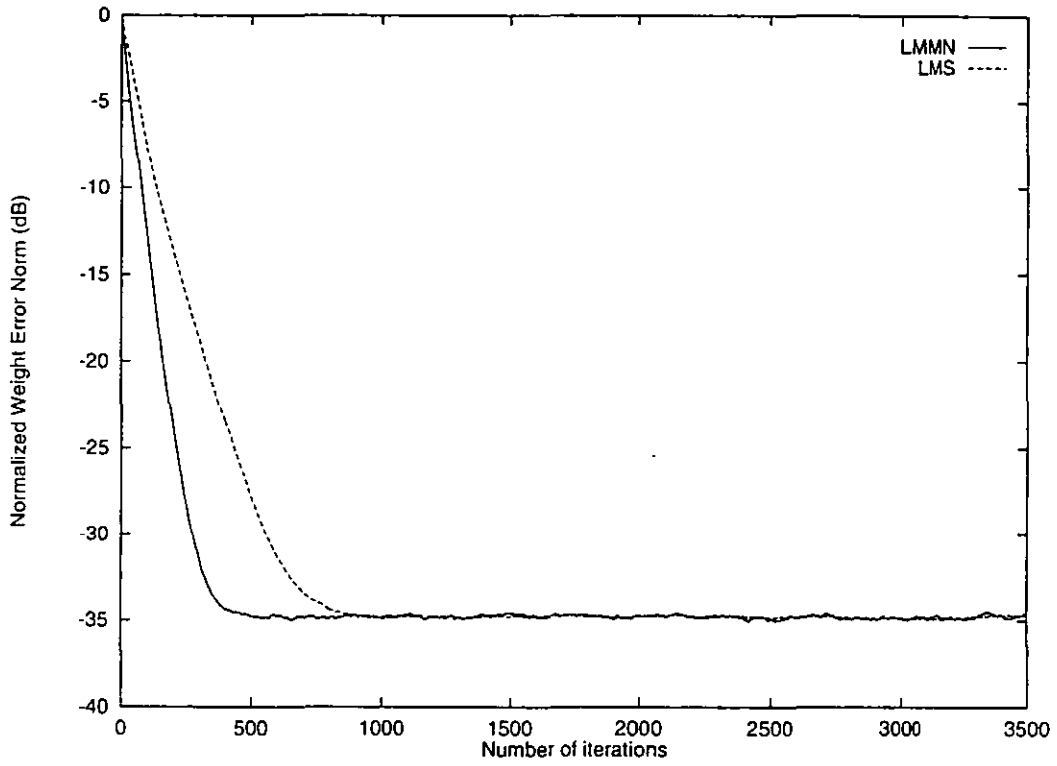


Figure 6.8: Learning curves for the LMMN and the LMS algorithms with the same steady state value used for channel 1.

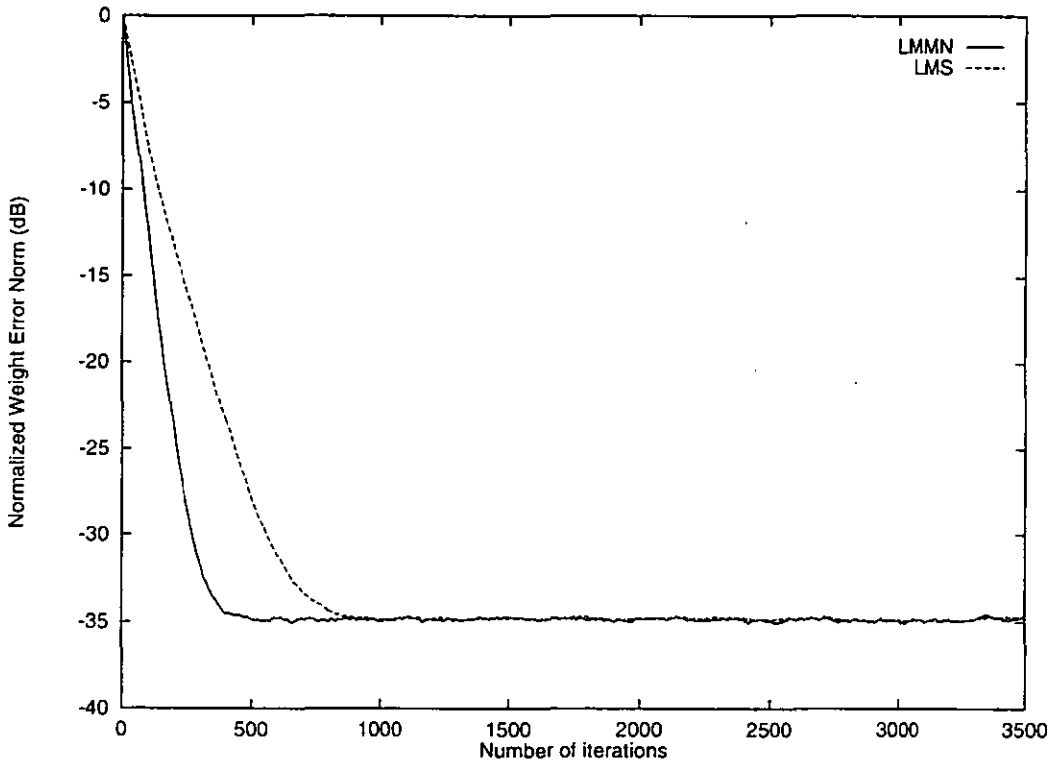


Figure 6.9: Learning curves for the LMMN and the LMS algorithms with the same steady state value used for channel 2.

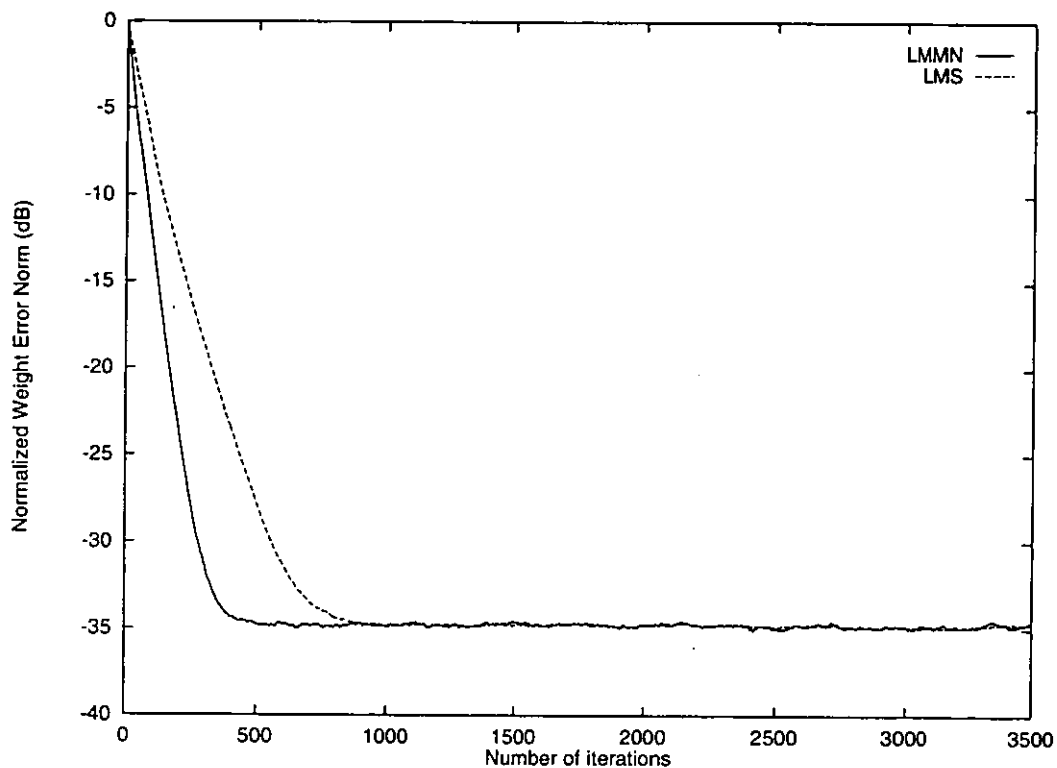


Figure 6.10: Learning curves for the LMMN and the LMS algorithms with the same steady state value used for channel 3.

The overall superiority of the LMMN algorithm over all others is no doubt due to its structure.

6.6.2 The effect of the noise variance

The better performance obtained by the LMMN algorithm over the others from both the misadjustment error and the convergence time points of view, is due to the fact that the noise variance is exactly known. However, this better performance cannot be guaranteed when the value of the noise variance used differs from the actual one. Unlike for the three algorithms (i.e., the LMF, the LMFS, and the LMMN), the noise variance is a parameter that is irrelevant to the LMS algorithm.

Since the effect of this parameter on the performance of the LMMN algorithm is important, proper care must therefore be given to it in order for the algorithm to converge to the optimum solution or as close as possible to it.

In general this value is not known a priori, and therefore as can be seen from the convergence factor of the far-end section derived previously in the last chapter and reported here again

$$0 < \mu_2 < \frac{2}{6N_2\sigma_x^2 E[w^2(n)]}, \quad (6.27)$$

that theoretically speaking if $E[w^2(n)]$ is greater than the actual value, convergence happens since this value keeps μ_2 in the range given by (6.27). However, if $E[w^2(n)]$ is smaller than the actual value, μ_2 will be larger than the upper bound of (6.27), hence instability might occur and the algorithm does not converge to the optimum solution. Indeed the simulations showed that the whole algorithm becomes unstable for very small values of $E[w^2(n)]$, even if μ_1 (the step size for the near-end section) is kept to within its range given by equation (5.54).

Figures 6.11, 6.12, and 6.13 depict the effect of the noise variance on the convergence behaviour of the LMMN algorithm, for channels 1, 2, and 3, respectively.

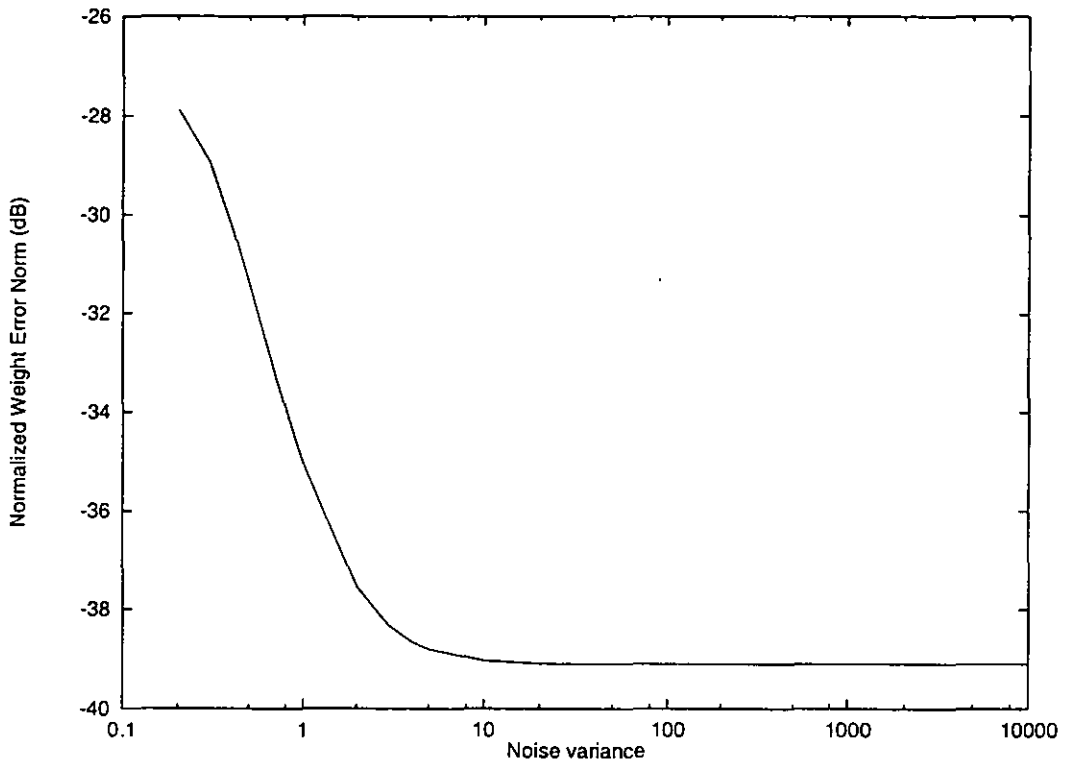


Figure 6.11: Effect of noise variance on the convergence behaviour of the LMMN algorithm for channel 1.

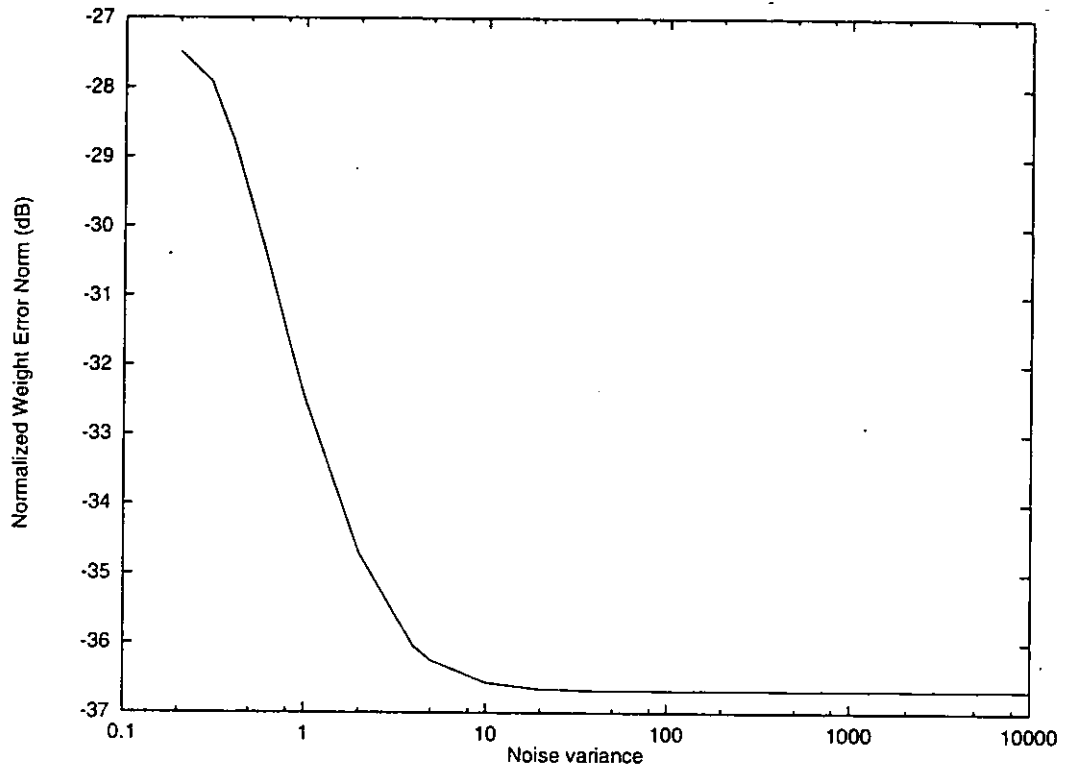


Figure 6.12: Effect of noise variance on the convergence behaviour of the LMMN algorithm for channel 2.

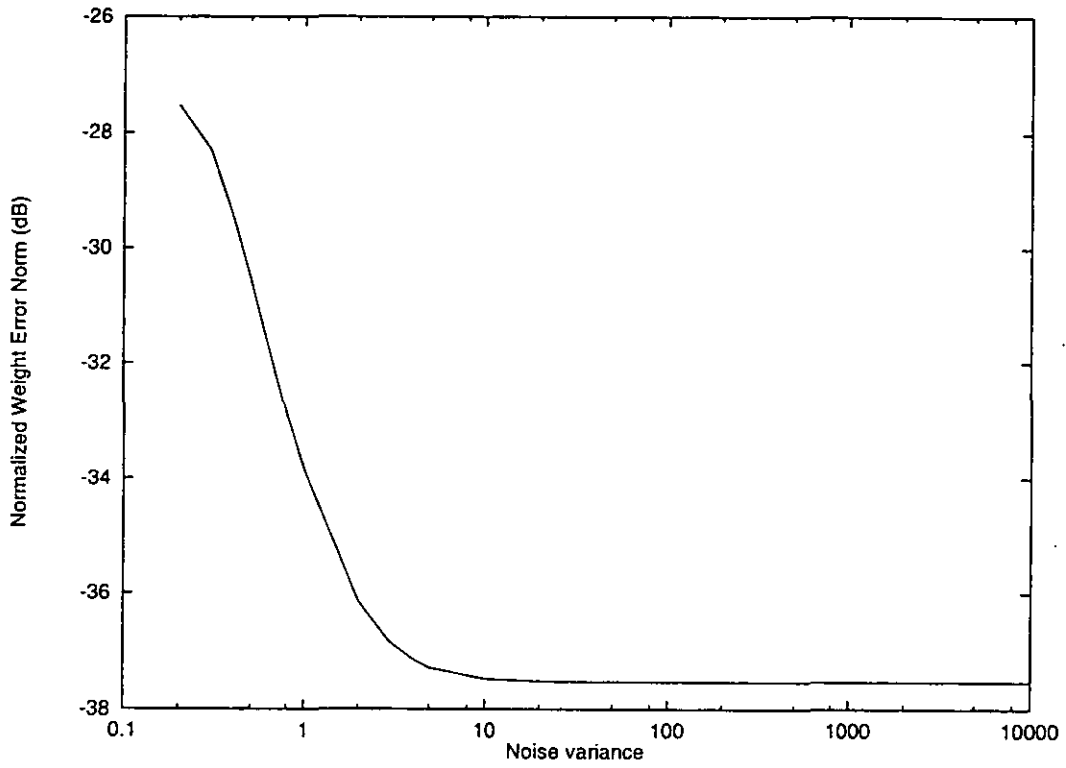


Figure 6.13: Effect of noise variance on the convergence behaviour of the LMMN algorithm for channel 3.

It is clear, from these figures, that when the value of the noise variance is equal to or greater than 10, the algorithm converges to the optimum solution. If a lesser value is used the algorithm will converge to some other non-optimal value.

Since the actual value of the noise variance is 1000, it is clear from these figures that as predicted by theory that when the noise variance is much greater than 1000 or close to it, the convergence of the algorithm to the optimum solution is always guaranteed. However, if this value is very small, the algorithm first does not converge to the optimum solution, and second instability is likely to happen. Using lesser value than 1000 for the noise variance will lead to both a lack of convergence and a likely instability of the algorithm.

In general a very large number for the noise variance can be safely used.

6.6.3 The effect of the noise distribution

It is known that when the noise distribution is Gaussian, the LMS algorithm outperforms the LMF algorithm [8] and its higher order versions based on the minimisation of the cost function $E[e^{2k}(n)]$, k being an integer greater than or equal to 2. However, the situation reverses when other noise distributions (e.g., uniform, sine wave, square wave) are used, i.e., the performance of the LMS algorithm deteriorates compared to that of the LMF algorithm and its higher order versions.

Since the LMMN algorithm is a hybrid algorithm, i.e, it uses the LMS algorithm for the near-end section of the echo canceller and the LMF algorithm for its far-end section, its performance will be affected by the noise distribution. Figures 6.14, 6.15, and 6.16, depict the effect of the noise distribution on the convergence behaviour of the LMMN algorithm, for channels 1, 2, and 3, respectively.

For simplicity, the misadjustment factor (5.79) for the LMMN algorithm can be written in the following form

$$M = \mu_1 N_1 \frac{J_{Nmin}}{E[w^2(n)]} E[x^2(n)] + \frac{2}{3} \mu_2 N_2 \frac{E[w^6(n)]}{\{E[w^2(n)]\}^2} E[x^2(n)]. \quad (6.28)$$

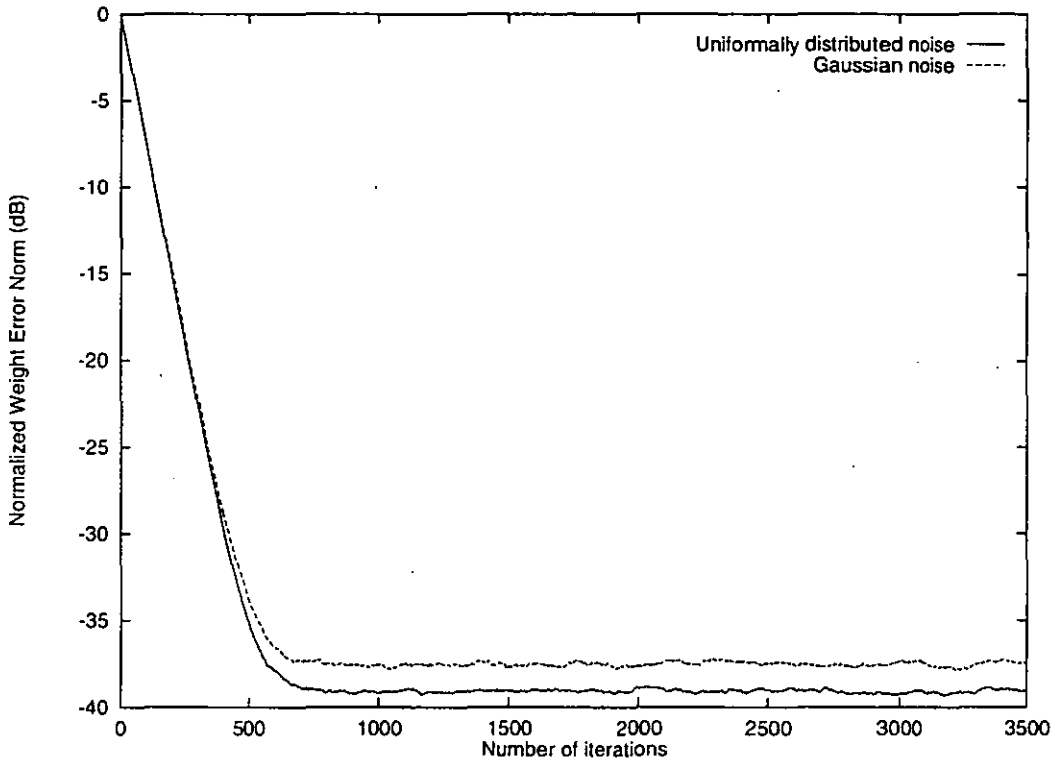


Figure 6.14: Effect of noise distribution on the convergence behaviour of the LMMN algorithm for channel 1.

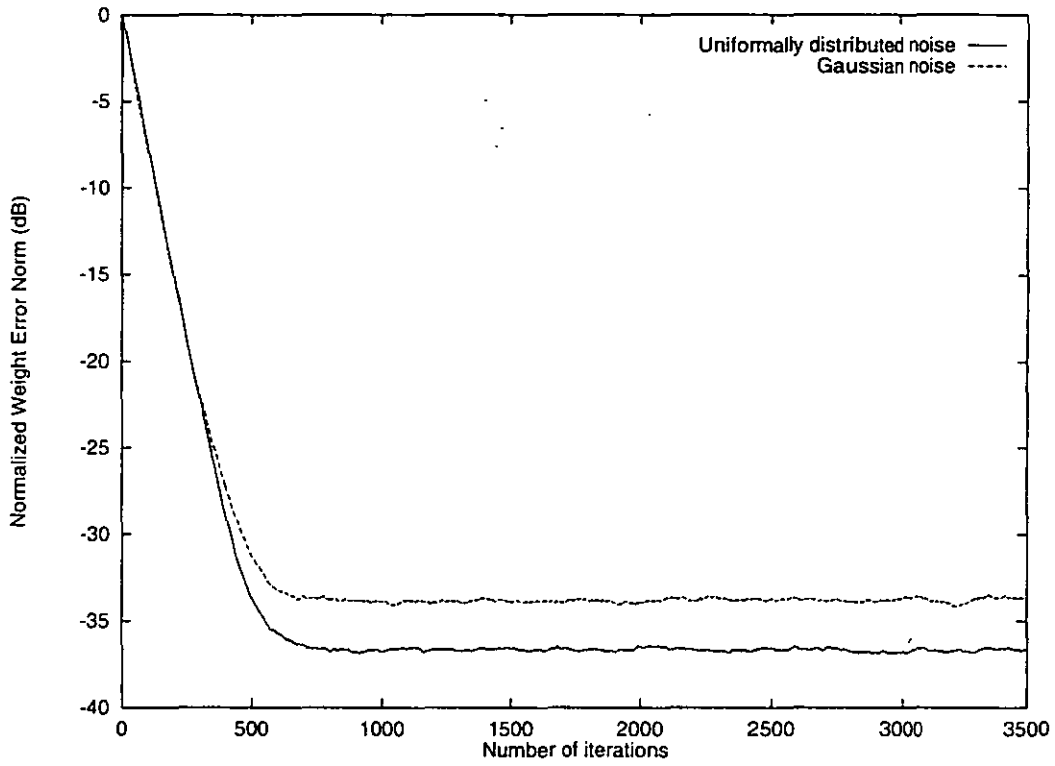


Figure 6.15: Effect of noise distribution on the convergence behaviour of the LMMN algorithm for channel 2.

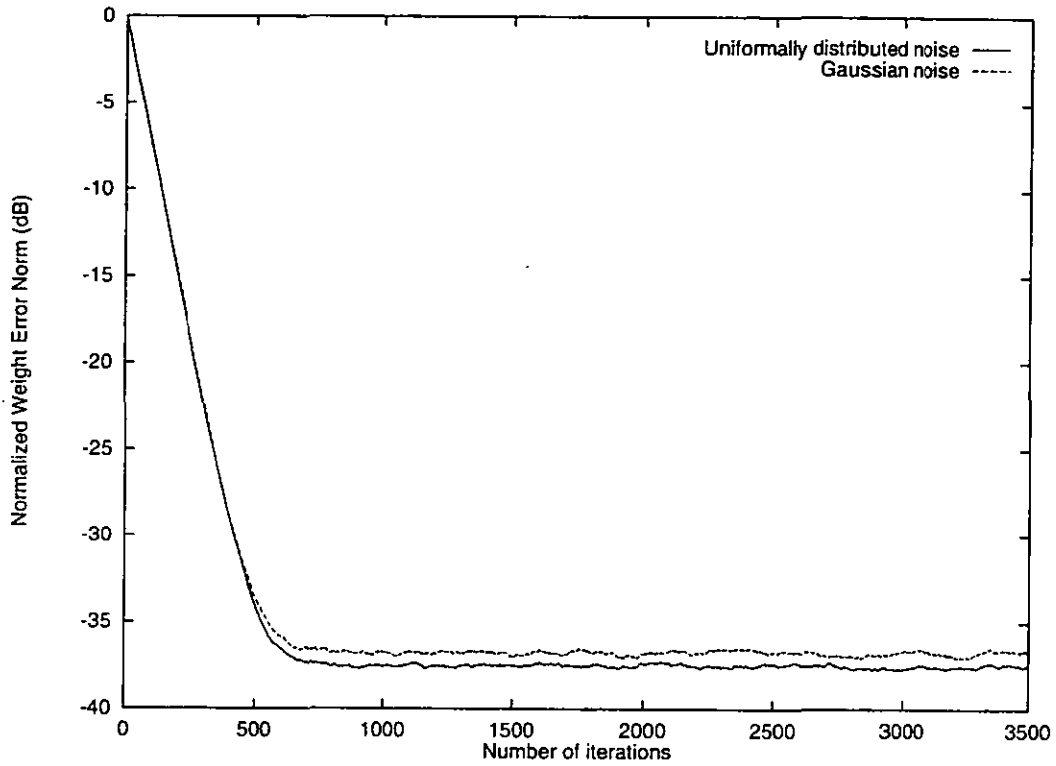


Figure 6.16: Effect of noise distribution on the convergence behaviour of the LMMN algorithm for channel 3.

When the noise distribution is uniform, it can be easily verified that the misadjustment factor for this algorithm is given by:

$$M = \mu_1 N_1 \frac{J_{Nmin}}{E[w^2(n)]} E[x^2(n)] + 2.6\mu_2 N_2 E[w^2(n)] E[x^2(n)]. \quad (6.29)$$

In the case of a Gaussian distribution, this factor becomes:

$$M = \mu_1 N_1 \frac{J_{Nmin}}{E[w^2(n)]} E[x^2(n)] + 10\mu_2 N_2 E[w^2(n)] E[x^2(n)]. \quad (6.30)$$

The values of $E[w^6(n)]$ and $E[w^2(n)]$ have been both evaluated [97] for the uniform and Gaussian distributions, respectively, and substituted in (6.28) to obtain (6.29) and (6.30), respectively.

As can be seen by (6.29) and (6.30), the uniform distribution is expected to give a lower misadjustment error than its Gaussian counterpart. The experimental results support this expectation. An average improvement of 1.7 dB in the normalized weight error norm is obtained with the uniform distribution.

6.6.4 The rescue condition

As was mentioned earlier, algorithms containing the term $E[e^4(n)]$ are expected to be destabilized if the absolute value of the error is greater than or equal to one. To prevent this from happening, a rescue condition is included. During the simulation of the LMMN algorithm, equation (5.5), was set as if it were minimising the following cost function for the far-end section:

$$J_F(n) = \beta E[e^2(n)] + (1 - \beta) E[e^4(n)], \quad (6.31)$$

where β is set to 1 when the absolute value of $e(n)$ becomes greater than or equal to 1, and zero otherwise. This will preserve the stability of the algorithm.

6.7 Summary

A new adaptive scheme for echo cancellation has been introduced. It was shown that minimising both the mean square criterion and the mean fourth one over

the near-end and the far-end sections of the echo canceller, respectively, leads to superior performance compared to all 3 algorithms, including the commonly-used LMS algorithm. It was also shown that the LMMN algorithm presented an extra three multiplications per update compared to the LMS algorithm.

It was first shown that the LMMN algorithm outperforms all other three algorithms as far as both the convergence behaviour and speed of convergence are concerned when these are given the same time constant and the same steady state value, respectively.

It was then observed that the LMMN algorithm was dependent upon both the noise variance and distribution. Whereas the LMS algorithm is dependent upon the noise distribution only. The expected performance of the LMMN algorithm for a uniformly distributed noise was confirmed by simulation.

Finally, since the actual value of the noise variance is usually not known a priori, it is shown that a choice of a large value for the noise variance, tends to safeguard both the convergence to the optimal solution (or close to it) and the stability of the algorithm.

Chapter 7

Conclusion

7.1 Achievements of the work

In this thesis, new algorithms for long data echo cancellers are proposed. The development of these algorithms was confined to the class of stochastic gradient algorithms only.

These algorithms are designed to improve the performance of the steady state behaviour of the LMS algorithm, i.e., reducing its misadjustment error, or its convergence speed, i.e., shortening its start up procedure. Two novel approaches were considered for these purposes.

In the first approach, the proposed algorithm was designed using a modification to the LMS algorithm. It was shown that the new algorithm improved the performance of the standard LMS algorithm when the delay between the near-end and far-end sections of the echo canceller is not accurately estimated. Such an inaccuracy in the bulk delay is known to adversely affect the performance of the algorithm, thus resulting in a large misadjustment error.

The suggested algorithm compensates for this miscalculation. The approach proposes a shaping for the transitions at the end of the near-end and at the beginning of far-end sections. Two transitions were considered in this work, i.e., a sharp

transition and a smooth one. In the latter transition, the uniform probability density function was assigned to the coefficients belonging to these locations. It was shown in the thesis that a substantial enhancement in performance is observed when this type of shaping is considered. Our algorithm therefore exhibits more robustness to the misadjustment error than does the LMS algorithm.

Moreover, the steady state and transient behaviours were also investigated, and the simulation results were found to be in a close agreement with the theoretical analysis. Also, the computational complexity of our algorithm compares well with that of the LMS.

In the second approach, a new adaptive scheme for echo cancellation was introduced, namely, the least mean-mixed norm (LMMN) algorithm. It was shown that minimising the mean square criterion and the mean fourth one over the near-end and the far-end sections of the echo canceller, respectively, leads to superior performance. However, reversing the order of the two minimisation criteria resulted in a poor performance of the resulting algorithm called the least mean fourth-square (LMFS).

The LMMN algorithm was compared to the LMS, the LMF and the LMFS algorithms. The LMMN algorithm outperformed all three of them in terms of convergence behaviour and speed. Next to the performance of the LMMN algorithm was that of the LMS. The LMS algorithm outperformed the LMF and the LMFS algorithms.

It was then observed that the LMMN algorithm was dependent upon both the noise variance and distribution. In this case it was suggested that the noise variance should be large to safeguard both the convergence to the optimal solution and the stability of the algorithm. Recall that the LMS algorithm, however, is dependent only upon the noise distribution.

The expected performance of the LMMN algorithm for a uniformly distributed noise was confirmed by simulation.

Finally, from a computational viewpoint, it was shown that the LMMN algo-

rithm presented only three extra multiplications per update compared to the LMS algorithm.

Another minimisation criterion was also proposed [98] to enhance the performance of the adaptive echo canceller. In [98], a mixed controlled-error-norm was proposed. It consists of minimising the following cost function:

$$J(n) = -\alpha E[e_1^2(n)] + (1 + \alpha)E[e_2^2(n)],$$

where $e_1(n)$ and $e_2(n)$ are the errors between the desired value and the outputs of the near-end and the far-end subcancellers, respectively, and α is the mixing parameter. However, only a minor improvement was obtained from this modification.

7.2 Summary of main contributions

The main contributions to the work carried out in this thesis can be summarised as follows:

- A development of a statistical analysis for long data echo cancellers.
- Novel statistical treatment of the bulk delay area of the adaptive echo canceller and the proposition and study of transitions.
- Development, analysis, and simulation of new mixed error-norms algorithms (LMMN and LMFS) using two different cost functions, one for each section of the echo canceller.
- Evaluation of the computational complexity of the proposed algorithms and their comparison with other well known algorithms.

7.3 Suggestions for further work

During the investigation of adaptive echo cancellers using the mixed norm approach, it was observed that both the LMMN and the LMS algorithms outperformed the LMFS algorithm. We can argue that when the MSE is applied to the

near-end section of the echo canceller and either the MFE or the MSE is applied to its far-end section, the performance of the resulting algorithm is better than that of the algorithm using the MFE in the near-end section of the echo canceller and either the MFE or the MSE in its far-end section.

A possible suggestion, then, would be to look at algorithms with the MSE in the near-end section of the echo canceller and any other cost function of the form $E[e^{2k}(n)]$, with $k \geq 3$, applied to its far-end section.

Also, the application of other algorithms used for long echo cancellers with different shaping than the ones given in this thesis, is worthy of further investigation.

Moreover, further investigations on how to improve the cost function defined in [98] such that an enhancement is provided to its corresponding algorithm needs to be pursued.

Other directions for further research include:

- The use of higher order statistics for the design of echo cancellers. A related work for noise cancellation has recently been reported in [99].
- Evaluation of the robustness of the proposed schemes in nonlinear and time varying environments.
- Design of IIR-based echo cancellers using the proposed algorithms.

Bibliography

- [1] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, "Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter," *Proc. IEEE*, vol. 64, No. 8, pp. 1151-1162, Aug. 1976.
- [2] H. W. Sorenson, "Least-Squares Estimation from Gauss to Kalman," *IEEE Spectrum*, vol. 7, pp. 63-68, July 1970.
- [3] D. G. Messerschmitt, "Echo Cancellation in Speech and Data Transmission," *IEEE J. Selected Areas in Comm.*, vol. SAC-2, No. 2, pp. 283-297, Mar. 1982.
- [4] R. W. Lucky, "Automatic Equalization for Digital Communication," *Bell Syst. Tech. J.*, vol., 44, No. 3, pp 547-588, Apr. 1965.
- [5] S. B. Weinstein, "Echo Cancellation on the Telephone Network," *IEEE Commun. Magazine*, vol. 16, No. 11, pp. 9-15, Jan. 1977.
- [6] A. Zerguine, C. F. N. Cowan, and M. Bettayeb, "Adaptive Echo Cancellation Using Statistical Shaping," *Twenty-ninth Annual, Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, California* Oct. 29-Nov. 1, 1995.
- [7] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Englewood Cliffs, NJ, 1991.

- [8] E. Walach and B. Widrow, "The Least Mean Fourth (LMF) Adaptive Algorithm and its Family," *IEEE Trans. Inf. Theory*, vol. IT-30, pp. 275-283, Feb. 1984.
- [9] D. I. Pazaitis and A. G. Constantinides, "LMS+F algorithm," *Electr. lett.*, vol. 31, No. 17, pp. 1423-1424, Aug. 1995.
- [10] J. A. Chambers, O. Tanrikulu, and A. G. Constantinides, "Least mean mixed-norm adaptive filtering," *Electr. lett.*, vol. 30, No. 19, pp. 1574-1575, Sep. 1994.
- [11] M. M. Sondhi, "An Adaptive Echo Canceller," *Bell Syst. Tech. J.*, vol. 46, No. 3, pp. 497-511, Mar. 1967.
- [12] K. Murano, "Adaptive Signal Processing Applied in Telecommunications," *IFAC Adaptive Systems in Control and Signal Processing* pp. 431-441, 1992.
- [13] S. U. H. Qureshi, "Adaptive Equalization," *Proc. IEEE*, vol. 73, No. 9, pp. 1349-1387, Sept. 1985.
- [14] B. Widrow, J. R. Glover, Jr., J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, E. Dong, Jr., and R. C. Goodlin, "Adaptive Noise Cancelling: Principles and Applications," *Proc. IEEE*, vol. 63, No. 12, pp. 1692-1716, Dec. 1975.
- [15] C. F. N. Cowan, J. Mavor, and J. W. Arthur, "Noise Cancellation and Inverse Filtering using a Compact High-Performance C.C.D Adaptive Filter," *Electr. lett.*, vol. 15, No. 1, pp. 35-37, Jan. 1979.
- [16] C. F. N. Cowan and P. M. Grant, *Adaptive Filters*, Prentice-Hall, Englewood Cliffs, NJ, 1985.
- [17] M. M. Sondhi and D. A. Berkley, "Silencing Echoes on the Telephone Network," *Proc. IEEE*, vol. 68, No. 8, pp. 948-963, Aug. 1980.

- [18] S. U. H. Qureshi, "Adaptive Equalization," *IEEE Commun. Magazine*, vol. 21, No. 2, pp. 9-16, Mar. 1982.
- [19] A. H. Gray and J. D. Markel, *Linear Prediction of Speech*, Springer Verlag, Berlin, 1976.
- [20] J. Makhoul, "Linear Prediction a Tutorial Review," *Proc. IEEE*, vol. 63, No. 4, pp. 561-580, April 1975.
- [21] J. G. Proakis, *Digital Communications*, McGraw-Hill, New York, 1983.
- [22] T. Kailath, "A View of three Decades of Linear Filtering Theory," *IEEE Trans. Inf. Theory*, vol. IT-20, pp. 146-181, March 1974.
- [23] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive Antenna Systems Array," *Proc. IEEE*, vol. 55, pp. 2143-2159, Dec. 1967.
- [24] G. Leitmann, *Mathematics in Science and Engineering vol 5: Optimization Techniques*, Academic Press, New York, 1962.
- [25] G. Leitmann, *Mathematics in Science and Engineering vol 31: Topics in Optimization*, Academic Press, New York, 1967.
- [26] D. Luenberger, *Optimization by Vector Space Methods*, John Wiley & Sons, New York, 1969.
- [27] N. Levinson, "The Wiener RMS (Root Mean Square) Error Criterion in Filter Design and Prediction," *J. Math. Phys.*, vol. 25 pp. 261-278, Jan. 1947.
- [28] J. M. Mendel, *Discrete Techniques Of Parameter Estimation*, Marcel Dekker, New York, 1973.
- [29] T. Kailath, *Lectures on Linear Least-Squares Estimation*, Springer-Verlag, New York, 1981.

- [30] D. D. Falconer and L. Ljung, "Application of fast Kalman Estimation to Adaptive Equalization," *IEEE Trans. Commun.*, vol. COM-26, pp. 1439-1446, June 1978.
- [31] G. Carayannis, D. G. Manolakis, and N. Kalouptsidis, "A Fast Sequential Algorithm for Least-Squares Filtering and Prediction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1394-1402, Dec. 1983.
- [32] J. M. Cioffi and T. Kailath, "Fast, Recursive-Least-Squares Transversal Filters for Adaptive Filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 304-337, 1984.
- [33] N. Wiener, *The Extrapolation, Interpolation, and Smoothing of stationary Time Series*, John Wiley & Sons, New York, 1962.
- [34] C. F. N. Cowan, "Performance Comparison of finite Linear Adaptive Filters," *IEE Proceedings Part F* vol. 134, No. 3, pp. 211-216, June 1987.
- [35] B. Mulgrew and C. F. N. Cowan, *Adaptive Filters and Equalizers*, Kluwer Academic Publishers, Norwell, MA, 1988.
- [36] A. V. Oppenheim and R. W. Schaffer, *Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1975.
- [37] R. A. Roberts and C. T. Mullis, *Digital Signal Processing*, Addison Wesley, Reading, MA, 1987.
- [38] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, Second edition, Macmillan, New York, 1992.
- [39] M. G. Larimore, J. R. Treichler, and C. R. Johnson, "SHARF; An Algorithm for Adapting IIR Digital Filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 1622-1624, Nov. 1976.

- [40] E. H. Satorius and S. T. Alexander, "Channel Equalization using Adaptive Lattice Algorithms," *IEEE Trans. Commun.*, vol. COM-27, No. 9, pp. 899-905, June 1979.
- [41] B. Friedlander, "Lattice Filters for Adaptive Processing," *Proc. IEEE*, vol. 70, pp. 829-867, Aug. 1982.
- [42] M. L. Honig and D. G. Messerschmitt, *Adaptive Filter: Structures, Algorithms, and Applications*, Kluwer Academic Publishers, Hingham, MA, 1984.
- [43] B. Widrow, "Adaptive Filters," in *Aspects of Network and System Theory* ed. R. E. Kalman and N. Declaris, Holt, Rinehart & Winston, New York, 1971.
- [44] G. Ungerboeck, "Theory of the Speed of Convergence in Adaptive Equalizers for Digital Communications," *IBM J. Res. Dev.*, vol. 16, pp. 546-552, 1972.
- [45] J. E. Mazo, "On the Independence Theory of Equalizer Convergence," *Bell Syst. Tech. Journ.*, No. 58, pp. 963-993, May-June 1979.
- [46] I. J. Nagumo and A. Noda, "A Learning Method for System Identification," *IEEE Trans. Automatic Control*, vol. AC-28, pp. 282-287, 1967.
- [47] R. D. Gitlin, H. C. Meadors, and S. B. Weinstein, "The Tap-Leakage Algorithm: An Algorithm for the Stable Operation of a Digitally Implemented, Fractionally Spaced Adaptive Equalizer," *Bell Syst. Tech. J.*, vol. 61, pp. 1817-1839, Oct. 1982.
- [48] R. W. Harris, D. M. Chabries, and F. A. Bishop "A Variable Step (VS) Adaptive Filter Algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 309-316, Feb. 1986.

- [49] G. A. Clark, S. K. Mitra, and S. R. Parker, "Block Implementations of Adaptive Digital Filters," *IEEE Trans. Circuits and Systems*, vol. CAS-28, pp. 584-592, Jun. 1981.
- [50] M. L. Honig, "Echo Cancellation of Voiceband Data Signals Using Recursive Least Squares and Stochastic Gradient Algorithms," *IEEE Trans. Commun.*, vol. COM-33, No. 1, pp. 65-73, Jan. 1985.
- [51] C. W. K. Gritton and D. W. Lin, "Echo Cancellation Algorithms," *IEEE ASSP Magazine*, pp. 30-37, April 1984.
- [52] L. Guidoux and B. Peuch, "Binary Passband Echo Canceller in a 4800 Bit/s Two-Wire Duplex Modem," *IEEE J. Selected Areas in Comm.*, vol. SAC-2, No. 5, pp. 711-721, Sept. 1984.
- [53] A. Lewis, "Adaptive Filtering - Applications in Telephony," *BT Technol. J.*, vol. 10, No. 1, pp. 49-63, Dec. 1992.
- [54] P. T. Brady and G. K. Helder, "Echo Suppressor Design in Telephone Communications," *Bell Syst. Tech. J.*, vol. 42, pp. 2893-2917, 1963.
- [55] CCITT Recommendation G. 164, "Echo Suppressors."
- [56] D. G. Messerschmitt, "An Electronic Hybrid with Adaptive Balancing for Telephony," *IEEE Trans. Commun.*, vol. COM-28, No. 8, pp. 1399-1407, Aug. 1980.
- [57] B. E. Dotter, Jr., A. De La Plaza, D. A. Hodges, and D. G. Messerschmitt, "Implementation of an Adaptive Balancing Hybrid," *IEEE Trans. Commun.*, vol. COM-28, No. 8, pp. 1408-1416, Aug. 1980.
- [58] K. Murano, S. Unagami, and F. Amano, "Echo Cancellation and Applications," *IEEE Commun. Magazine*, vol. 28, No. 1, pp. 49-55, Jan. 1990.

- [59] O. Agazzi, D. A. Hodges, and D. G. Messerschmitt, "Nonlinear Echo Cancellation of Data Signals," *IEEE Trans. Commun.*, vol. COM-30, No. 11, pp. 2421-2433, Nov. 1982.
- [60] M. J. Smith, C. F. N. Cowan, and P. F. Adams, "Nonlinear Echo Cancellers Based on Transposed Distributed Arithmetic," *IEEE Trans. Cir. and Syst.*, vol. CAS-35, No. 1, pp. 6-18, Jan. 1988.
- [61] K. Yamazaki, S. Aly, and D. Falconer, "Convergence Behavior of a Jointly Adaptive Transversal and Memory-Based Echo Canceller," *IEE Proceedings-F*, vol. 138, No. 4, pp. 361-370, Aug. 1991.
- [62] N. A. M. Verhoeckx, H. C. Van Den Elzen, F. A. M. Sniijders, and P. J. van Gerwen, "Digital Echo Cancellation for Baseband Data Transmission," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, No. 6, pp. 768-781, Dec. 1979.
- [63] S. B. Weinstein, "A Passband Data-Driven Echo Canceller for Full-Duplex Transmission on Two-Wire Circuits," *IEEE Trans. Commun.*, vol. COM-25, No. 7, pp. 654-666, July 1977.
- [64] N. Holte and S. Stueflotten, "A New Digital Echo Canceler for Two-Wire Subscriber Lines," *IEEE Trans. Commun.*, vol. COM-29, No. 11, pp. 1573-1581, Nov. 1981.
- [65] V. G. Koll and S. B. Weinstein, "Simultaneous Two-Way Data Transmission over A Two-Wire Circuit," *IEEE Trans. Commun.*, vol. COM-21, No. 2, pp. 143-147, Feb. 1973.
- [66] K. H. Mueller, "A New Digital Echo Canceler for Two-Wire Full-Duplex Data Transmission," *IEEE Trans. Commun.*, vol. COM-24, No. 9, pp. 956-962, Sep. 1976.

- [67] J. M. Cioffi and J. J. Werner, "Effects of Biases on Digitally Implemented Data-Driven Echo Cancelers," *AT&T Techn. J.*, vol. 64, No. 1, pp. 115-138, Jan. 1985.
- [68] P. F. Adams, "Adaptive Filters in Telecommunications," ed. C. F. N. Cowan and P. M. Grant, Prentice-Hall, Englewood Cliffs, NJ, 1985.
- [69] O. Agazzi, D. A. Hodges, and D. G. Messerschmitt, "Large-Scale Integration of Hybrid-Method Digital Subscriber Loops," *IEEE Trans. Commun.*, vol. COM-30, No. 9, pp. 2095-2108, Sep. 1982.
- [70] K. Shenoi, *Digital Signal Processing in Telecommunications*, Prentice-Hall, Englewood Cliffs, NJ, 1995.
- [71] H. Fan and W. K. Jenkins, "An Investigation of an Adaptive IIR Echo Canceler: Advantages and Problems," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-36, No. 12, pp. 1819-1834, Dec. 1988.
- [72] E. R. Ferrara, Jr., "Frequency Domain Adaptive Filtering," ed. C. F. N. Cowan and P. M. Grant, Prentice-Hall, Englewood Cliffs, NJ, 1985.
- [73] F. Ling, "Echo Cancellation," ed. N. Kalouptsidis and S. Theodoridis, Prentice-Hall, Englewood Cliffs, NJ, 1993.
- [74] D. L. Duttweiler and Y. S. Chen, "A Single Chip VLSI Echo Canceller," *Bell Syst. Tech. J.*, vol. 59, No. 2, pp. 149-161, Feb. 1980.
- [75] J. J. Shynk, "Adaptive IIR Filtering," *IEEE ASSP Magazine*, vol. 28, No. 1, pp. 4-21, Apr. 1989.
- [76] C. R. Johnson, Jr., "Adaptive IIR Filtering: Current Results and Open Issues," *IEEE Trans. Inform. Theory*, vol. IT-30, No. 2, pp. 237-250, Mar. 1984.

- [77] G. Long, D. Shwed, and D. D. Falconer, "Study of a Pole-Zero Adaptive Echo Canceller," *IEEE Trans. Circuits Systems*, vol. CAS-34, No. 7, pp. 765-769, Jul. 1987.
- [78] T. A. C. M. Classen and W. F. G. Mecklenbrauker, "Comparisons of the Convergence of two Algorithms for the Adaptive FIR Digital Filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 670-678, June 1981.
- [79] H. Sari, "Performance Evaluation of three Adaptive Equalization Algorithms," in *Proc. ICASSP'82*, pp. 1385-1388.
- [80] C. J. Macleod, E. Ciapala, and Z. J. Jelonek, "Quantisation in non-recursive Equalisers for Data Transmission," *PROC. IEE*, Vol 122, No. 10, pp. 1105-1110, Oct. 1975.
- [81] P. Xue and B. Liu, "Adaptive Equalizer Using Finite-Bit Power-of-Two Quantizer," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1603-1611, December 1986.
- [82] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Trans. ASME J. Basic Eng.*, vol. 82D, pp. 35-45, March 1960.
- [83] R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction Theory" *Trans. ASME J. Basic Eng.*, vol. 83, pp. 95-108, 1961.
- [84] D. N. Godard, "Channel Equalization using a Kalman Filter for Fast Data Transmission," *IBM J. Res. Dev.*, vol. 18, pp. 267-273, 1976.
- [85] M. Morf, A. Vieira, and D. T. Lee, "Ladder Forms for Identification and Speech Processing," *Proc. IEEE CDC'77*, pp. 1074-1078.
- [86] M. Morf and D. T. Lee, "Recursive Least Squares Ladder Forms for Fast Parameter Tracking," *Proc. IEEE CDC'78*, pp. 1362-1367.

- [87] F. Ling and J. G. Proakis, "Numerical Accuracy and Stability: Two Problems of Adaptive Estimation Algorithms Caused by Round-Off Error," *Proc. IEEE ICASSP'84*, pp. 30.3.1-30.3.4.
- [88] F. Ling, D. Manolakis, and J. G. Proakis, "New Forms of LS Lattice Algorithms and an Analysis of their Round-Off Error Characteristics," *Proc. IEEE ICASSP'85*, pp. 1739-1742.
- [89] I. K. Proudler, J. G. McWhirter, and T. J. Shepherd, *QRD-Based Lattice Filter Algorithms*, SPIE, San Diego, Calif., 1989.
- [90] P. A. Regalia and M. G. Bellanger, "On the Duality between Fast QR Methods and Lattice Methods in Least Squares Adaptive Filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-39, pp. 879-891, 1991.
- [91] G. C. Goodwin and R. L. Payne, *Dynamic System Identification: Experiment Design and Data Analysis*, Academic Press, New York, 1977.
- [92] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1985.
- [93] O. Macchi and E. Eweda, "Second-Order Convergence Analysis of Stochastic Adaptive Linear Filtering," *IEEE Trans. Automat. Contr.*, vol. AC-28, pp. 76-85, Jan. 1983.
- [94] R. R. Bitmead, "Convergence in Distribution of LMS-Type Adaptive Parameter Estimates," *IEEE Trans. Automat. Contr.*, vol. AC-28, pp. 54-60, Jan. 1983.
- [95] W. A. Gardner, "Learning Characteristics of Stochastic-Gradient-Descent Algorithms: A general Study, Analysis and Critique," *Signal Processing*, vol. 6, pp. 113-133, 1984.

- [96] C. L. Nikias and A. P. Petropulu, *High-Order Spectra Analysis: A Nonlinear Signal Processing Framework*, Prentice-Hall, Englewood Cliffs, NJ, 1993.
- [97] A. Papoulis, *Probability, Random Variables, and Stochastic Processes, Second Ed.*, McGraw-Hill Publishing Co., New York, 1984.
- [98] A. Zerguine, C. F. N. Cowan, and M. Bettayeb, "Adaptive Echo Cancellation Using the L_2 Norm of both Near-End and Far-End Errors," *Proceedings of the First Communications Conference*, pp. 212-216, Muscat, Oman, Mar. 1996.
- [99] D. C. Shin and C. L. Nikias, "Adaptive Interference Canceller for Narrowband and Wideband Interferences Using Higher Order Statistics," *IEEE Trans. Signal Processing*, vol. SP-42, pp. 2715-2728, Oct. 1994.

Appendix A

Publications

- [1] A. Zerguine, C. F. N. Cowan, and M. Bettayeb, "Adaptive Echo Cancellation Using Statistical Shaping," *Twenty-ninth Annual, Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, California* Oct. 29-Nov. 1, 1995.
- [2] A. Zerguine, C. F. N. Cowan, and M. Bettayeb, "Adaptive Echo Cancellation Using the L_2 Norm of both Near-End and Far-End Errors," *Proceedings of the First Communications Conference*, pp. 212-216, Muscat, Oman, Mar. 1996.

