LMS-BASED ALGORITHMS FOR ECHO CANCELLATION IN TELECOMMUNICATIONS

by

PRASAD SRISTI
B.Tech., Indian Institute of Technology, Bombay, 1997

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

We accept this dissertation as conforming
to the required standard

[redacted]

Dr. A. Antoniou, Co-Supervisor, Dept. of Elect. & Comp. Eng.

[redacted]

Dr. W.-S. Lu, Co-Supervisor, Dept. of Elect. & Comp. Eng.

[redacted]

Dr. M. Serra, Member, Dept. of Computer Science

[redacted]

Dr. L. T. Bruton, External Examiner, University of Calgary

Supervisors: Dr. A. Antoniou and Dr. W.-S. Lu

ABSTRACT

The application of various popular least-mean-squares (LMS) algorithms to echo cancellation in telecommunication systems is studied. To start with, an analysis of the echo path impulse response in an HDSL2 environment is presented in order to provide a framework for studying the performance of the adaptation algorithms.

Then three categories of LMS algorithms, namely, subband, variable step-size, and block LMS adaptation algorithms are researched. An extensive study of the effects of the filter banks used on the performance of the various configurations is carried out through mathematical analysis and simulations. A new method of designing optimum filter banks for usage in subband adaptive filters is described and the superior performance of the designed filter banks over other types of filter banks is illustrated through simulations.

Various existing variable step-size adaptation algorithms are described and simulated and their performance is studied in different echo cancellation scenarios. A new variable step-size LMS algorithm is proposed and simulations are included to show that superior performance of the proposed algorithm is possible through the proposed algorithm in conventional as well as subband echo cancellation configurations with respect to existing variable step-size algorithms. Block adaptive filtering algorithms in time and frequency domain are then discussed and simulated.

The thesis concludes with comparisons between block LMS and conventional LMS adaptive filtering in terms of computational complexity and echo cancellation performance which are based on mathematical analysis and simulations.

Examiners:

[redacted]

Dr. A. Antoniou, Co-Supervisor, Dept. of Elect. & Comp. Eng.

[redacted]

Dr. W.-S. Lu, Co-Supervisor, Dept. of Elect. & Comp. Eng.

[redacted]

Dr. M. Serra, Member, Dept. of Computer Science

[redacted]

Dr. L. T. Bruton, External Examiner, University of Calgary

Table of Contents

List of Figures

List of Tables

Acknowledgement

I would like to thank my supervisors Dr. A. Antoniou and Dr. W.-S. Lu for their guidance and support throughout my association with UVic. I would also like to thank my friends and colleagues at PMC-Sierra Inc., Hamish Dobson and Stuart Robinson, who at various points in my career have encouraged me in my graduate work. Finally, this acknowledgement would be incomplete without the mention of the unconditional support of my parents.

Dedication

To my parents

List of Abbreviations

ADC	Analog to digital conversion
ADSL	Asymmetric digital subscriber line
BER	Bit error rate
BLMS	Block-LMS algorithm
CSA	Carrier service area
DAC	Digital to analog conversion
DSL	Digital subscriber line
FBLMS	Frequency domain block-LMS algorithm
FFT	Fast Fourier transform
FIR	Finite impulse response
FSS	Fixed step size
FTTC	Fiber to the curb
HDSL	High bit-rate digital subscriber line
HDSL2	High bit-rate digital subscriber line, generation 2
IFFT	Inverse fast Fourier transform
ITU	International telecommunication union
LMS	Least mean squares
MSE	Mean squares error
MVSS	Modified variable step size
SHDSL	Single-line high bit-rate digital subscriber line
SNR	Signal to noise ratio
VDSL	Very high bit-rate digital subscriber line
VSS	Variable step size

Chapter 1

Introduction

1.1 Introduction

The world-wide subscriber loop plant for telephone service has grown enormously to encompass billions of miles of copper wires. This copper plant remains one of the world's greatest and most valued technological asset. Although fiber deployment continues to make greater bandwidth available, investment in existing copper wire ensures that wire will still be an important communications medium during the transition to optical fiber over the next few decades. The telecommunication industry, eager to extend the useful lifetime of the existing copper plant and at the same time provide high-speed digital servies, has a keen interest in the deployment and implementation of technologies that allow low-cost high-speed services over the copper loop plant.

An important wideband service is the T1 service normally used by businesses for voice multiplexing and digital data transmission. It uses 24 voice channels each operating at 64 Kbps thus providing a data rate of 1.544 Mbps. To overcome the signal loss resulting from attenuation in long cables, repeater sections are needed for every 6000 ft of 22-gauge cable. Moreover, in some cases, bridged taps must be removed in the lines and pair selection needs to be done. As clearly outlined in [1], these engineering requirements can result in long lead times for the installation of T1 service and can be expensive.

The recently emerging digital subscriber line (DSL) techologies enable near T1-rate services over existing unconditioned, repeaterless subscriber lines, thereby circumventing the problems mentioned above. In the various DSL systems, which are commonly referred to as xDSL systems, the procedure of bridge tap removal and re-

peater installation are eliminated by improving the performance of the transcievers. This improvement has been possible by technological advances in VLSI circuit design as well as the efficient use of digital signal processing algorithms [2]. The various existing and emerging xDSL techologies are described briefly in the next section.

1.2 Overview of xDSL Technologies

1.2.1 ADSL

Asymmetric digital subscriber line (ADSL) technology converts an existing telephone line into an access path for multimedia and high-speed data communications. In this technology most of the bandwidth is devoted to the downstream direction, sending data to the user. Only a small portion of bandwidth is available for upstream or user-interaction messages. Such asymetric network traffic is inherent in graphics intensive Internet usage. ADSL operates at up to 6.1 Mbps in the downstream direction and up to 640 Kbps in the upstream direction. The presence of a frequency *splitter* enables low-bandwidth voice traffic to be present on the same line. ADSL provides dedicated bandwidth to the end user. Unlike Internet over cable service which requires a user to share the same bandwidth with other users and is dependent on the network traffic, ADSL provides dedicated bandwidth to the end user.

1.2.2 ADSL Lite

ADSL Lite is a sub-rated ADSL solution that entails reduced digital signal processing capabilities relative to those of the full-rate ADSL system. The ADSL Lite specification is now in final review by the International Telecommunications Union (ITU) for standardization under the name of g.lite. Because of the absence of the frequency splitter, which reduces the installation time at the customer premises, ADSL Lite is seen as the key to mass deployment of ADSL services.

1.2.3 HDSL

The earliest variation of DSL is the high bit-rate DSL (HDSL) which is used for wide-band digital transmission within a corporate site and between a telephone company

and the customer. The main characteristic of HDSL is that the data transmission is symmetrical with an equal amount of bandwidth available in both the upstream and downstream direction. The HDSL system uses two twisted-pair copper lines to provide a duplex data rate of 1.544 Mbps, thus achieving the T1 rate on two twisted-pair lines.

1.2.4 HDSL2

High bit-rate DSL, generation 2 (HDSL2) is an advanced technology with the ability to support full T1 service deployment over a single pair of copper wires. Not only does HDSL2 have all the advantages of HDSL, namely, lower cost and longer reach, it also offers the same features and spectral compatibility on a standards-based solution with multiple vendor interoperability on a single pair thus increasing the available bandwidth on a single twisted-pair by a factor of 2. The performance of HDSL2 allows service providers to offer guarenteed symmetric upstream and downstream data rates of 1.544 Mbps, at a *bit error rate* (BER) of 10^{-7}. In addition, HDSL2 is spectrally friendly allowing it to coexist with ADSL and other services. The HDSL2 system development is one of the most challenging in terms of the DSP technologies involved because of the need to handle echoes, intersymbol interference, crosstalk, and the attenuation of a 12,000 ft line.

1.2.5 SHDSL

Single-line high bit-rate DSL (SHDSL) is a multirate version of HDSL2 and is driven by the G.shdsl standard which was finalized in April 2000. SHDSL tranceivers are capable of symmetric data rates from 192 Kbps to 2306 Kbps in steps of 8 Kbps. The rate of transmission is decided at startup and is negotiated between the central office and the remote transceivers using a handshake procedure. SHDSL is poised to become the most popular of symmetric DSL solutions because of its versatility.

1.2.6 VDSL

Very high bit-rate DSL (VDSL) is intended to provide very high asymmetrical bandwidth - up to 52 Mbps in the downstream direction and up to 2 Mbps in the upstream

direction. In some sense, VDSL can be considered as next generation ADSL and is proposed to meet the broadband access requirements over a *fiber to the curb* (FTTC) network, addressing the last section of copper cabling to the customer premises. Typical distance and implementation of VDSL is 3000 ft at 26 Mbps upstream data rate. VDSL standards are still in the process of evolution. The features of the various network access services are summarized in Table 1.1.

Table 1.1. *Features of various network access services.*

Service	Downstream rate	Upstream rate	Copper pairs	Distance to CO in feet
ADSL	1.5 - 6.1 Mbps	16 - 640 Kbps	1 copper pair	18,000
HDSL	1.544 Mbps	1.544 Mbps	2 copper pairs	12,000
HDSL2	1.544 Mbps	1.544 Mbps	1 copper pair	12,000
SHDSL	192 - 2360 Kbps	192 - 2360 Kbps	1 copper pair	11,500 - 22,000
VDSL	12.96 - 55.2 Mbps	1.5 - 6 Mbps	1 copper pair	1,000 - 4,500

1.3 Echo Cancellation

A certain amount of echo was considered as a positive feature for voice applications in the past. However, with the advent of satellite communications, the long delays introduced make echos an unpleasant experience. Voice echo cancellers are usually installed in central offices to significantly reduce the magnitude of the echo. For voice applications, a reduction in the level of echo between 20 to 30 dB is typically required [3]. However, full-duplex data transmission over a single pair of twisted loops can be realized only with the use of echo cancellation and hence the echo canceller forms an important part of the transceiver [4].

The echo for DSL systems occurs at the hybrid circuit where the transmit and the receive paths are joined together and connected to the twisted-pair loop [5]. Since the received signal strength is relatively low because of the high attenuation caused due to the typically long loops, the echo strength can be much higher than the received signal strength and hence the required reduction in the echo level can be as high as 60

dB. The hybrid circuit can be designed to provide certain amount of echo attenuation. However, because of the large range of loop impedances, digital adaptive filters need to be used to cancel out the echo. The echo canceller is required to be in parallel with the echo path as shown in Fig. 1.1 and it simulates the echo path including the digital to analog converter (DAC), the transmit filter, the hybrid circuit, the receiver filter, and the analog to digital converter (ADC). The hybrid circuit is introduced in greater detail in Chapter 2. An echo canceller is intended to produce an echo replica with the use of the transmit data.

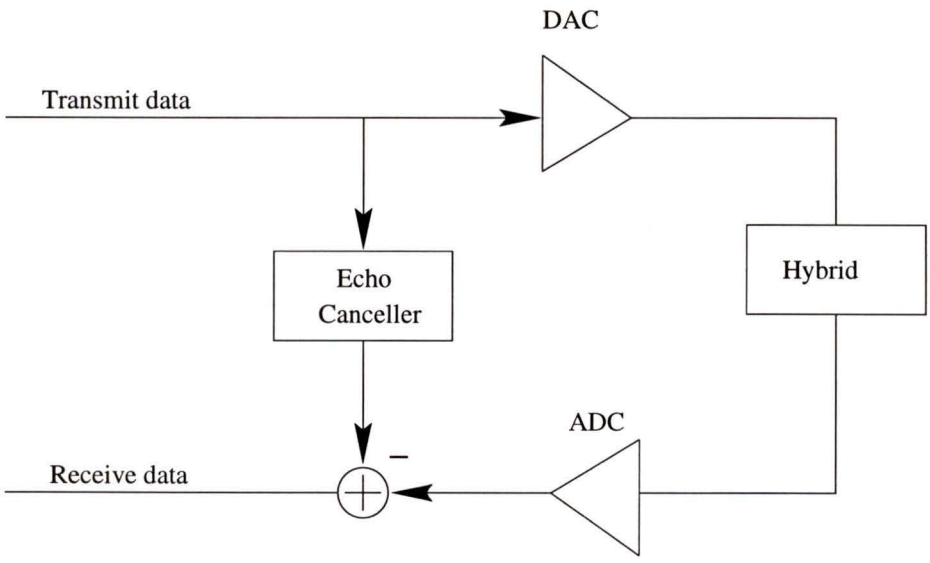

Figure 1.1. *Adaptive echo cancellation for duplex communication using a hybrid circuit.*

1.4 Adaptive Filters

An adaptive filter is used when the statistical characteristics of the data to be processed are not known a priori. In such a situation, the adaptive filter relies on a recursive algorithm for its operation, which makes it possible for the filter to perform satisfactorily in an environment where complete knowledge of the relevant signal characteristics is unknown. The algorithm starts from a set of initial conditions and converges to the optimum Weiner solution.

As a direct consequence of the application of a recursive algorithm whereby the parameters of the filter are updated in every iteration, the parameters become data dependant. Therefore, an adaptive filter is a nonlinear device in the sense that it does not obey the principle of superposition. However, adaptive filters are commonly classified as linear or nonlinear depending upon the procedure used to update the filter parameters. An adaptive filter is said to be *linear* if the estimate of the parameter of interest is calculated as a linear combination of the available information. Otherwise, the adaptive filter is said to be nonlinear.

A wide variety of recursive algorithms have been described in the literature for the operation of linear adaptive filters. The choice of one algorithm over another is mainly determined by one or more of the following factors:

Rate of convergence: This is defined as the number of iterations required for the algorithm to converge *close enough* to the optimum Weiner solution in the mean-square sense in response to stationary inputs. A fast rate of convergence allows the algorithm to adapt rapidly to a stationary environment of unknown statistics.

Misadjustment: For an algorithm of interest, this parameter provides a quantitative measure of the amount by which the final value of the mean-squared error, averaged over an ensemble of adaptive filters, deviates from the minimum mean squares error produced by the Weiner filter.

Computational requirements: The issues of concern for the computational requirements include the number of arithmetic operations such as multiplications, divisions, additions, and subtractions to complete one iteration of the algorithm. The size of memory required to store the data and the program can also be important depending on whether the algorithm is being implemented in hardware or software.

Structure: This refers to the structure of information flow in the algorithm, determining the manner in which it is implemented in hardware form. An algorithm whose structure exhibits high modularity and repetitive configurations is well suited for implementation using very large-scale integration (VLSI) circuits.

Tracking: When an adaptive algorithm operates in a nonstationary environment, the algorithm is required to track statistical variations in the environment. The

tracking performance of the algorithm, however, is influenced by the rate of convergence and steady-state fluctuations or misadjustment.

Robustness: For an adaptive filter to be robust, small disturbances can only result in small estimation errors. The disturbances may arise from a variety of factors, internal or external to the filter.

One such algorithm which is favored because of its computational requirements, structure, and robustness is the *least-mean-squares* (LMS) algorithm, which is based on the method of steepest descent, a well-known technique in optimization theory. In this work, we are mainly concerned with the performance of different variations of the LMS algorithm as applied to echo cancellation.

1.5 Contributions and Organization of the Thesis

The thesis is organised as follows. In Chapter 2, the basic configuration of an echo canceller is illustrated and the process of deriving the echo path impulse response for HDSL2 CSA test loops is described and simulated. The LMS algorithm is also introduced in this chapter. The rest of Chapter 2 deals with LMS subband echo cancellation configurations and role of the filter banks in subband echo cancellation. The structure of the LMS subband echo canceller with cross-adaptive filters is described. Simulations results to compare the rates of convergence of an LMS-based full-band echo canceller and an LMS-based subband echo canceller are described. Both echo cancellation configurations result in the residual echo converging to the same value. However, the subband echo canceller shows a slower rate of convergence because of the presence of cross-adaptive filters. Cross-adaptive filters are necessary in a subband echo canceller to cancel the aliasing caused by the filter banks used. In order to eliminate the cross-adaptive filters, the filter banks used in a subband echo canceller need to result in minimal aliasing. The problem of optimum filter bank design for use in subband echo cancellation is thus formulated. An approach leading to a new solution to the formulated problem is also explained in Chapter 2. The performance of the optimized filter bank designed using the new approach is studied using simulations of subband echo cancellation. The optimized filter bank is shown to offer improved performance in terms of residual echo over other filter banks. The subband

echo cancellation system using the optimized filter banks and no cross-adaptive filters also converges faster than a system that uses cross-adaptive filters.

Chapter 3 focuses on variable step-size adaptive algorithms and their performance in various scenarios with special application to xDSL echo cancellation. Two existing variable step-size adaptation LMS algorithms, namely, variable step-size (VSS) and modified variable step-size (MVSS), are described and simulated in conventional full-band echo cancellation configurations and a performance comparison is given. The MVSS algorithm is shown to have a faster rate of convergence over the VSS algorithm in low SNR scenarios. However, the MVSS algorithm is shown to result in a larger final misadjustment in the case of long adaptive filters. In order to overcome this problem of large misadjustment, a new VSS algorithm based on appropriate modifications to the existing VSS algorithms is proposed in Chapter 3. The new proposed algorithm is shown to reduce the steady-state misadjustment over the MVSS algorithm while maintaining the faster rate of convergence over the VSS algorithm. The proposed algorithm is also simulated in a scenario where an abrupt change occurs in the echo path impulse reponse and has been shown to converge to the new echo path impulse response coefficients. Such performance is a required characteristic of variable step-size adaptation algorithms. Chapter 3 also deals with the application of the proposed algorithm to subband LMS adaptive filtering. Simulation results are provided to demonstrate that the proposed algorithm yields a lower steady-state misadjustment in each subband when compared to a fixed step-size LMS algorithm.

In Chapter 4, block LMS adaptive filtering algorithms, which are typically employed to reduce computation complexity of the conventional LMS algorithm, are described. Two types of block-LMS algorithms in the frequency domain, namely, the constrained frequency-domain block-LMS algorithm (CFBLMS) and the unconstrained frequency-domain block-LMS algorithm (UFBLMS), are described and simulations are performed in echo cancellation scenarios for comparison of rate of convergence and residual echo. The CFBLMS and UFBLMS algorithms are found to require different step sizes and are shown to converge to the same residual echo. The UFBLMS algorithm, however, shows a slightly degraded rate of convergence over the CFBLMS algorithm. Chapter 4, deals also with time-domain block adaptive filtering algorithms and relevant simulations. Two popular LMS algorithms employing partial

coefficient updating schemes, the periodic and sequential LMS algorithms, are explained and simulated for comparison of the rate of convergence and computational complexity. It is shown that the rate of convergence of both algorithms is reduced by approximately the same factor by which the computational complexity is reduced. A time-domain inexact LMS algorithm employing partial updates is then introduced. This algorithm is shown to bring about a large reduction in the computational complexity of an echo canceller in the absence of a receive signal without a corresponding decrease in the rate of convergence.

Finally, conclusions from the work are summarized in Chapter 5 along with some suggestions for future work.

Chapter 2

Subband Adaptive Filtering for Echo Cancellation

2.1 Introduction

Echoes are a serious problem in data and voice transmission since they affect the quality of the received signal. In voice transmission, the echo is caused by the round-trip delays while in data transmission, the main source of echo is the leakage of the transmit signal to the receiver, as explained in Chapter 1. Adaptive filtering has been successfully used since the 1960's in cancelling echoes. Typically adaptive filter is used to model the unknown echo path in a communication channel and then as far as possible cancel the echo signals. A typical system used for echo cancellation is shown in Fig. 2.1. It is the same as the configuration used for system identification since echo cancellation is a specific case of system identification where one tries to estimate the transfer function of the echo path.

The objective of the system shown in Fig. 2.1 is to cancel echo $y(n)$ caused by signal $x(n)$, where W* represents the echo path and W is an FIR filter that attempts to model the echo path using $x(n)$ and error signal $e(n)$. This chapter is concerned mainly with the study of *subband* echo cancellation and its performance in comparison to conventional full-band echo cancellation. The chapter is organized as follows.

In order to study the performance of an echo cancellation system, a model of the echo path and the adaptive algorithm are needed. In Section 2.2, we describe the echo path in a single-line high bit-rate digital subscriber line (HDSL2) environment. Some of these echo path models are used in simulations presented later in the thesis.

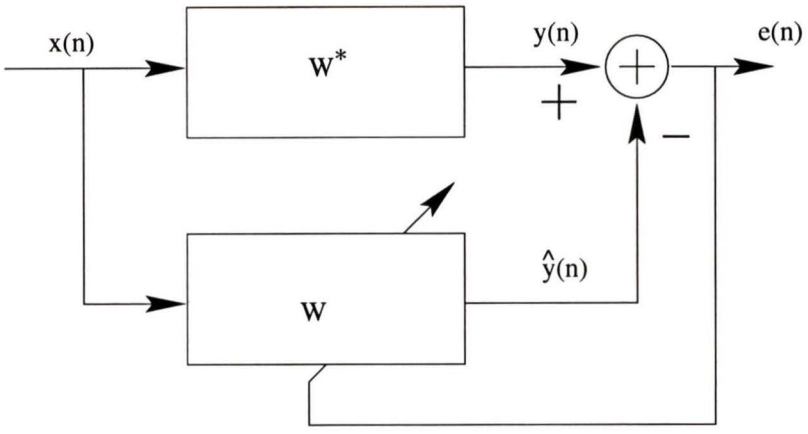

Figure 2.1. *Echo cancellation.*

Section 2.3 describes the structure of a subband echo canceller and the algorithm used in subband echo cancellation. Simulation results are presented and the performance of a subband echo canceller is compared with that of the conventional full-band echo canceller.

In Section 2.4, we formulate the problem of designing the filter banks to be used in subband echo cancellation and propose a new method of designing optimum orthogonal filter banks. Simulation results are presented to demonstrate that improved performance can be achieved in an echo canceller through the use of optimally designed filter banks.

2.2 Echo Path Modeling for HDSL2 Environment

The HDSL2 system uses a single twisted pair copper loop in full-duplex mode to achieve the required bit rate of 1.544 Mbps over the full *carrier serving area* (CSA). The full-duplex scheme uses a four-wire to two-wire conversion hybrid for simultaneous transmission in both directions on the two-wire pair. In effect, the hybrid circuit is used to match the input impedance of the loop to which it is connected. Due to the variation in loop makeups, it is impossible to design a hybrid circuit that will provide an exact match with every possible loop used for providing HDSL2 service. Therefore, part of the transmitted signal will appear at the receiving circuit as echo,

thus necessitating the use of an echo canceller [6, 7]. To investigate the performance of the echo canceller system in the HDSL2 environment, we need to model the echo path.

The model of a hybrid circuit used in voice-band modem applications is shown in Fig. 2.2. It is simply an electrical bridge with one branch being a twisted pair loop coupled through a line transformer. A line transformer model is shown in Fig. 2.3 and the parameters used for simulating this model are listed in Table 2.1. These parameter values are taken from reference [8].

The transfer function of the echo path $H(s)$ is given by

$$\frac{V_R}{V_T} \tag{2.1}$$

where V_R and V_T are the voltages shown in Fig. 2.2, and can be calculated as

$$V_R = V_1 - V_2 = \frac{10Z_bal}{10Z_{bal} + 10Z_0}V_T - \frac{Z_{in}}{Z_{in} + Z_0}V_T \tag{2.2}$$

Hence

$$H(s) = \frac{Z_{bal}}{Z_0 + Z_{bal}} - \frac{Z_{in}}{Z_0 + Z_{in}} \tag{2.3}$$

where Z_{bal} is the balance impedance (110 ohms), Z_{in} is the input impedance of the loop reflected through the line transformer as seen from the bridge side, and Z_0 is the series impedance of the line transformer.

From Eq. (2.3) we can see that once the parameters of the line transformer are fixed, the echo path transfer function is entirely dependant on the input impedance of the loop. The HDSL2 draft standard lists 8 CSA test loops. These are illustrated in Fig. 2.4 where AWG denotes the *American wire gauge*. In this section we derive the echo path transfer functions for these 8 CSA loops.

Table 2.1. *Parameters for the line transformer*

R_p	0.85 Ω
L_p	3 mH
R_c	10 kΩ
R_s	0.85 Ω
L_l	2 μH

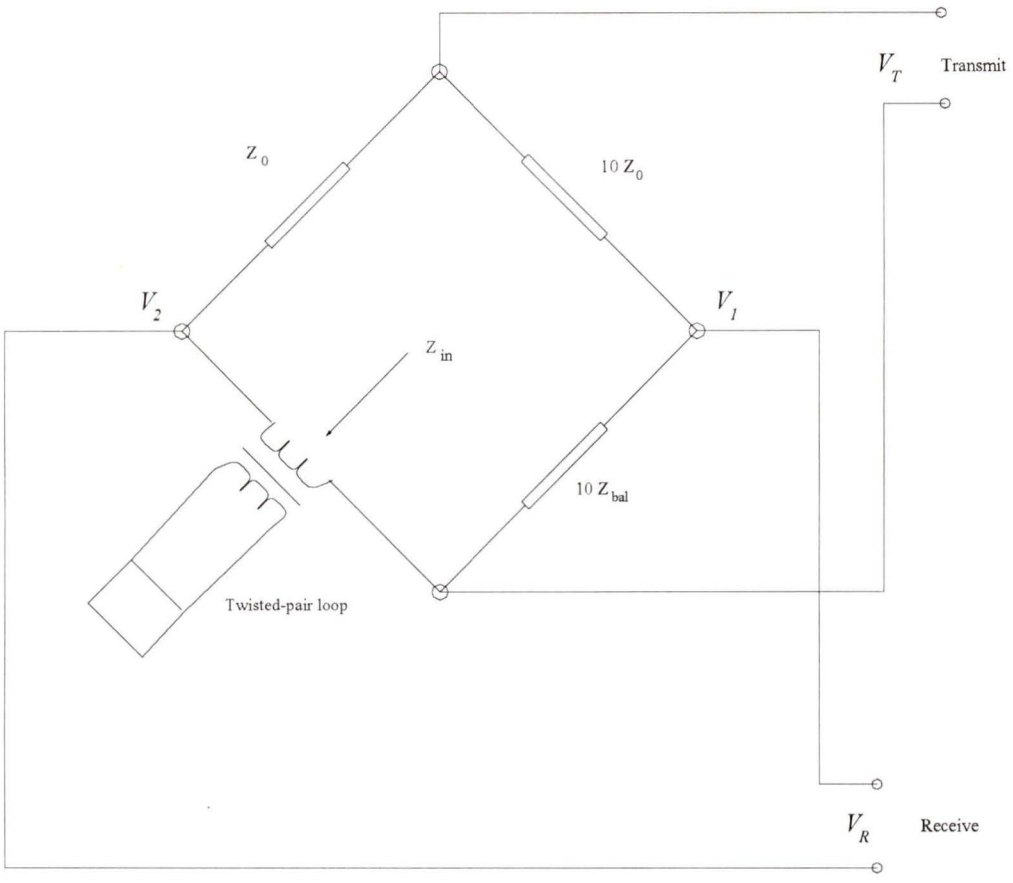

Figure 2.2. *Equivalent hybrid circuit.*

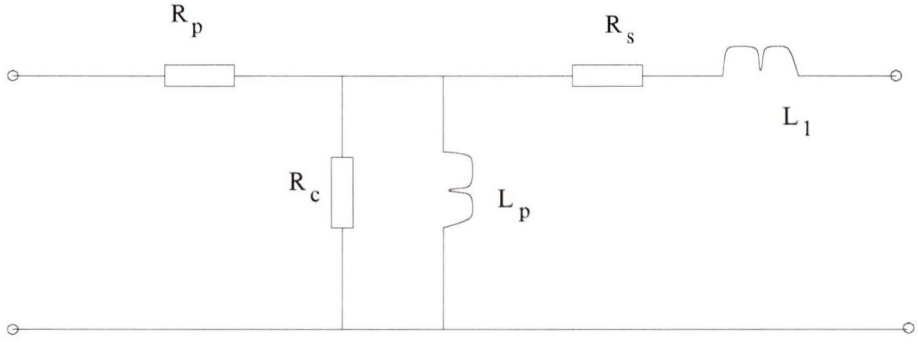

Figure 2.3. *Transformer equivalent circuit.*

The input impedance of the loop is calculated using the frequency-dependent ABCD or chain parameters. These parameters are obtained using the propagation constants of the specific gauge cable. The loop can be made up of different sections

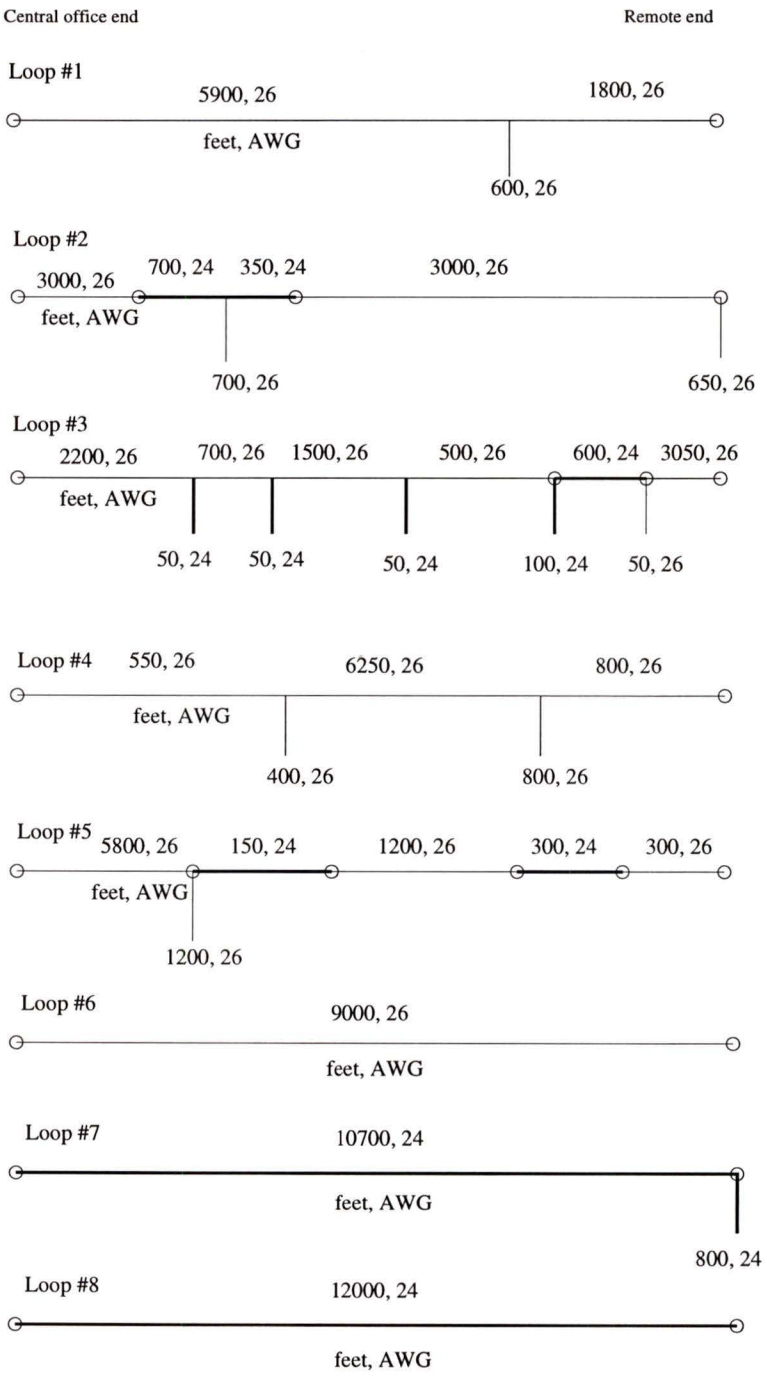

Figure 2.4. *CSA test loops for HDSL2.*

of different gauges and bridged taps at various points along the loop. The use of the ABCD parameters is convenient because the ABCD representation for the entire cable can be readily obtained by multiplying the chain matrices of the individual cascaded sections of the cable.

When the ABCD parameters A_l, B_l, C_l, D_l of the entire cable with the line transformers at either end are determined, the input impedance of the loops can be calculated as

$$Z_{in} = \frac{A_l + \frac{B_l}{Z_l}}{C_l + \frac{D_l}{Z_l}} \tag{2.4}$$

where Z_l is a resistive load impedance of 110 ohms. These values can be plugged into Eq. (2.3) to obtain the echo path transfer function.

An alternative and simpler way of computing the echo path transfer function, $H(s)$, is given in [8]. According to this method,

$$H(s) = H_{eco}(s)\frac{1 + \frac{Z_n}{Z_i}}{1 + \frac{Z_d}{Z_i}} \tag{2.5}$$

where $H_{eco}(s)$ is the echo path transfer function without any loop, Z_n is said to be the *null impedance* at the loop connection, Z_d is said to be the *zero impedance* at the same loop connection, and Z_i is the input impedance of the particular CSA loop. The exact formulas for determining $H_{eco}(s)$, Z_n, and Z_d are given in [8], and Z_i has been calculated using the ABCD parameters of the 8 CSA test loops. The real and imaginary parts of each impedance Z_i as seen at the central office end and the remote end are plotted against frequency in Figs. 2.5-2.8.

The impulse response of the echo path can then be obtained by taking the inverse Fourier transform of the echo-path transfer functions. The impulse responses for the 8 CSA loops at the central office end and the remote customer end are plotted in Figs. 2.9-2.12.

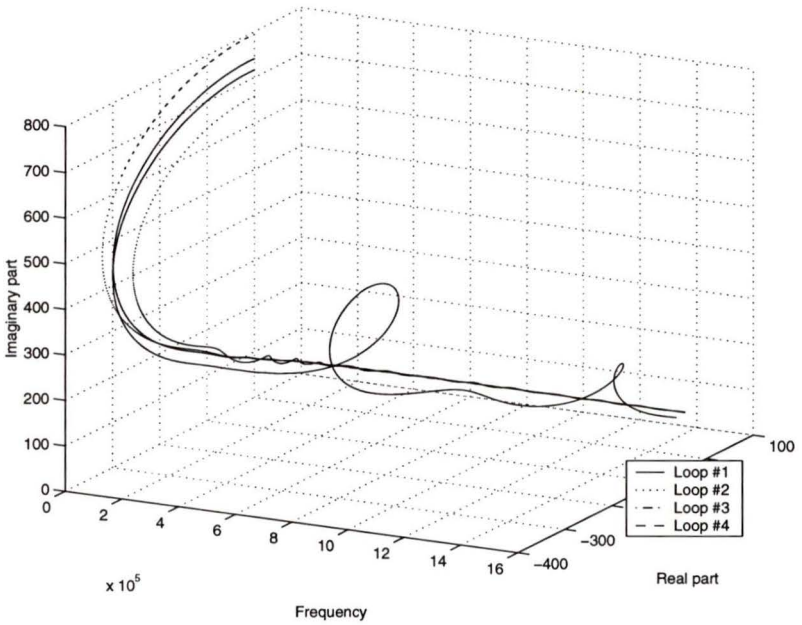

Figure 2.5. *CSA loop impedances: Central office end.*

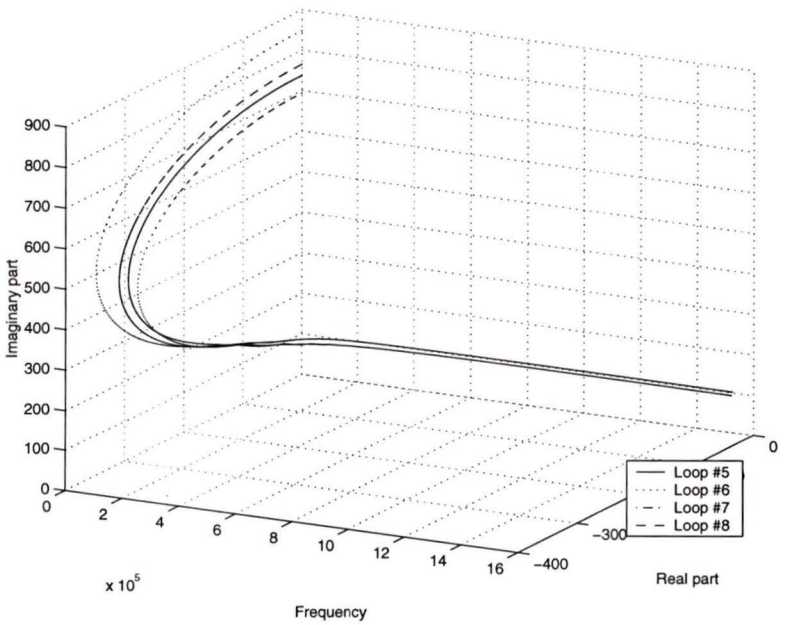

Figure 2.6. *CSA loop impedances: Central office end.*

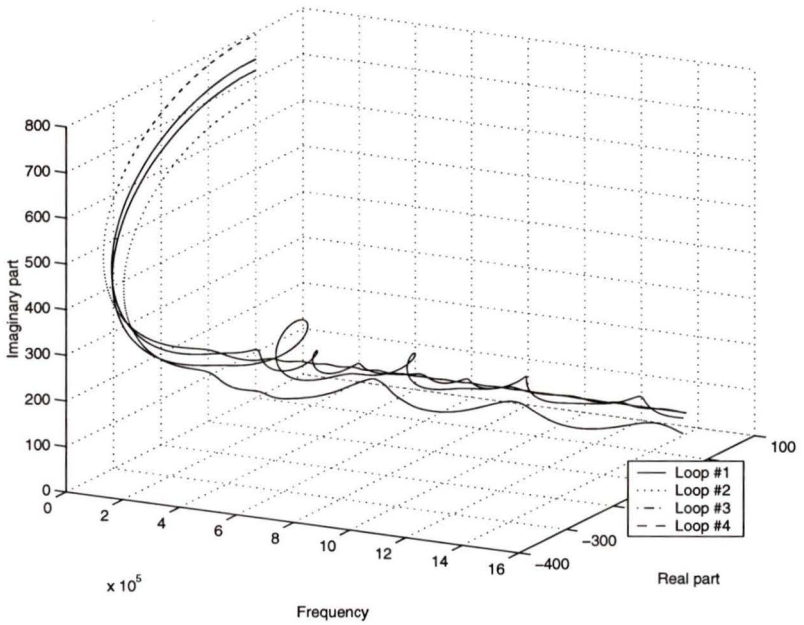

Figure 2.7. *CSA loop impedances: Remote end.*

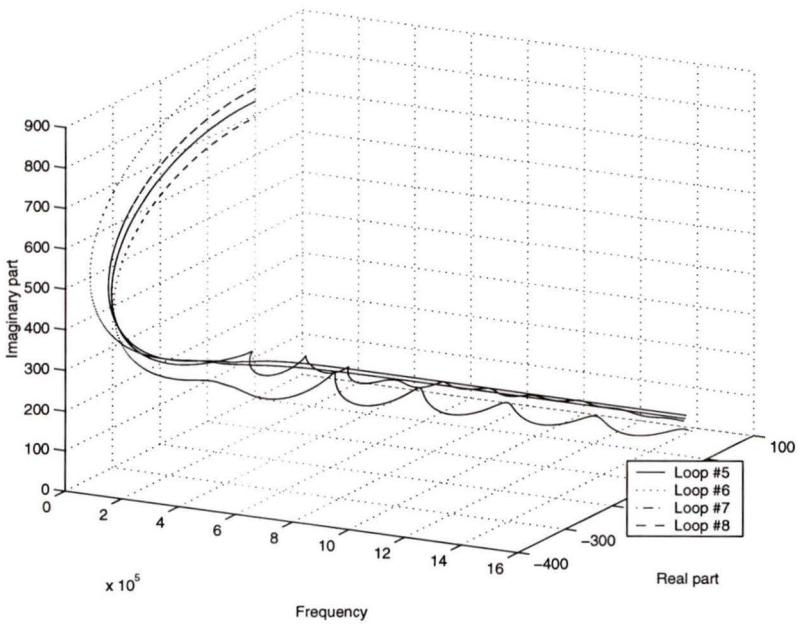

Figure 2.8. *CSA loop impedances: Remote end.*

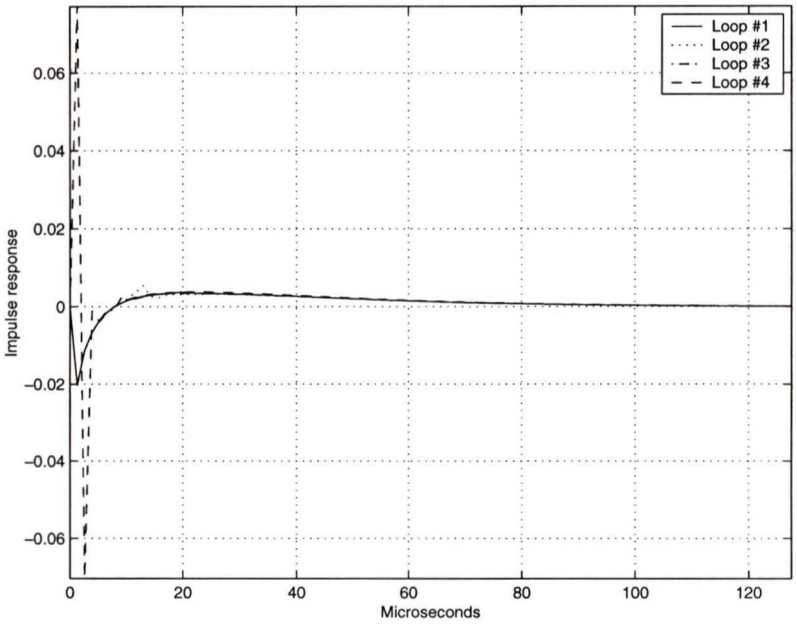

Figure 2.9. *Echo path impulse responses: Central office end.*

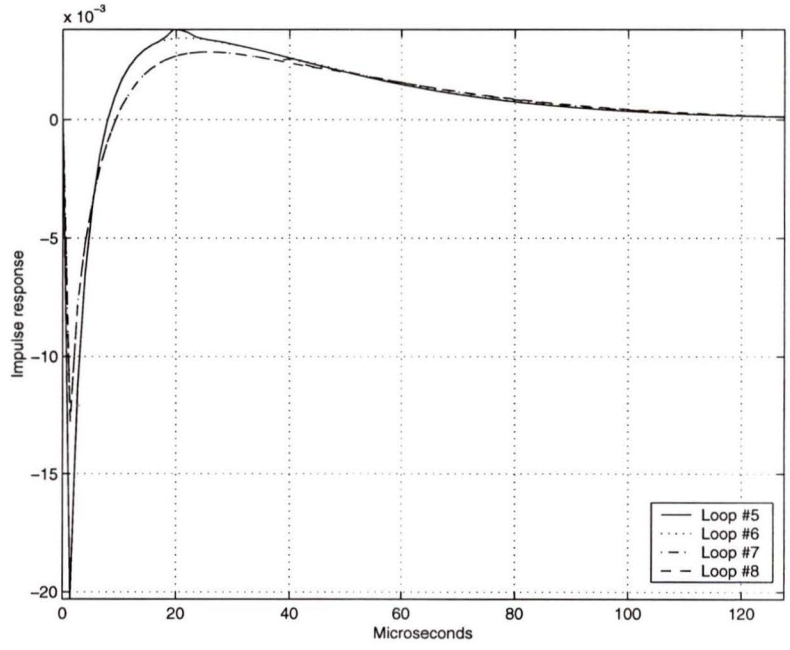

Figure 2.10. *Echo path impulse responses: Central office end.*

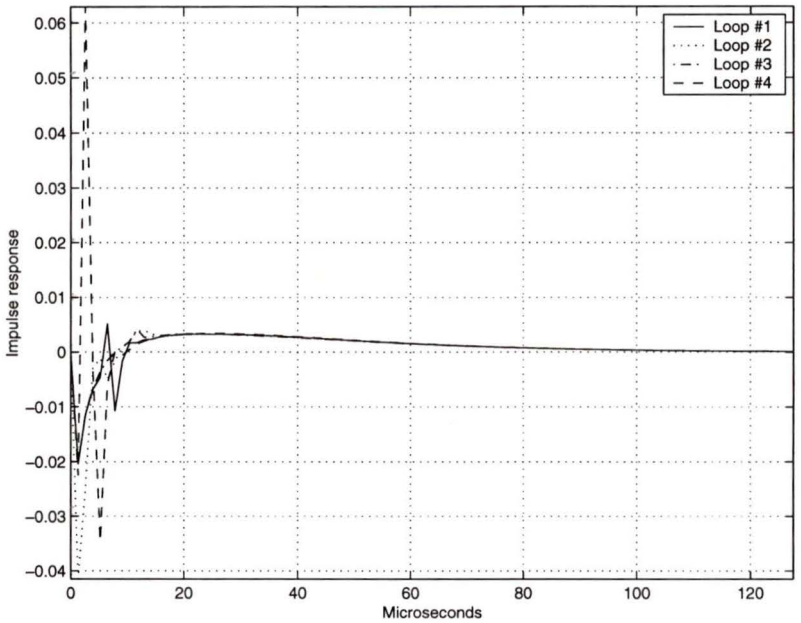

Figure 2.11. *Echo path impulse responses: Remote end.*

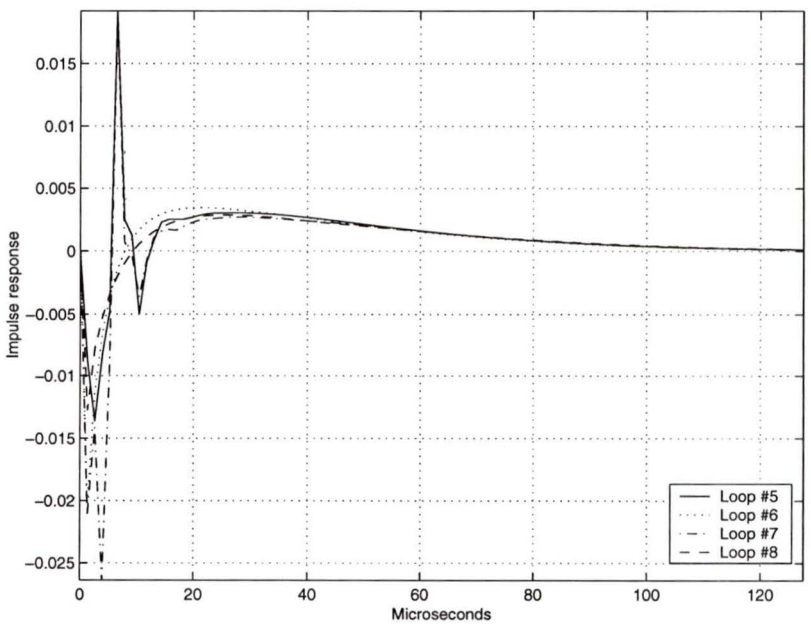

Figure 2.12. *Echo path impulse responses: Remote end.*

2.3 Adaptive Filter Structures and Algorithms

As mentioned in Section 2.2, the echo path can be modelled using an FIR filter of length N. One of the most popular performance measures for adaptive filters is the *mean-square error* (MSE) defined by

$$\begin{aligned} \xi(n) &= E[e(n)e^*(n)] \\ &= E\{(y(n)-\hat{y}(n))(y(n)-\hat{y}(n))^*\} \\ &= E[|y(n)|^2] - \mathbf{w^T p} - \mathbf{p^T w} + \mathbf{w^T R w} \end{aligned} \tag{2.6}$$

where $\mathbf{w}$ is a column vector of the adaptive filter coefficients, $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$ is the input autocorrelation matrix, $\mathbf{p} = E[\mathbf{x}(n)y(n)]$ is the crosscorrelation vector, and $\mathbf{x}(n) = [x(n-N+1) \ \cdots \ x(n-1) \ x(n)]^T$ is the input vector.

We now consider an algorithm for adjusting the adaptive-filter coefficients so as to minimize the MSE. The minimum error occurs at the point at which the gradient (first-order derivatives of the MSE) with respect to the filter coefficients is equal to zero. Taking the gradient of the MSE in (2.6), we have

$$\bigtriangledown [\xi(n)] = 2\mathbf{Rw} - \mathbf{2p} \tag{2.7}$$

Assuming that $\mathbf{R}$ is nonsingular, we can then set the gradient to zero and solve for the optimum filter coefficients as

$$\mathbf{w_{opt}} = \mathbf{R^{-1}p} \tag{2.8}$$

where $\mathbf{R^{-1}}$ is the inverse of $\mathbf{R}$. Thus, if $\mathbf{R}$ is known, we can evaluate $\mathbf{w_{opt}}$ using (2.6). This is called the Wiener filter solution. However, in many cases like ours, either $\mathbf{R}$ is unknown or is time varying and thus a direct solution for $\mathbf{w}$ may not be feasible or may not be optimal at all times. In these cases it is desirable to develop an adaptation algorithm for updating $\mathbf{w}$ that can start at any initial value and converge to $\mathbf{w_{opt}}$.

2.3.1 LMS Algorithm

Gradient-based algorithms use gradient information to determine the location of the minimum. These algorithms are based on the recursive relation

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \delta \bigtriangledown \xi(n) \tag{2.9}$$

where δ is a convergence-related parameter that controls the step size taken at each iteration along the performance surface.

In real-time applications, we do not know $\mathbf{R}$ and $\mathbf{p}$, and in addition they may be changing with time. Thus we use an estimate of the gradient instead.

The classical *least-mean square* (LMS) algorithm [9, 12] uses the *instantaneous* estimate of the gradient. The principle behind the LMS algorithm is to ignore the expectation operator in the computation of $\bigtriangledown\xi(n)$. That is, we let

$$\begin{aligned} \bigtriangledown\xi(n) &= \bigtriangledown e(n)^2 \\ &= -2\, e(n)\mathbf{x}(n) \end{aligned} \tag{2.10}$$

Hence from Eqs. (2.7) and (2.8), we have

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}(n) \tag{2.11}$$

where μ is known as the step size and the performance of the adaptive filter is greatly dependant on the choice of μ. If μ is not chosen small enough, then the filter coefficients may fail to converge to the optimum values; however, if it is chosen too small, then the filter coefficients may exhibit slow convergence [9, 10]. The LMS algorithm and the influence of the choice of the step-size μ on the convergence of the algorithm is considered in detail in Chapter 3.

In the next section we describe the use of the LMS algorithm in subband adaptive filtering.

2.3.2 Adaptive Filtering in Subbands

Because of its simplicity, the classical LMS algorithm has been used extensively in the past. However, in the case of cancellation of long echoes, slow convergence limit its performance.

These problems have motivated a new approach in which adaptive filtering is performed in subbands for the purpose of reducing the computational complexity and improving the convergence speed.

2.3.2.1 Structure of a Subband Echo Canceller

In subband adaptive filtering, the input and output signals are split into adjacent frequency subbands by analysis filter banks; then each subband signal is subsampled, and an adaptive filtering algorithm is applied to the subband signals. Subsampling leads to a greater computational efficiency than the usual full-band scheme since all the required multiplications and additions can be done at a reduced rate.

The structure of a subband echo canceller is illustrated Fig. 2.13. Let $s(n)$ be the received signal which consists of the echo $y(n)$ generated due to input $x(n)$ filtered by the echo path E, and the signal transmitted from the far end $u(n)$. The aim of the echo canceller is to eliminate the echo $y(n)$ from the received signal $s(n)$.

Fig. 2.13 shows a two-band, one-level subband decomposition scheme in which both the input signal $x(n)$ and the received signal $s(n)$ are decomposed into adjacent frequency bands using the analysis filters H_0 and G_0 represented by the transfer functions $H_0(z)$ and $G_0(z)$. This results in two sets of signals, $\{s_0(n), x_0(n)\}$ at the outputs of filters H_0, and $\{s_1(n), x_1(n)\}$ at the outputs of filters G_0. In this scheme, a pair of adaptive filters is used to cancel the echo in each set of signals. Thus, adaptive filter 1 and the cross adaptive filter 3 are used to cancel the echo in $s_0(n)$ using $x_0(n)$ and $x_1(n)$, while adaptive filter 2 and cross adaptive filter 4 are used to cancel the echo in $s_1(n)$ using $x_1(n)$ and $x_0(n)$. The resultant error signals $e_0(n)$ and $e_1(n)$ pass through the synthesis filters H_1 and G_1 represented by the transfer functions $H_1(z)$ and $G_1(z)$, resulting in $e(n)$ which is an estimate of $u(n)$.

2.3.2.2 Simulation

It has been proved in [13] that with the use of the configuration shown in Fig. 2.13 for echo cancellation, the MSE approaches zero asymptotically. This has been verified by simulation. The echo path was simulated using the impulse response of CSA loop #2 at the central office end. The impulse response was truncated to 124 μs to avoid

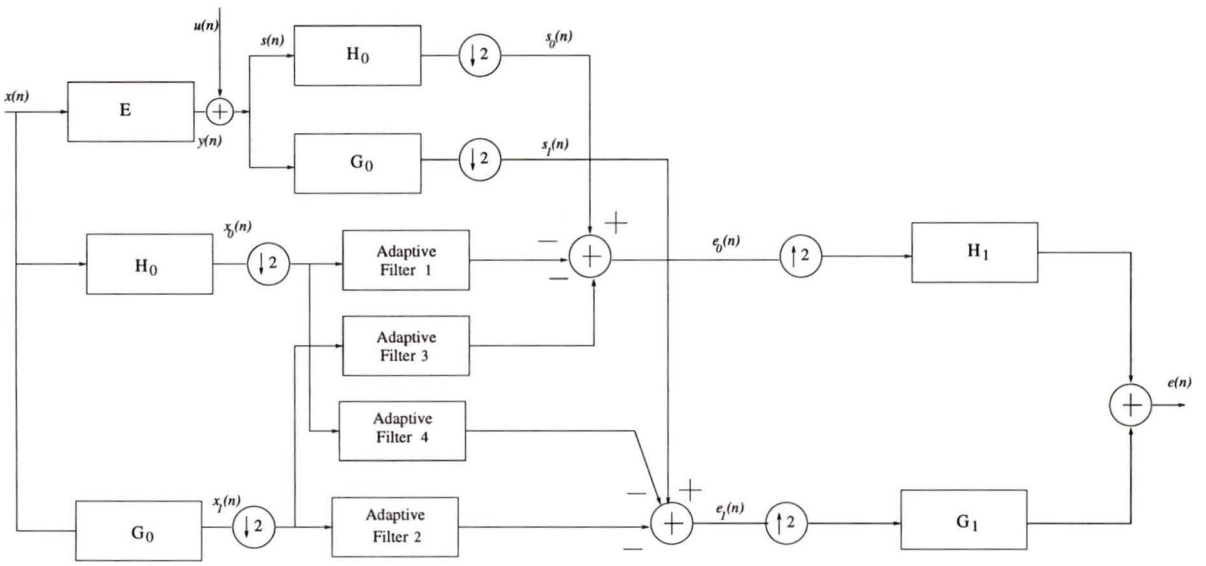

Figure 2.13. *Subband echo cancellation: A two-band one-level decomposition scheme.*

residual echo inherent with untruncated impulse responses. The sampling interval was set at 1 μs which resulted in the impulse response vector having a length of 124. The input signal to the echo path was a zero-mean white Gaussian noise with a variance of 0.25. The conventional full-band echo cancellation was simulated using an LMS adaptive filter of length 124. The subband echo cancellation system was simulated using the configuration shown in Fig. 2.13. Daubechies wavelet filters [14] of length 16 were used to decompose the input signal as well as the echo. Adaptive filters 1, 2, 3, and 4 were of length 76 each as required [13, 15]. The simulation was performed in MATLAB with floating-point precision. Fig. 2.14 shows that the MSE of the subband echo canceller approaches the MSE of the full-band conventional echo canceller. From the figure, a degradation in the convergence speed due to the presence of the cross adaptive filters can be observed.

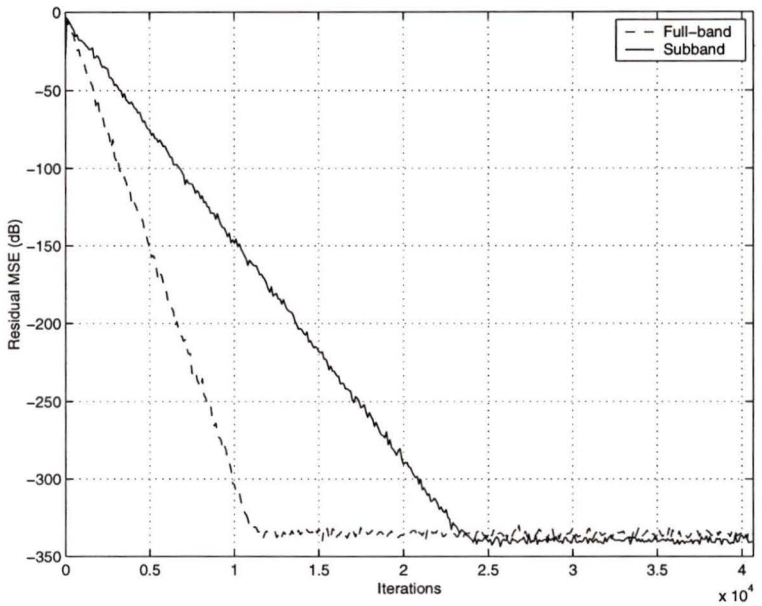

Figure 2.14. *Comparison of subband and conventional echo cancellation.*

2.4 Design of Filter Banks for Subband Echo Cancellation

Orthogonal wavelet filters are an ideal choice for the design of the filter banks because of the orthogonality property. Orthogonality is a desirable property since it results in perfect-reconstruction filter banks. In such filter banks, the aliasing caused at the analysis filter bank is cancelled at the synthesis filter bank [16].

Let $H_0(z)$ and $G_0(z)$ be the lowpass and highpass transfer functions of the analysis filter bank, respectively. Similarly, let $H_1(z)$ and $G_1(z)$ be the corresponding transfer functions in the synthesis filter bank. Alias cancellation or the perfect-reconstruction property is achieved if the equality

$$H_1(z)H_0(z) + G_1(z)G_0(z) = 0$$

holds for all z.

2.4.1 Problem Formulation

In the case of subband echo cancellation, the signal undergoes adaptive filtering after the analysis filter banks and hence it is not possible for the aliasing to be cancelled at the synthesis filter banks even with the use of a perfect-reconstruction filter bank. This aliasing is cancelled using the cross-adaptive filters which degrade the speed of convergence of an echo cancellation system. In order to eliminate the cross adaptive filters, we attempt to minimize the aliasing caused at the analysis stage and to this end we need to use perfect half-band filters at the analysis and synthesis stages. Therefore, $H_0(e^{j\omega T})$ and $G_0(e^{j\omega T})$ are designed to approximate the desired frequency responses

$$H_0(e^{j\omega T}) = 0, \quad |\omega| \geq \frac{\pi}{2} \tag{2.12}$$

and

$$G_0(e^{j\omega T}) = 0, \quad |\omega| \leq \frac{\pi}{2} \tag{2.13}$$

where T is assumed to be unity.

Let $\{h_n\}$ and $\{g_n\}$ be the lowpass and highpass FIR filter coefficients in the discrete domain, respectively. The conditions for optimization are as follows:

- $\{h_n\}$ is chosen to satisfy (2.12);
- $\{g_n\}$ is chosen to satisfy (2.13); and
- $\{h_n\}$ satisfies the bi-orthogonal condition

$$\sum_n h_{n-2l} h_n = \delta_l \tag{2.14}$$

where δ_l is the Kronecker delta function. Based on the above optimization criterion, the problem can be converted into the maximization of a penalty function [17], i.e.,

$$\underset{h}{\text{maximize}} \left[\frac{\mathbf{h^T E h}}{\mathbf{h^T h}} + \sum_{l=1}^{\frac{L-1}{2}} \kappa_l \frac{\left(\sum_{n=2l}^{L-1} h_{n-2l} h_n\right)^2}{\mathbf{h^T h}} \right] \tag{2.15}$$

where $\mathbf{h} = [h_0 \; h_1 \; \cdots \; h_{L-1}]^T$, $\mathbf{E} = [\frac{\sin(n-m)\pi/2}{(n-m)\pi}]_{L\times L}$, and κ_l are the Lagrange multipliers for $l = 0, \ldots, (L-1)/2$. Note that the above function does not impose the strict condition of orthogonality and hence the resulting lowpass filter coefficient vectors obtained are not orthogonal.

2.4.2 Design of Optimized Orthogonal Wavelets using Linear Parameterization

Reference [18] describes a method of optimizing the filter bank without loosing the orthogonality property.

Since the filters used in a two-band scheme are half-band filters, their transfer function can be expressed as

$$P(z) = H_0(z)H_0(z^{-1}) \tag{2.16}$$

where $P(z)$ satisfies the condition

$$P(z) + P(-z) = 2 \quad \forall z \tag{2.17}$$

When $H_0(z)$ has at least L zeros at $\omega = \pi$, then it takes the form

$$H_0(z) = \left(\frac{1+z^{-1}}{2}\right)^L B_1(z) \tag{2.18}$$

where $B_1(z)$ is a Kth-order polynomial in z^{-1}. $P(z)$ characterizes an orthogonal, lowpass, analysis wavelet filter, which corresponds to a wavelet that has at least L vanishing moments. In such a case, $P(z)$ takes the form

$$P(z) = \left(\frac{z^{-1}}{4} + \frac{1}{2} + \frac{z}{2}\right)^L B(z) \tag{2.19}$$

with

$$B(z) = \sum_{k=-K}^{K} \hat{b}_k z^k, \; b_{2K-k} = b_k \tag{2.20}$$

To have (2.16) and (2.18) satisfied, $B(z)$ should be factorizable as

$$B(z) = B_1(z)B_1(z^{-1}) \tag{2.21}$$

This allows us to write $B(z)$ as

$$B(z) = z^{-k} \sum_{k=0}^{2K} b_k z^k, \; b_{2K-k} = b_k \tag{2.22}$$

and

$$\left(\frac{z^{-1}}{4} + \frac{l}{2} + \frac{z}{4}\right)^L = z^{-L} \sum_{l=0}^{2L} a_l z_l, \; a_{2L-l} = a_l \tag{2.23}$$

Reference [18] goes on to show that the condition for $P(z)$ to satisfy (2.17) is

$$\sum_{\substack{l+k=s \\ l\geq 0, g\geq 0}} a_l b_k = \begin{cases} 1, & s = N \\ 0, & s = 1, 3, \ldots, N-2 \end{cases} \tag{2.24}$$

where $N = K + L$ is assumed to be odd. We can now write (2.24) as

$$\mathbf{Ab} = \mathbf{m} \tag{2.25}$$

where $\mathbf{A}$ is an $(N+1)/2 \times (K+1)$ matrix determined by the a_l's, $\mathbf{b}$ is a column vector of b_k's of length $K+1$, and $\mathbf{m} = [0 \cdots 1]^T$ is a vector of length $(N+1)/2$. The number of degrees of freedom in (2.24) is $\eta = (K - L + 1)/2$ and it depends on the number of vanishing moments we would like the wavelet to have. Assuming that $\eta \geq 1$, the system of linear equations in (2.24) has solutions of the type

$$\mathbf{b} = \mathbf{b_0} + \mathbf{V}_\eta \phi \tag{2.26}$$

Now, the frequency response of the half-band filter can be expressed as

$$P(e^{j\omega}) = 2^{-L}(1 + \cos\omega)^L \mathbf{h}_1(\omega)^T (\mathbf{b}_0 + V_\eta \phi) \tag{2.27}$$

where

$$\mathbf{h}_1(\omega) = [1 \; 2\cos\omega \; \cdots \; 2\cos K\omega]^T \tag{2.28}$$

We can take out the constant component of $P(e^{j\omega})$ and write

$$P(e^{j\omega}) = P_0(e^{j\omega}) + \mathbf{h}^T(\omega)\phi \tag{2.29}$$

where

$$P_0(e^{j\omega}) = 2^{-L}(1 + \cos\omega)^L \mathbf{h}_1(\omega)\mathbf{b}_0 \tag{2.30}$$

and

$$\mathbf{h}(\omega) = 2^{-L}(1 + \cos\omega)^L \mathbf{V}_\eta^T \mathbf{h}_1(\omega) \tag{2.31}$$

2.4.3 Design of the Orthogonal Wavelet Filter for Echo Cancellation

The condition for factorization of $B(z)$ to exist is that $P(e^{j\omega}) \geq 0$ for $\omega \in [0, \pi]$. The discretized version of this condition is given in [18] as

$$\mathbf{D}\phi \geq \mathbf{p} + \epsilon\mathbf{e} \tag{2.32}$$

with

$$\mathbf{D} = \begin{bmatrix} \mathbf{h}_1^T(\omega_1) \\ \mathbf{h}_1^T(\omega_2) \\ \vdots \\ \mathbf{h}_1^T(\omega_M) \end{bmatrix} \mathbf{V}_\eta, \ \mathbf{p} = \begin{bmatrix} \mathbf{h}_1^T(\omega_1) \\ \mathbf{h}_1^T(\omega_2) \\ \vdots \\ \mathbf{h}_1^T(\omega_M) \end{bmatrix} \mathbf{b}_0 \tag{2.33}$$

where $\{\omega_i, \ 0 \leq i \leq M-1\}$ are the frequencies on the interval $[0, \pi]$ at which the continous-time equation is evaluated. The term ϵ is a small positive scalar and $\mathbf{e} = [1 \ 1 \cdots 1]^T$

As explained in the previous section, we need to maximize the quantity

$$E = \frac{1}{2\pi} \int_{-\pi/2}^{\pi/2} |H_0(e^{j\omega})|^2 d\omega \tag{2.34}$$

From (2.16) and (2.32), we have

$$E = \frac{1}{2} \int_{-\pi/2}^{\pi/2} P(e^{j\omega}) d\omega = \mathbf{c}^T \phi + \lambda \tag{2.35}$$

where

$$\mathbf{c} = -\frac{1}{2\pi} \int_{-\pi/2}^{\pi/2} \mathbf{h}(\omega) d\omega \tag{2.36}$$

Therefore, the optimization problem can be formulated as the linear programming problem

$$\text{minimize } \mathbf{c}^T \phi \tag{2.37}$$

subject to:

$$\mathbf{D}\phi \geq \mathbf{p} + \epsilon \mathbf{e} \tag{2.38}$$

By using the primal Newton-barrier algorithm as described in [18], we can find the optimum ϕ and on using (2.25) we can calculate $\mathbf{b}$. Then using (2.20), $B(z)$ can be constructed and $B_1(z)$ can be obtained using spectral factorization; then $H_0(z)$ can be obtained.

2.4.3.1 Modified Objective Function

In order to force most of the energy to be concentrated in $[-\pi/2, \ \pi/2]$, we have previously tried to maximize the quantity

$$\int_0^{\pi/2} |H_0(e^{j\omega})|^2 \, d\omega \tag{2.39}$$

further improvement can be achieved if we squeeze in more energy by subtracting the energy in $[\pi/2,\ \pi]$ from the above quantity. We can add a weight to the second term, i.e., try to maximize the quantity

$$\int_0^{\pi/2} |H_0(e^{j\omega})|^2\, d\omega - w \int_{\pi/2}^{\pi} |H_0(e^{j\omega})|^2\, d\omega \tag{2.40}$$

where w is the weight assigned based on the importance of the second term. It is easy to see that w will be meaningful if it takes values from 0 to 1. Hence we could further optimize the half-band filter by finding the value of w that gives the maximum energy concentration in $[-\pi/2,\ \pi/2]$ and then calculating the half-band filter corresponding to that w.

A good measure of the optimization is the fraction of energy content of a filtered signal in $[-\pi/2,\ \pi/2]$, over the total energy of the filtered signal, which can be calculated as

$$E_f = \frac{\int_{-\pi/2}^{\pi/2} |H_0(e^{j\omega})|^2\, d\omega}{\int_{-\pi}^{\pi} |H_0(e^{j\omega})|^2\, d\omega} \tag{2.41}$$

Optimizing a filter of order 19 with 9 degrees of freedom, which is a filter corresponding to a wavelet that has only one vanishing moment, we obtain a minimum E_f of 0.9699, which is very close to the value of 0.974 obtained using the method described in [17]. This is a small difference given that the resulting filter with this method is *orthogonal*. Furthermore, the method used for optimization is linear programming which is much faster than unconstrained maximization of the penalty function described in [17]. The half-band filter coefficients otained are given in Table 2.2.

Fig. 2.15 shows the amplitude responses of the optimized half-band filter of length 20 and the Daubechies orthogonal filter of the same length.

2.4.4 Simulation

To demonstrate the improved performance of the wavelet filter designed in the preceeding section, two subband echo cancellers with similar input signals and echo path impulse responses were simulated. The structures of the echo cancellers were the same as shown in Fig. 2.13 but without the cross adaptive filters 3 and 4. The echo-path impulse response was the same as that used in Section 2.3.2.2 and the input signal

Table 2.2. *Filter Coefficients*

n	Designed in [17]	Optimum orthogonal
0	-0.0160	0.00278766
1	0.0367	-0.00810576
2	-0.0142	0.00076205
3	-0.0387	0.02097110
4	0.0309	-0.01573315
5	0.0411	-0.02924597
6	-0.0445	0.03446170
7	-0.0435	0.03726315
8	0.0588	-0.05770701
9	0.0598	-0.04798513
10	-0.0762	0.08744309
11	-0.0829	0.06629413
12	0.0970	-0.12529291
13	0.1304	-0.10885254
14	-0.1202	0.17666459
15	-0.2418	0.22383742
16	0.1101	-0.21986416
17	0.5976	-0.65945582
18	0.6492	-0.59173515
19	0.2829	-0.20293388

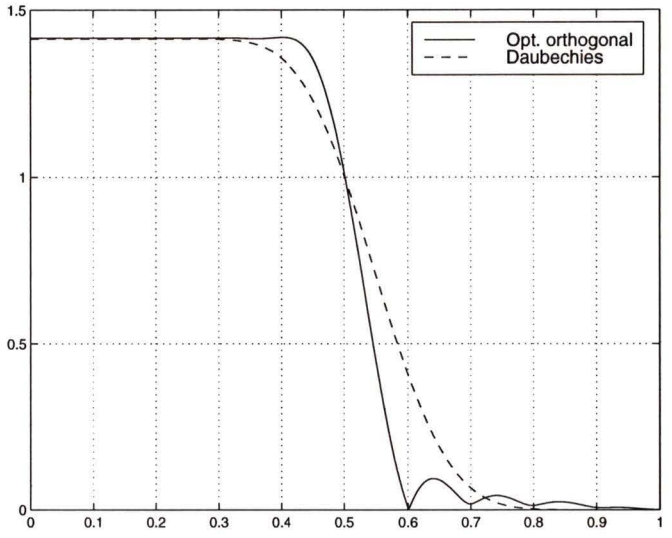

Figure 2.15. *Amplitude responses of optimized orthogonal filter of length 20 and Daubechies orthogonal wavelet filter.*

was again a white Gaussian noise with a variance of 0.25. The two subband echo cancellers were different only in the use of filter banks. One of them was simulated using a Daubechies wavelet filter of length 20 and the other was simulated using the filter bank designed in the preceeding section. Fig. 2.16 shows that the subband echo canceller that uses the designed filter bank has a better performance than the one that uses the Daubechies wavelet filter in terms of the residual echo power.

In order to illustrate the effect of the cross adaptive filters on the rate of convergence, the MSE for the first 2000 iterations for 3 different echo cancellers is shown in Fig. 2.17. We see that the rate of convergence of the subband echo canceller without the cross filters is better than that of the subband echo canceller with cross filters.

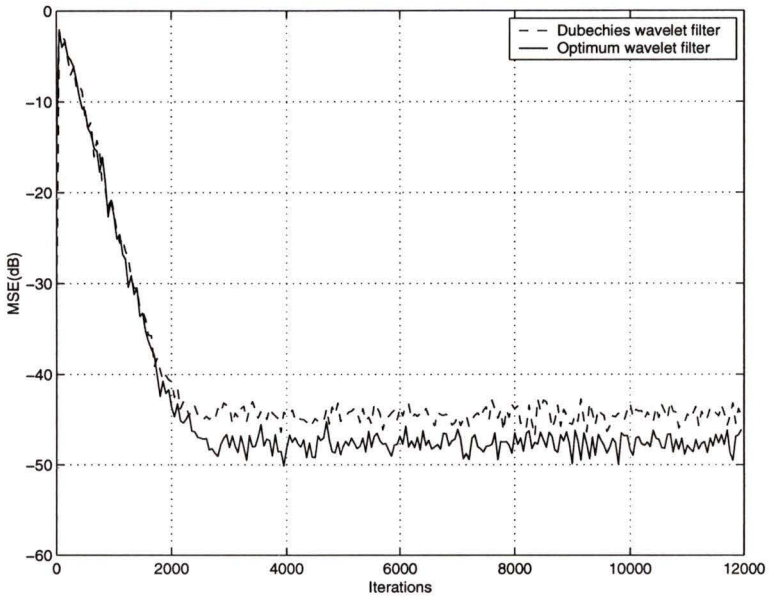

Figure 2.16. *Comparison of performances of Daubechies wavelet filter and optimum orthogonal wavelet filter.*

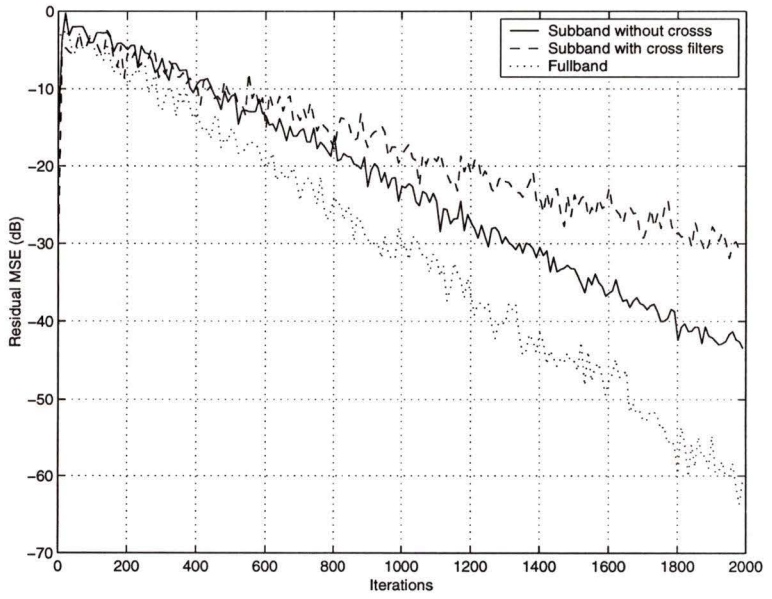

Figure 2.17. *Comparison of rate of convergence.*

2.5 Conclusions

The process of subband adaptive filtering has been described and simulated. All the simulations have been performed on the echo-path impulse response of one of 8 CSA test loops for HDSL2.

A procedure for obtaining the echo impulse responses for the CSA loops has been described in Section 2.2. The input impedances as well as the echo path impulse responses for the 8 CSA test loops for HDSL2 have been given.

Section 2.3 describes the structure of an adaptive filter and verifies the asymptotic convergence of the MSE in a subband echo canceller with that of a conventional full-band echo canceller. The subband echo canceller was found to exhibit a poorer rate of convergence when compared wth that of the conventional echo canceller.

The problem of optimal design of filter banks that can be used in a subband echo-canceller without cross adaptive filters has been formulated in Section 2.4. Two methods of optimization of the filter bank have been described in detail. The first one optimizes the filter but loses the orthogonality in the process whereas the second yields an orthogonal filter bank. These methods have been applied and results were presented for a filter of length 20.

The optimized wavelet filter has been shown to offer improved performance through simulation as described in Section 2.4 and an improved rate of convergence has been achieved due to the absence of cross adaptive filters.

Chapter 3

Variable Step-Size LMS Algorithms

3.1 Introduction

The LMS algorithm uses an adaptation error and a step size to update the adaptive filter coefficients. The value of the step size determines the performance of the algorithm [9]. Various analyses have been performed on the effect of the choice of the step size on the performance characteristics of the algorithm in terms of the rate of convergence and steady-state misadjustment [9, 10]. In general, a smaller step size results in a small misadjustment but also a slower rate of convergence. On the other hand, a larger step size results in a faster rate of convergence but a large steady-state misadjustment. One popular approach to eliminate this trade-off between the rate of convergence and misadjustment is to use a variable-step-size LMS algorithm which uses a large step size in the initial stages of convergence of the algorithm thus speeding up convergence but uses a smaller step size when the algorithm is close to convergence thus resulting in a small misadjustment [10, 19, 20, 21].

3.2 Known Algorithms

As was stated in Sec. 2.3.1, an LMS algorithm is characterized by the equation

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu(n)e(n)\mathbf{x}(n) \tag{3.1}$$

where $\mathbf{w}(n)$ is the coefficient vector at time n and $\mu(n)$, $e(n)$, and $\mathbf{x}(n)$ are the step size, adaptation error, and input vector, respectively, at time n. In the case of a

fixed-step-size (FSS) LMS algorithm, $\mu(n)$ is chosen to be a constant. On the other hand, the variable-step-size (VSS) LMS algorithm described in [19] uses the equation

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

for updating the step size where $0 < \alpha < 1$, $\gamma > 0$, and $\mu(n+1)$ is bounded by μ_{max} from above and by μ_{min} from below. Typically, the value of μ_{max} is selected to provide the maximum possible rate of convergence while the value of μ_{min} is chosen based on the desired level of the steady-state misadjustment. As the algorithm converges, the step size decreases in value since the LMS algorithm causes the mean-square error (MSE) to decrease. However, the VSS algorithm results in an undesirable noisy step size because of the presence of the $e^2(n)$ term in Eq. (3.2) [22]. Furthermore, in the case of low signal-to-noise ratio (SNR), when the steady-state MSE of the algorithm is high, the VSS algorithm usually results in a large steady-state misadjustment [22].

In order to overcome the problems associated with noisy step size and large steady-state misadjustment, a modified variable step size (MVSS) has been proposed in [22]. The algorithm updates the step size based on the time-averaged estimate of the autocorrelation of $e(n)$ and $e(n-1)$. The MVSS algorithm can be described by

$$\mu(n+1) = \alpha\mu(n) + \gamma p(n)^2 \tag{3.3}$$

where

$$p(n) = \beta p(n-1) + (1-\beta)e(n)e(n-1) \tag{3.4}$$

The limits on $\mu(n+1)$, α, and γ are the same as those in the VSS LMS algorithm. The positive constant β $(0 < \beta < 1)$ is an exponential weighting parameter that governs the averaging time constant, and $p(n)$ is the time-averaged estimate of the autocorrelation of $e(n)$ and $e(n-1)$.

In what follows, the FSS, VSS, and MVSS algorithms are simulated for the sake of comparison in a system identification setup. The unknown moving average system has four time invariant coefficients, and the FIR adaptive filter is of order three. Both the unknown system and the FIR adaptive filter are excited by zero-mean, white Gaussian signal of unity variance. The MVSS algorithm is implemented using the parameter $\alpha = 0.97$, as suggested in [22] and $\beta = 0.99$. In accordance with Eq.(33) in [22], γ is set to 10^{-3} to produce a steady-state excess MSE of about -34 dB. The

VSS algorithm is used with $\alpha = 0.97$, as required by [19] and $\gamma = 1 \times 10^{-5}$ is used to obtain the same level of steady state excess MSE. The FSS algorithm is used with a step-size value of 3.5×10^{-4}. Fig. 3.1 shows the better performance of the MVSS algorithm over that of the VSS and the FSS algorithms. This is confirmed by the plot of the mean behaviour of one of the four coefficients of the adaptive filter, shown in Fig. 3.2. The results shown in Figs. 3.1 and 3.2 have been obtained by averaging the MSE over 100 independant runs for each of the three algorithms under comparison.

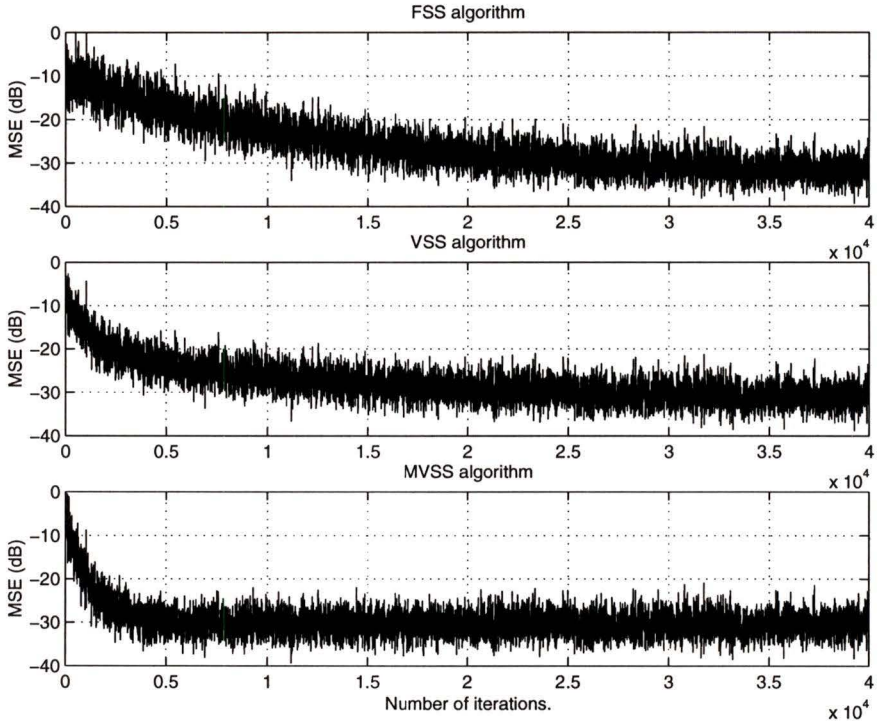

Figure 3.1. *Comparison of MSE using FSS, VSS, and MVSS algorithms.*

3.3 New Variable Step-Size Algorithm

The development of the MVSS algorithm is based on the assumption that the steady-state error $e(n)$ is uncorrelated and hence results in a small value for the step size after convergence. However, in the case of high-order adaptive filters, which are typically encountered in xDSL echo cancellation applications [8], the adaptation error $e(n)$

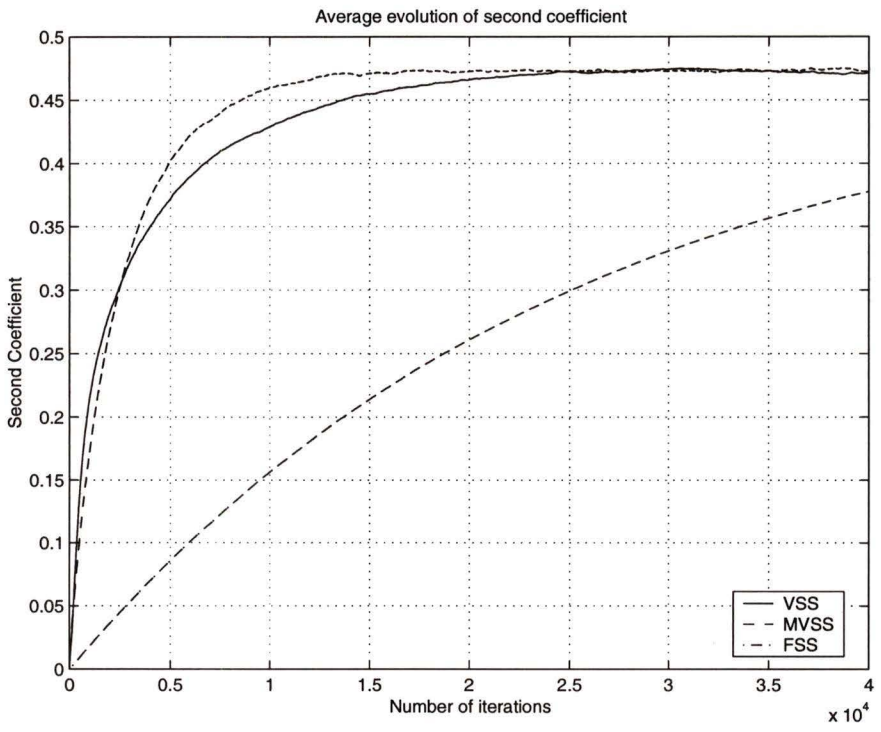

Figure 3.2. *Comparison of coefficient evolution using FSS, VSS, and MVSS algorithms.*

may not be correlated during convergence. This would reduce the step size to the minimum value before convergence. In other words, even though the steady-state analysis of the MVSS algorithm holds good, the algorithm requires a larger number of iterations to reach the steady state.

Since the MSE is a critical measure of the convergence of the LMS algorithm, we propose to update the step size based on the time-averaged estimate of the autocorrelation of $e(n)$ and $e(n-1)$ and that of the MSE. Under these circumstances, the update equation for $p(n)$ takes the form

$$p(n) = \beta p(n-1) + (1-\beta)[e(n)e(n-1) + e(n)^2] \tag{3.5}$$

By rewriting Eq. (3.5) as

$$p(n) = \beta p(n-1) + (1-\beta)\{e(n)[e(n-1) + e(n)]\} \tag{3.6}$$

we notice that no additional multiplications are required for the update equation and the marginal increase in the computational complexity over the MVSS algorithm is

due to the addition of $e(n)$ and $e(n-1)$ in Eq. (3.6). In addition, the averaging of $e(n)^2$ results in the reduction of the noisy nature of the step size inherent with the VSS algorithm. In summary, the proposed LMS algorithm is described by

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu(n)e(n)\mathbf{x}(n) \tag{3.7a}$$

$$\mu(n+1) = \alpha\mu(n) + \gamma p(n)^2 \tag{3.7b}$$

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

where $\mu(0) = \mu_{max}$ and $p(0) = \mu(0)$. In Section 3.3.1, the performance of the proposed algorithm is compared with that of the known MVSS LMS algorithm through simulations.

3.3.1 Simulations

For the purpose of comparison, the proposed and the MVSS algorithms were used in an echo cancellation configuration. The echo path was simulated using the echo path impulse response of CSA loop #2 from the set of HDSL2 test loops, at the central office end. The echo path impulse response was truncated to 124 μs to avoid the residual echo inherent in untruncated impulse responses. The sampling interval was set to 1 μs which resulted in an impulse response vector of length of 124.

The echo path was excited with a zero-mean, white Gaussian signal to generate the echo. The echo signal thus generated was contaminated with the received signal which was also a zero-mean, white Gaussian signal to result in an SNR of 0 dB. The two algorithms were simulated with the following parameter values:

$$\begin{aligned} \alpha &= 0.97 \\ \gamma &= 0.08 \\ \beta &= 0.999 \\ \mu_{max} &= 0.1 \\ \mu_{min} &= 0.01 \end{aligned}$$

To measure the performance of the two algorithms, the value of the misadjustment $||\mathbf{w}^* - \mathbf{w}(n)||$, where $\mathbf{w}^*$ is the echo path impulse response vector and $\mathbf{w}(n)$ is the estimated coefficient vector of the LMS algorithm at the nth iteration, was calculated and plotted against the number of iterations in Fig. 3.3. As can be observed, the proposed algorithm yields a lower steady-state misadjustment.

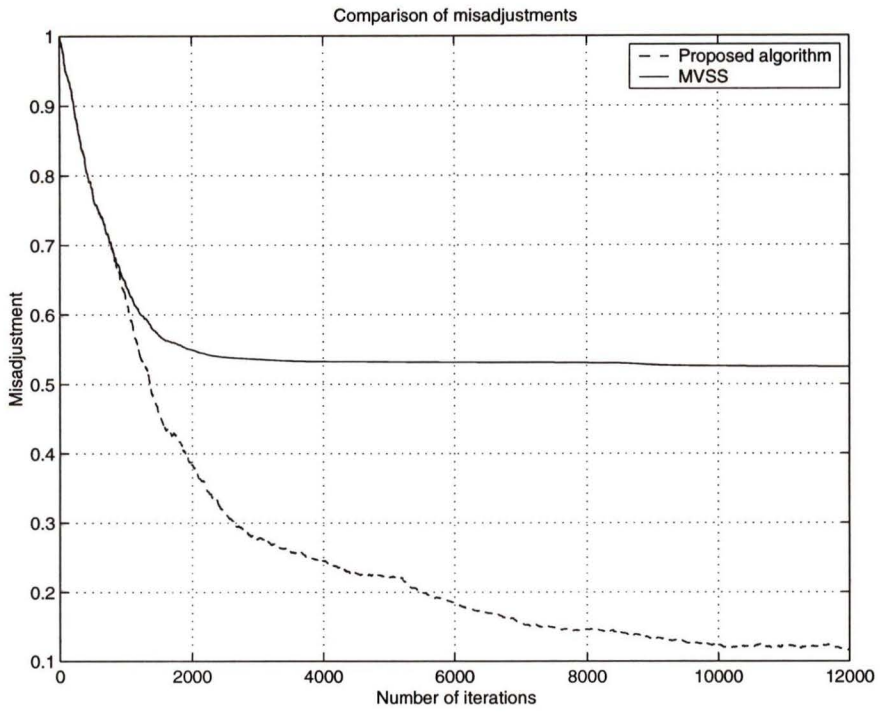

Figure 3.3. *Comparison of the misadjustments using the MVSS and proposed VSS algorithms.*

The use of the proposed VSS algorithm results in the step size taking a small value after convergence. Hence, one of the scenarios in which the performance of the proposed VSS algorithm is of concern is when the echo path impulse response of the channel under consideration undergoes an abrupt change after the step-size has taken a small value. In such a case, the convergence of the algorithm to the changed impulse response would be much slower than the convergence to the impulse response before the abrupt change unless the step-size assumes a large value after the change in the echo path impulse response. In order to study the performance of the proposed VSS algorithm in such a scenario, the setup of the preceding simulation was repeated and the echo path impulse response was scaled by 0.4 at iteration 12288, which was enough for the algorithm to converge and for the step-size to go to a small value. Fig. 3.4 shows the MSE and the step-size profile of the proposed algorithm. We notice that the step-size assumes the maximum permissible value, i.e., μ_{max}, with the change and hence the algorithm converges to the new optimum. The MSE and

the step-size shown in Fig. 3.4 have been averaged over 20 independant runs of the algorithm.

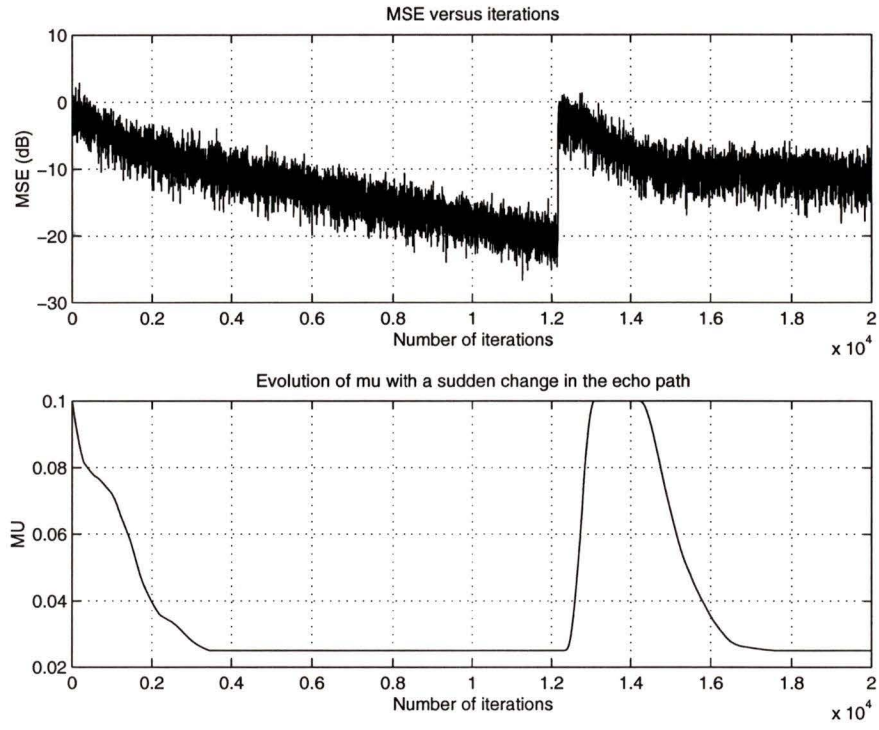

Figure 3.4. *Performance of the proposed VSS algorithm with a abrupt change in the echo path impulse response.*

3.4 Application to Subband Adaptive Filtering

In subband adaptive filtering, the input and output signals are split into adjacent frequency subbands by analysis filter banks; then each subband signal is subsampled, and the adaptive filtering algorithm is applied to each of these signals. In this chapter, we consider the subband echo cancellation system shown in Fig. 3.5 in which the transmit and receive signals are decomposed into two subbands and adaptive echo cancellation is performed independantly in the two subbands without the presence of the cross adaptive filters as described in [13].

When the FSS LMS algorithm is used for adaptive filtering in each subband, the

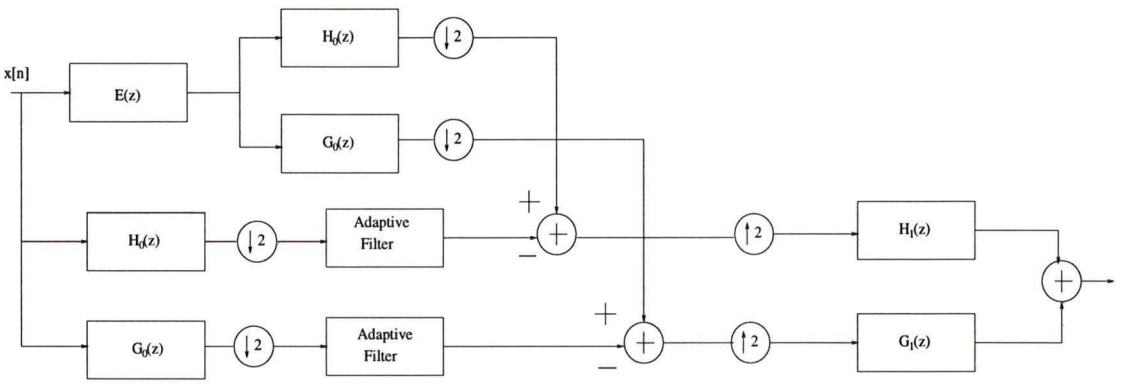

Figure 3.5. *Echo cancellation using filter banks: A two-band decomposition scheme.*

value of the step size is chosen to be the same for each subband since we do not know the signal characteristics in each subband *a priori*. However, using the proposed VSS algorithm, we expect the step size to vary differently in each subband thereby resulting in a smaller steady-state misadjustment and hence a smaller steady-state MSE.

In order to measure the performance of the LMS algorithms in subband adaptive filtering, the misadjustments in the two subbands are defined as $||\mathbf{w_H}^* - \mathbf{w_H}(n)||$ and $||\mathbf{w_G}^* - \mathbf{w_G}(n)||$ where $\mathbf{w_H}(n)$ and $\mathbf{w_G}(n)$ are the coefficient vectors of the adaptive filters in the nth iteration in the subbands generated by the filters H_0 and G_0, respectively, and $\mathbf{w_H}^*$ and $\mathbf{w_G}^*$ are the optimal coefficient vectors in the two subbands that can be calculated from [13] as

$$\mathbf{w_H}^* = \mathcal{D}(\mathbf{e} * \mathbf{h}_0) \tag{3.8}$$

and

$$\mathbf{w_G}^* = \mathcal{D}(\mathbf{e} * \mathbf{g}_0) \tag{3.9}$$

by neglecting the interaction between adjacent subbands. In (3.8) and (3.9), $\mathbf{e}$ denotes the echo path impulse response vector and is the same as vector $\mathbf{w}^*$ in the full-band case, $\mathcal{D}$ denotes the operation of downsampling and, in the case of a two-subband scheme, $\mathcal{D}$ denotes downsampling by a factor of 2. In Section 3.4.1, we simulate the subband echo cancellation using the proposed algorithm and discuss the results.

3.4.1 Simulations

For the purpose of simulation, the echo path chosen for the subband case was a lowpass FIR filter with a length of 124. The synthesis and analysis filter banks shown in Fig. 3.5 were designed using the method described in [23] which results in optimum orthogonal filter banks. The filters involved, namely H_0, G_0, H_1, and G_1, were FIR filters of length 20. The proposed VSS algorithm has been used with $\mu_{min} = 0.01$ and $\mu_{max} = 0.3$. The rest of the parameters in Eq. 3.7 were chosen to be the same as the values given in Section 3.3.1. The FSS LMS algorithm was simulated with a step size of 0.3. The misadjustments in each subband for the two algorithms are plotted in Fig. 3.6. Fig. 3.7 shows the average MSE of 20 independant runs of the proposed and the FSS algorithms. We notice that the proposed algorithm results in a smaller steady-state MSE. Also, we can notice from Fig. 3.8 that the step-size evolution in each subband is different which contributes to the better performance over that of the FSS algorithm.

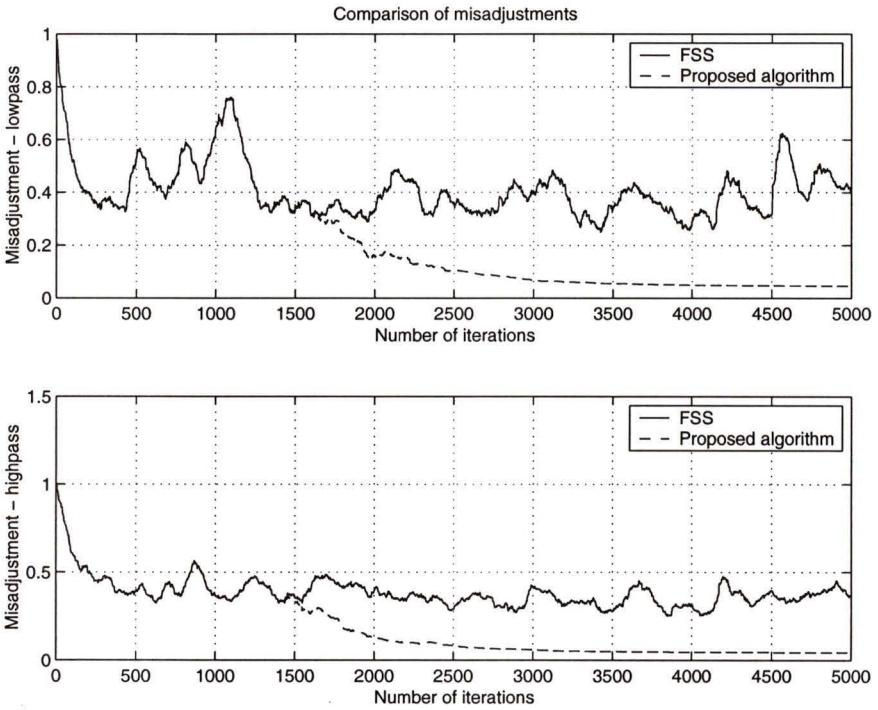

Figure 3.6. *Comparison of the misadjustment in the subbands using the FSS LMS and the proposed VSS algorithms.*

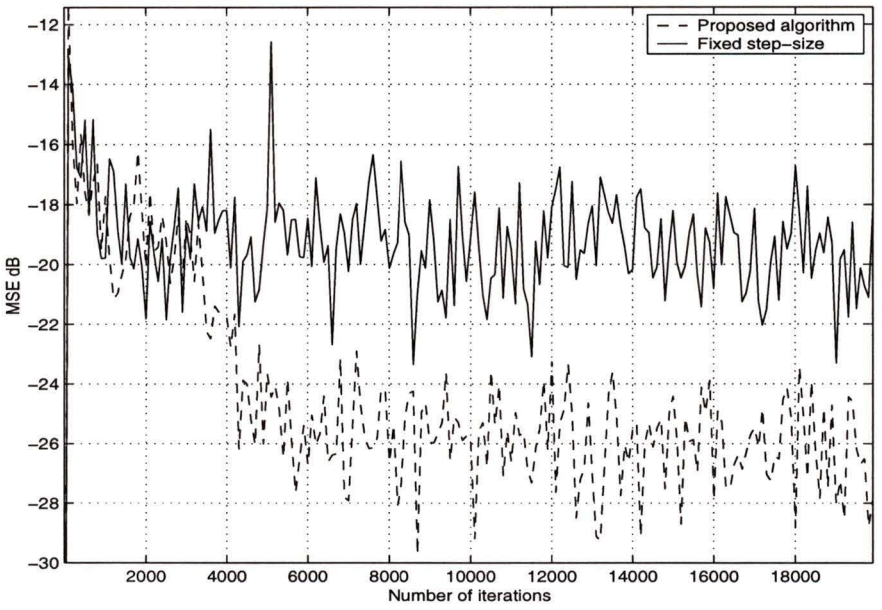

Figure 3.7. *Comparison of the MSE using the FSS LMS and the proposed VSS algorithms in subband echo cancellation.*

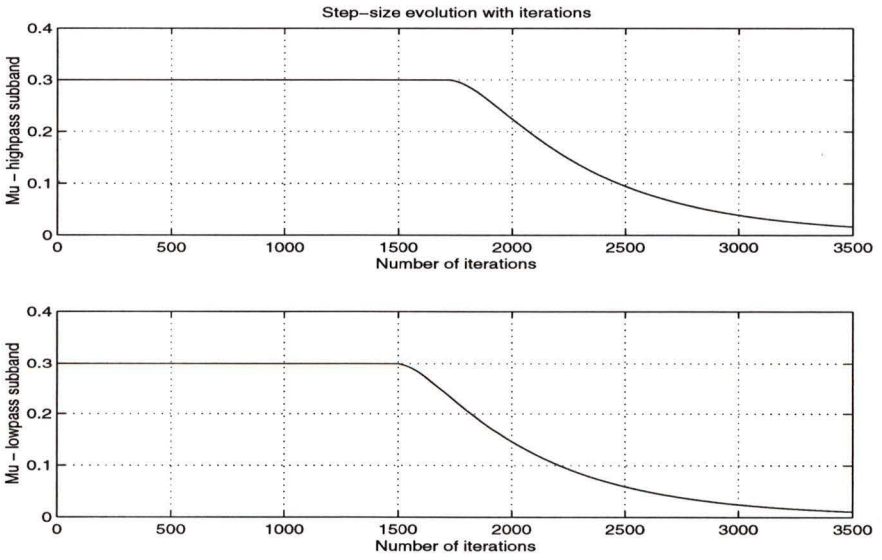

Figure 3.8. *Evolution of the step size in the two subbands using the proposed VSS algorithm.*

3.5 Conclusions

Two existing variable step-size algorithms, namely, the VSS and the MVSS LMS algorithms, have been described and simulations have been performed to study their performance. The MVSS algorithm exhibits superior performace over the VSS algorithm in low SNR scenarios. However, in the case of long adaptive filters, the MVSS algorithm results in a large final misadjustment. A new VSS algorithm has been proposed in Section 3.3 to overcome this problem and the proposed VSS algorithm has demonstrated superior performance over the MVSS algorithm in the case of long adaptive filters. The performance of the proposed VSS algorithm has also been studied in a scenario where a abrupt change in the echo path impulse response occurs. The step-size of the proposed algorithm has been found to assume the maximum permissible value after the change in the echo path impulse response thus resulting in a faster convergence of the adaptive filter coefficients to the new echo path impulse response coefficients.

The application of the proposed algorithm to subband LMS adaptive filtering has been investigated in Section 3.4. Simulation results show that the proposed algorithm yields a lower steady-state misadjustment in each subband as well as a lower residual MSE when compared with the FSS subband LMS algorithm. It has also been noticed that improved system performance can only be achieved with a distinct step-size adaptation for each individual subband.

Chapter 4

Block Adaptive Filtering for Echo Cancellation

4.1 Introduction

One of the approaches for reducing the computational complexity of long adaptive filters is to incorporate *block* updating strategies whereby fast Fourier transforms (FFT) algorithms efficiently implement the filter convolution and the gradient correlation [24, 25]. These techniques reduce the computational complexity because the filter output and the adaptive weights are computed only after a large block of input data has been accumulated. Depending on the organization of the input data into blocks, the block approaches may introduce some degradation in performance, including an end-to-end delay and possibly a reduction in the stable range of the algorithm step size.

The basic operation underlying a frequency-domain adaptive filter is the tranformation of the input signal into a more desirable form before adaptive processing. This transformation of the input signal is accomplished by using one or more discrete Fourier transforms (DFTs). The tranformation is non-adaptive and corresponds to a simple preprocessing step which is independent of the data. Figs. 4.1 and 4.2 show two formulations for the frequency-domain adaptive filter. In Fig. 4.1, the error signal $e(n)$ is computed in the time-domain by using the filter output $y(n)$ and the desired signal $d(n)$ and it is then tranformed into the frequency domain for use in the coefficient update process. In Fig. 4.2, the desired response $d(n)$ is first tranformed into the frequency domain and thus the error is computed directly in the frequency domain. For adaptive algorithms where the error is a linear function of the input data (e.g.,

the LMS algorithm), these two approaches may yield similar results [25].

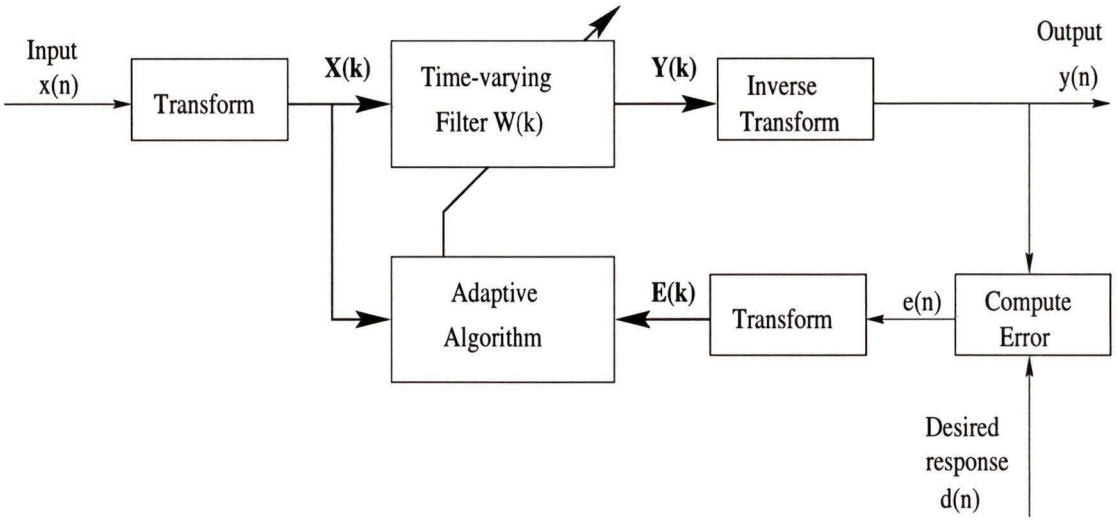

Figure 4.1. *Frequency-domain adaptive filtering with error calculated in the time-domain.*

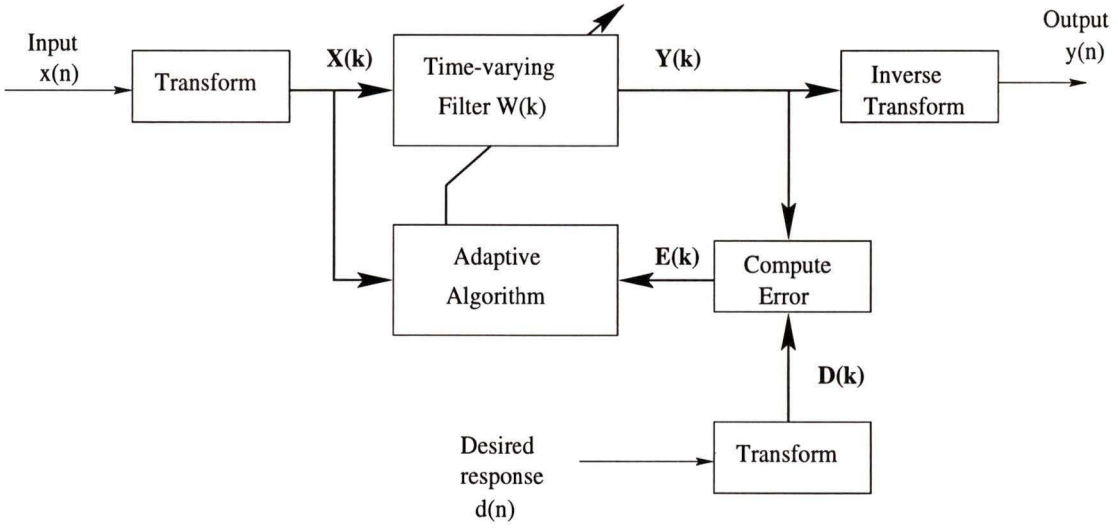

Figure 4.2. *Frequency-domain adaptive filtering with error calculated in the frequency domain.*

The block adaptive filtering technique is a general method that can be applied to various adaptive algorithms. However, in this chapter we are concerned with

the block implementation of the LMS algorithm, both in the time and frequency domains. In Section 4.2, the block implementation of the LMS algorithm in the time-domain is described. Section 4.3 deals with the frequency-domain block-LMS algorithm and its relation to the time-domain block-LMS algorithm. Simulations were carried out to study the performance of the frequency domain algorithms. In Section 4.4, a time-domain fast LMS algorithm is described. In Section 4.5, a time-domain LMS algorithm with partial coefficient updates is described and performance of the algorithm is evaluated in terms of computational complexity and convergence, by simulations.

4.2 Block-LMS Adaptive Filtering

In a block adaptive filter of the type illustrated in Fig. 4.3, the input data $u(n)$ are sectioned into L-point blocks by means of a serial-to-parallel converter, and an FIR filter of length N is applied to the blocks of the input data so produced, one block at a time. The coefficients of the FIR filter are held fixed over each block of data, so that the adaptation of the filter occurs on a block-by-block basis rather than on a sample-by-sample basis as in the conventional LMS algorithm. Let $\mathbf{w}(k)$ denote the coefficient vector at the kth block processing instant. Since the FIR filter is of length N,

$$\mathbf{w}(k) = [w_0(k)\ w_1(k)\ \cdots\ w_{N-1}(k)]^T \tag{4.1}$$

Index n is used to denote the sample time of the input data. Since each block is of length L,

$$n = kL + i, \quad i = 0, 1, \ldots, N-1 \tag{4.2}$$

Similarly we define the input data vector $\mathbf{u}(n)$ as

$$\mathbf{u}(n) = [u(n)\ u(n-1)\ \cdots\ u(n-N+1)]^T \tag{4.3}$$

Hence at time n, the filter output $y(n)$ is given by the inner product

$$y(n) = \mathbf{w}^T(k)\mathbf{u}(n) \tag{4.4}$$

Equivalently,

$$\begin{aligned} y(kL+i) &= \mathbf{w}^T(k)\mathbf{u}(kL+i) \\ &= \textstyle\sum_{l=0}^{N-1} w_l(k)u(kL+i-l), \quad i = 0, 1, \ldots, N-1 \end{aligned} \tag{4.5}$$

Let $d(n)$ denote the desired response of the filter at time n. The error signal $e(n)$ is given by

$$e(n) = d(n) - y(n) \tag{4.6}$$

or equivalently, by

$$e(kL + i) = d(kL + i) - y(kL + i) \tag{4.7}$$

That is, the error signal is permitted to vary at the sampling rate of the input signal as in the conventional LMS algorithm. A sectioning mechanism is used to section the sequence $e(n)$ into L-point blocks in a synchronous manner as illustrated in Fig. 4.3. Since the error signal is allowed to vary at the same sampling rate as the input signal,

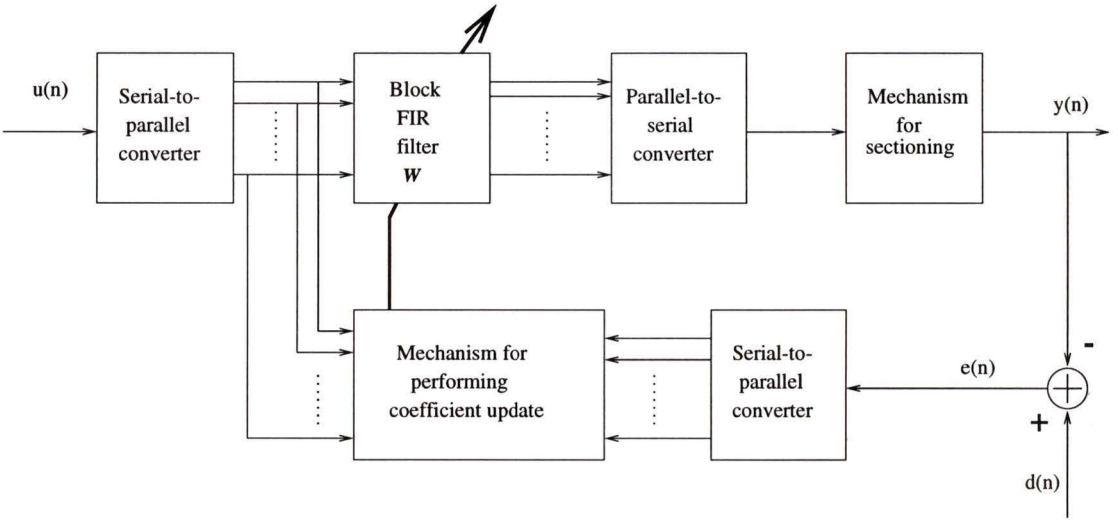

Figure 4.3. *Block adaptive filtering.*

for each block of input data, we have different values of the error signal for use in the coefficient update process. Reference [26] defines the LMS update equation for the block adaptive filter as

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu \sum_{i=0}^{L-1} \mathbf{u}(kN + i)e(kN + i) \tag{4.8}$$

where μ is the step-size parameter similar to the step-size of the conventional LMS algorithm. Eq. (4.8) can be rewritten as

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\phi(k) \tag{4.9}$$

where the N-by-1 vector is the cross correlation vector defined by

$$\phi(k) = \sum_{i=0}^{L-1} \mathbf{u}(kN+i)e(kN+i) \tag{4.10}$$

A distinctive feature of the above block-LMS algorithm is that an averaged estimate of the gradient vector is used instead of the instantaneous estimate used in Eq. (2.9) for the coefficient update. In the next section, we describe and simulate the frequency domain block-LMS algorithm and study its relation to the above described time-domain block-LMS algorithm.

4.3 Frequency-Domain Block-LMS Algorithms

We notice from the previous section that Eq. (4.5) implements linear convolution and Eq. (4.10) implements linear correlation. The basic principle of the frequency-domain block-LMS (FBLMS) algorithm is to use it to calculate the linear convolution and the linear correlation, respectively. Since the DFT-based calculation actually implements circular convolution, in order to obtain a result that is equivalent to linear convolution, a data vector $\mathbf{u}_k$ that has a larger dimension than the filter length N must be used [27]. In this section, the subscript is used to denote block processing instant. Hence a $2N$ length data vector, at the k-th block processing instant, $\mathbf{u}_k$ is formulated as

$$\mathbf{u}_k = [u(2Nk)\ u(2Nk+1)\ \cdots\ u(2Nk+2N-1)]^T \tag{4.11}$$

where the last N samples in $\mathbf{u}_k$ are taken from the current block of input data and the first N samples are taken from the previous input data block. In order to maintain consistent dimensions, the length of the weight vector must also be increased to length $2N$ by appending N zeros as indicated below:

$$\mathbf{w}_k = [\overbrace{w_k(0)w_k(1)\ldots w_k(N-1)}^{\mathbf{w}_{k,c}}\overbrace{0\cdots 0}^{N}]^T \tag{4.12}$$

If $\mathbf{F}$ denotes the $2N \times 2N$ DFT matrix with elements $F_{mn} = exp(-j2\pi mn/2N)$, then the output of the adaptive filter in the frequency domain is given by

$$\mathbf{Y}_k = \mathbf{U}_k\mathbf{W}_k \tag{4.13}$$

where $\mathbf{U}_k = \text{diag}(\mathbf{F}\mathbf{u}_k)$ and $\mathbf{W}_k = \mathbf{F}\mathbf{w}_k$. Since the last N elements of $\mathbf{F}^{-1}\mathbf{Y}_k$ coincide with the result of linear convolution, only these N elements should be used in calculating the error vector. Furthermore, if we define

$$\mathbf{d}_k = [\overbrace{0 \cdots 0}^{N} \ \overbrace{d_k(0) \ d_k(0) \ d_k(1) \ \cdots \ d_k(N-1)}^{\mathbf{d}_{k,c}}]^T \tag{4.14}$$

$$\mathbf{y}_k = [\overbrace{0, \ldots, 0}^{N}, \ \overbrace{\text{last } N \text{ elements of } \mathbf{F}^{-1}\mathbf{Y}_k}^{\mathbf{y}_{k,c}}]^T \tag{4.15}$$

then the error signal in the frequency domain is

$$\mathbf{E}_k = \mathbf{F}(\mathbf{d}_k - \mathbf{y}_k) \tag{4.16}$$

In order to guarentee that there are only N nonzero terms in the coefficient vector of the adaptive filter, a gradient constraint should be used to force the last N elements of the gradient in the time domain to be zero. Because of this constraint, the above FBLMS algorithm is known as the *constrained* FBLMS algorithm. Fig. 4.4 illustrates the constrained FBLMS algorithm in a flow chart. The weight update equation for this algorithm is given by

$$\mathbf{W}_{k+1} = \mathbf{W}_k + 2\mu\mathbf{F}(\boldsymbol{\Delta}(k)^T, 0, \ldots, 0)^T \tag{4.17}$$

where $\boldsymbol{\Delta}(k)$ the first N elements of $\mathbf{F}^{-1}\mathbf{U}_k^H\mathbf{E}_k$ and ^{H} denotes the complex-conjugate transpose. From Fig. 4.1, we can see that three FFT and two IFFT operations are required for each iteration of the algorithm. However, at any point of time only one FFT or IFFT operation is performed on the data and hence only one FFT and one IFFT blocks are needed if the algorithm is implemented by hardware. In order to reduce the computational complexity of the constrained FBLMS algorithm, an unconstrained FBLMS algorithm has been developed in reference [24] by simply removing the constraint from the constrained FBLMS equation so that the coefficient updating formula becomes

$$\mathbf{W}_{k+1} = \mathbf{W}_k + 2\mu\mathbf{U}_k^H\mathbf{E}_k \tag{4.18}$$

With the constraint removed, the gradient is no longer the result of linear correlation but rather the result of circular correlation. However, this simplified FBLMS algorithm requires only two FFT and one IFFT operations. The block diagram for the unconstrained FBLMS algorithm is shown in Fig. 4.5.

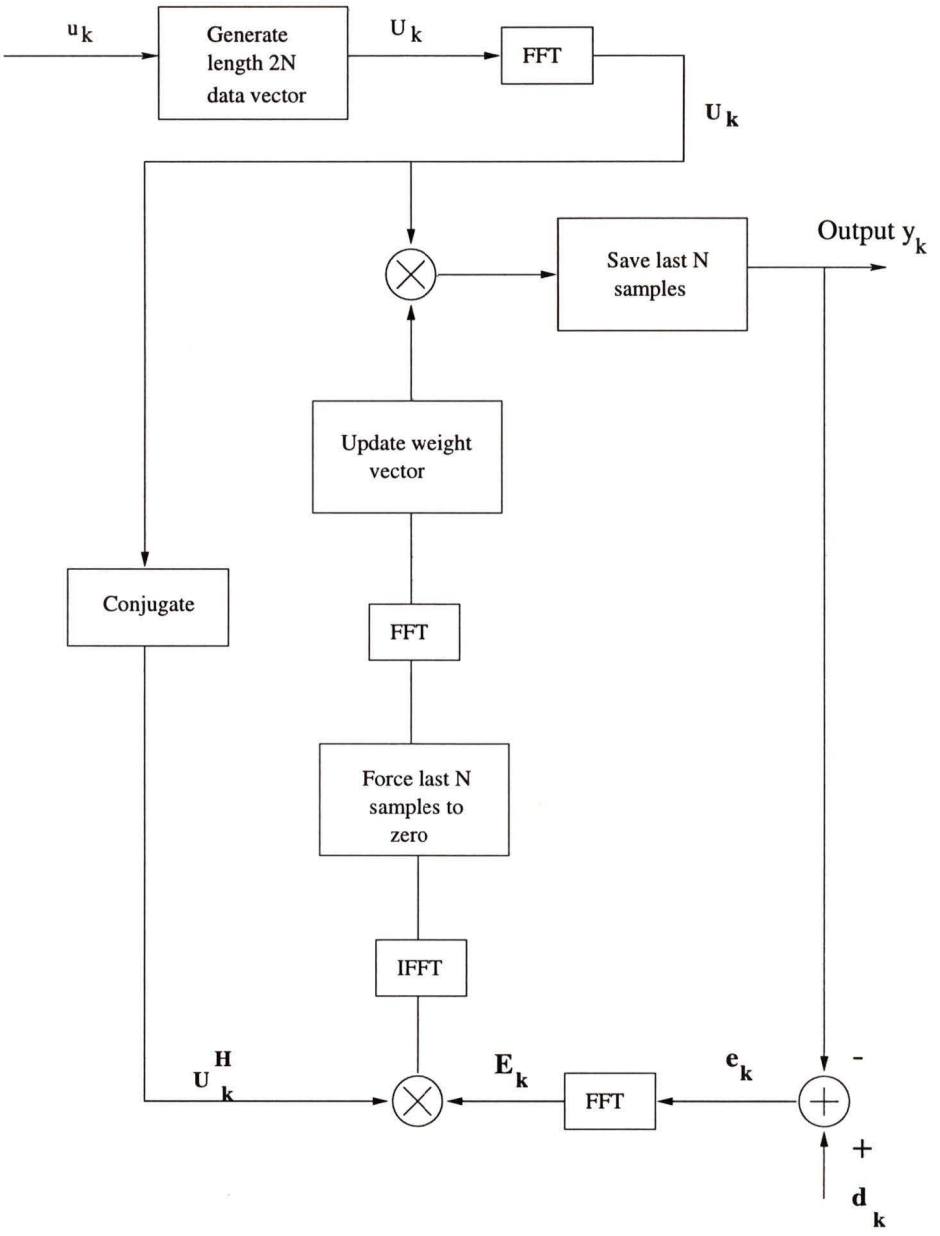

Figure 4.4. *Constrained FBLMS algorithm.*

In order to study the relation between the two FBLMS algorithms and the time-domain block-LMS algorithms, we note that the last N elements of $\mathbf{F}^{-1}\mathbf{Y}_k = \mathbf{F}^{-1}\mathbf{U}_k\mathbf{W}_k$ are actually the result of circular convolution of the two sequences $\mathbf{u}_k$ and $\mathbf{w}_k$. The

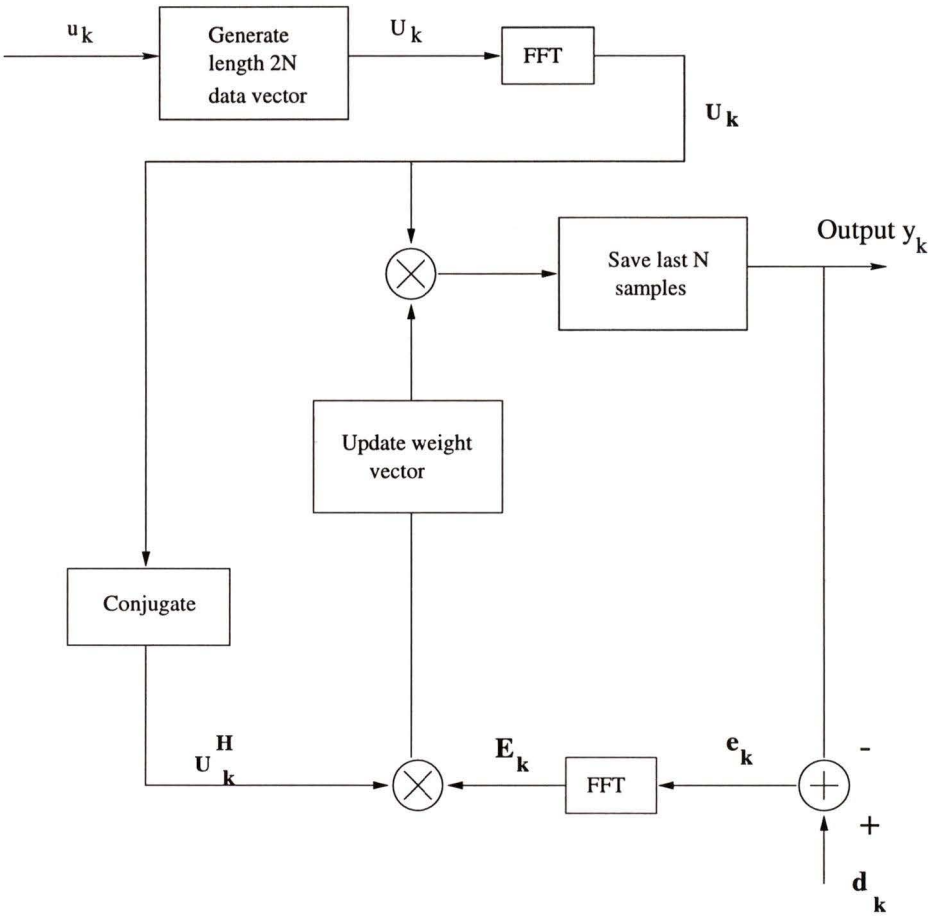

Figure 4.5. *Unconstrained FBLMS algorithm.*

output of the constrained FBLMS adaptive filter can now be written as [28]

$$\mathbf{F}^{-1}\mathbf{Y}_k = \mathbf{F}^{-1}\mathbf{U}_k\mathbf{W}_k = \chi_k\mathbf{w}_k \tag{4.19}$$

where

$$\chi_k = \begin{vmatrix} u(2Nk) & u(2Nk+2N-1) & \cdots & u(2Nk+2) & u(2Nk+1) \\ u(2Nk+1) & x(2Nk) & \cdots & u(2Nk+3) & u(2Nk+2) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ u(2Nk+2N-2) & u(2Nk+2N-3) & \cdots & u(2Nk) & u(2Nk+2N-1) \\ u(2Nk+2N-1) & u(2Nk+2N-2) & \cdots & u(2Nk+1) & u(2Nk) \end{vmatrix} \tag{4.20}$$

Since χ_k is a circulant matrix, it can be shown [28] that $\mathbf{F}\chi_k\mathbf{F}^{-1}$ is a diagonal matrix

and its diagonal elements are equal to the DFT of the column vector $\mathbf{x}_k$. That is,

$$\mathbf{U}_k = \text{diag}(\mathbf{F}\mathbf{x}_k) = \mathbf{F}\chi_k\mathbf{F}^{-1} \tag{4.21}$$

The error vector $\mathbf{E}_k$ in the frequency domain can be written as

$$\mathbf{E}_k = \mathbf{F}(\mathbf{d}_k - \mathbf{y}_k) = \mathbf{F}(\mathbf{d}_k - \mathbf{h}_l\mathbf{F}^{-1}\mathbf{U}_k\mathbf{W}_k) \tag{4.22}$$

where

$$\mathbf{H}_l = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{4.23}$$

Hence the coefficient update equation for the constrained FBLMS algorithm is

$$\mathbf{W}_{k+1,c} = \mathbf{W}_{k,c} + \mu\mathbf{F}\mathbf{H}_u\mathbf{F}^{-1}\mathbf{U}_k^H\mathbf{E}_k \tag{4.24}$$

where

$$\mathbf{H}_u = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{4.25}$$

and the corresponding coefficient update equation for the unconstrained FBLMS algorithm is

$$\mathbf{W}_{k+1,u} = \mathbf{W}_{k,u} + \mu\mathbf{U}_k^H\mathbf{E}_k \tag{4.26}$$

Substituting Eq. (4.24) into (4.25) and (4.26), we obtain

$$\mathbf{W}_{k+1,c} = \mathbf{W}_{k,c} + \mu\mathbf{F}\mathbf{H}_u\mathbf{F}^{-1}\mathbf{U}_k^H[\mathbf{D}_k - \mathbf{F}\mathbf{H}_l\mathbf{F}^{-1}\mathbf{U}_k\mathbf{W}_{k,c}] \tag{4.27}$$

$$\mathbf{W}_{k+1,u} = \mathbf{W}_{k,u} + \mu\mathbf{U}_k^H[\mathbf{D}_k - \mathbf{F}\mathbf{H}_l\mathbf{F}^{-1}\mathbf{U}_k\mathbf{W}_{k,u}] \tag{4.28}$$

In order to provide a relation to the time-domain block-LMS algorithm, we transform the two updating equations, namely, (4.27) and (4.28), into time domain by premultiplying with $\mathbf{F}^{-1}$, resulting in

$$\mathbf{w}_{k+1,c} = \mathbf{w}_{k,c} + \mu\mathbf{H}_u\chi_k^H[\mathbf{d}_k - \mathbf{H}_l\chi_k^H\mathbf{w}_{k,c}] \tag{4.29}$$

$$\mathbf{w}_{k+1,u} = \mathbf{w}_{k,u} + \mu\chi_k^H[\mathbf{d}_k - \mathbf{H}_l\chi_k^H\mathbf{w}_{k,u}] \tag{4.30}$$

In (4.29), the last N elements of the weight vector $\mathbf{w}_{k,c}$ are constrained to be zero. Hence the algorithm is equivalent to a length-N, block LMS adaptive filter. The unconstrained FBLMS algorithm, on the other hand, implements a length-$2N$ adaptive filter in the time domain and its data matrix $\chi_{k,u}^H$ is circularly generated from the reversed data vector. $\mathbf{x}_k$.

4.3.1 Computational Complexity

In this section, we compare computational complexity of the constrained and the unconstrained FBLMS algorithms with that of the conventional LMS algorithm. Since the number of multiplications involved forms a major portion of the complexity of an adaptive algorithm, in this analysis, we compare the number of multiplications required for each iteration of the constrained and unconstrained FBLMS and the conventional LMS algorithms.

Consider the conventional LMS algorithm of length N. In this case, N multiplications are performed to compute the output and an additional N multiplications are performed to update the coefficients. Thus a total of $2N$ multiplications per iterations are required. Hence in order to produce N output samples, the total number of multiplications required is $2N^2$. For the case of the constrained FBLMS algorithm, each $2N$-point FFT or IFFT operation requires approximately $2N \log_2 2N$ real multiplications. Hence five such operations account for $10N \log_2 2N$ multiplications. In addition, the computation of the frequency-domain output vector requires $8N$ multiplications and an additional $8N$ multiplications are required for the computation of the cross-correlations relating to the gradient vector estimation. Hence, the total number of multiplications required to produce N output samples is

$$10N \log_2 2N + 16N = 10N \log_2 N + 26N \tag{4.31}$$

Similarly, for the unconstrained FBLMS algorithm the number of multiplications required is

$$6N \log_2 2N + 16N = 6N \log_2 N + 22N \tag{4.32}$$

Therefore, the ratio of complexity for the constrained FBLMS to the conventional LMS algorithm is

$$\frac{5 \log_2 N + 13}{N} \tag{4.33}$$

and the ratio for the unconstrained FBLMS algorithm is

$$\frac{3 \log_2 N + 11}{N} \tag{4.34}$$

In the next section, simulations are given to compare the performance of the constrained and unconstrained FBLMS algorithms.

4.3.2 Simulations

Two different simulations have been performed on the two FBLMS algorithms. The first case considered is an echo cancellation setup with an echo path simulated using an FIR lowpass filter of length 128 resulting in a block size of 256. The constrained and the unconstrained FBLMS algorithms were coded using the standard block-set of SIMULINK in MATLAB. The SIMULINK models are shown in Fig. 4.6 and Fig. 4.7. The transmit signal was a white Gaussian noise with unity variance. The receive signal was chosen to produce a nominal SNR of 50 dB at the receiver. The step-size values were chosen to maximize the rate of convergence by the trial-and-error method which resulted in values of 0.0075 and 0.0055, respectively, for the constrained and unconstrained FBLMS algorithms. Fig. 4.8 shows the MSE versus iterations for the two FBLMS algorithms. It can be observed that both FBLMS algorithms converge to the same value of residual MSE and that the unconstrained FBLMS shows a slightly degraded performance in terms of the rate of convergence.

The second simulation was performed using the truncated echo path impulse response of CSA test loop #2 used in Sections 2.4.4 and 3.4.1. The impulse response was padded with zeros to result in a vector of length 128 in order for the adaptive filter length to be a power of 2, for ease in FFT and IFFT calculations. The step-size values for both the algorithms was chosen to be 0.035 which was the best step size obtained for the constrained FBLMS algorithm by trial and error. Fig. 4.9 shows the MSE versus the number of iterations for the two algorithms and we can notice that the unconstrained FBLMS algorithm tends to be unstable. This confirms the results in [28] that removing the gradient constraint from the update equations results in a reduced range of stable step-size values for the algorithm.

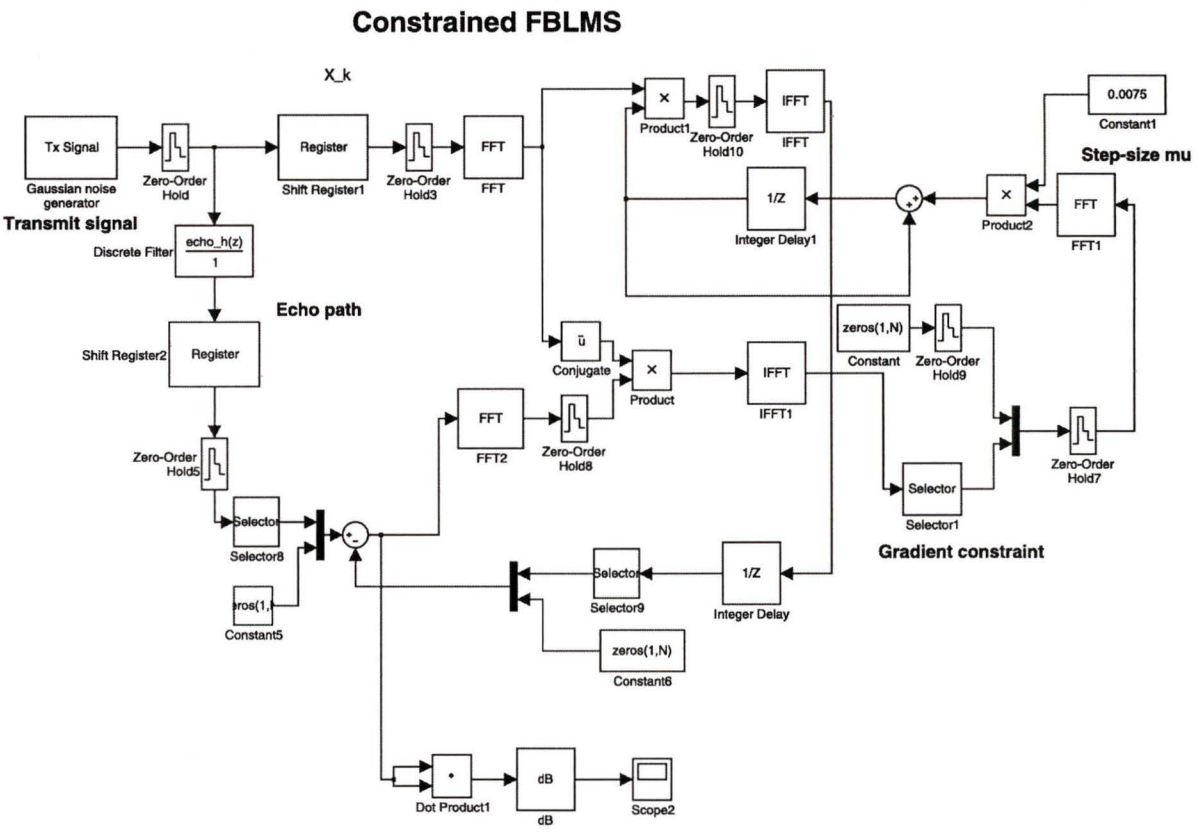

Figure 4.6. *Constrained FBLMS implementation in MATLAB SIMULINK.*

4.4 A Time-Domain Fast Exact Block-LMS Algorithm

The block-LMS algorithm given by Eq. (4.8) and the subsequent frequency-domain implementations of the algorithm described in Section 4.3 are not mathematically equivalent to the conventional LMS algorithm. Hence they suffer from problems such as reduced range of step size for stable performance [29]. In this section, we describe a time-domain block-LMS algorithm which is mathematically equivalent to the conventional LMS algorithm but differs only in terms of implementation.

In an exact-block algorithm, the filtering errors are computed at every time in-

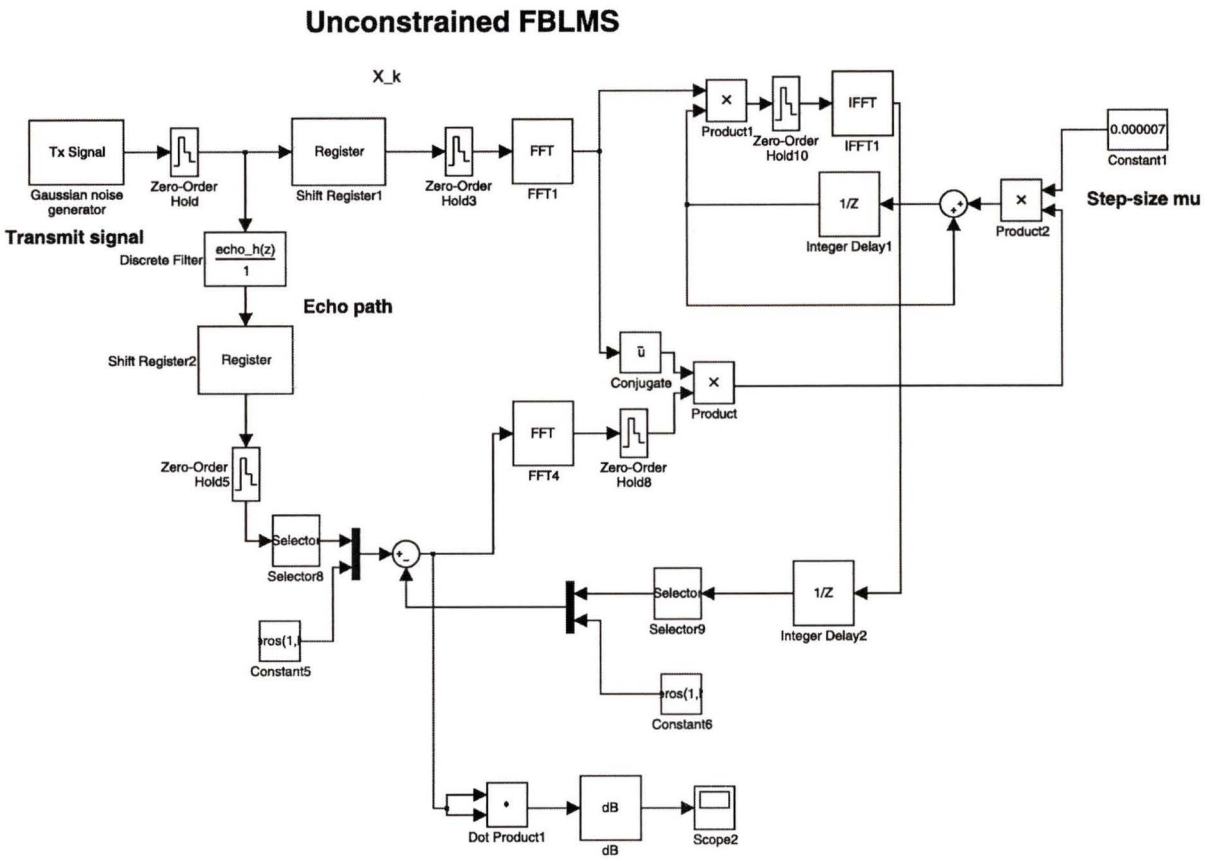

Figure 4.7. *Unconstrained FBLMS implementation in MATLAB SIMULINK.*

stant, while the coefficient vector is updated every M time steps, where M is the block length [30, 31]. However, these coefficient vectors are equal to the respective quantities computed by the corresponding conventional step-by-step adaptive algorithm. Thus, the same estimates are obtained at a substantially reduced complexity. Moreover, the block size need not be a factor of the filter length and hence the block size can be selected based on other requirements such as the size of memory, etc.

In this section, the coefficient vector at the nth sampling time instant is denoted by $\mathbf{w}(n)$. A similar notation is used for the input data vector $\mathbf{u}(n)$ and the error signal $e(n)$. In the fast-exact LMS(FELMS) algorithm, for a block of data of length M, the goal is to compute $\mathbf{w}(n)$ based on $\mathbf{w}(n-M)$, i.e., the coefficient vector estimate

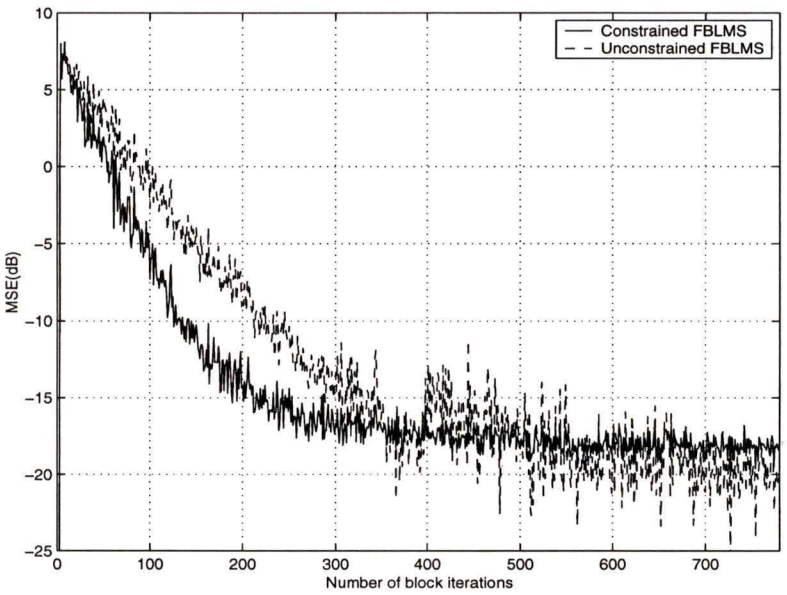

Figure 4.8. *Comparison of the MSE using the constrained and unconstrained FBLMS algorithms using step-size values determined by trial and error method.*

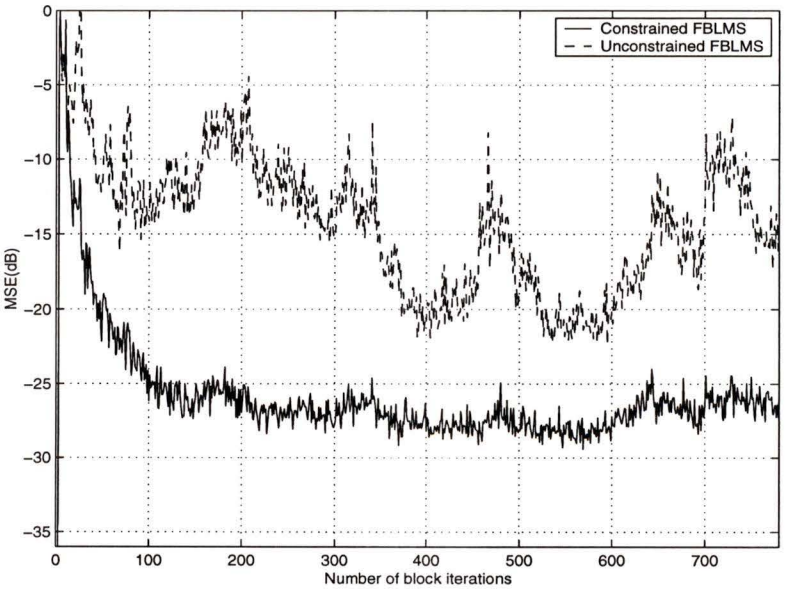

Figure 4.9. *Comparison of the MSE using constrained and unconstrained FBLMS algorithms using equal step-size values.*

corresponding to the beginning of the current block, and the input/output samples at all time instants $n, n-1, \ldots, n-M$, within the current data block.

By writing the equations for the error signal at time instants $n-M+1, \ldots, n$, we have

$$\begin{aligned} e(n-M+1) &= y(n-M+1) - \mathbf{u}^T(n-M+1)\mathbf{w}(n-M) && (4.35)\\ &\vdots && (4.36)\\ e(n) &= y(n) - \mathbf{u}(n)\mathbf{w}(n-1) && (4.37) \end{aligned}$$

Expressing the error at time instant n in terms of the coefficient vector at time instant $n-M$ in a matrix relation, we get [29]

$$\tilde{\mathbf{e}}(n) = \tilde{\mathbf{y}}(n) - \tilde{\mathbf{U}}(n)\mathbf{w}(n-M) - \mathbf{S}(n)\tilde{\mathbf{e}}(n) \tag{4.38}$$

where

$$\tilde{\mathbf{e}}(n) = [e(n-M+1)\cdots e(n)]^T$$

$$\tilde{\mathbf{y}}(n) = [y(n-M+1)\cdots y(n)]^T$$

and

$$\tilde{\mathbf{U}}(n) = [\mathbf{u}(n-M+1)\cdots \mathbf{u}(n)]$$

The matrix $\mathbf{S}(n)$ is given by

$$\mathbf{S}(n) = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ s_1(n-M+2) & 0 & \cdots & 0 & 0 \\ s_2(n-M+3) & s_1(n-M+3) & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ s_{M-1}(n) & s_{M-2}(n) & \cdots & s_1(n) & 0 \end{bmatrix} \tag{4.39}$$

where $s_i(n) = \mu\mathbf{u}^T(n)\mathbf{u}(n-i)$. Hence the error vector update equation is given by

$$\tilde{\mathbf{e}}(n) = \mathbf{G}(n)[\tilde{\mathbf{y}}(n) - \tilde{\mathbf{U}}^T(n)\mathbf{w}(n-M)] \tag{4.40}$$

where $\mathbf{G}(n) = [\mathbf{S}(n) + \mathbf{I}]^{-1}$. The coefficient update equation is given by

$$\mathbf{w}(n) = \mathbf{w}(n-M) + \mu\tilde{\mathbf{U}}(n)\tilde{\mathbf{e}}(n) \tag{4.41}$$

Eqs. (4.38) and (4.41) define the fast-exact LMS algorithm. The computational savings result from the efficient computation of the matrix and vector products $\tilde{\mathbf{U}}(n)\mathbf{w}(n-M)$ in Eq. (4.38) and $\tilde{\mathbf{U}}(n)\tilde{\mathbf{e}}(n)$ in Eq. (4.41). Reference [29] describes a computationally efficient implementation of the above products using FIR filters to calculate the matrix and vector products. No simulation studies have been performed on the fast-exact LMS algorithm since its performance is equivalent to that of the conventional LMS algorithm.

4.4.1 Computational Complexity

The computational complexity of the FELMS algorithm as given by [32] is significantly reduced compared with that of the LMS algorithm. The number of multiplications per iteration is given by

$$2\left(\frac{3}{2}\right)^{\nu}\frac{M}{N}+\left(\frac{3.2^{\nu}-5}{2}\right)+\left(\frac{1}{2}\right)^{\nu} \tag{4.42}$$

where $N=2^{\nu}$ is the length of the adaptive filter and the number of additions is given by

$$2\left[2\left(\frac{3}{2}\right)^{\nu}-1\right]\frac{M}{N}+4\left(\frac{3}{2}\right)^{\nu}+2(2^{\nu}-3) \tag{4.43}$$

4.5 LMS Algorithm Employing Partial Updates

In this section, we explore LMS algorithms that update only a portion of the coefficient vector at each sample time. Strictly speaking, such algorithms do not belong to the class of block-LMS algorithms since updating of the coefficient vector is done at each sample time. In block-LMS algorithms, typically, the entire coefficient vector is updated after a block of input data is accumulated. However, some of the LMS algorithms employing partial updates have close correspondence to the time-domain block-LMS algorithms described previously and hence are considered in this chapter.

The main motivation behind the utlilization of the adaptive algorithms employing partial updates is to reduce computational complexity for long adaptive filters. For example, an LMS-based, N-coefficient, echo canceller employing N multiplication/addition, N data signal read, N coefficient read, and N coefficient write operations could require expensive processor and memory to meet the computational

requirements for large N and high sampling rates [33, 34]. The LMS algorithm employing a partial update strategy, however, can be used to reduce the computational requirements in such a case without a major change to the VLSI architectures of the conventional LMS algorithms. This is in contrast to the subband LMS algorithms described in Chapter 2 that operate in parallel, and are useful for high sampling rates since the coefficient updating is performed at subsampled rates, but require considerable changes to the architecture of the conventional LMS implemenations.

Two of the most popular partial update LMS algorithms are the *periodic LMS algorithm* and the *sequential LMS algorithm* [35]. The coefficient update equation in the periodic LMS algorithm is given by

$$w_{i,k+1} = \begin{cases} w_{i,k} + \mu e_l u_{l-i+1} & \text{if } (k+i) \bmod L = 0 \text{ and } l = L\lfloor k/L \rfloor \\ w_{i,k} & \text{otherwise} \end{cases} \tag{4.44}$$

where $\mathbf{w}_k = [w_{1,k} \;\; w_{2,k} \;\; \ldots \;\; w_{N,k}]^T$ is the coefficient vector of the adaptive filter at sampling time instant k and $\lfloor P \rfloor$ denotes the truncation operator on the real number P. It can be noticed from the above equation that a fraction of the coefficients are updated each iteration in a *round-robin* fashion. Hence for a round-robin factor of L, all the coefficients are updated once in every L iterations. For $L = 1$, the algorithm becomes the conventional LMS algorithm. For $L > 1$, the number of multiplications and coefficient memory accesses required are fewer than those required by the conventional algorithm. Comparing the update in Eq. (4.44) with Eq. (3.1), we observe that the conventional LMS algorithm uses the current value of the error signal e_k for the coefficient update; the periodic LMS algorithm, on the other hand, uses several past few error samples e_l for the coefficient update process. Hence, an additional memory for the error signal is required for implementing this algorithm. By considering L iterations of the updates in Eq. (4.44), it can be shown that the update equation of this algorithm is mathematically equivalent to the following update equation

$$\mathbf{w}_{k+L} = \mathbf{w}_k + \mu e_k \mathbf{u}_k \tag{4.45}$$

where $\mathbf{u} = [u_k \;\; u_{k-1} \;\; \ldots \;\; u_{k-N+1}]^T$ is the input data vector. Hence the periodic LMS algorithm is an alternative implementation of the conventional LMS algorithm in which the coefficient updates are performed once in every L iterations.

The *sequential LMS algorithm* is characterized by the equation

$$w_{i,k+1} = \begin{cases} w_{i,k} + \mu e_k u_{k-i+1} & \text{if } (k-i+1) \bmod L = 0 \\ w_{i,k} & \text{otherwise} \end{cases} \tag{4.46}$$

For $L = 1$, the algorithm reduces to the conventional LMS algorithm. Similar to the periodic LMS algorithm, this algorithm employs a regular processing strategy in which an equal number of coefficients (N/L) are updated in each iteration and hence the computational complexity remains constant for each iteration. However, the sequential LMS algorithm is not mathematically equivalent to Eq. (4.45). Table 4.1 gives the number of multiplications involved in the conventional, periodic, and sequential LMS algorithms.

Table 4.1. *Comparison of complexities for the conventional, periodic, and sequential LMS algorithms(number of operations per iteration).*

Algorithm	*Multiplications*	*Additions*	*Data memory access*
Conventional LMS	$2N$	2N	2N
Periodic LMS	$\frac{1}{L}(2N+1)+\frac{1}{L}$	$\frac{2N}{L}$	$2N+1$
Sequential LMS	$\left(1+\frac{1}{L}\right)N+1$	$\left(1+\frac{1}{L}\right)N$	2N

4.5.1 Simulations

The truncated echo path impulse response of CSA loop # 2 was again used to simulate the echo path in an echo cancellation setup. Both the periodic and sequential LMS algorithms were simulated in MATLAB SIMULINK as *s-functions* with the same value for the step-size parameters. The convergence behaviour of the algorithms was found to be very similar for the same value of L and step size. Hence only one algorithm, the periodic LMS, was chosen for comparison with the conventional algorithm. It has been observed that for $L = 1$, the convergence performance of the periodic LMS algorithm matches that of the conventional LMS algorithm as expected. For $L = 2$ and $L = 4$, Fig. 4.8 shows the mean MSE in dB for 10 independant runs of the periodic and conventional LMS algorithms. It can be observed that while the conventional LMS algorithm converges to -60 dB in aproximately 2500 iterations, the

periodic LMS algorithm takes approximately 5000 and 10000 iterations with $L = 2$ and $L = 4$, respectively, for convergence of the mean MSE to -60 dB. This confirms that the periodic and sequential LMS algorithms decrease the computational complexity by a factor of L; however, these algorithms also reduce the convergence rate simultaneously by the same factor [35]. In the next section we describe an algorithm that takes advantage of the partial coefficient update method while achieving a rate of convergence which is comparable to that of the conventional LMS algorithm.

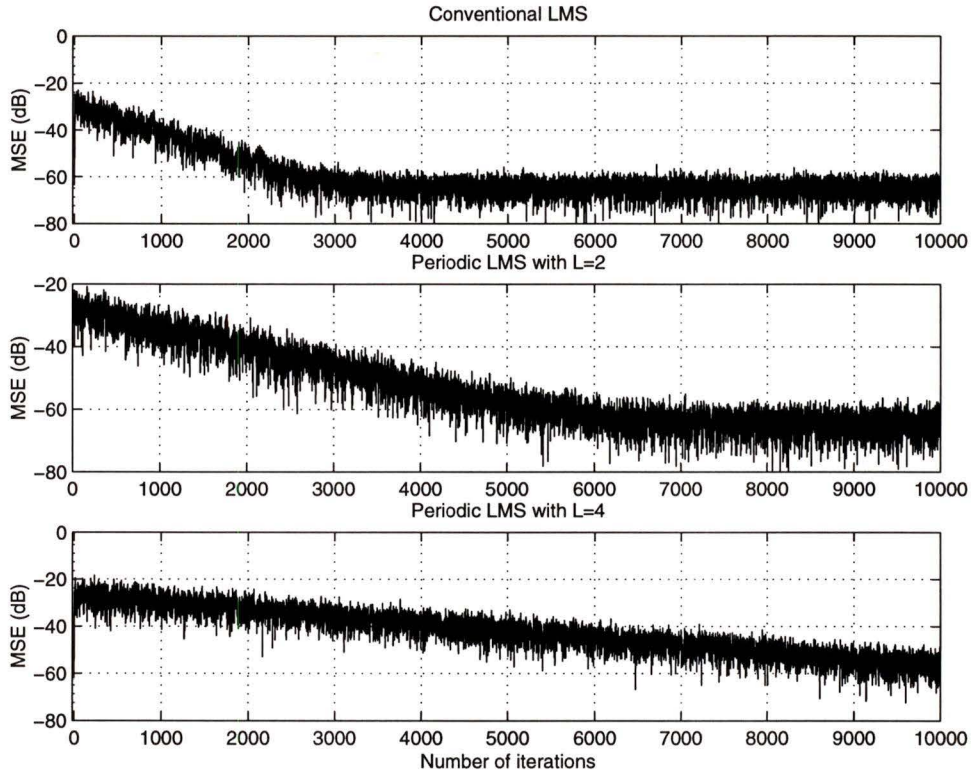

Figure 4.10. *Comparison of the performance of the periodic LMS algorithm with that of the conventional LMS algorithm.*

4.6 Time-Domain Inexact BLMS Algorithm Employing Partial Updates

In the previous section, we considered the periodic LMS algorithm whose L iterations are mathematically equivalent to Eq. (4.45). It has been demonstrated by means of simulations that such an approach would reduce the convergence rate by the same ratio by which the computational complexity is reduced. In this section, we consider the partial update implementation of the block LMS update equation described by Eq. (4.8). Evidently, such an equation would be

$$w_{i,k+1} = \begin{cases} w_{i,k} + \mu \sum_{k=(i-1)L+1}^{k=iL} e_k u_{k-i+1} & \text{if } (k-i+1) \bmod L = 0 \\ w_{i,k} & \text{otherwise} \end{cases} \tag{4.47}$$

From the above equation, we can notice that the number of multiplications required for each coefficient update is L. The number of coefficients being updated in each iteration is L/N. Hence the total number of multiplications in each iteration is $L \times N/L = N$, which is the same as the number of multiplications per iteration of the conventional LMS algorithm. However, in the inexact approach, instead of using L past error samples, only M past error samples are used in the update equation, where M is typically chosen to be less than L. Hence the number of multiplications required per iteration in the inexact approach is $M \times N/L$ which is less than N.

4.6.1 Simulations

In order to study the effects of the inexact approach to the BLMS algorithm, echo cancellation simulations have been performed with the echo path chosen to be the same as that used in Section 4.5.1. Two different SNR scenarios, i.e., SNR $= \infty$ and SNR $= 25$ dB, have been considered. The computational complexity $\mathcal{F}$ of a particular combination of M and L has been calculated as the fraction of the number of flops required for the same number of iterations of the conventional LMS algorithm. For the purpose of quantifying the rate of convergence, the number of iterations required for the conventional LMS algorithm to reach a misadjustment of -30 dB has been used as reference. The rate of convergence $\mathcal{C}$ has been calculated as the percentage increase in number of iterations required for the algorithm to reach the -30 dB level

of misadjustment. A uniform step size was chosen for all the algorithms in order for the effect of the step-size to be factored out of the results. Table 4.2 shows the simulation results for various combinations of M and L. A value of ∞ for $\mathcal{C}$ denotes that the algorithm converges to a misadjustment larger than -30 dB. It can be observed that in the case of infinite SNR, the inexact algorithm offers a large amount of savings in computation with insignificant degradation in the rate of convergence. However, for a nominal SNR of 25 dB, the degradation in the rate of convergence with respect to the reduction in computational complexity is significant. Hence, the inexact BLMS algorithm can be used with huge savings in computational complexity, in the absence of a receive signal resulting in an infinity SNR. Such a scenario is typical in communication systems where a *silent* training period is available for the convergence of the echo canceller.

Table 4.2. *Computational complexity and convergence comparison.*

M	L	$\mathcal{F}$	$\mathcal{C}$ (SNR = ∞)	$\mathcal{C}$ (SNR = 25 dB)
4	16	0.3762	15%	∞
8	16	0.5386	6%	16%
4	32	0.2034	12%	∞
8	32	0.2864	5%	46%
16	32	0.4467	5%	18%
4	64	0.1169	100%	∞
8	64	0.1575	20%	∞
16	64	0.2386	5%	91%
32	64	0.4002	-12%	2%
48	64	0.5612	-14%	-5%

4.7 Conclusions

Block adaptive filter algorithms have been described. The constrained and unconstrained frequency-domain block LMS (FBLMS) algorithms have been explained in Section 4.3 and simulated using MATLAB SIMULINK. The relationship between the

FBLMS algorithms and the time-domain BLMS algorithms has been given. The constrained FBLMS algorithm was found to be computationally more intensive than the unconstrained FBLMS but also has a better rate of convergence as well as a larger range of stable step-size values.

A time-domain fast exact implementation of the block LMS algorithm has been given in Section 4.4. Two popular LMS algorithms employing partial coefficient update schemes, the periodic and sequential LMS algorithms, have been explained and simulated in Section 4.5. Comparison studies of the algorithms employing partial update schemes have shown that the rate of convergence is reduced by approximately the same factor by which the computational complexity is reduced. In order to overcome this tradeoff, the time-domain inexact LMS algorithm employing partial updates has been introduced in Section 4.6. This algorithm has been found to reduce the computational complexity by a large amount in an echo cancellation setup in the absence of a receive signal, without decreasing the rate of convergence. However, the rate of convergence has been found to deteriorate as the receive signal power is increased and hence we can conclude that this algorithm can be used with advantage in echo cancellers where a silent training period is available.

Chapter 5

Conclusions and Suggestions for Further Work

5.1 Conclusions

The work presented in this thesis is the result of investigation into various LMS-based adaptive algorithms and their performance in an xDSL echo cancellation environment. Extensive simulations of echo cancellation using the algorithms under study have been performed, described, and analyzed.

The procedure for obtaining the echo impulse responses for the CSA loops has been described in Chapter 2. The input impedances as well as the echo path impulse responses for the 8 CSA test loops for HDSL2 have been given. The echo path impulse response for the test loops considered has been found to be approximately 130 μs long.

Subband echo cancellation has been introduced and simulation results have been given in Chapter2. The subband LMS echo canceller was found to exhibit a poorer rate of convergence than the conventional LMS echo canceller. A new method of design of the filter banks, which can be used in a subband echo canceller without cross adaptive filters, has been given. Filter banks designed using the proposed method have been found to be superior to the filter banks designed using the existing methods in literature.

Various exisiting variable step-size algorithms, namely, the VSS and the MVSS LMS algorithms, have been described in Chapter 3 and simulations results have been given to demonstrate their performance. The MVSS algorithm exhibits superior performace over the VSS algorithm in low SNR scenarios. However, in the case of long adaptive filters, the MVSS algorithm results in a large final misadjustment. A

new VSS algorithm has been proposed to overcome this problem and the proposed VSS algorithm has demonstrated superior performance over the MVSS algorithm in the case of long adaptive filters. The performance of the proposed VSS algorithm was also found to be satisfactory in a scenario where an abrupt change in the echo path impulse response occurs.

The application of the proposed algorithm to subband LMS adaptive filtering has been investigated. Simulations results show that the proposed algorithm yields a lower steady-state misadjustment in each subband as well as a lower residual MSE when compared with the FSS subband LMS algorithm. It has also been observed that improved system performance can only be achieved with a distinct step-size adaptation for each individual subband.

Block adaptive filter algorithms have been described in Chapter 4. The constrained and unconstrained frequency-domain block LMS (FBLMS) algorithms have been explained and simulated using MATLAB SIMULINK. The relationship between the FBLMS algorithms and the time-domain BLMS algorithms has been given. The constrained FBLMS algorithm was found to be computationally more intensive than the unconstrained FBLMS but also has better rate of convergence as well as a larger range of stable step-size values.

Two popular LMS algorithms employing partial coefficient update schemes, the periodic and sequential LMS algorithms, have been explained and simulated. Comparison studies of the algorithms employing partial update schemes have shown that the rate of convergence is reduced by approximately the same factor by which the computational complexity is reduced. In order to overcome this tradeoff, the time-domain inexact LMS algorithm employing partial updates has been introduced. This algorithm has been found to reduce the computational complexity by a large amount in an echo cancellation setup in the absence of a receive signal, without decreasing the rate of convergence.

5.2 Suggestions for Further Work

The subband echo cancellation schemes considered have been limited to two-band architectures where the input and the echo are decomposed into high- and low-frequency

components. A study of the effect of filter bank characteristics on multi-band architectures would be interesting. A study of subband echo cancellation using other popular adaptive algorithms such as *recursive least-squares* (RLS) algorithm will be beneficial in gaining a better understanding of the advantages of subband based architectures.

Variable step-size LMS algorithms have been studied in this thesis with application to echo cancellation architectures only. Application of the variable step-size algorithms to other architectures such as adaptive equalization and comparison with the performance in echo cancellation architectures would be more insightful.

The frequency- and time-domain block-LMS algorithms have been studied with the usage of a fixed step-size. An investigation into variable step-size implementations of the block-LMS algorithms would be useful and interesting, although such a study is expected to be quite involved.

The algorithms considered in this thesis have been compared with each other using the rate of convergence and compuational complexity as the main criteria. However, in practice, the suitability of an algorithm's structure for VLSI implementations is also of importance. Therefore, inclusion of comparisons of various algorithms in terms of their amenability toVLSI implementations would certainly make the simulation studies more valuable.

Bibliography

[1] H. Cravis and T. V. Crater, "Engineering of T1 carrier system repeatered lines," *Bell Syst. Tech. J.*, vol. 42, pp. 431-486, Aug. 1991.

[2] J. G. Proakis, *Digital Communications*, McGraw-Hill, 1995.

[3] M. Sondhi and D. A. Berkely, "Silencing echos on the telephone network," *Proc. IEEE*, vol. 68, pp. 948-963, Mar. 1991.

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

[5] J. A. C. Bingham, *The Theory and Practice of Modem Design*, Wiley, 1985.

[6] W. Y. Chen, *DSL: Simulation Techniques and Standards Development for Digital Subscriber Line Systems*, Indianapolis, IN, Macmillan Techinical Publishing, 1998.

[7] "Special issue on high bit-rate digital subscriber lines," *IEEE J. Sel. Areas Commun.*, vol. 9, Aug. 1991.

[8] W. Y. Chen, J. L. Dixon, and D. L. Waring, "High bit rate digital subscriber line echo cancellation," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 6, pp. 848-860, Aug. 1991.

[9] S. J. Orfanidis, *Optimum Signal Processing*, New York, Macmillan, 1988.

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

[11] S. Haykin, *Adaptive Filter Theory*, New Jersey, Prentice Hall, 1996.

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

[13] A. Gilloire and M. Vetterli, "Adaptive filtering in subbands with critical sampling: analysis, experiments, and applications to acoustic echo cancellation," *IEEE Trans. on Signal Processing*, vol. 40, pp. 1862-1875, Aug. 1992.

[14] W.-S. Lu, *Lecture Notes: ELEC 639A*, University of Victoria, Victoria, 1999.

[15] S. Hosur and A. H. Tewfik, "Wavelet transform domain adaptive filtering," *IEEE Trans. on Signal Processing*, vol. 45, pp. 617-630, Mar. 1997.

[16] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley, MA: Wellesley-Cambridge Press, 1996.

[17] Q. Jin, K. M. Wong, and Z. Q. Luo, "Design of an optimum wavelet for cancellation of long echoes in telephony," *Proc. IEEE-SP Time-Frequency and Time-Scale Analysis*, pp. 488-491, Oct. 1994.

[18] W.-S. Lu and A. Antoniou, "Optimized orthogonal and biorthogonal wavelets using linear parameterization of halfband filters," *Proc. ISCAS* '98, vol. 5, pp. 98-101, May 1998.

[19] R. H. Kwong and E. W. Johnston, "A variable step size LMS algorithm," *IEEE Trans. Signal Processing*, vol. 40, no. 7, pp. 1633-1642, July 1992.

[20] S. B. Gelfand, Y. Wei, and J. V. Krogmeier, "The stability of variable step-size LMS algorithms," *IEEE Trans. on Signal Processing*, vol. 47, no. 12, pp. 3277-3288, Dec. 1999.

[21] S.-S. Ahn and P. J. Voltz, "Convergence of the delayed normalized LMS algorithm with decreasing step-size," *IEEE Trans. on Signal Processing*, vol. 44, no. 12, pp. 3008-3016, Dec. 1996.

[22] T. Aboulnasr and K. Mayyas, "A robust variable step-size LMS-type algorithm: analysis and simulation," *IEEE Trans. Signal Processing*, vol. 45, no. 3, pp. 631-639, Mar. 1997.

[23] P. Sristi, W.-S. Lu, and A. Antoniou, "Design and application of an optimum orthogonal wavelet filter for echo cancellation," *Proc. PACRIM* '99, pp. 79-82, Aug. 1999.

[24] D. Mansour and A. H. Gray. Jr, "Unconstrained frequency-domain adaptive filter," *IEEE Trans. Acoust. Speech, Sig. Proc.*, vol. ASSP-30, pp. 726-734, Oct. 1982.

[25] J. J. Shynk, "Frequency-domain and multi-rate adaptive filtering," *IEEE Signal Processing Mag.*, vol. 9, no. 1, pp. 14-37, Jan. 1992.

[26] G. A. Clark and S. K. Mitra, "Block implementation of adaptive digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 584-592, June 1981.

[27] N. K. Jablon, "On the complexity of frequency-domain adaptive filtering," *IEEE Trans. on Signal Processing*, vol. 39, pp. 2331-2334, Oct. 1991.

[28] X. Li and W. K. Jenkins, "The comparison of constrained and unconstrained frequency-domain block-LMS adaptive algorithms," *IEEE Trans. on Signal Processing*, vol. 44, no. 7, pp. 1813-1816, July 1996.

[29] G.-O. Glentis, K. Berberidis, and S. Theodoridis, "Efficient least-square adaptive algorithms for FIR tranversal filters," *IEEE Signal Processing Mag.*, July 1999.

[30] K. Berberidis and S. Theodoridis, "A new fast block adaptive algorithm," *IEEE Trans. on Signal Processing*, vol. 47, no. 1, pp. 75-87, Jan. 1999.

[31] J. Benesty and P. Duhamel, "A fast exact LMS adaptive algorithm," *IEEE Trans. on Signal Processing*, vol. 40, no. 12, pp. 2904-2920, Dec. 1992.

[32] P. Montazeri and P. Duhamel, "A set of algorithms linking NLMS and block RLS algorithms," *IEEE Trans. on Signal Processing*, vol. 43, no. 2. pp. 444-453, Feb. 1995.

[33] H. Herzberg, R. Haimi-Cohen, and Y. Be'ery, "A systolic array realization of an LMS adaptive filter and the effects of delayed adaptation," *IEEE Trans. on Signal Processing*, vol. 40, no. 11, pp. 2799-2803, Nov. 1992.

[34] C.-L. Wang, "Bit-serial VLSI implementation of delayed LMS adaptive FIR filters," *IEEE Trans. on Signal Processing*, vol. 42, no. 8, pp. 2169-2175, Aug. 1994.

[35] S. C. Douglas, "Adaptive filters employing partial updates," *IEEE Trans. Circuits Syst.-II: Analog and Digital Signal Processing*, vol. 44, no. 3, pp. 209-216.

[36] B. Farhang-Boroujeny, "Analysis and efficient implementation of partioned block LMS adaptive filters," *IEEE Trans. on Signal Processing*, vol. 44, no. 11, pp. 2865-2868, Nov. 1996.

VITA

Surname: Sristi ***Given Names:*** Prasad

Place of Birth: India ***Date of Birth:*** August 17, 1976

Educational Institutions Attended

University of Victoria	1998 to 2003
Indian Institute of Technology, Bombay	1993 to 1997

Degrees Awarded

B.Tech	Indian Institute of Technology, Bombay	1997

Conference Publications

1. P. Sristi, W.-S. Lu, and A. Antoniou, "Design and application of an optimum orthogonal wavelet filter for echo cancellation," *Proc. PACRIM* '99, pp. 79-82, Aug. 1999.
2. P. Sristi, W.-S. Lu, and A. Antoniou, "A new variable-step-size adaptive LMS algorithm and its application in subband adaptive filtering for echo cancellation," *IEEE ISCAS*, 2001

PARTIAL COPYRIGHT LICENSE

I hereby grant the right to lend my dissertation to users of the University of Victoria Library, and to make single copies only for such users or in response to a request from the Library of any other university, or similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of this dissertation for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this dissertation for financial gain shall not be allowed without my written permission.

Title of Dissertation: ADAPTIVE LMS-BASED ALGORITHMS FOR ECHO CANCELLATION IN TELECOMMUNICATIONS.

Author: 

PRASAD SRISTI
July 2003