TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY
Julkaisu 470 • Publication 470

Radu Ciprian Bilcu

On Adaptive Least Mean Square FIR Filters:
New Implementations and Applications

Tampere 2004

Tampereen teknillinen yliopisto. Julkaisu 470
Tampere University of Technology. Publication 470

Radu Ciprian Bilcu

On Adaptive Least Mean Square FIR Filters:
New Implementations and Applications

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB 103, at Tampere University of Technology, on the 23rd of June 2004, at 12 o´clock noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2004

ISBN 952-15-1178-8 (printed)
ISBN 952-15-1823-5 (PDF)
ISSN 1459-2045

Abstract

Among many adaptive algorithms that exist in the open literature, the class of approaches which are derived from the minimization of the mean squared error between the output of the adaptive filter and some desired signal, seems to be the most popular. Probably the simplest algorithm belonging to this class is the Least Mean Squared (LMS) algorithm which has the advantage of low complexity and simplicity of implementation. One of the main concerns in all practical situations is to develop algorithms which provide fast convergence of the adaptive filter coefficients and in the same time good filtering performance. There are four main classes of applications where the adaptive filters were applied with success, namely: system identification, inverse modeling, prediction and interference canceling. In this thesis we develop new algorithms for the first two classes of applications although they can be implemented also for prediction and interference canceling.

In this thesis several new algorithms for adaptive filtering are introduced. The main goal is to improve the performances of the existing algorithms, in terms of convergence speed and filtering performance and also to introduce some new approaches. The new algorithms are classified into several classes each of them addressing a certain application.

It is well known that the LMS algorithm has a slow convergence for correlated inputs. Moreover its filtering performance and convergence speed are inversely related through a single parameter, the step-size. An adaptive filter implementing the LMS might have stability problems operating in a non-Gaussian environment due to the use of instantaneous gradient to update the coefficients. In applications belonging to the class of system identification, not only the values of the coefficients of the model are of interest, but also the length of the model. Therefore algorithms for length adaptation might be of equal interest. Another situation, that can appear in identification applications is when the coefficients of the model are time-varying. The adaptive algorithm should provide a mechanism to track the changes of the model.

This thesis contains three main parts which are concentrated on time domain implementations, transform domain implementations and applications respectively. At the beginning of the first part two new variable step-size LMS algorithms are introduced which show good convergence speed. The dependence between the speed of adaptation and filtering performance is reduced and the setup of the parameters is very easy as compared with other existing approaches. The problem of length estimation is addressed later on and an algorithm to iteratively adjust the length of the adaptive filter toward the length of the model is proposed. This algorithm is derived for system identification application. Next, the problem of tracking time-varying systems is discussed and the analytical expressions for the steady-state mean squared error and mean squared coefficient error are revised. Based on these expressions a new algorithm in which the step-size is iteratively modified toward the optimum is introduced. An important feature of the proposed algorithm is the fact that the user does not need to know any information about the statistics of the optimum model.

At the end of the first part the class of order statistics LMS algorithms is discussed and a new algorithm belonging to this class is introduced. The new algorithm uses an adaptive filter to smooth the gradient such that it does not require the knowledge of the noise distribution in order to be implemented.

The second part of the thesis is dedicated to the transform domain implementation of the LMS algorithm and its variants. First, three new algorithms belonging to the class of variable step-size LMS algorithm in transform domain are introduced. To the best of our knowledge the idea of step-size adaptation in transform domain, based on the output error was not addressed so far in the open literature. The existing approaches, assume a time-varying step-size due to the power estimates of the transform coefficients, whereas in our implementations, the step-size is adapted by the output error. We continue with the problem of time-varying modeling using the transform domain LMS and we introduce a new algorithm. The aim of this algorithm is to increase the convergence speed of its time domain counterpart and also to reduce its complexity.

At the end of the second part the scrambled LMS is briefly presented and it is compared with the LMS and transform domain LMS. The chosen framework is the digital data transmission over a telephone line. The analytical expressions of the mean squared error and mean squared coefficient error are derived for this special case and a discusion about their convergence speed and steady-state error is given. The aim of this discusion is to provide some useful information about the utility of each algorithm.

In the first two parts of the thesis, computer experiments, showing the performances of all the proposed algorithms, are provided for system identification application. Since many of the algorithms can be implemented also in other applications, in the third part of this thesis, channel equalization, CDMA multiuser detection and echo cancellation applications are also addressed.

Acknowledgments

The work presented in this thesis has been carried out at the Institute of Signal Processing from Tampere University of Technology, Finland.

My sincere gratitude goes to Prof. Pauli Kuosmanen, for giving me the chance to work in the great atmosphere at the Institute of Signal Processing, also for his supervision and entire support during my work. I want to express my special thanks to Prof. Karen Egiazarian for his supervision during this work and for many excellent ideas and fruitful discussions.

I am grateful to Prof. Corneliu Rusu from Technical University of Cluj-Napoca, Romania, the person who introduced me for the first time in the wonderful world of signal processing.

I would like to warmly thank Prof. Jaakko Astola and the personnel from Tampere International Center for Signal Processing for the opportunity to attend excellent seminars and workshops. I wish to thank my colleagues from Institute of Signal Processing for the creative atmosphere which has been a continuous source of inspiration for me. The financial support provided by Tampere Graduate School in Information Science and Technology (TISE) is greatfully acknowledged.

I wish to thank the noisy gang from the E corridor for keeping me awake, especially Prof. Tapio Saramäki, the head of the "Allergy Group".

I would also like to thank the Romanian community in Tampere, especially: Maria, Ion, Daniela and Marian Crisan for being my family here in Finland. My thanks goes to Anca, Simona, Florin and Mircea for their fun company and for being my friends.

I am grateful to my parents Eugenia and Mircea, my brother Alin and my sister-in-law Elena, as well as to my parents-in-law Viorica and Andrei for their endless love, continuous support and for being my family. I will always be indebted to the surgeons who saved my mother's life in March 2003. My warmest thanks go to my wonderful wife Enikö Beatrice for her continuous love, support and never ending patience as well as for her great help in my work. Without you Betty I could have never achieved this important goal of my life.

Tampere, December 2003 Radu Ciprian Bilcu

Contents

List of Figures

List of Tables

Abbreviations

AGC	Automatic Gain Control
AFIR	Adaptive Finite Impulse Response
ALMS	Average Least Mean Square
AOSLMS	Adaptive Order Statistics Least Mean Square
CDMA	Code Division Multiple Access
CP-LMS	Complementary Pair Least Mean Square
CP-VSLMS	Complementary Pair Variable Step-size Least Mean Square
DCT	Discrete Cosine Transform
FIR	Finite Impulse Response
IID	Independent Identical Distributed
IIR	Infinite Impulse Response
KLT	Karhunen-Loeve Transform
LMS	Least Mean Square
MAI	Multiple Access Interference
MLMS	Median Least Mean Square
MSC	Mean Squared Coefficients error
MSE	Mean Squared Error
MxLMS	Trimmed Mean Least Mean Square
NBI	Narrow Band Interference
NCVSLMS	Noise Constrained Variable Step-size Least Mean Square
NEE	Normalized Estimation Error
OS	Order Statistics
OSLMS	Order Statistics Least Mean Square
OxLMS	Outer mean Least Mean Square
RVSLMS	Robust Variable Step-size Least Mean Square
SCLMS	Scrambled Least Mean Square
SNR	Signal-to-Noise Ratio
TDCPVSLMS	Transform Domain Complementary Pair Variable Step-size Least Mean Square
TDLMS	Transform Domain Least Mean Square
TDNCVSLMS	Transform Domain Noise Constrained Variable Step-size Least Mean Square

TDVSLMS	Transform Domain Variable Step-size Least Mean Square
VLLMS	Variable Length Least Mean Square
VSSLMS	Variable Step-Size Least Mean Square

Notations

$\mathbf{h}$	a vector with constant elements
$\mathbf{h}(n)$	a vector with time-varying elements
$\widehat{\mathbf{h}}(n)$	the estimate of the time-varying vector $\mathbf{h}(n)$
$\mathbf{h}^t$	the transpose of the vector $\mathbf{h}$
$\Delta\mathbf{h}(n)$	the difference of two vectors
$\mathbf{h}_o$	the optimum vector
$\|\|\mathbf{h}\|\|$	the norm of the vector $\mathbf{h}$
$\mathbf{h}_N$	a vector of length N
$\mathbf{R}$	a constant matrix
$\mathbf{R}(n)$	a time-varying matrix
$\mathbf{R}^{-1}$	the inverse of a matrix
$\mathbf{I}$	the identity matrix
$tr\,[\mathbf{A}]$	the trace of matrix $\mathbf{A}$
$\mathbf{R}_{N\times M}$	a matrix with dimensions $N \times M$
y	a scalar
$y(n)$	a time-varying scalar
$\widehat{y}(n)$	an estimate of a time-varying scalar
σ_v^2	the variance of $v(n)$
μ	a fixed step-size
$\mu(n)$	a time-varying step-size
$\nabla J(n)$	the gradient of $J(n)$
$E\{A\}$	the expected value of A
$\mathcal{M}$	the steady-state misadjustment

Chapter 1

Introduction

During the last decades the adaptive filters have attracted the attention of many researchers due to their property of *self-designing* [41]. In applications where some a priori information about the statistics of the data is available, a linear filter optimal for that application can be designed in advance (e.g. the Wiener filter which minimizes the mean squared error between the output of the filter and some desired signal). In the absence of this *a priori* information a solution is to use adaptive filters which possesses the ability to adapt their coefficients to the statistics of the signals involved. As a consequence, the adaptive filters and algorithms were successfully implemented in a wide variety of devices for diverse application fields such as communications, control, radar and biomedical engineering, to mention a few.

Adaptive filtering comprises two basic operations: the filtering process and the adaptation process. In the filtering process an output signal is generated from an input data signal using a digital filter, whereas the adaptation process consists of an algorithm which adjusts the coefficients of the filter to minimize a desired cost function. There is a large variety of filter structures and algorithms used in adaptive filtering, each of them being more suitable for a certain application. We first classify the adaptive filters into two main categories: the *Adaptive Finite Impulse Response* (AFIR) and the *Adaptive Infinite Impulse Response* (AIIR) filters. Moreover, in the class of AFIR filters there are three different filter structures, namely: the transversal filter depicted in Fig. 1.1, the lattice predictor as shown in Fig. 1.2 and the systolic array (see Fig. 1.3) [41]. There are other FIR structures such as, subband FIR adaptive filters and frequency-domain adaptive filters, to mention a few. The first part of this dissertation addresses the algorithms for transversal adaptive FIR filters as depicted in Fig. 1.4, where $\hat{h}_1(n), \ldots, \hat{h}_N(n)$ are the coefficients of the adaptive filter at time instant n, $x(n)$ is the input sequence, $y(n)$ is

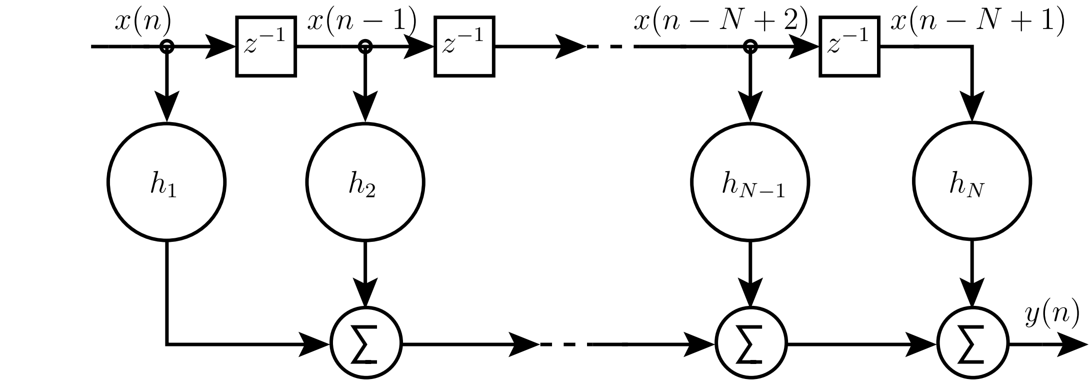

Figure 1.1: The block diagram of a transversal filter.

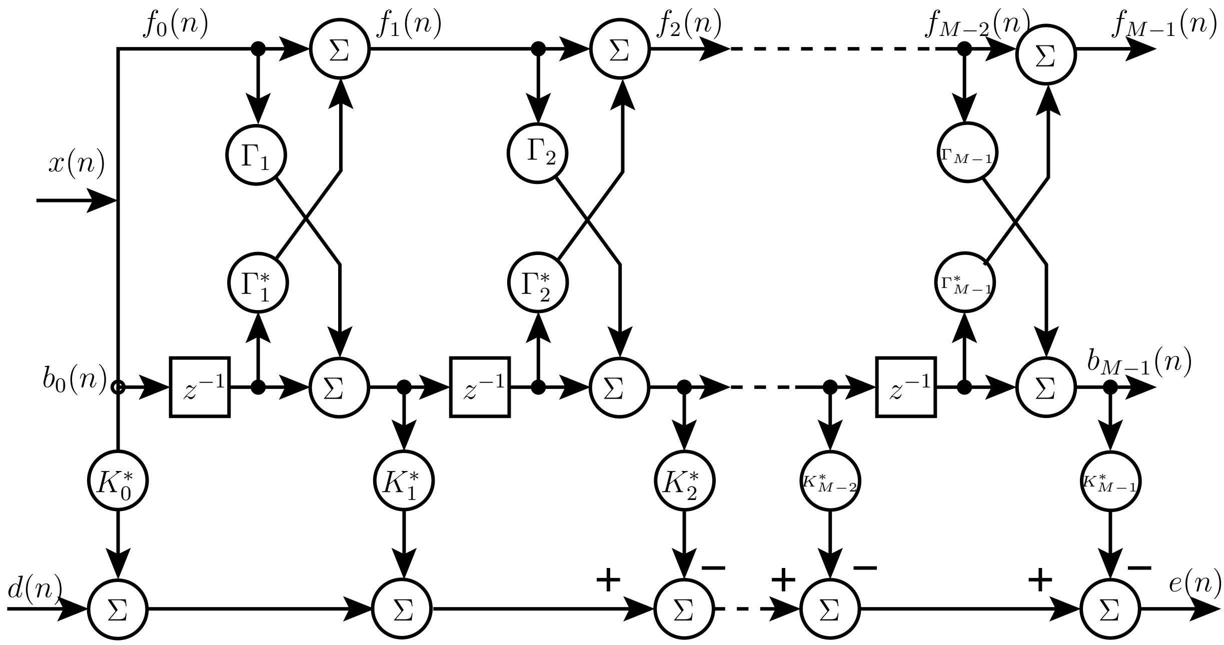

Figure 1.2: The block diagram of a lattice predictor.

the output sequence, $d(n)$ is the desired signal and $e(n)$ represents the output error. The second part of the dissertation is dedicated to the transform domain adaptive filters which can be directly obtained from the structure shown in Fig. 1.4 including an orthogonal transform at the input of the adaptive filter.

In connection to Fig. 1.4, the coefficients $\hat{h}_i(n)$ are changed at each iteration by means of an adaptive algorithm. Among many adaptive algorithms, probably the most known is the *Least Mean Squared* (LMS), which was derived from the minimization of the mean

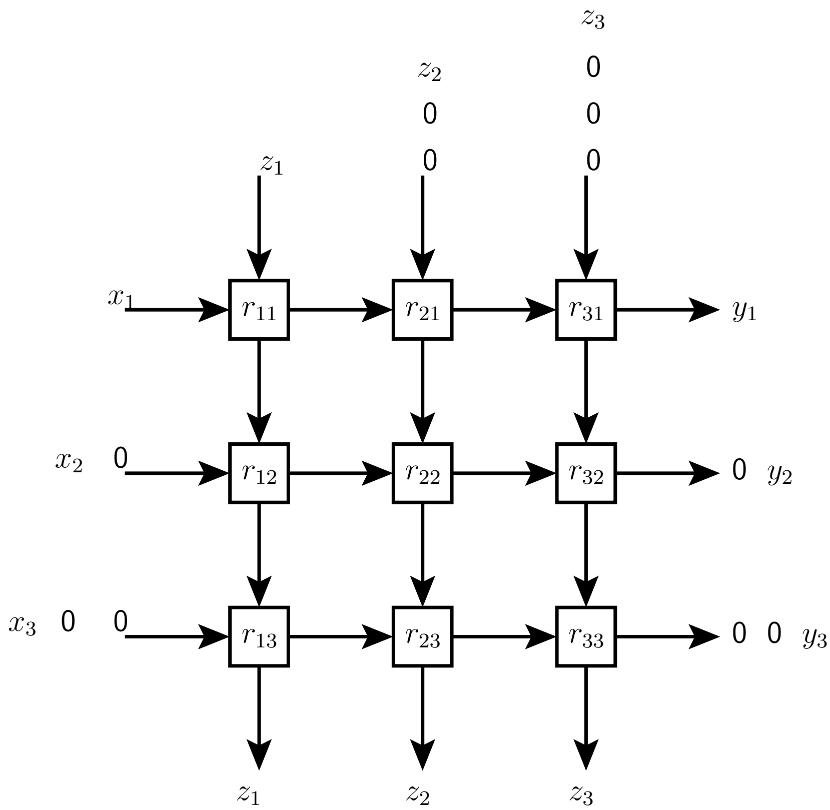

Figure 1.3: The block diagram of a rectangular systolic array.

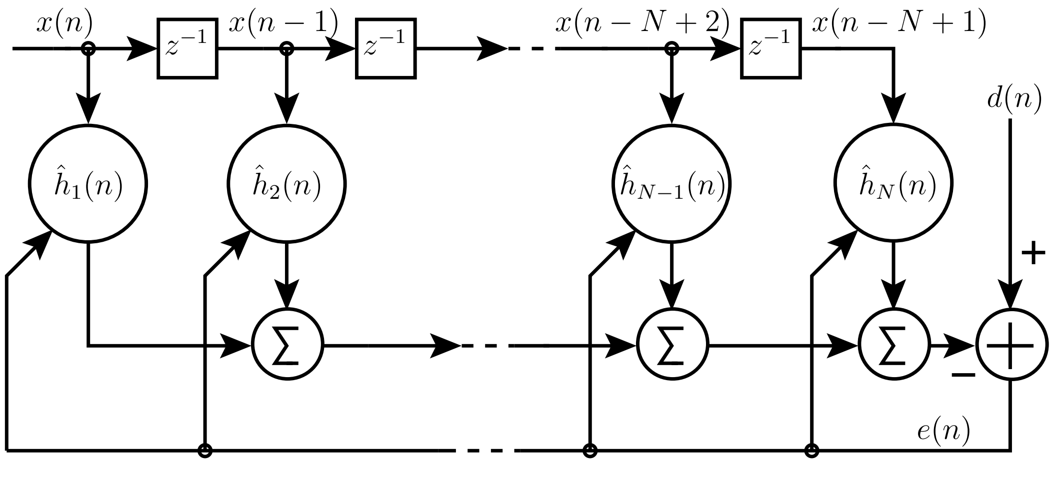

Figure 1.4: The block diagram of an adaptive transversal FIR filter.

squared error [25], [41], [43], [78], [83]:

$$J(n) = E\left\{e^2(n)\right\} \tag{1.1}$$

Many other adaptive algorithms based on minimization of other cost function also exist such as the Least Mean Fourth algorithm [76], Least Mean P-Power algorithm [62] and algorithms with adaptive cost function [71], [72], to mention just few. This dissertation addresses the class of adaptive algorithms derived from the minimization of the mean squared error which include the LMS algorithms and its variants. When the LMS is used to adapt the filter coefficients, the following update equation is implemented at each iteration:

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu e(n)\mathbf{x}(n). \tag{1.2}$$

where $\hat{\mathbf{h}}(n) = \left[\hat{h}_1(n), \ldots, \hat{h}_N(n)\right]^t$ is a $N \times 1$ vector which contains the filter coefficients, $\mathbf{x}(n) = [x(n), \ldots, x(n-N+1)]^T$ is the input vector containing the present and past $N-1$ samples from the input sequence and μ is a constant called the step-size which controls the convergence and the stability of the algorithm.

An AFIR filter using (1.2), in the adaptation process, converges close to the Wiener filter after a number of iterations called the transient period [41], [78]. In communication applications, during the transient period, the transmission of data is not possible therefore, a fast adaptation is one of the main concerns. The LMS algorithm described in (1.2) has a very low computational complexity (number of additions, subtractions, divisions, multiplications per iteration) and memory load, which makes it very attractive for practical implementations. It is well known that the step-size μ influences the performances of the adaptive filter [41]. Despite its low complexity, the LMS has also some drawbacks which influence its performance in terms of convergence speed and accuracy. Much research has been done during the last decades in order to develop algorithms which eliminate or at least reduces the drawbacks of the LMS [1], [4], [5], [7], [9], [11], [12], [14]-[19], [21], [23], [27], [28], [30], [33], [38], [39], [41], [46], [48], [58], [65], [80].

We shall review here some of the main inconveniences of the LMS algorithm. Based on this, we classify the adaptive algorithms in terms of their goal and the problem they address to.

1. First drawback of the LMS is its trade-off between small steady-state mean squared error and fast convergence. For small values of μ in (1.2), the convergence of the filter coefficients is very slow, but the steady-state mean squared error (MSE) is small. On the contrary, for larger μ the convergence speed is increased and also the steady-state MSE. It follows that, in the case of the LMS algorithm, it is not

possible to obtain fast convergence and small steady-state error at the same time. In all practical applications, the main goal is to obtain accurate and fast convergence, therefore some new adaptive algorithms which increase the convergence speed of the LMS while maintaining a low level of the MSE are necessary.

2. It was established in the open literature, that the speed of convergence of the LMS depends on the eigenvalue spread of the input autocorrelation matrix [4], [25], [30], [41], [51], [56], [63]. For large eigenvalue spread, the LMS has a slow convergence while faster convergence is obtained for eigenvalue spread close to unity.

3. The coefficients of the adaptive filter are updated using an estimate of the cost function gradient, $e(n)\mathbf{x}(n)$. In all applications, the signals involved (see Fig. 1.4) might be corrupted by additive noise. When the noise is present in the desired sequence $d(n)$ or in the input sequence $x(n)$, will interfere also in the coefficients adaptation process through the term $e(n)\mathbf{x}(n)$. Due to this fact, in applications where the distribution of the noise is highly impulsive, the LMS might have low convergence and stability problems.

4. In the block digram shown in Fig. 1.4, we have assumed that the optimum length of the adaptive filter is a priori known. This is not always true in practice, since the statistics of the signals involved are unknown, and so the optimum number of the coefficients of the adaptive filter. In some applications, an estimate of the optimum length may be of interest.

5. Tracking capability is another very important characteristic of an adaptive filter. In stationary environments, there is a linear dependence between the step-size μ and the steady-state MSE. In this case, the optimum Wiener filter has constant coefficients and by decreasing the step-size, the steady-state MSE is reduced. When the environment is non-stationary, such as, fading communication channels, the dependence between the steady-state MSE and μ is not linear anymore. Actually, this nonlinear function has a minimum which is obtained for a certain μ_{opt}. The value of μ_{opt} depends on the statistical variation of the environment. It follows that, in non-stationary environments, in order to have small adaptation errors, the step-size must be closer to the optimum μ_{opt}.

There have been many adaptive algorithms introduced in the open literature which try to solve one or more of these inconveniences of the LMS. Depending on the problem they addres one can make the following classification of adaptive LMS algorithms:

C1 Variable Step-Size LMS (VSSLMS) algorithms which uses in (1.2) a time-varying step-size $\mu(n)$ instead of a fixed one. Their main goal is to speed up the convergence while maintaining a small level of the steady-state MSE and also to reduce the trade-off between steady-state MSE and convergence speed;

C2 Order Statistic LMS (OSLMS) algorithms which improve the convergence of the adaptive filter in non-Gaussian noise environments;

C3 Variable Length LMS (VLLMS) algorithms which estimate not only the coefficients of the adaptive filter also its optimum length;

C4 Variable Step-Size LMS algorithms for time-varying environments. The algorithms from this class are different from the VSSLMS in C1 due to the fact that their primary goal is to adapt the step-size $\mu(n)$ toward the optimum μ_{opt} which minimize the steady-state MSE;

C5 Transform Domain LMS (TDLMS) algorithms which use an orthogonal transformation at the input, such that the input autocorrelation matrix is diagonalized. This class of algorithms was introduced to improve the convergence of the LMS in applications where the input sequence is highly correlated;

C6 Transform Domain Variable Step-size (TDVSLMS) algorithms which are the combination of VSSLMS and TDLMS and possesses high convergence speed for both correlated and uncorrelated input sequences. They also reduce the trade-off between steady-state MSE and convergence speed;

C7 Transform Domain LMS algorithms for time-varying environments, which were introduced to increase the convergence speed and to simplify the structure of the algorithms in C4;

C8 Other decorrelation techniques, such as, the scrambled LMS (SCLMS), which performs the decorrelation of the input sequence by means of a scrambling device. However the SCLMS was first introduced in applications where secure data transmission was necessary.

1.1 Overview of the thesis

The dissertation consists of three main parts. According to the above classification of the adaptive algorithms, all classes from C1 to C8 are discussed in Chapter 2 and Chapter 3.

Chapter 2 deals with the time domain implementations of adaptive algorithms belonging to classes C1 to C4. First, we start with a brief theoretical analysis of the standard LMS and some algorithms with variable step-size that were proposed in the open literature. Next, two novel Variable Step-Size LMS algorithms are introduced and described in detail. A discussion about the effects of the miss-estimation of the filter length is given in the sequel and based on this, a variable length LMS algorithm for uncorrelated input sequences is presented. Subsequently, a theoretical analysis of the LMS for non-stationary environments is described in detail and based on this analysis a novel adaptive algorithm with optimum step-size is introduced. Finally the class of Order Statistics LMS algorithms is presented and a new OSLMS algorithm is introduced.

Chapter 3 is dedicated to the algorithms belonging to classes C5 to C8. First, the Transform Domain LMS algorithm is described together with some of its variants introduced in the literature. Next, three new algorithms which are combinations of VSSLMS and TDLMS are presented in more details. A transform domain algorithm with optimum step-size for time-varying environments is introduced as an alternative to the time domain implementation from Chapter 2. At the end of this chapter, a brief introduction to the class of Scrambled LMS algorithms is presented. The theoretical comparison in terms of mean squared error, mean squared coefficient error and convergence speed between the LMS, TDLMS and SCLMS is also discussed for the problem of digital data transmission through a telephone line. The analytical results are supported by simulations performed for two types of correlated input sequences.

Chapter 4 shows the implementations of some of the algorithms from Chapter 2 and Chapter 3 in echo cancellation, channel equalization and CDMA frameworks. For the channel equalization framework the cases of Gaussian and non-Gaussian noise are addressed.

1.2 Author's contribution

The author's contribution to the existing theory is mainly in Chapters 2-4. To the author's knowledge, no work has been done before towards combining the VSSLMS and TDLMS adaptive algorithms to obtain a new class of TDVSLMS algorithms. In the open literature, the step-size of the TDLMS is considered time-varying due to the power normalization and different techniques which improve the decorrelation of the input sequence are discussed. The step-size of the algorithms from the novel class of TDVSLMS depends also on the output error which highly increases their convergence speed. In addition to that, various novel VSSLMS and VLLMS for time domain and transform domain are introduced. The problem of optimum step-size estimation in time-varying environments for time domain and transform domain is addressed and two new algorithms are derived. A new Order Statistic LMS (OSLMS) algorithm suitable in applications with unknown noise distribution is also presented.

The main contribution of this thesis is in the following points:

1. Variable step-size LMS algorithms for time domain: in Proceedings of *IEEE Wireless Communications and Networking Conference EUROCOMM 2000* [18] and presented at *X European Signal Processing Conference EUSIPCO 2000* [19].

2. Variable Length LMS algorithm for time domain, presented at *IEEE International Conference on Electronics, Circuits and Systems, ICECS 2002*, [8].

3. A new Order Statistic LMS algorithm presented at *IEEE International Conference on Audio, Speech and Signal Processing, ICASSP 2002*, [12].

4. Novel adaptive algorithms with optimum step-size for time-varying environments (in both time and transform domain), presented at: *7th WSEAS International Conference on Circuits, Systems, Communications and Computers, WSEAS/CSCC 2003*, [15] and *IEEE International Symposium on Image and Signal Processing and Analysis ISISPA 2003*, [14].

5. A new class of Transform Domain Variable Step-Size LMS algorithms, published in *IEEE Signal Processing Letters*, [9], *WSEAS Transactions on Circuits*, [10] and presented at: *XI European Signal Processing Conference, EUSIPCO 2002*, [11] and *IEEE International Conference on Electronics, Circuits and Systems, ICECS 2001*, [7].

6. Comparative study between Transform Domain LMS and Scrambled LMS in echo cancellation framework, presented at *International TICSP Workshop on Spectral Methods and Multirate Signal Processing, SMMSP 2003*, [6]

The author has done the basic derivations and most of the experimental and writing work in all these publications. The author fulfilled the publications task with the supervisor and the co-authors of the papers. Other results related with one part or another of the thesis were published in [13], [16] and [26].

Chapter 2

Time domain implementations

This chapter considers the time domain implementations of four classes of adaptive algorithms: Variable Step-Size LMS (VSSLMS), Variable Length LMS (VLLMS), the adaptive LMS algorithm for time-varying environments and the Order Statistic LMS (OSLMS) algorithms.

In the first section, the standard LMS algorithm [41], [78] is reviewed and its theoretical analysis is considered. In the analysis, we follow three main directions: first the transient and steady-state behavior in terms of the mean squared error is presented for the case of a stationary environment in which the Wiener filter has time invariant coefficients. Second, the problem of length mismatch between the adaptive filter and the unknown filter is studied in a system identification framework. The analytical expressions are derived for the case when the unknown system has constant coefficients for both correlated and uncorrelated input sequences. Finally, the analysis of the LMS algorithm for tracking a time-varying system with fixed length is presented and the formula for the optimum step-size which minimizes the steady-state MSE is obtained.

In the second section, the class of Variable Step-Size LMS algorithms is addressed. First we describe few of the most cited algorithms in the open literature and next based on the results outlined in the first section, two new VSSLMS algorithms are presented. Simulations results showing the behavior of these new algorithms for the problem of system identification are given in the sequel. The section ends with a comparison in terms of computational complexity, memory load and setup simplicity of the mentioned VSSLMS algorithms.

The class of Variable Length LMS algorithms is detailed in Section 2.3 that starts with a brief description of some algorithms that were published in the literature. Next we introduce a new VLLMS algorithm for system identification, which adjusts the length

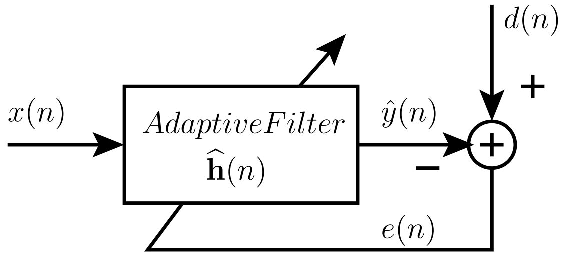

Figure 2.1: Simplified block diagram of an adaptive FIR filter.

of the adaptive filter toward the length of the unknown filter. In the derivation of this algorithm we use some theoretical results presented in the first section. The simulations results of the proposed VLLMS algorithm for the problem of system identification for uncorrelated input sequences, are presented at the end of this section. Length adaptation for the case of correlated input sequence is addressed in Chapter 4.

Section 2.4 is dedicated to the problem of tracking time-varying systems. From the theoretical results shown in the first section we directly derive an adaptive algorithm in which the step-size is time-varying and converge near the optimum. The new algorithm is implemented in the time-varying system identification framework and the simulation results are shown.

The last section deals with the class of Order Statistics LMS algorithms (OSLMS). We first briefly review the algorithms from this class and next we introduce a new algorithm. The section ends with comparative simulation results of the OSLMS algorithms for the problem of system identification for various noise distributions.

2.1 The Least Mean Square algorithm

The simplified block diagram of a transversal adaptive FIR filter is depicted in Fig. 2.1 where the block denoted by $AdaptiveFilter$ comprises an adaptive filter $\widehat{\mathbf{h}}(n) = \left[\widehat{h}_1(n), \widehat{h}_2(n), \ \ldots, \widehat{h}_N(n)\right]^t$ and algorithm, $x(n)$ is the input sequence from which the input vector $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^t$ is obtained, $e(n)$ is the output error, $\widehat{y}(n)$ is the output of the adaptive filter and $d(n)$ is the desired signal. All the theoretical derivations from the present section are referred to this figure.

In connection with Fig. 2.1 the output of the adaptive filter can be written as follows:

$$\widehat{y}(n) = \widehat{\mathbf{h}}^t(n)\mathbf{x}(n) = \mathbf{x}^t(n)\widehat{\mathbf{h}}(n) = \sum_{i=1}^{N} \widehat{h}_i(n)x(n-i+1). \tag{2.1}$$

where t is the transposition operator.

The output error is expressed by the following equation [41]:

$$e(n) = d(n) - \widehat{y}(n). \tag{2.2}$$

The coefficients of the adaptive filter are updated to minimize the output mean squared error defined as follows:

$$J(n) = E\left[e^2(n)\right] = E\left\{[d(n) - \widehat{y}(n)]^2\right\} \tag{2.3}$$

The optimum filter coefficients in the mean square sense (the optimum Wiener solution) are those coefficients for which the partial derivatives of $J(n)$ equals to zero. Denoting the vector of the optimum coefficients as $\mathbf{h}_o = [h_{o_1}, \ldots, h_{o_N}]^t$, the system of equations which gives $\mathbf{h}_o$ is obtained as in the sequel:

$$\begin{aligned}\frac{\delta J(n)}{\delta h_{o_i}} &= \frac{\delta E\left[(d(n) - y(n))^2\right]}{\delta h_{o_i}} = \\ &= -2E\left\{x(n-i+1)\left[d(n) - y_o(n)\right]\right\} = \\ &= -2E\left\{x(n-i+1)e_o(n)\right\} = 0, \qquad \forall i = 1, \ldots, N;\end{aligned} \tag{2.4}$$

where

$$e_o(n) = d(n) - \mathbf{h}_o^t \mathbf{x}(n) \tag{2.5}$$

is the minimum output error obtained when the coefficients of the adaptive filter equals the coefficients of the optimum Wiener filter.

Equation (2.4) can be written in a more compact form as follows:

$$E\left[\mathbf{x}(n)e_o(n)\right] = 0. \tag{2.6}$$

It follows from (2.6) that the optimum error is orthogonal to the input vector at each time instant n, and this represents the well known principle of orthogonality.

From (2.4), the Wiener-Hopf equations which give the coefficients of the optimum filter are represented by:

$$\begin{cases} h_{o_1} r(1-1) + h_{o_2} r(1-2) + \cdots + h_{o_N} r(1-N) = p(0) \\ h_{o_1} r(2-1) + h_{o_2} r(2-2) + \cdots + h_{o_N} r(2-N) = p(1) \\ \ldots \\ h_{o_1} r(N-1) + h_{o_2} r(N-2) + \cdots + h_{o_N} r(N-N) = p(N) \end{cases} \leftrightarrow \quad \mathbf{R}\mathbf{h}_o = \mathbf{p} \tag{2.7}$$

where $r(i-j) = E\left[x(n-i)x(n-j)\right]$, $p(i) = E\left[d(n)x(n-i)\right]$ and $\mathbf{p} = [p(1), p(2), \ldots, p(N)]^t$.

We note that the terms $r(i-j) = r(j-i)$ and $r(i-i) = r(j-j) = r(0)\ \forall i, j$, therefore the matrix $\mathbf{R}$ can be written as:

$$\mathbf{R} = \begin{bmatrix} r(0) & r(1) & r(2) & \dots & r(N-1) \\ r(1) & r(0) & r(1) & \dots & r(N-2) \\ \vdots & \vdots & \ddots & \vdots & \\ r(N-1) & r(N-2) & r(N-3) & \dots & r(0) \end{bmatrix}. \tag{2.8}$$

When the matrix $\mathbf{R}$ is invertible and its elements can be estimated, the optimum Wiener filter can be easily obtained from (2.7) as:

$$\mathbf{h}_o = \mathbf{R}^{-1}\mathbf{p}. \tag{2.9}$$

In situations when the elements of the matrix $\mathbf{R}$ are not available an iterative algorithm can be applied to the adaptive filter which transforms its coefficients toward $\mathbf{h}_o$. One simple adaptive algorithm is the Steepest Descent (SD) algorithm, which updates the coefficients of the adaptive filter at each iteration in the opposite direction of the cost function gradient. In the case of the SD, the update formula for the filter coefficients is:

$$\widehat{\mathbf{h}}(n+1) = \widehat{\mathbf{h}}(n) - \frac{1}{2}\mu \bigtriangledown J(n), \tag{2.10}$$

where $\bigtriangledown J(n) = \left[\frac{\delta J(n)}{\delta \widehat{h}_1(n)}, \frac{\delta J(n)}{\delta \widehat{h}_2(n)}, \dots, \frac{\delta J(n)}{\delta \widehat{h}_N(n)}\right]^t$ and

$$\frac{\delta J(n)}{\delta \widehat{h}_i(n)} = -2E\left[x(n-i+1)e(n)\right]. \tag{2.11}$$

In order to compute the elements of the gradient in (2.11), the expectation operator must be used. A simpler alternative is to use the instantaneous gradient instead of the true gradient and the obtained algorithm is called the Least Mean Square (LMS). As a consequence, the LMS algorithm uses the following coefficient update formula:

$$\widehat{\mathbf{h}}(n+1) = \widehat{\mathbf{h}}(n) + \mu e(n)\mathbf{x}(n). \tag{2.12}$$

where the step-size μ was introduced to control the stability of the algorithm.

Finally, the LMS algorithm can be described by the following four steps:

Least Mean Square algorithm:

At every iteration n do:

1. Form the input vector $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^t$ from the input sequence $x(n)$.
2. Compute the output of the adaptive filter: $\widehat{y}(n) = \mathbf{x}^t(n)\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}^t(n)\mathbf{x}(n)$;
3. Compute the output error: $e(n) = d(n) - \widehat{y}(n)$;
4. Update the coefficients of the adaptive filter: $\widehat{\mathbf{h}}(n+1) = \widehat{\mathbf{h}}(n) + \mu e(n)\mathbf{x}(n)$;

2.1.1 Analysis of the LMS for stationary environments

We point out some important results of the transient and steady-state analysis of the mean square error of the LMS algorithm for stationary environments which will be used further in the subsequent sections. The following fundamental assumptions are used in order to make the convergence analysis of the LMS mathematically tractable[1] [41]:

a. The input vectors $\mathbf{x}(1)$, $\mathbf{x}(2)$, ..., $\mathbf{x}(n)$ are statistically independent from each other;

b. The input vector $\mathbf{x}(n)$ at time instant n is independent of all previous samples of the desired sequence, $d(1)$, $d(2)$, ..., $d(n-1)$;

c. The desired response $d(n)$ at time instant n depends on the corresponding input vector $\mathbf{x}(n)$, but it is statistically independent of all previous samples of the desired response;

d. The input vector $\mathbf{x}(n)$ and the desired response $d(n)$ consist of mutually Gaussian-distributed random variables at all time instants n.

We first start with the analysis of the coefficient error vector defined as:

$$\Delta\mathbf{h}(n) = \widehat{\mathbf{h}}(n) - \mathbf{h}_o, \tag{2.13}$$

where $\mathbf{h}_o$ is the vector of the optimum coefficients given in (2.9).

Subtracting the optimum coefficients vector from (2.12) and using (2.13), one obtains:

$$\Delta\mathbf{h}(n+1) = \widehat{\mathbf{h}}(n+1) - \mathbf{h}_o = \widehat{\mathbf{h}}(n) - \mathbf{h}_o + \mu e(n)\mathbf{x}(n) = \Delta\mathbf{h}(n) + \mu e(n)\mathbf{x}(n), \tag{2.14}$$

[1]These four assumptions form the so called *independence theory* which, even though is violated in many practical applications, was proved to retain sufficient information about the adaptive process such that to allow the derivation of quite reliable design guidelines.

Subsequently, the formula to compute the output error can be rewritten as follows:

$$e(n) = d(n) - \widehat{\mathbf{h}}^t(n)\mathbf{x}(n) = d(n) - \mathbf{h}_o^t\mathbf{x}(n) + \mathbf{h}_o^t\mathbf{x}(n) - \widehat{\mathbf{h}}^t(n)\mathbf{x}(n) = e_o(n) - \Delta\mathbf{h}(n)^t\mathbf{x}(n), \tag{2.15}$$

where $e_o(n)$ is the minimum error defined in (2.5).

From (2.14) and (2.15) the coefficient error vector can be obtained as follows:

$$\begin{aligned}\Delta\mathbf{h}(n+1) &= \Delta\mathbf{h}(n) - \mu\mathbf{x}(n)\mathbf{x}^t(n)\Delta\mathbf{h}(n) + \mu\mathbf{x}(n)e_o(n)\\ &= \left[\mathbf{I} - \mu\mathbf{x}(n)\mathbf{x}^t(n)\right]\Delta\mathbf{h}(n) + \mu\mathbf{x}(n)e_o(n),\end{aligned} \tag{2.16}$$

where $\mathbf{I}$ is the $N \times N$ identity matrix.

Taking the mathematical expectation on both sides of (2.16) we obtain:

$$E\left[\Delta\mathbf{h}(n+1)\right] = E\left\{\left[\mathbf{I} - \mu\mathbf{x}(n)\mathbf{x}^t(n)\right]\Delta\mathbf{h}(n)\right\} + \mu E\left[\mathbf{x}(n)e_o(n)\right]. \tag{2.17}$$

The last term in (2.17) vanishes due to the principle of orthogonality expressed by (2.6). From (2.12) it follows that the coefficient error vector $\Delta\mathbf{h}(n)$ is independent of $\mathbf{x}(n)$ and as a consequence, (2.17) simplifies as follows:

$$E\left[\Delta\mathbf{h}(n+1)\right] = \left[\mathbf{I} - \mu\mathbf{R}\right]E\left[\Delta\mathbf{h}(n)\right]. \tag{2.18}$$

The stability condition for the step-size μ, which ensures the convergence of the adaptive filter coefficients in the mean can be obtained from (2.18) after some mathematical manipulations as follows (see [41] for more details):

$$0 < \mu < \frac{2}{\lambda_{max}}. \tag{2.19}$$

where λ_{max} is the maximum eigenvalue of $\mathbf{R}$.

To obtain an analytical expression for the output mean squared error, we first compute the correlation matrix of the coefficient error vector $\mathbf{C}(n)$. The matrix $\mathbf{C}(n)$ is defined as $\mathbf{C}(n) = E\left[\Delta\widehat{\mathbf{h}}(n)\Delta\widehat{\mathbf{h}}^t(n)\right]$ and it is obtained from (2.18) as follows (see [41] for detailed analytical derivations):

$$\mathbf{C}(n+1) = \mathbf{C}(n) - \mu\left[\mathbf{R}\mathbf{C}(n) + \mathbf{C}(n)\mathbf{R}\right] + \mu^2\mathbf{R}tr\left[\mathbf{R}\mathbf{C}(n)\right] + 2\mu^2\mathbf{R}\mathbf{C}(n)\mathbf{R} + \mu^2 J_{min}\mathbf{R}. \tag{2.20}$$

where $J_{min} = E\left[e_o^2(n)\right]$ is the minimum MSE obtained in the case of perfect adaptation.

Recalling equation (2.3), the MSE can be further detailed as in the sequel:

$$\begin{aligned}J(n) &= E\left[\left(d(n) - \widehat{\mathbf{h}}^t(n)\mathbf{x}(n)\right)\left(d(n) - \mathbf{x}^t(n)\widehat{\mathbf{h}}(n)\right)\right] =\\ &= E\left[\left(e_o(n) + \mathbf{h}_o^t\mathbf{x}(n) - \widehat{\mathbf{h}}^t(n)\mathbf{x}(n)\right)\left(e_o + \mathbf{x}^t(n)\mathbf{h}_o - \mathbf{x}^t(n)\widehat{\mathbf{h}}(n)\right)\right] =\\ &= E\left[\left(e_o(n) - \Delta\mathbf{h}^t(n)\mathbf{x}(n)\right)\left(e_o - \mathbf{x}^t(n)\Delta\mathbf{h}(n)\right)\right]\end{aligned} \tag{2.21}$$

where $\Delta\mathbf{h} = \widehat{\mathbf{h}}(n) - \mathbf{h}_o$ and $\mathbf{h}_o$ is the optimum Wiener solution.

Finally, taking into account the above assumptions and after some mathematical manipulations, the MSE can be expressed as (see [41] for more details):

$$J(n) = J_{min} + tr\left[\mathbf{RC}(n)\right], \tag{2.22}$$

where $\mathbf{R}$ is the input autocorrelation matrix, $\mathbf{C}(n)$ is the crosscorrelation matrix of the coefficient error vector and J_{min} is the minimum MSE which is obtained in the case of perfect adaptation (if $\widehat{\mathbf{h}}(n) = \mathbf{h}_o$).

The steady-state value of the mean squared error, can be expressed by combining (2.20) and (2.22) for $n \to \infty$, and the following analytical expression results [41]:

$$J_{st} = J_{min} + J_{min}\frac{\sum\limits_{i=1}^{N}\frac{\mu\lambda_i}{2-2\mu\lambda_i}}{1-\sum\limits_{i=1}^{N}\frac{\mu\lambda_i}{2-2\mu\lambda_i}}, \tag{2.23}$$

which for small values of the step-size μ becomes:

$$J_{st} = J_{min} + \frac{\mu}{2}J_{min}\sum_{i=1}^{N}\lambda_i = J_{min} + \frac{\mu}{2}J_{min}N\sigma_x^2 = J_{min} + J_{ex}. \tag{2.24}$$

where λ_i is the i^{th} eigenvalue of $\mathbf{R}$ and $\sigma_x^2 = r(0)$ is the variance of the input sequence $x(n)$.

It results from (2.23) and (2.24), that the steady-state MSE contains two components. First component J_{min} is the minimum MSE which can be achieved in the ideal case of perfect adaptation, when the coefficients of the adaptive filter equals the coefficients of the Wiener filter. The second component J_{ex}, called the excess MSE, depends upon the step-size μ and it is due to the missadaptation of the filter coefficients. We shall retain these two equations which will be recalled later during this thesis.

The misadjustment is defined as the ratio between the excess MSE and the minimum MSE and it is given by [41], [78]:

$$\mathcal{M} = \frac{J_{ex}}{J_{min}} = \frac{\sum\limits_{i=1}^{N}\frac{\mu\lambda_i}{2-2\mu\lambda_i}}{1-\sum\limits_{i=1}^{N}\frac{\mu\lambda_i}{2-2\mu\lambda_i}}, \tag{2.25}$$

and for small step-size the above expression can be simplified to:

$$\mathcal{M} = \frac{\mu N}{2}\sigma_x^2, \tag{2.26}$$

The average learning curve of the LMS algorithm can be approximated with an exponential with time constant τ. In this case the misadjustment can be expressed as follows:

$$\mathcal{M} = \frac{\mu N \lambda_{av}}{2} = \frac{N}{4\tau}. \tag{2.27}$$

where $\lambda_{av} = \frac{1}{N}\sum_{i=1}^{N}\lambda_i$ is the average eigenvalue of $\mathbf{R}$, and

$$\tau = \frac{1}{2\mu\lambda_{av}}. \tag{2.28}$$

is the average time constant which is proportional to the length of the transient period of the algorithm.

We should note, that the condition for the convergence in the mean square sense (the condition which ensures $J(n) \to J_{st}$) is given by [41]:

$$0 < \mu < \frac{2}{3\sum_{i=1}^{N}\lambda_i} = \frac{2}{3N\sigma_x^2}. \tag{2.29}$$

It results from (2.27) and (2.28), that for a given filter length N, the misadjustment is direct proportional with the step-size and the convergence time is inverse proportional to μ. As a consequence, if two adaptive filters with the same length N and different step-size are compared, the faster convergence is obtained with the filter that has larger step-size. However, the filter with smaller step-size will converge to a smaller level of the misadjustment $\mathcal{M}$. This is a very important conclusion which is the basis for derivation of the class of Variable Step-Size LMS algorithms.

2.1.2 Mean squared error as a function of the adaptive filter length

In the previous section, we pointed out some important analytical expressions for the optimum Wiener solution, MSE and coefficient error vector of an adaptive FIR filter that uses the LMS algorithm to update its coefficients. From the analytical expression (2.24) of the steady-state MSE we see that it depends upon J_{min} which is the minimum MSE in the case of perfect adaptation. In some applications, such as system identification, the desired signal $d(n)$ in Fig. 2.1 might be generated by an FIR filter of a certain length (for instance in echo cancellation the sequence $d(n)$ represents the echo generated by the echo path which is modelled as an FIR filter). Usually, in this kind of applications the length of the filter generating the desired sequence is not known and the length of the adaptive

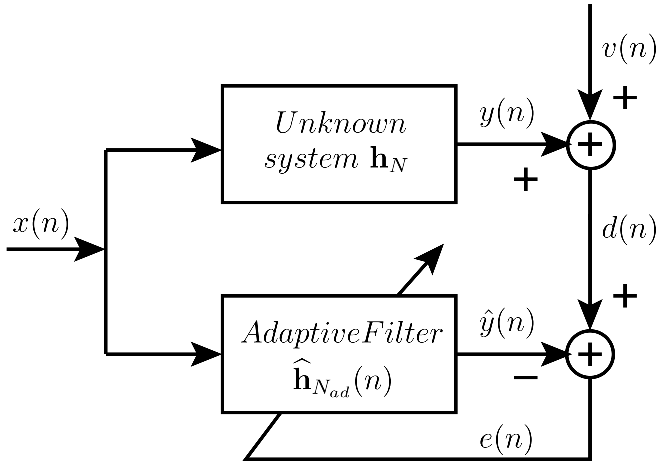

Figure 2.2: System identification block diagram for the case when the unknown filter and the adaptive filter have different lengths.

filter is chosen by the user. As a consequence, when there is a mismatch between the lengths of the two adaptive filters, there is a bias term which increases the term J_{min} in (2.24) and also the steady-state value of the MSE. In this section, we extend the previous analysis with an emphasis in the effect of the length adaptation. We focus on the problem of system identification as shown in Fig. 2.2 in which the desired sequence $d(n)$ is obtained at the output of an FIR filter (called unknown system) of length N while the adaptive filter has different length N_{ad}.

To this end, we first change some notations in order to emphasize the difference between the lengths of the adaptive filter and the unknown system. We denote by N, $\mathbf{h}_N$, N_{ad} and $\widehat{\mathbf{h}}_{N_{ad}}$, the length of the unknown system, the coefficients vector of the unknown system, the length of the adaptive filter and the coefficients vector of the adaptive filter respectively.

With reference to Fig. 2.2, the adaptive filter coefficients are updated by the following formula:

$$\widehat{\mathbf{h}}_{N_{ad}}(n+1) = \widehat{\mathbf{h}}_{N_{ad}}(n) + \mu \mathbf{x}_{N_{ad}}(n) e(n); \tag{2.30}$$

where μ is the step-size and $\mathbf{x}_{N_{ad}}(n)$ is the vector of the past N_{ad} input samples.

The output error is computed as:

$$e(n) = y(n) - \widehat{y}(n) + v(n); \tag{2.31}$$

where $v(n)$ is the output noise,

$$y(n) = \mathbf{h}_N^t \mathbf{x}_N(n) = \sum_{i=1}^{N} h_i x(n-i+1), \tag{2.32}$$

$$\widehat{y}(n) = \widehat{\mathbf{h}}_{N_{ad}}^t(n) \mathbf{x}_{N_{ad}}(n) = \sum_{i=1}^{N_{ad}} \widehat{h}_i(n) x(n-i+1), \tag{2.33}$$

where N and N_{ad} respectively indicate the lengths of the corresponding vectors.

The coefficients vectors, $\mathbf{h}_N$ and $\widehat{\mathbf{h}}_{N_{ad}}(n)$, are denoted as follows:

$$\mathbf{h}_N = [h_1, h_2, \ldots, h_N]^t, \quad \text{and} \quad \widehat{\mathbf{h}}_{N_{ad}}(n) = \left[\widehat{h}_1(n), \widehat{h}_2(n), \ldots, \widehat{h}_{N_{ad}}(n)\right]^t. \tag{2.34}$$

and the input vectors are written as:

$$\mathbf{x}_N(n) = [x(n), \ldots, x(n-N+1)]^t \quad \text{and} \quad \mathbf{x}_{N_{ad}}(n) = [x(n), \ldots, x(n-N_{ad}+1)]^t \tag{2.35}$$

We should emphasize that the lengths N and N_{ad} are usually not equal since the length N of the unknown filter is not known and the length N_{ad} of the adaptive filter is chosen by the user. Of course, in applications where the length estimation is not of primary concern, a long adaptive filter might be implemented without introducing a bias term in the steady-state MSE as will be more clear in the sequel.

The optimum coefficients $\mathbf{h}_{o_{Nad}} = [h_{o_1}, \ldots, h_{o_{Nad}}]$, which minimize the mean squared error, are obtained from the Wiener-Hopf equation. We recall here equation (2.27) in which we explicitly write the terms $p_i(n)$ and we assume that the output noise $v(n)$ is a random Gaussian-distributed, zero mean sequence independent on $x(n)$:

$$\begin{cases} h_{o_1} r(0) + h_{o_2} r(1) + \cdots + h_{o_{Nad}} r(N_{ad}-1) &= h_1 r(0) + h_2 r(1) + \cdots + h_N r(N-1) \\ h_{o_1} r(1) + h_{o_2} r(0) + \cdots + h_{o_{Nad}} r(N_{ad}-2) &= h_1 r(1) + h_2 r(0) + \cdots + h_N r(N-2) \\ \cdots & \\ h_{o_1} r(N_{ad}-1) + \ldots\ldots\ldots\ldots + h_{o_{Nad}} r(0) &= h_1 r(N_{ad}-1) + \ldots + h_N r(N-N_{ad}) \end{cases} \tag{2.36}$$

Subsequently, (2.36) can be written in a compact form:

$$\mathbf{R}_{Nad \times Nad} \mathbf{h}_{o_{Nad}} = \mathbf{R}_{Nad \times N} \mathbf{h}_N. \tag{2.37}$$

where

$$\mathbf{R}_{Nad \times Nad} = E\left[\mathbf{x}_{Nad}(n) \mathbf{x}_{Nad}^t(n)\right] = \begin{bmatrix} r(0) & r(1) & \ldots & r(N_{ad}-1) \\ r(1) & r(0) & \ldots & r(N_{ad}-2) \\ \cdots & \cdots & \cdots & \cdots \\ r(N_{ad}-1) & r(N_{ad}-2) & \ldots & r(0) \end{bmatrix} \tag{2.38}$$

$$\mathbf{R}_{Nad\times N} = E\left[\mathbf{x}_{Nad}(n)\mathbf{x}_N^t(n)\right] = \begin{bmatrix} r(0) & r(1) & \dots & r(N-1) \\ r(1) & r(0) & \dots & r(N-2) \\ \dots & \dots & \dots & \dots \\ r(N_{ad}-1) & r(N_{ad}-2) & \dots & r(N-N_{ad}) \end{bmatrix} \quad (2.39)$$

and the subscripts indicate the sizes of the corresponding vectors and matrices.

We note that the matrix $\mathbf{R}_{Nad\times Nad}$ in (2.37) is a square matrix, whereas $\mathbf{R}_{Nad\times N}$ has different number of rows and columns. As a consequence, in order to compute the optimum coefficients, three cases must be taken into account namely $N < N_{ad}$, $N = N_{ad}$ and $N > N_{ad}$ as follows:

Case 1 : $N < N_{ad}$ In order to obtain the optimum coefficients, the vector of the unknown system is padded with zeros and we obtain the following vector of length N_{ad}:

$$\mathbf{h}_{N_{ad}} = [h_1, \dots, h_N, 0, \dots, 0]^t_{N_{ad}} \quad (2.40)$$

Subsequently, (2.37) can be written as follows:

$$\mathbf{R}_{Nad\times Nad}\mathbf{h}_{o_{Nad}} = \mathbf{R}_{Nad\times Nad}\mathbf{h}_{Nad}, \quad (2.41)$$

and the optimum solution is given by the following equation:

$$\mathbf{h}_{o_{Nad}} = \mathbf{h}_{Nad}. \quad (2.42)$$

with $\mathbf{h}_{Nad}$ given in (2.40).

Case 2 : $N = N_{ad}$ The optimum solution for this case is the same as in the previous case with the difference that the vector of the unknown filter is not padded with zeros since $\mathbf{h}_{o_{Nad}}$ and $\mathbf{h}_N$ are of the same length.

Case 3 : $N > N_{ad}$ To compute the optimum coefficients vector, first we rewrite (2.37) in an equivalent form:

$$\mathbf{R}_{Nad\times Nad}\mathbf{h}_{o_{Nad}} = \mathbf{R}_{Nad\times Nad}\mathbf{h}_{Nad} + \mathbf{R}_{Nad\times(N-Nad)}\mathbf{h}_{N-Nad}, \quad (2.43)$$

where the vectors and the matrices from the right-hand side of (2.43) are given by:

$$\mathbf{h}_{Nad} = [h_1, \dots, h_{Nad}], \qquad \mathbf{h}_{N-Nad} = [h_{Nad+1}, \dots, h_N] \quad (2.44)$$

$$\mathbf{R}_{Nad\times(N-Nad)} = \begin{bmatrix} r(N_{ad}) & r(N_{ad}+1) & \dots & r(N-1) \\ r(N_{ad}-1) & r(N_{ad}) & \dots & r(N-2) \\ \dots & \dots & \dots & \dots \\ r(1) & r(2) & \dots & r(N-N_{ad}) \end{bmatrix} \quad (2.45)$$

and $\mathbf{R}_{Nad\times Nad}$ is given in (2.38).

Multiplying (2.43) by $\mathbf{R}^{-1}_{Nad\times Nad}$ at left, the vector of the optimum coefficients is obtained:

$$\mathbf{h}_{o_{Nad}} = \mathbf{h}_{Nad} + \mathbf{R}^{-1}_{Nad\times Nad}\mathbf{R}_{Nad\times(N-Nad)}\mathbf{h}_{N-Nad}. \tag{2.46}$$

Finally, combining (2.42) and (2.46) the optimum Wiener solution can be expressed in a compact form as follows:

$$\mathbf{h}_{o_{Nad}} = \begin{cases} [h_1,\ldots,h_N,0,\ldots,0]^t_{Nad} & \text{if} \quad N < N_{ad} \\ [h_1,\ldots,h_N]^t_N & \text{if} \quad N = N_{ad} \\ \mathbf{h}_{Nad} + \mathbf{R}^{-1}_{Nad\times Nad}\mathbf{R}_{Nad\times(N-Nad)}\mathbf{h}_{N-Nad} & \text{if} \quad N > N_{ad} \end{cases} \tag{2.47}$$

As a result, when the length N_{ad} of the adaptive filter exceeds the length N of the unknown filter, the vector of the optimum coefficients is obtained by padding with $N-N_{ad}$ zeros the vector $\mathbf{h}_N$. On the contrary, when the length of the adaptive filter is smaller than the length of $\mathbf{h}_N$, the optimum coefficients are obtained adding a bias term to the corresponding coefficients of the unknown filter as shown in (2.47).

In the case of uncorrelated inputs, the elements of the matrix $\mathbf{R}_{Nad\times(N-Nad)}$ are all zeros and in (2.46) we have $\mathbf{h}_{o_{Nad}} = \mathbf{h}_{Nad}$ with $\mathbf{h}_{Nad}$ given in (2.44). Consequently, for uncorrelated input sequences the vector of the optimum coefficients is given by:

$$\mathbf{h}_{o_{Nad}} = \begin{cases} [h_1,\ldots,h_N,0,\ldots 0]^t_{Nad}, & \text{if} \quad N < N_{ad} \\ [h_1,\ldots,h_N]^t_N & \text{if} \quad N_{ad} = N \\ [h_1,\ldots,h_{N_{ad}}]^t_{Nad}, & \text{if} \quad N > N_{ad}. \end{cases} \tag{2.48}$$

To complete the steady-state analysis of the LMS algorithm for the case of length missmatch, we shall study also the output mean squared error. To this end, we recall equation (2.22) which gives the output MSE where we explicitly denote the sizes of the matrices and vectors involved:

$$J(n) = J_{min} + tr\left[\mathbf{R}_{Nad\times Nad}\mathbf{C}_{Nad\times Nad}(n)\right], \tag{2.49}$$

where $\mathbf{R}_{Nad\times Nad}$ is given in (2.38),

$$\mathbf{C}_{Nad\times Nad}(n) = E\left\{\Delta\widehat{\mathbf{h}}_{N_{ad}}(n)\Delta\widehat{\mathbf{h}}^t_{N_{ad}}(n)\right\}, \tag{2.50}$$

$$\Delta\widehat{\mathbf{h}}_{N_{ad}}(n) = \widehat{\mathbf{h}}_{N_{ad}}(n) - \mathbf{h}_{o_{Nad}}, \tag{2.51}$$

and $\mathbf{h}_{o_{Nad}}$ is given by (2.47) for correlated inputs and (2.48) for uncorrelated inputs.

The steady-state MSE is obtained following similar derivations as in the previous section and we obtain:

$$J_{st} = E\{e^2(\infty)\} = J_{min}\left(1 + \frac{\sum\limits_{i=1}^{N_{ad}} \frac{\mu\lambda_i}{2-2\mu\lambda_i}}{1 - \sum\limits_{i=1}^{N_{ad}} \frac{\mu\lambda_i}{2-2\mu\lambda_i}}\right), \tag{2.52}$$

where λ_i are the eigenvalues of the input signal autocorrelation matrix $\mathbf{R}_{Nad\times Nad}$.

For small values of the step-size μ, (2.52) simplifies to:

$$J_{st} = J_{min}\left(1 + \frac{\mu}{2}\sum_{i=1}^{N_{ad}}\lambda_i\right) = J_{min}\left(1 + \frac{\mu}{2}tr\left[\mathbf{R}_{Nad\times Nad}\right]\right) = J_{min} + J_{min}\frac{\mu}{2}N_{ad}\sigma_x^2, \tag{2.53}$$

From (2.53) we see that the misadjustment $\mathcal{M} = \frac{\mu N_{ad}\sigma_x^2}{2}$ do not depend on the difference between the lengths of the to filters (unknown system and the adaptive filter). Intuitively we can explain this by the following fact: if the length of the adaptive filter is larger than the length of the unknown system, the extra coefficients converge to zero. However, they will no be exactly equal to zero but will oscillate around zero. These small oscillations of the extra coefficients will generate misadjustment exactly as the other non-zero coefficients. When the length of the adaptive filter is smaller than the length of $\mathbf{h}_N$, the optimum coefficients are obtained adding a bias term to the corresponding coefficients from $\mathbf{h}_N$. The bias terms are non-zero if the input is correlated and they are zero for uncorrelated inputs. However the misadjustment is the measure of adaptation to the optimum solution which is biased and not to the coefficients of the unknown system.

To obtain the final expression of the steady-state MSE, the value of J_{min} must be expressed for three different cases: $N_{ad} < N$, $N_{ad} = N$ and $N_{ad} > N$.

Analysis of the minimum mean squared error

The value of the minimum mean squared error, J_{min}, in (2.52) and (2.53) is obtained in the case of perfect adaptation (when the coefficients of the adaptive filter equals the optimum coefficients) and it is given by:

$$\begin{aligned} J_{min} &= E\left\{e_o^2(n)\right\} = E\left\{\left[y(n) + v(n) - y_o(n)\right]^2\right\} = E\left\{v^2(n)\right\} + E\left\{\left[y(n) - y_o(n)\right]^2\right\}, \\ J_{min} &= \sigma_v^2 + E\left\{\left[\sum_{i=1}^{N} h_i x(n-i+1) - \sum_{i=1}^{N_{ad}} h_{o_i} x(n-i+1)\right]^2\right\} \end{aligned} \tag{2.54}$$

where $y_o(n) = \sum_{i=1}^{N_{ad}} h_{o_i} x(n-i+1)$ is the optimum output and $v(n)$ is a zero mean random Gaussian-distributed sequence with variance σ_v^2 and independent of $x(n)$.

It follows from (2.47), that J_{min} for the case of a correlated input sequence, can be written as:

$$J_{min} = \begin{cases} \sigma_v^2 & \text{if} \quad N \leq N_{ad}, \\ \sigma_v^2 + E\left\{\left[\sum_{i=1}^{N_{ad}} \left(h_i - h_{o_i}\right) x(n-i+1) + \sum_{i=N_{ad}+1}^{N} h_i x(n-i+1)\right]^2\right\} & \text{if} \quad N > N_{ad} \end{cases} \tag{2.55}$$

For uncorrelated input sequence $x(n)$, $E\left\{x(n-i+1)x(n-j+1)\right\} = 0$ for $i \neq j$, and the vector of the optimum coefficients is given by (2.48). As a result, equation (2.55), in the case of uncorrelated input, simplifies to:

$$J_{min} = \begin{cases} \sigma_v^2, & \text{if} \quad N \leq N_{ad}, \\ \sigma_v^2 + \sum_{i=N_{ad}+1}^{N} h_i^2 r(0), & \text{if} \quad N > N_{ad} \end{cases} \tag{2.56}$$

Simulations and results

In order to test the above analytical results two sets of simulations were done. In the first set, the behavior of the MSE and the behavior of the coefficients in the mean, for correlated input sequence is studied. By doing this analysis we are interested to check the validity of (2.47) and (2.55). The second set of simulations was done for uncorrelated input sequence with a view to verify the validity of (2.48) and (2.56). Both sets of simulations were done in system identification framework as depicted in Fig. 2.2 for the case when the the length of the adaptive filter was smaller than the length of the unknown system ($N_{ad} < N$) and also for the situation when the adaptive filter has more coefficients than the unknown system ($N_{ad} > N$).

In all simulations, a number of 100 independent Monte-Carlo simulations were done, each of them consisting of a number of 5×10^4 iterations and the results were averaged. The coefficients of the adaptive filter were initialized with ones although in practice, when there is no information about the optimum solution, usually they are initialized with zeros. We have chosen this kind of initialization to show that the extra coefficients of the adaptive filter, (when $N_{ad} > N$) are adapted toward zero independently of initialization.

Correlated input sequence:

1. $N > N_{ad}$: In this case, the length of the unknown FIR filter $\mathbf{h}_N$ in Fig. 2.2 was $N = 9$, the length of the adaptive filter was $N = 5$ and the input sequence $x(n)$ was

generated by the following model:

$$x(n) = 1.79x(n-1) - 1.85x(n-2) + 1.27x(n-3) - 0.41x(n-4) + \eta(n) \quad (2.57)$$

where $\eta(n)$ is a random zero mean Gaussian-distributed sequence with variance chosen, such that the variance of $x(n)$ is unity ($\sigma_x^2 = 1$).

The trace of the input autocorrelation matrix $\mathbf{R}_{Nad \times Nad}$ was $tr\left[\mathbf{R}_{Nad \times Nad}\right] = N_{ad}\sigma_x^2 = 5$. It follows, that the condition for the convergence in the mean square sense in (2.29) is $\mu < 0.13$. A step-size $\mu = 10^{-2}$ which satisfy this condition, was chosen.

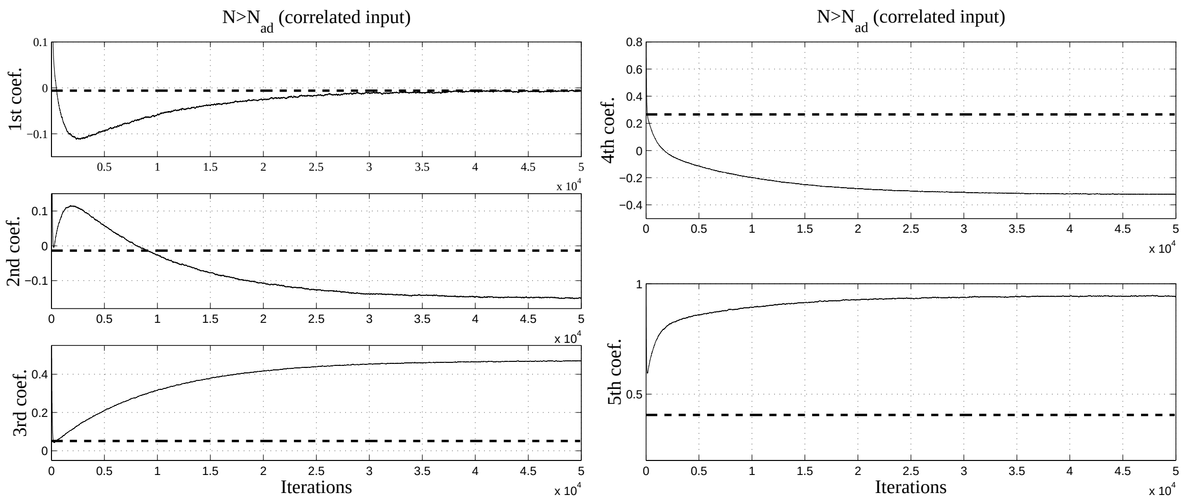

Figure 2.3: The coefficients $\widehat{h}_1(n)$ to $\widehat{h}_5(n)$ of the adaptive filter, during the adaptation (continuous line) and the corresponding coefficients of the unknown system (dashed line) for correlated input sequence and $N > N_{ad}$.

The output MSE is shown in Fig. 2.4 together with the steady-state MSE obtained when the adaptive filter and the unknown filter are of equal length ($N_{ad} = N = 9$). We can see from this figure, that the obtained steady-state level of the MSE is higher than the one obtained for equals lengths, which verifies the analytical result of (2.55). It follows that for a correlated input sequence and $N > N_{ad}$, the steady-state MSE has a bias term that is due to the extra coefficients of the unknown system which does not have correspondences into the adaptive filter. Unfortunately, there is no proof that larger differences in the length of the filters increase the bias term. In fact, it is possible in some cases, that the extra coefficients of the unknown system have negative values and therefore, the MSE bias can decrease when the difference between the lengths of the two filters increases.

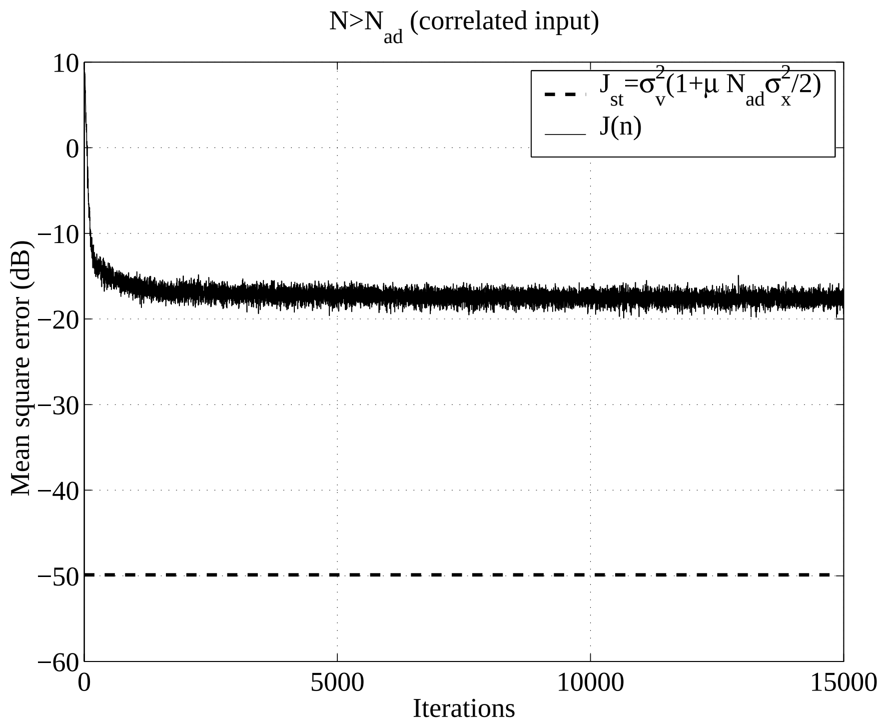

Figure 2.4: Mean squared error for correlated input sequence and $N > N_{ad}$.

The coefficients of the adaptive filter during the adaptation are plotted in Fig. 2.3, where the corresponding coefficients of the unknown system are also shown. The plotted learning curves clearly show that, at the steady-state there is a bias in every coefficient of the adaptive filter which verifies the analytical result of (2.47).

We can conclude that, there is a bias term in every adaptive filter coefficient, in the case of correlated input sequences when the adaptive filter is smaller that the unknown system. These bias terms prevent the coefficients of the adaptive filter to converge close to the corresponding coefficients of the unknown system. In other words, the coefficients of the optimum filter are not equal with the coefficients of the unknown system. The misadjustment of the adaptive filter (that is a measure of the adaptation to the optimum filter) is not influenced by the bias. On the other hand, the minimum mean squared error J_{min} is affected by the bias terms of the coefficients and by the non-zero elements of the matrix $\mathbf{R}_{Nad\times(N-Nad)}$.

2. $\underline{N < N_{ad}}$: In the simulations performed for this case we have used the same unknown system of length $N = 9$ and the same correlated input sequence given by the model in (2.57). The length of the adaptive filter was $N_{ad} = 11$ and the step-size used to

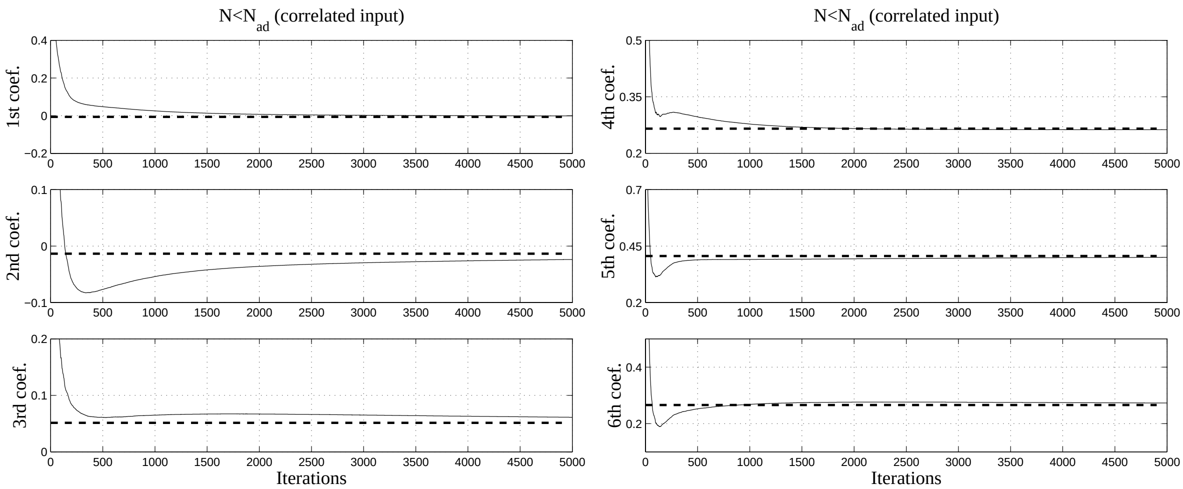

Figure 2.5: The coefficients $\widehat{h}_1(n)$ to $\widehat{h}_6(n)$ of the adaptive filter, during adaptation (continuous line) and the corresponding coefficients of the unknown system (dashed line) for correlated input sequence and $N < N_{ad}$.

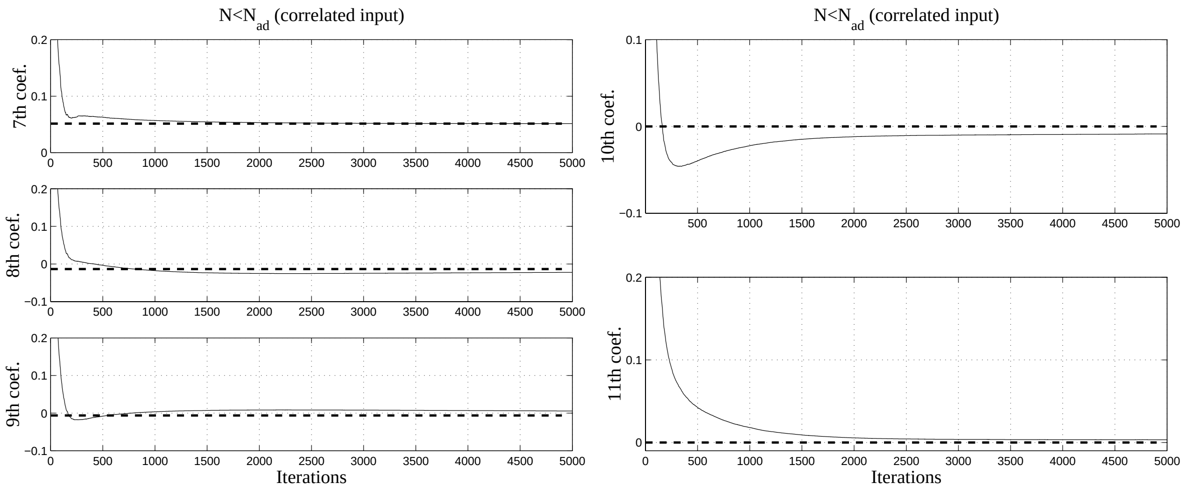

Figure 2.6: The coefficients $\widehat{h}_7(n)$ to $\widehat{h}_{11}(n)$ of the adaptive filter, during the adaptation (continuous line) and the corresponding coefficients of the unknown system (dashed line) for correlated input sequence and $N < N_{ad}$.

update its coefficients was $\mu = 10^{-2}$ which satisfies the stability condition in the mean square sense.

The behaviors of the adaptive filter coefficients are shown in Fig. 2.5 and Fig. 2.6. In both figures, the corresponding coefficients of the unknown system are also plotted as dashed lines. Since the adaptive filter has larger length than the unknown

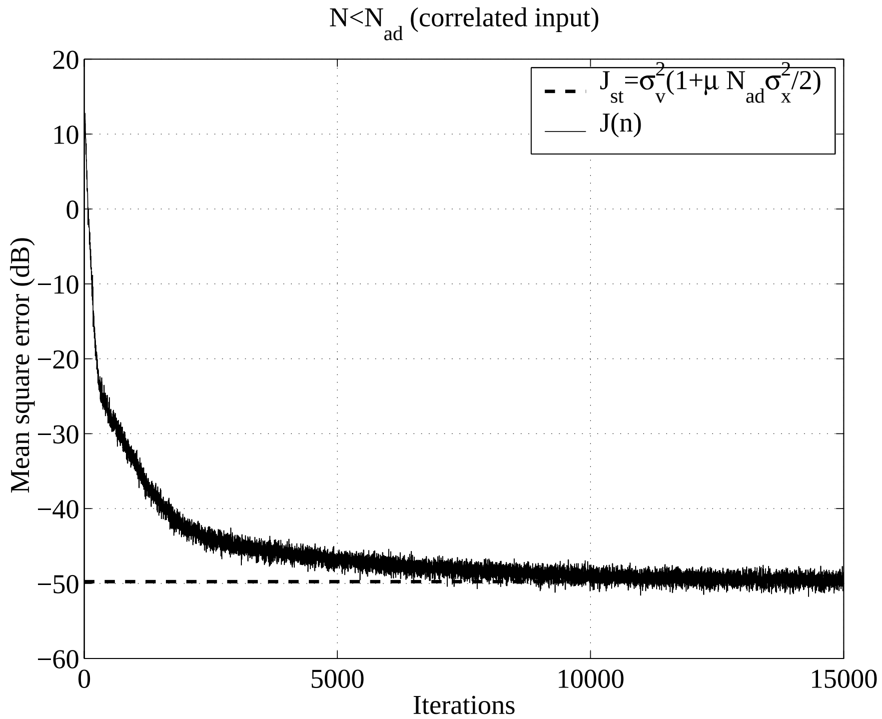

Figure 2.7: Mean squared error for correlated input sequence and $N < N_{ad}$.

system, in the last to plots of Fig. 2.6 we have shown the zero level with dashed line. From Fig. 2.5 and Fig. 2.6 we can see that the first 9 coefficients converge to $\mathbf{h}_N$ whereas the last two coefficients of the adaptive filter converge to zero, which demonstrate the analytical result of (2.47).

The MSE during the adaptation is shown in Fig. 2.7 together with the steady-state MSE obtained when the length of the adaptive filter equals the length of the unknown system. From this figure, we can see that the convergence of the MSE for $N < N_{ad}$ and for $N = N_{ad}$ are the same. This is due to the fact that the extra coefficients of the adaptive filter converges to zero as shown in Fig. 2.6.

Uncorrelated input sequence:

1. $N > N_{ad}$: The behavior of an adaptive filter using the LMS algorithm is studied in the system identification setup depicted in Fig. 2.2, for the case when the length N_{ad} of the adaptive filter is smaller than the length N of the unknown system and the input $x(n)$ is uncorrelated. To this end, we have chosen $N = 9$, $N_{ad} = 5$ and a zero mean random Gaussian distributed input sequence with unity variance. In this case, the trace of the input autocorrelation matrix is $tr\left[\mathbf{R}_{Nad\times Nad}\right] = 5$ and a

step-size $\mu = 10^{-2}$ which fulfill the condition for the MSE convergence was used.

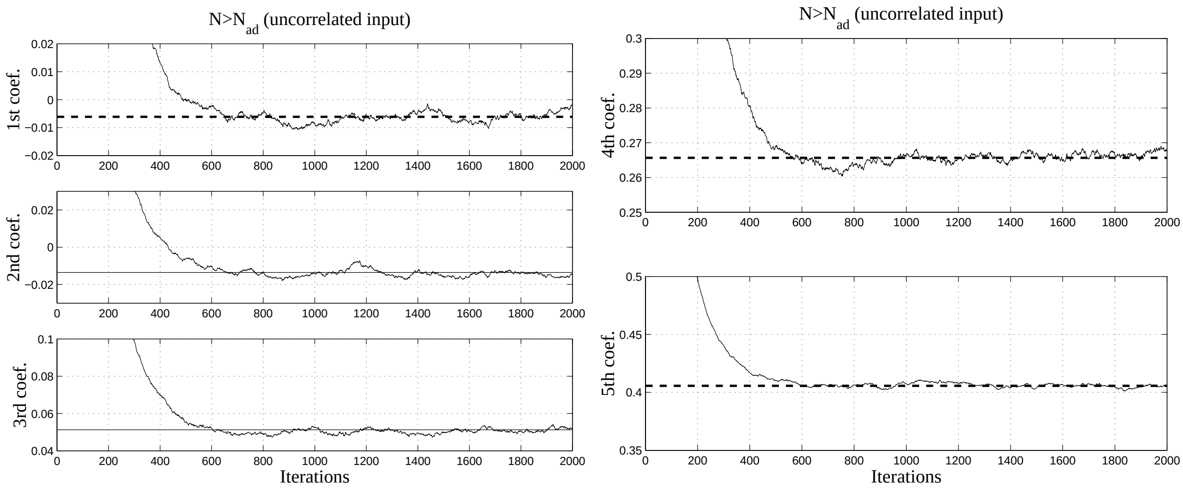

Figure 2.8: Coefficients for $\widehat{h}_1(n)$ to $\widehat{h}_5(n)$ of the adaptive filter, during adaptation (continuous line) and the corresponding coefficients of the unknown system (dashed line) for uncorrelated input sequence and $N > N_{ad}$.

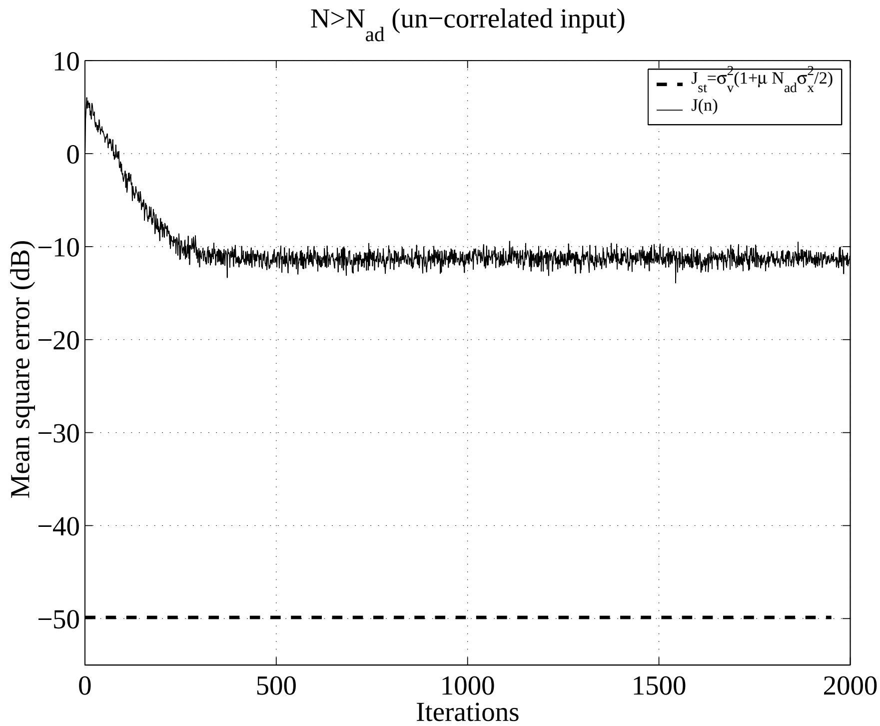

Figure 2.9: Mean squared error for uncorrelated input sequence and $N > N_{ad}$.

The values of the coefficients during the adaptation are shown in Fig. 2.8 together with the values of the first five coefficients of the unknown system (dashed line). We see that the coefficients of the adaptive filter converge close to their corresponding coefficients of the unknown system.

The mean squared error $J(n)$ during adaptation is shown in Fig. 2.9 together with the value of the steady-state MSE, denoted as J_{st}, obtained in the case of equal lengths. Clearly, there is a positive bias term due to the extra coefficients of the unknown filter (see (2.56)).

2. $\underline{N < N_{ad}}$: In order to study the MSE and the coefficient behavior for the case when the adaptive filter has more coefficients than the unknown system the same simulations were repeated for $N = 9$ and $N_{ad} = 11$. The coefficients of the unknown system were the same as in the previous experiments, the input sequence $x(n)$ was a zero mean random Gaussian-distributed sequence with variance $\sigma_x^2 = 1$ and a step-size $\mu = 10^{-2}$ was chosen.

   The coefficients of the adaptive filter versus the coefficients of the unknown system, during the adaptation are shown in Fig. 2.10 and Fig. 2.11. Due to the fact that the adaptive filter has more coefficients than $\mathbf{h}_N$, in the last two plots of Fig. 2.11, the zero levels are shown with dashed lines. We can see from these plots that the first 9 coefficients of the adaptive filter converge close to the corresponding coefficients of the unknown system, whereas the extra coefficients converge to zero.

   The MSE at the output of the adaptive filter is plotted in Fig. 2.12 together with the value of the steady-state MSE obtained for equals lengths (dashed line). We can see that, although the adaptive filter has more coefficients, the steady-state value of the MSE is unbiased. This result is the consequence of the fact that the extra coefficients of the adaptive filter converge to zero and the others converge to $\mathbf{h}_N$.

In conclusion, the analysis of the adaptive FIR filter using the LMS algorithm for the problem of system identification was presented. In the analysis, we have been interested in studying the effect of the mismatch between the lengths of the unknown system and the adaptive filter. The aim of this study is to provide a theoretical basis for the development of the class of Variable Length LMS algorithms in which not only the estimation of the coefficients is of interest but also the length estimation.

From the presented analysis, we can conclude the following: when the input sequence is correlated and the adaptive filter is smaller than the unknown system the coefficients of the adaptive filter converges to the biased values of the first N_{ad} coefficients of $\mathbf{h}_N$. In this case, the steady-state MSE is also biased as compared with the case of equal lengths.

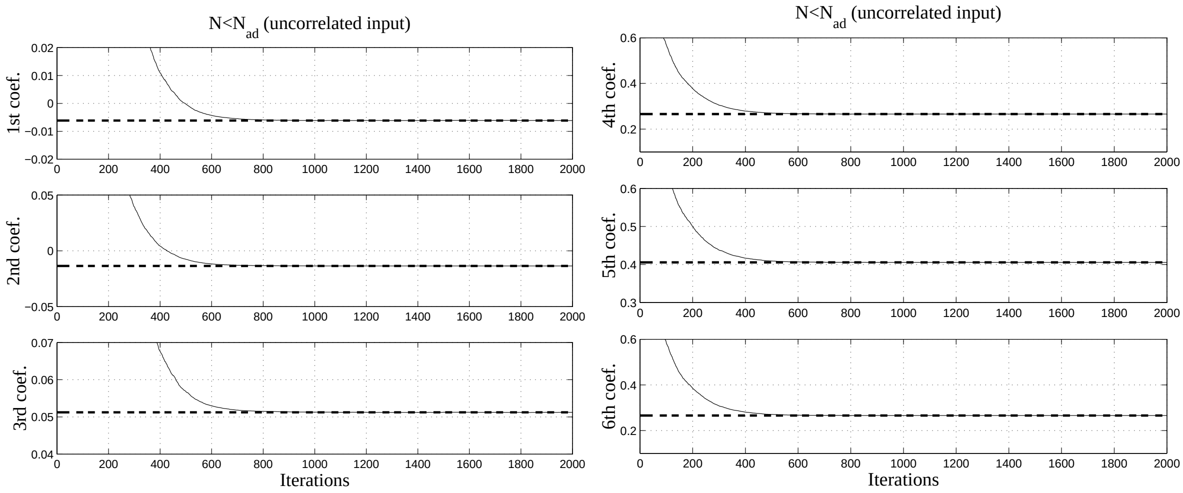

Figure 2.10: Coefficients for $\widehat{h}_1(n)$ to $\widehat{h}_6(n)$ of the adaptive filter, during adaptation (continuous line) and the corresponding coefficients of the unknown system (dashed line) for uncorrelated input sequence and $N < N_{ad}$.

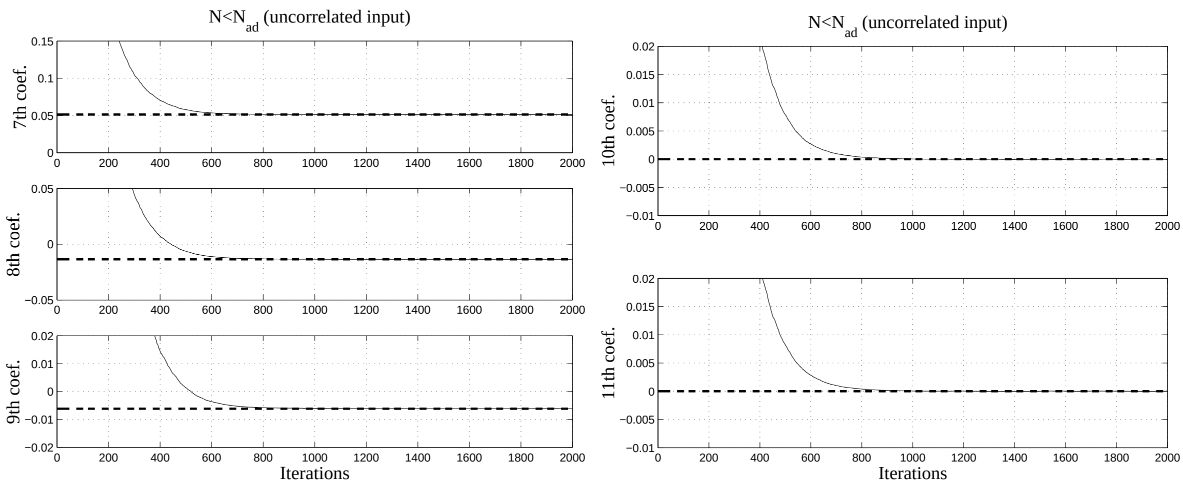

Figure 2.11: Coefficients for $\widehat{h}_7(n)$ to $\widehat{h}_{11}(n)$ of the adaptive filter, during adaptation (continuous line) and the corresponding coefficients of the unknown system (dashed line) for uncorrelated input sequence and $N < N_{ad}$.

When the length of $\widehat{\mathbf{h}}_{Nad}$ is larger than the length of the unknown system, the first N coefficients of the adaptive filter converge to $\mathbf{h}_N$ and the others to zero. In this case the steady-state MSE is the same as in the case of $N = N_{ad}$. Unfortunately, for correlated input sequence there is no direct dependence between the value of the MSE bias and the difference between the filter lengths. In some cases, the extra coefficients of the unknown

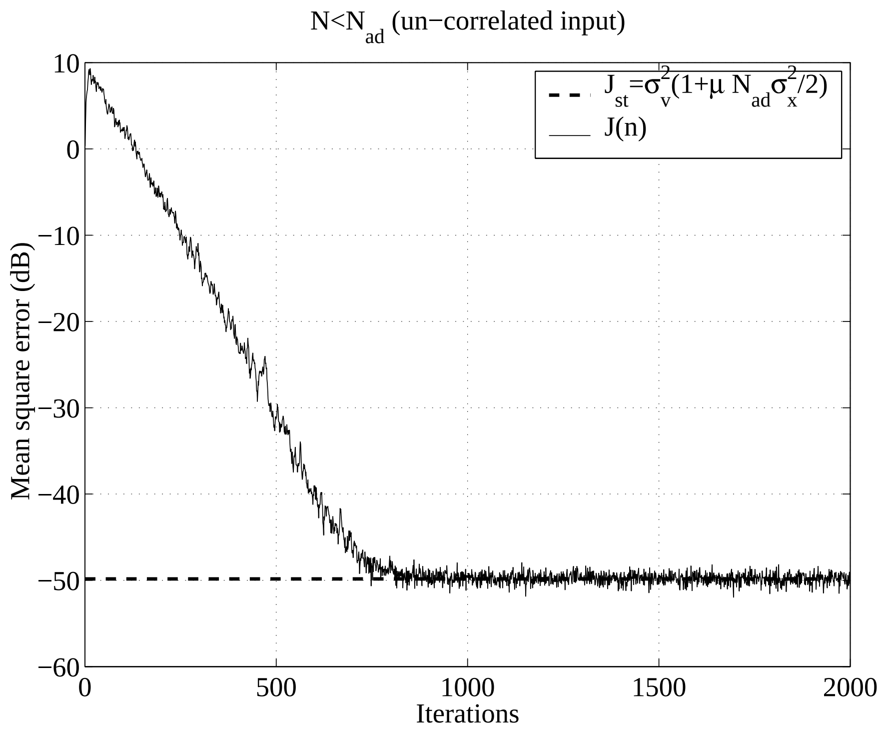

Figure 2.12: Mean squared error for uncorrelated input sequence and $N < N_{ad}$.

system can have negative values therefore, there is no guarantee that for larger mismatch of the lengths the MSE bias is increased.

For uncorrelated input sequence, the optimum coefficients are equal to the first N_{ad} coefficients of the unknown system in the case $N_{ad} < N$ whereas for $N_{ad} \geq N$ the first N coefficients of the optimum solution are equal to the coefficients of $\mathbf{h}_N$ and the others are zero. As a consequence the steady-state output MSE is biased when $N_{ad} < N$ and unbiased for $N_{ad} \geq N$. More than that, there is a direct dependence between the bias level and the length mismatch of the filters. When the adaptive filter has length closer to the length of the unknown system the level of the bias is smaller whereas when the difference between these two lengths is increased the bias increases.

2.1.3 Optimum step-size for time-varying environments

In this section, we study the behavior of the LMS adaptive algorithm for a time-varying environment. With reference to Section 2.1.1, we derive here the equivalent Wiener-Hopf equation (2.7) for time-varying environment. To this end, we start from the minimization of the MSE defined in (2.3). The optimum coefficients are obtained making the partial derivatives of $J(n)$ equal to zero and we obtain:

$$\begin{cases} h_{o_1}(n)r_{1-1}(n) + h_{o_2}(n)r_{1-2}(n) + \cdots + h_{o_N}(n)r_{1-N}(n) = p_0(n) \\ h_{o_1}(n)r_{2-1}(n) + h_{o_2}(n)r_{2-2}(n) + \cdots + h_{o_N}(n)r_{2-N}(n) = p_1(n) \\ \dots \\ h_{o_1}(n)r_{N-1}(n) + h_{o_2}(n)r_{N-2}(n) + \cdots + h_{o_N}(n)r_0(n) = p_N(n) \end{cases} \tag{2.58}$$

where $r_{i-j}(n) = E\left[x(n-i+1)x(n-j+1)\right]$, $p_i(n) = E\left[d(n)x(n-i+1)\right]$ and $\mathbf{p}(n) = \left[p_1(n), p_2(n), \dots, p_N(n)\right]^t$.

In (2.58), the input sequence $x(n)$ and the desired sequence $d(n)$ were assumed to be non-stationary as opposed with (2.3), where $x(n)$ and $d(n)$ were assumed stationary. As a consequence, the expectation operator of $x(n-i+1)x(n-j+1)$ depends on the time instant n, such that, $E\left\{x(n-i+1)x(n-j+1)\right\} \neq E\left\{x(m-i+1)x(m-j+1)\right\}$ for $n \neq m$. This is why the time instant n appears explicitly in (2.58). The indexes of the elements $r_{i-j}(n)$ represent the difference between the time instants of $x(n-i+1)$ and $x(n-j+1)$ and they express the fact that the cross-correlations between different inputs of the adaptive filter are computed. For the same reason, the elements of the vector $\mathbf{p}(n)$ contain information about the time instant at which they were computed (since $x(n)$ and $d(n)$ are assumed to be non-stationary). Since the autocorrelation matrix $\mathbf{R}(n)$ and the vector $\mathbf{p}(n)$ are time-varying, also the optimum vector $\mathbf{h}_o(n)$ is time-varying and (2.58) can be written in a compact form as follows:

$$\mathbf{R}(n)\mathbf{h}_o(n) = \mathbf{p}(n). \tag{2.59}$$

where:

$$\mathbf{R}(n) = \begin{bmatrix} r_0(n) & r_1(n) & \dots & r_{N-1}(n) \\ r_1(n) & r_0(n) & \dots & r_{N-2}(n) \\ \dots & \dots & \dots & \dots \\ r_{N-1}(n) & r_{N-2}(n) & \dots & r_0(n) \end{bmatrix} \tag{2.60}$$

We note that, (2.59) includes all the non-stationary situations that can appear in practice as follows: if $x(n)$ is non-stationary but $d(n)$ is stationary the autocorrelation matrix and the cross-correlation vector are both time-varying. When $x(n)$ and $d(n)$ are both non-stationary, $\mathbf{R}(n)$ and $\mathbf{p}(n)$ are again time-varying. In applications, where the input sequence $x(n)$ is stationary and the desired sequence $d(n)$ is non-stationary, just the cross-correlation vector $\mathbf{p}(n)$ is time-varying and the matrix $\mathbf{R}$ has fixed coefficients. As a result, in all the above mentioned cases the optimum vector $\mathbf{h}_o(n)$ has time-varying coefficients.

Here we address the problem of time-varying system identification depicted in Fig. 2.13, where $\mathbf{h}(n) = \left[h_1(n), \dots, h_N(n)\right]^t$ is a linear time-varying channel whose coefficients

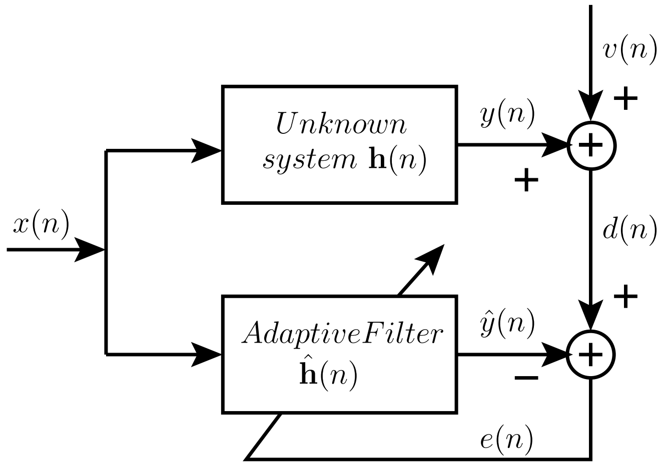

Figure 2.13: Block diagram of time-varying system identification using an adaptive FIR filter.

must be estimated by an adaptive FIR filter with the same number of coefficients $\widehat{\mathbf{h}}(n) = \left[\widehat{h}_1(n), \widehat{h}_2(n) \ldots, \widehat{h}_N(n)\right]^t$, $x(n)$ is the stationary input sequence, $d(n)$, $e(n)$ and $v(n)$ are the desired sequence, the output error and the output noise respectively.

We note that the difference between Fig. 2.2 and Fig. 2.13 consists in the fact that the unknown system does not have fixed coefficients but they are time-varying. However, the length of the unknown system here is assumed to be known and an adaptive FIR filter with the same length is implemented. Due to the fact that the input sequence $x(n)$ is stationary and the desired sequence $d(n)$ is obtained at the output of a time-varying FIR filter, the input autocorrelation matrix is time-invariant and the cross-correlation vector $\mathbf{p}(n)$ is time-varying. Therefore, we restrict our discussion to the applications where the non-stationarity appears into the desired sequence $d(n)$, and (2.59) simplifies to:

$$\mathbf{R}\mathbf{h}_o(n) = \mathbf{p}(n). \tag{2.61}$$

The other situations, when the non-stationarity appears in the input sequence or in both $x(n)$ and $d(n)$, are left beyond the scope of this thesis. Although, we address here just this case, there are many practical applications in which the analytical results developed in this section and the algorithm introduced in the subsequent are of great interest. One example of such application is echo cancellation where the echo path can be modelled by a time-varying FIR filter.

With reference to Fig. 2.13, the noisy observation at time n is given by:

$$d(n) = y(n) + v(n) = \mathbf{h}^t(n)\mathbf{x}(n) + v(n). \quad (2.62)$$

where $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^t$ is the tap input vector.

To obtain the optimum coefficients vector $\mathbf{h}_o(n)$ we express the cross-correlation vector $\mathbf{p}(n)$ in terms of autocorrelation matrix and vector $\mathbf{h}(n)$. Assuming that the output noise $v(n)$ is zero mean and independent to $x(n)$, (2.61) can be written:

$$\mathbf{R}\mathbf{h}_o(n) = \mathbf{R}\mathbf{h}(n) \quad \rightarrow \quad \mathbf{h}_o(n) = \mathbf{h}(n). \quad (2.63)$$

As a result, the optimum coefficients vector at each time instant n equals the vector of the time-varying coefficients of the unknown system. The minimum output error is obtained in the ideal case when the vector of the adaptive filter coefficients equals $\mathbf{h}_o(n)$ and it can be expressed as follows:

$$e_o(n) = \mathbf{h}^t(n)\mathbf{x}(n) - \mathbf{h}_o^t(n)\mathbf{x}(n) + v(n) = v(n), \quad (2.64)$$

From (2.64), the minimum MSE is obtained as follows:

$$J_{min} = E\left[e_o^2(n)\right] = E\left[v^2(n)\right] = \sigma_v^2. \quad (2.65)$$

The coefficients of the adaptive filter $\widehat{\mathbf{h}}(n)$ are modified to minimize the output mean squared error. When the LMS algorithm is used for adaptation, the update formula for the coefficients of the adaptive filter is:

$$\widehat{\mathbf{h}}(n+1) = \widehat{\mathbf{h}}(n) + \mu e(n)\mathbf{x}(n). \quad (2.66)$$

where μ is the step-size.

In order to make the theoretical analysis more tractable, the following assumptions are commonly used in the open literature (see [28] and [41]):

1. $\mathbf{x}(n)$ and $e_o(n)$ are zero-mean, stationary, jointly normal and with finite moments. Where $e_o(n)$ is the output error in the case of perfect adaptation (when $\widehat{\mathbf{h}}(n) = \mathbf{h}_o(n)$).

2. The successive increments $\epsilon(n) = \mathbf{h}(n+1) - \mathbf{h}(n)$ of the channel coefficients are independent to each other. However, the elements of $\epsilon(n)$ might be statistically dependent for a given n. The sequence $\epsilon(n)$ is zero-mean and stationary such that the covariance matrix of the filter coefficients increments $\mathbf{Q} = E\left[\epsilon(n)\epsilon^t(n)\right]$ is time invariant.

3. $\widehat{\mathbf{h}}(n)$ is independent to $e_o(n)$ and $\mathbf{x}(n)$. This assumption is satisfied when the value of the step-size μ is small enough.

4. $e_o(n)$, $\epsilon(n)$ and $\mathbf{x}(n)$ are independent to each other.

According to Assumption 2, we restrict our discussion to the tracking of time-varying systems given by the following model:

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \epsilon(n), \tag{2.67}$$

with $\epsilon(n)$ being the $N \times 1$ vector of the channel increments.

However, the time-varying model in (2.67) is not the only model which satisfy the Assumption 2. It was shown in [25] that also the Markov model $\mathbf{h}(n+1) = \alpha\mathbf{h}(n) + \epsilon(n)$ satisfy Assumption 2 provided that the constant α is close to unity.

Subtracting $\mathbf{h}_o(n) = \mathbf{h}(n)$ from both sides of (2.66) and using (2.67), the coefficient error vector is obtained as follows:

$$\begin{aligned} \Delta\mathbf{h}(n+1) &= \widehat{\mathbf{h}}(n+1) - \mathbf{h}(n+1) = \widehat{\mathbf{h}}(n) - \mathbf{h}(n) - \epsilon(n) + \mu e(n)\mathbf{x}(n), \\ \Delta\mathbf{h}(n+1) &= \Delta\mathbf{h}(n) - \epsilon(n) + \mu e(n)\mathbf{x}(n) \end{aligned} \tag{2.68}$$

The output error is obtained as:

$$\begin{aligned} e(n) &= y(n) - \widehat{y}(n) + v(n) = \mathbf{h}^t(n)\mathbf{x}(n) - \widehat{\mathbf{h}}^t(n)\mathbf{x}(n) + v(n), \\ e(n) &= \mathbf{h}^t(n)\mathbf{x}(n) - \mathbf{h}_o^t(n)\mathbf{x}(n) + \mathbf{h}_o^t(n)\mathbf{x}(n) - \widehat{\mathbf{h}}^t(n)\mathbf{x}(n) + v(n), \\ e(n) &= v(n) - \Delta\mathbf{h}^t(n)\mathbf{x}(n). \end{aligned} \tag{2.69}$$

where $\widehat{y}(n) = \widehat{\mathbf{h}}(n)^t\mathbf{x}(n)$.

From (2.68) and (2.69) it follows, that the coefficient error vector can be written in the following manner:

$$\Delta\mathbf{h}(n+1) = \Delta\mathbf{h}(n) - \epsilon(n) - \mu\mathbf{x}(n)\mathbf{x}^t(n)\Delta\mathbf{h}(n) + \mu v(n)\mathbf{x}(n) \tag{2.70}$$

The mean of the coefficient error vector is obtained taking the expectation operator in (2.70) and using the Assumptions 1 and 2:

$$E\left[\Delta\mathbf{h}(n+1)\right] = \left(\mathbf{I} - \mu\mathbf{R}\right) E\left[\Delta\mathbf{h}(n)\right]. \tag{2.71}$$

Following a similar derivation as in [25], [28] and [41], it can be shown that for large n the left hand side of (2.71) converges to zero provided that the step-size satisfies the following condition:

$$0 < \mu < \frac{2}{\lambda_{max}}. \tag{2.72}$$

where λ_{max} is the maximum eigenvalue of $\mathbf{R}$.

At this point we should make two important remarks:

- The expected value of the coefficient error vector $E[\Delta\mathbf{h}(n)]$, for large n converges to zero in view of the Assumption 2 in which the increments $\epsilon(n)$ of the unknown filter coefficients are assumed zero mean. If this assumption is violated, in the right-hand side of (2.71) a non-zero term $E[\epsilon(n)]$ appears which prevents the vector $E[\Delta\mathbf{h}(n+1)]$ from converging to zero.

- The convergence condition of (2.72) is valid for a time-invariant autocorrelation matrix $\mathbf{R}$. When the autocorrelation matrix is time-varying (due to the non-stationarity of $x(n)$), the stability condition of (2.72) becomes:

$$0 < \mu < \frac{2}{\lambda_{max}(n)}. \tag{2.73}$$

  which means that the step-size μ should be smaller than the inverse of the maximum eigenvalue of $\mathbf{R}(n)$ computed at each time instant n.

The cross-correlation matrix $\mathbf{C}(n) = E[\Delta\mathbf{h}(n)\mathbf{h}^t(n)]$, of the coefficient error vector can be computed multiplying (2.70) with its transpose and taking the expectation operator, as follows:

$$\begin{aligned}\mathbf{C}(n+1) &= E\left[\Delta\mathbf{h}^t(n+1)\Delta\mathbf{h}(n+1)\right] = \\ &= \mathbf{C}(n) + \mathbf{Q} - \mu\mathbf{R}\mathbf{C}(n) - \mu\mathbf{C}(n)\mathbf{R} + \mu^2\sigma_v^2\mathbf{R} + \mu^2\mathbf{R}tr\left[\mathbf{R}\mathbf{C}(n)\right] + 2\mu^2\mathbf{R}\mathbf{C}(n)\mathbf{R}.\end{aligned} \tag{2.74}$$

For small value of the step-size μ the last two terms in (2.74) can be neglected (see [28]) and the cross-correlation matrix $\mathbf{C}(n+1)$ becomes:

$$\mathbf{C}(n+1) = \mathbf{C}(n) + \mathbf{Q} - \mu\mathbf{R}\mathbf{C}(n) - \mu\mathbf{C}(n)\mathbf{R} + \mu^2\sigma_v^2\mathbf{R}, \tag{2.75}$$

At the steady-state, for large values of n we have $\lim_{n\to\infty}\mathbf{C}(n+1) = \lim_{n\to\infty}\mathbf{C}(n)$ and (2.75) can be written as in the sequel:

$$\mu\mathbf{R}\mathbf{C}(\infty) + \mu\mathbf{C}(\infty)\mathbf{R} = \mathbf{Q} + \mu^2\sigma_v^2\mathbf{R}, \tag{2.76}$$

The steady-state mean square coefficient error defined as $\Theta_{st} = tr[\mathbf{C}(\infty)]$ is obtained by pre-multiplying (2.76) with $\mathbf{R}^{-1}$ and taking the trace, as follows:

$$\begin{aligned}2\mu\Theta_{st} &= tr\left[\mathbf{R}^{-1}\mathbf{Q}\right] + \mu^2\sigma_v^2 tr\left[\mathbf{I}\right], \\ \Theta_{st} &= \frac{1}{2}\left[\mu\sigma_v^2 N + \frac{1}{\mu}tr\left[\mathbf{R}^{-1}\mathbf{Q}\right]\right]\end{aligned} \tag{2.77}$$

where we have used the fact that $tr\left[\mathbf{AB}\right] = tr\left[\mathbf{BA}\right]$.

Using Assumptions 1 to 4 and following a development similar to [28], [41], which is not detailed here, the steady-state MSE is obtained as follows:

$$J_{st} = J_{min} + tr\left[\mathbf{RC}(n)\right], \tag{2.78}$$

where J_{min} is the minimum MSE in the case of perfect adaptation and it is expressed by (2.65).

Finally, after some mathematical manipulations, the steady-state MSE can be explicitly expressed by the following formula:

$$J_{st} = \sigma_v^2 + \frac{1}{2}\left[\mu\sigma_v^2 tr\left[\mathbf{R}\right] + \frac{tr\left[\mathbf{Q}\right]}{\mu}\right] \tag{2.79}$$

The steady-state MSE given by (2.79) is a nonlinear function of the step-size μ and it possesses a minimum (see [28] and [41]) which corresponds to an optimum step-size μ_{opt}. The value of μ_{opt} can be computed taking the derivative of (2.79) with respect to μ equal to zero.

$$\frac{\delta J_{st}}{\delta\mu} = 0 \Rightarrow \mu_{opt}^{mse} = \sqrt{\frac{tr\left(\mathbf{Q}\right)}{\sigma_v^2 tr\left[\mathbf{R}\right]}} \tag{2.80}$$

Also the steady-state value of the mean square coefficient error Θ_{st} expressed by (2.77) is a nonlinear function of the step-size μ and its minimum is obtained for:

$$\mu_{opt}^{coef} = \sqrt{\frac{tr\left[\mathbf{R}^{-1}\mathbf{Q}\right]}{\sigma_v^2 N}}. \tag{2.81}$$

In conclusion, the steady-state MSE and the steady-state mean square coefficients (MSC) error, for a time-varying system identification are nonlinear functions on the step-size μ. This effect is different from the case of a time-invariant system identification where the dependency between the steady-state MSE and steady-state MSC are linear functions on the step-size μ. As a consequence, for applications in which minimization of the output error is of primary interest, a step-size close to μ_{opt}^{mse} should be used. On the contrary, when the minimization of the coefficient error represents the main goal, the step-size used in the adaptation process must be close to μ_{opt}^{coef}. Another interesting conclusion is that the class of Variable Step-Size LMS algorithms which were introduced based on the analytical results obtained for a time-invariant environment might not be suitable for time-varying systems. The VSSLMS which were derived based on the linear dependence between the steady-state MSE and the step-size might not give good results in time-varying environments in the sense that if the step-size is decreased as the algorithm goes close to the steady-state the MSE may actually increase. However, the optimum step-size μ_{opt}^{mse} can be accommodated in the VSSLMS such that the speed of convergence is increased while maintaining a small output missadjustment.

In order to use (2.80), for computation of the optimum step-size, one needs to know $tr\,[\mathbf{Q}]$, $tr\,[\mathbf{R}]$, and σ_v^2. Although the trace of $\mathbf{R}$ can be estimated during the adaptation, the noise variance and the trace of $\mathbf{Q}$ are not known in practice. The common method in the present literature is to estimate the channel parameters and the channel noise prior to adaptation and to compute the optimum step-size using (2.80). The problem is when the channel statistics cannot be easily and accurately estimated or when they change during the adaptation. In such cases, the optimum step-size is impossible to be estimated in advance. Iterative methods are therefore necessary which adaptively changes the step-size toward the optimum, such that the steady-state MSE or steady-state MSC error are minimized. An iterative algorithm for step-size adaptation toward μ_{opt}^{mse} is introduced in Section 2.3 of this thesis.

2.1.4 Simulations and results

At this point, we will perform some computer simulations with the aim to verify the analytical results of (2.77) and (2.79). To this end, we have implemented an adaptive FIR filter in time-varying system identification framework as depicted in Fig. 2.13. The length of the unknown system $\mathbf{h}(n)$ was $N = 4$ and the length of the adaptive filter $\widehat{\mathbf{h}}(n)$ is equal. The coefficients of the unknown system were modelled as in (2.67) and the elements of $\epsilon(n)$ were zero mean random Gaussian-distributed with identical variances $\sigma_\epsilon^2 = 10^{-5}$, such that the cross-correlation matrix equal $\mathbf{Q} = \sigma_\epsilon^2 \mathbf{I}$ with $\mathbf{I}$ being the identity matrix. The input sequence $x(n)$ used in the simulations was given by the following model:

$$x(n) = \alpha x(n-1) + \eta(n).$$

where $\alpha = 0.75$ and $\eta(n)$ was zero mean Gaussian distributed with variance chosen, such that the variance of $x(n)$ was unity.

The optimum step-sizes μ_{opt}^{mse} and μ_{opt}^{coef} which minimize the steady-state MSE and Θ_{st} are obtained from (2.77) and (2.79) as follows:

$$\mu_{opt}^{mse} = 10^{-2} \quad \text{and} \quad \mu_{opt}^{coef} = 0.0171$$

We have conducted several experiments using the above system setup, and for each experiment the step-size used to update the adaptive filter coefficients was constant.

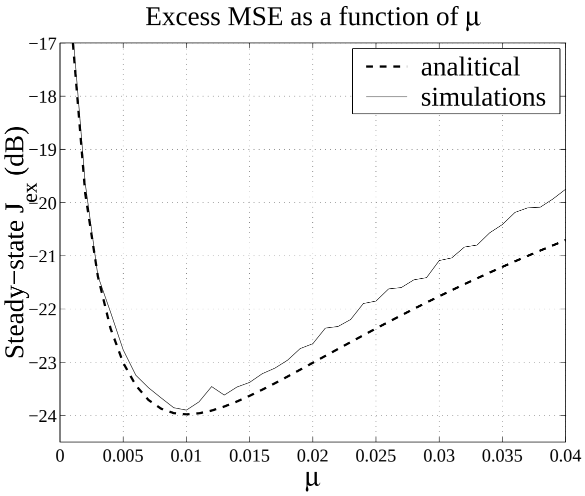

Figure 2.14: Steady-state excess MSE vs. the step-size for a time-varying system identification: experimental results (continuous line) and theoretical results (dashed line).

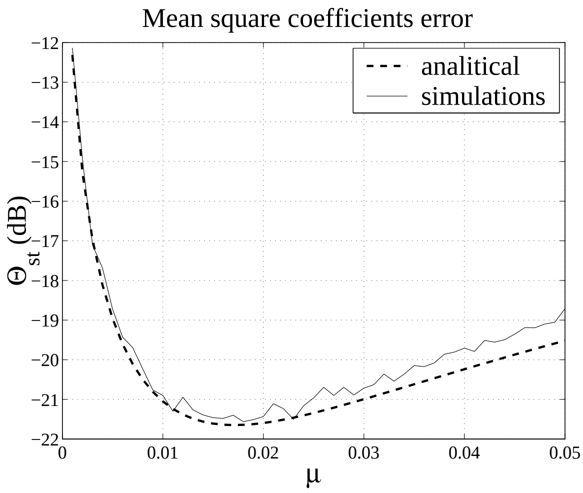

Figure 2.15: Steady-state mean square coefficient error vs. the step-size for a time-varying system identification: experimental results (continuous line) and theoretical results (dashed line).

However, the step-sizes used in different experiments were not equals. For each experiment a number of 100 independent runs of 10^4 iterations were performed and the results were averaged. In every experiment the steady-state MSE and the steady-state mean square coefficient error are computed and they are plotted in Fig. 2.14 and Fig. 2.15 respectively together with their theoretical values obtained by estimation of (2.77) and (2.79) for different step-sizes.

A good agreement between the theory and practice can be observed in both Fig. 2.14 and Fig. 2.15.

2.2 Variable step-size Least Mean Square algorithms

As we outlined in Section 2.1, in practical applications adaptive algorithms which possess high convergence speed while maintaining small convergence error are of great interest. For instance, in channel equalization during the transient period, the frequency characteristic of the adaptive equalizer is far from the inverse of the frequency response of the channel therefore, the data transmitted during this time will be corrupted. In echo cancellation application, if the coefficients of the adaptive canceler are not close to the coefficients of the FIR filter which models the echo path the resulting echo signal is not attenuated. Actually, it is possible in this application, that during the transient period, the echo will be actually amplified. As a consequence, the transient period of the adaptive filter must be as small as possible for most of the practical applications in order to improve the overall quality of the system.

The Least Mean Square algorithm has a small computational complexity therefore, it is very simple to be implemented in practice. Although its simplicity, one of its main drawbacks is the fact that the speed of convergence and steady state error depends on the same parameter, the step-size μ. This conclusion was pointed out in Section 2.1.1 equations (2.27) and (2.28), where we have seen that there is a direct dependence between the step-size and the misadjustment, while the convergence speed is inverse proportional with μ. In conclusion, when a constant step-size is used in the LMS, there is a tradeoff between the steady-state error and the convergence speed, which prevent a fast convergence when the step-size is chosen to be small for small output error. In order to deal with this problem, a simple idea is to use a step-size which is time-varying during the adaptation. At early stages of the adaptation, when the adaptive filter is far from the optimum, a larger value of the step-size should be used. This will shorten the transient period and increase the convergence speed of the adaptive filter. As the adaptive filter goes close to the optimum Wiener solution, the step-size should be decreased and so the misadjustment expressed by (2.26). The adaptive algorithms derived from the LMS, which uses time-varying step-size modified as described above, belong to the class of Variable Step-Size LMS (VSSLMS) algorithms. We should note that there are other adaptive algorithms with time-varying step-size, which we do not include in the class of VSSLMS due to the fact that they use step-size adaptation for other purposes, such as, finding the optimum step-size for time-varying environments.

In this section, we emphasize on the class of VSSLMS which uses time-varying step-size to improve the convergence speed and also to reduce the tradeoff between the misadjustment and the convergence time. Other algorithms with time-varying step-size are discussed in subsequent sections.

2.2.1 Existing approaches

We review here some of the most cited algorithms from the class of VSSLMS and after that in the next two sections we introduce two new VSSLMS algorithms. All the algorithms from this section are described with reference to Fig. 2.1.

The VSSLMS algorithm

The VSSLMS algorithm first introduced by Kwong and Johnston in [48] uses the following update formula for the adaptive filter coefficients:

$$\widehat{\mathbf{h}}(n+1) = \widehat{\mathbf{h}}(n) + \mu(n)e(n)\mathbf{x}(n). \tag{2.82}$$

where $\widehat{\mathbf{h}}(n)$ is the $N \times 1$ vector of the adaptive filter coefficients, $\mathbf{x}(n)$ is the vector of the past N samples from the input sequence $x(n)$, $\mu(n)$ is a time-varying step-size and $e(n)$ is the output error.

The time-varying step-size is also adapted as in the following equation:

$$\begin{aligned} \mu'(n+1) &= \alpha\mu'(n) + \gamma e^2(n), \\ \mu(n+1) &= \begin{cases} \mu_{max} & \text{if } \mu'(n+1) > \mu_{max}, \\ \mu_{min} & \text{if } \mu'(n+1) < \mu_{min}, \\ \mu'(n+1) & \text{otherwise.} \end{cases} \end{aligned} \tag{2.83}$$

with $0 < \alpha < 1$ and $\gamma > 0$ being some constant parameters and μ_{max} and μ_{min} being the upper and lower bounds of the time-varying step-size.

The constant parameter μ_{max} which is normally selected close to the instability point of the conventional LMS algorithm is used to increase the convergence speed, while the parameter μ_{min} is chosen to provide a good compromise between the steady-state misadjustment and the tracking capability of the algorithm. The parameter γ is used to control the convergence time and also the steady-state level of the misadjustment. The behavior of the step-size as described in (2.83) is the following: at early stages of the adaptation (when the coefficients $\widehat{\mathbf{h}}(n)$ are far from the optimum solution) the step-size is increased due to the large value of the output error. As the algorithm goes closer to the steady-state the value of $e(n)$ decreases which decrease the step-size $\mu(n)$.

The following approximate analytical expression for the steady-state misadjustment of the VSSLMS algorithm was derived in [48]:

$$\mathcal{M} = \frac{J_{ex}}{J_{min}} = \frac{1 - \sqrt{1 - 2\frac{(3-\alpha)\gamma J_{min}}{1-\alpha^2} tr\left[\mathbf{R}\right]}}{1 + \sqrt{1 - 2\frac{(3-\alpha)\gamma J_{min}}{1-\alpha^2} tr\left[\mathbf{R}\right]}} \tag{2.84}$$

Clearly, the steady-state misadjustment depends on the parameter γ and on the minimum value of the MSE J_{min}. Since the speed of convergence of the algorithm depends also on the parameter γ we can conclude that there is still a dependence between the misadjustment and the convergence time. Another drawback of this algorithm is the fact that the steady-state misadjustment depends also on J_{min}. For instance in system identification applications, the minimum MSE equals the output noise variance, therefore the steady-state misadjustment depends on the system noise.

The robust variable step-size LMS algorithm

In order to reduce the influence of J_{min} in the steady-state misadjustment (see (2.84)), the Robust Variable Step-Size LMS (RVSLMS) algorithm was proposed in [1]. The RVSLMS uses an estimate of the autocorrelation between the output error at adjacent time instants $e(n)$ and $e(n-1)$ to control the step-size. The same objective, to increase the step-size at early stages of the adaptation and to decrease $\mu(n)$ when the algorithm approaches the steady-state, is followed.

The following update equation is used for the step-size:

$$\begin{aligned} p(n) &= \beta p(n-1) + (1-\beta)\, e(n)e(n-1), \\ \mu'(n+1) &= \alpha\mu'(n) + \gamma p^2(n), \\ \mu(n+1) &= \begin{cases} \mu_{max} & \text{if } \mu'(n+1) > \mu_{max}, \\ \mu_{min} & \text{if } \mu'(n+1) < \mu_{min}, \\ \mu'(n+1) & \text{otherwise.} \end{cases} \end{aligned} \tag{2.85}$$

where the parameters α, γ, μ_{min} and μ_{max} are the same as those used in the VSSLMS algorithm from [48] and $0 < \beta < 1$ is an exponential weighting factor which controls the quality of the estimation of $p(n)$.

The misadjustment of the RVSLMS was derived in [1] for both stationary and non-stationary environments. For stationary environment the steady-state misadjustment is given by the following expressions [1]:

$$\mathcal{M} = \frac{\gamma\alpha J_{min}^2 (1-\beta)}{(1-\alpha^2)(1+\beta)} tr\left[\mathbf{R}\right]. \tag{2.86}$$

Although the misadjustment of the RVSLMS still depends on J_{min}, the parameter β which is introduced in addition together with γ controls the steady-state misadjustment. The parameter β can be chosen, such that, a small $\mathcal{M}$ is obtained while maintaining a large γ which increases the convergence speed.

The complementary pair LMS algorithm

In order to obtain an algorithm which improves the convergence speed while maintaining a small steady-state error, in [61] the so called Complementary Pair LMS (CP-LMS) algorithm was proposed. The block diagram of an adaptive filter implementing this algorithm

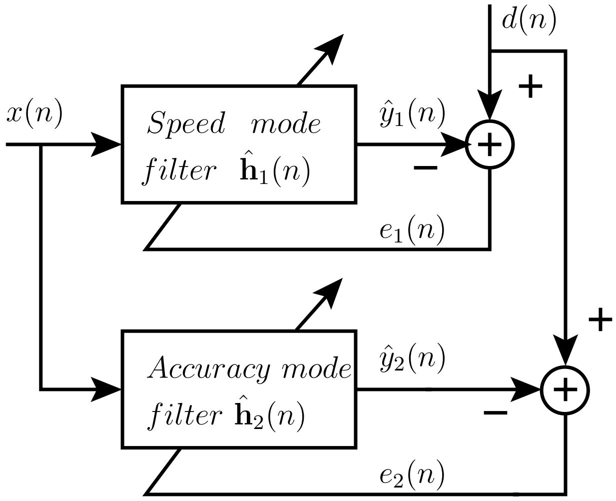

Figure 2.16: The block diagram of the complementary pair LMS.

is depicted in Fig. 2.16, where the speed mode filter $\widehat{\mathbf{h}}_1(n)$ and the accuracy mode filter $\widehat{\mathbf{h}}_2(n)$ represent two adaptive filters whose coefficients are adapted by the LMS algorithm with constant step-sizes μ_1 and μ_2 respectively. The speed mode filter uses a large step-size and it is used to increase the convergence speed while the accuracy mode filter, which uses a small step-size, is implemented to obtain a small steady-state error. Actually the filter of interest is the accuracy mode filter and the other adaptive filter is used just to increase the speed as it will be more clear in the sequel. Although this adaptive filtering structure does not use a time-varying step-size in the update equation, we have chosen to include it in the class of VSSLMS because the coefficients of the accuracy filter are not adapted using only the step-size μ_2 but they are also adapted by μ_1. The coefficients of the accuracy and speed mode filters are updated as in the following equations:

- The update equation of the accuracy mode filter coefficients:

$$\widehat{\mathbf{h}}_2(n+1) = \begin{cases} \widehat{\mathbf{h}}_1(n), & \text{if} \begin{cases} n = T, 2T, 3T, \ldots \\ \text{and} \\ \prod_{j=1}^{J} Q(n-jT) = 1 \end{cases} \\ \widehat{\mathbf{h}}_2(n) + \mu_2 e_2(n)\mathbf{x}(n), & \text{otherwise;} \end{cases} \tag{2.87}$$

where Q is computed as follows:

$$Q(m) = \begin{cases} 1, & \text{if } \sum_{i=m}^{m+T} e_1^2(i) < \sum_{i=m}^{m+T} e_2^2(i) \\ 0 & \text{otherwise;} \end{cases} \tag{2.88}$$

- The coefficients of the speed mode filter are updated as follows:

$$\widehat{\mathbf{h}}_1(n+1) = \widehat{\mathbf{h}}_1(n) + \mu_1 e_1(n)\mathbf{x}(n). \tag{2.89}$$

  where $\widehat{\mathbf{h}}_1(n)$ is the coefficients vector of the speed mode filter, $\widehat{\mathbf{h}}_2(n)$ is the coefficients vector of the accuracy mode filter, μ_1 and μ_2 are the step-sizes of the speed mode filter and accuracy mode filter respectively; $e_1(n)$ and $e_2(n)$ are the errors of the speed mode filter and accuracy mode filter which are computed as follows (see Fig. 2.16):

$$e_1(n) = d(n) - \widehat{y}_1(n) \quad \text{and} \quad e_2(n) = d(n) - \widehat{y}_2(n). \tag{2.90}$$

As we can see from the equations (2.87), (2.88) and (2.89), the CP-LMS algorithm presented in [61] consists in two adaptive filters that operate in parallel one with large step-size, called speed mode filter and another with small step-size, called accuracy mode filter. Both adaptive filters use a fixed step-size in the adaptation process and their coefficients are updated using the standard LMS algorithm for a number of T consecutive iterations which represents the test interval. At the end of each test interval the local averages of the square errors of both adaptive filters are computed and the coefficients of the accuracy mode filter are updated accordingly. If the local average of square error of the accuracy mode filter for L consecutive test intervals is larger than the local average of the square error of the speed mode filter, the coefficients of the accuracy mode filter are updated with the coefficients of the speed mode filter. The reason of this update is the following: when $\sum_{i=m}^{m+T} e_1^2(n) < \sum_{i=m}^{m+T} e_2^2(n)$, it means that the speed mode filter is closer to the optimum solution than $\widehat{\mathbf{h}}_2(n)$ and its coefficients should be used. As both $\widehat{\mathbf{h}}_1(n)$ and $\widehat{\mathbf{h}}_2(n)$ are close to the optimum solution, the accuracy mode filter will perform better than $\widehat{\mathbf{h}}_1(n)$ and its coefficients are adapted using the LMS with a small and fixed step-size.

2.2.2 Complementary Pair Variable Step-Size LMS algorithm

In the CP-LMS algorithm, the coefficients of the accuracy mode filter are re-initialized with the coefficients of the speed mode filter all the time when the local sum of the square error $e_1(n)$ is less than the local sum of the square error $e_2(n)$ for L consecutive test intervals. This is because in that case the coefficients $\widehat{\mathbf{h}}_1(n)$ are closer to the optimum. This observation can be extended to the step-size as well. For a training interval in which the speed mode filter performs better than the accuracy mode filter not only its coefficients are closer to the optimum but also its larger step-size is a better choice. As a consequence, the step-size of the accuracy mode filter must be increased. On the contrary, when the accuracy mode filter is closer to the optimum solution, its step-size is decreased in order to obtain a desired level of the steady-state missadjustment. The new algorithm, called Complementary Pair Variable Step-size LMS (CP-VSLMS) uses the idea of the CP-LMS algorithm proposed in [61], but the difference between the two algorithms consists in the fact that μ_2 is time-varying.

In the case of the CP-VSLMS algorithm, the coefficients of the filter with step $\mu_2(n)$ are re-initialized in the same way as in the CP-LMS algorithm when the local average of the $e_2^2(n)$ is larger than the local average of the $e_1^2(n)$. In the same time, the step $\mu_2(n+1)$ is changed to the value $\frac{\mu_1+\mu_2(n)}{2}$, which increases the convergence speed of the algorithm. When the filter with step $\mu_2(n)$ is closer to the optimum, the step $\mu_2(n)$ is decreased in order to obtain a small steady-state error. As a consequence, the CP-VSLMS algorithm can be described by the following steps:

1. Adaptation of the speed filter coefficients:

$$\widehat{\mathbf{h}}_1(n+1) = \widehat{\mathbf{h}}_1(n) + \mu_1 e_1(n)\mathbf{x}(n), \tag{2.91}$$

   where $e_1(n) = d(n) - \widehat{y}_1(n)$.

2. Adaptation of the coefficients $\widehat{\mathbf{h}}_2(n)$:

$$\widehat{\mathbf{h}}_2(n+1) = \begin{cases} \widehat{\mathbf{h}}_1(n+1), & \text{if } \begin{cases} n = T, 2T, 3T, \dots \\ \text{and} \\ \prod_{i=1}^{L} Q(n-iT) = 1 \end{cases} \\ \widehat{\mathbf{h}}_2(n) + \mu_2(n)e_2(n)\mathbf{x}(n), & \text{otherwise.} \end{cases} \tag{2.92}$$

   and $e_2(n) = d(n) - \widehat{y}_2(n)$.

3. The re-initialization of the variable step:

$$\mu_2(n+1) = \begin{cases} \frac{\mu_1+\mu_2(n)}{2}, & \text{if } \begin{cases} n = T, 2T, 3T, \dots \\ \text{and} \\ \prod_{i=1}^{L} Q(n-iT) = 1; \end{cases} \\ \max\left\{\alpha\mu_2(n), \mu_3\right\}, & \text{otherwise} \end{cases} \tag{2.93}$$

where:

$$Q(m) = \begin{cases} 1, & \text{if} \sum_{i=m}^{m+T} e_1^2(i) > \sum_{i=m}^{m+T} e_2^2(i); \\ 0, & \text{otherwise.} \end{cases} \tag{2.94}$$

The filter of interest is $\widehat{\mathbf{h}}_2(n)$ and its step-size is increased when it converges slowly than $\widehat{\mathbf{h}}_1(n)$. The step-size $\mu_2(n)$ is decreased when the filter $\widehat{\mathbf{h}}_2(n)$ is near the steady-state. The coefficients of the adaptive filter $\widehat{\mathbf{h}}_2(n)$ are re-initialized as in the CP-LMS algorithm. The minimum value of the step $\mu_2(n)$ is obtained when the algorithm is at the steady-state and it is close to μ_3. The maximum value of $\mu_2(n)$ is obtained when the algorithm is far from the steady-state and can be in some cases very close to μ_1, but always will be smaller than μ_1.

The parameters μ_1 and μ_3 which are the upper and the lower bounds of $\mu_2(n)$ must be chosen to ensure the convergence. Moreover, if μ_1 is close to the value of μ_3, the number of changes in the step-size $\mu_2(n)$ is small. In order to provide a large number of modifications of $\mu_2(n)$ we must choose $\mu_3 \ll \mu_2$. As a result, the parameters μ_1 and μ_3 must be chosen to satisfy the following condition:

$$\mu_3 \ll \mu_1 < \frac{2}{3tr\left[\mathbf{R}\right]}. \tag{2.95}$$

where $\mathbf{R}$ is the input autocorrelation matrix.

The value of the parameter α in equation (2.93) must be in the interval $(0, 1)$, such that the step $\mu_2(n)$ is decreased when $\widehat{\mathbf{h}}_2(n)$ performs better than $\widehat{\mathbf{h}}_1(n)$. The parameter L, that is the number of consecutive test intervals, where the sum of the square errors are computed, is used in order to avoid missadaptations of the step-size and the coefficients. The convergence speed of the CP-VSLMS algorithm can be modified by μ_1, α and T and the steady-state level of the misadjustment is obtained selecting the value of μ_3.

Theoretical analysis

Is is easy to show, writing the Wiener-Hopf equations, that both adaptive filters converge to the same optimum solution given by the following equation:

$$\mathbf{h}_o = \mathbf{R}^{-1}\mathbf{p}. \tag{2.96}$$

where $\mathbf{R}$ is the input autocorrelation matrix and $\mathbf{p}$ is the cross-correlation vector between the desired sequence and the input.

To examine the behavior of the coefficients of the CP-VSLMS algorithm we analyze both algorithms during one test interval. Suppose, that the analysis is made on the interval from $(k-1)T$ to kT and at $(k-1)T$ the coefficients $\widehat{\mathbf{h}}_2((k-1)T)$ were re-initialized with $\widehat{\mathbf{h}}_1((k-1)T)$. During one test interval the adaptive filters have constant step-sizes μ_1 and $\mu_2((k-1)T)$. At the end of the test interval, the average of the coefficient error vectors are obtained as follows (see Section 2.1.1):

$$\begin{aligned} E\left[\Delta\widehat{\mathbf{h}}_1(kT)\right] &= [\mathbf{I}-\mu_1\mathbf{R}]\,E\left[\Delta\widehat{\mathbf{h}}_1(kT-1)\right], \\ E\left[\Delta\widehat{\mathbf{h}}_2(kT)\right] &= [\mathbf{I}-\mu_2((k-1)T)\mathbf{R}]\,E\left[\Delta\widehat{\mathbf{h}}_2(kT-1)\right]. \end{aligned} \tag{2.97}$$

where k is an integer.

Writing the eigenvalue decomposition $\mathbf{R}=\mathbf{Q\Lambda Q}^t$ of the matrix $\mathbf{R}$ equation (2.97) can be written as follows for every coefficient:

$$\begin{aligned} E\left[v_{1_i}(kT)\right] &= [1-\mu_1\lambda_i]^T\,E\left[v_{1_i}((k-1)T)\right], \quad i=\overline{1,N} \\ E\left[v_{2_i}(kT)\right] &= [1-\mu_2((k-1)T)\lambda_i]^T\,E\left[v_{2_i}((k-1)T)\right], \quad i=\overline{1,N}. \end{aligned} \tag{2.98}$$

where $\mathbf{v}_2(kT)=\mathbf{Q}\Delta\widehat{\mathbf{h}}_2(kT)$, $\mathbf{v}_1(kT)=\mathbf{Q}\Delta\widehat{\mathbf{h}}_1(kT)$, λ_i is the i^{th} eigenvalue of $\mathbf{R}$ and $v_{1_i}(n)$ and $v_{2_i}(n)$ are the i^{th} elements of $\mathbf{v}_1(n)$ and $\mathbf{v}_2(n)$ respectively.

From (2.98) it is clear, that the convergence of the coefficients of the speed mode filter to the optimum is much faster than the convergence of $\widehat{\mathbf{h}}_2(n)$ due to the fact that the step-size $\mu_2(n)$ is less than μ_1 for all time instants n from $(k-1)T$ to kT and $v_{1_i}((k-1)T)=v_{2_i}((k-1)T)$[2] for all i.

The MSE at the end of the test interval can be obtained taking into account that the step-sizes μ_1 and $\mu_2(n)$ are constant for $n=\overline{(k-1)T,kT}$, as follows:

$$\begin{aligned} J_1(kT) &= J_{min}+tr\left[\mathbf{R}\mathbf{C}_1(kT)\right] \\ J_2(kT) &= J_{min}+tr\left[\mathbf{R}\mathbf{C}_2(kT)\right] \end{aligned} \tag{2.99}$$

where the minimum MSE for both adaptive filters is the same $J_{min}=\lim_{n\to\infty}E\left[e_o^2(n)\right]=\lim_{n\to\infty}E\left[(d(n)-\mathbf{h}_o\mathbf{x}(n))^2\right]$ and the cross-correlation matrices of the coefficient error vectors are expressed as $\mathbf{C}_1(n)=E\left[\Delta\widehat{\mathbf{h}}_1(n)\Delta\widehat{\mathbf{h}}_1^t(n)\right]$ and $\mathbf{C}_2(n)=E\left[\Delta\widehat{\mathbf{h}}_2(n)\Delta\widehat{\mathbf{h}}_2^t(n)\right]$.

In [20] p. 156, the following approximate expression for the transient MSE of the LMS with fixed step-size is given.

$$J(n)\approx J_{min}+\sum_{i=1}^{N}\lambda_i\left(1-\mu\lambda_i\right)^{2n}v_i^2(0) \tag{2.100}$$

[2]This is due to the fact that $\widehat{\mathbf{h}}_2((k-1)T)=\widehat{\mathbf{h}}_1((k-1)T)$, which implies $\mathbf{v}_1((k-1)T)=\mathbf{v}_2((k-1)T)$

where λ_i is the i^{th} eigenvalue of the input autocorrelation matrix, $v_i(0)$ is the i^{th} element of the vector $\mathbf{v}(0) = \mathbf{Q}^t \left[\hat{\mathbf{h}}(0) - \mathbf{h}_o\right]$ and the columns of $\mathbf{Q}$ represent the eigenvectors of the input autocorrelation matrix.

Taking into account that during one test interval the step-sizes of both adaptive filters are constant, the following equations can be obtained from (2.100):

$$\begin{aligned} J_1(kT) &\approx J_{min} + \sum_{i=1}^{N} \lambda_i \left(1 - \mu_1 \lambda_i\right)^{2T} v_{1_i}^2 \left((k-1)T\right), \\ J_2(kT) &\approx J_{min} + \sum_{i=1}^{N} \lambda_i \left(1 - \mu_2\left((k-1)T\right) \lambda_i\right)^{2T} v_{2_i}^2 \left((k-1)T\right) \end{aligned} \tag{2.101}$$

where λ_i is the i^{th} eigenvalue of $\mathbf{R}$, $v_{1_i}\left((k-1)T\right)$ is the i^{th} element of $\mathbf{v}_1\left((k-1)T\right) = \mathbf{Q}^t \Delta\widehat{\mathbf{h}}_1\left((k-1)T\right) = \widehat{\mathbf{h}}_1\left((k-1)T\right) - \mathbf{h}_o$ and $v_{2_i}\left((k-1)T\right)$ is the i^{th} element of $\mathbf{v}_2\left((k-1)T\right) = \Delta\widehat{\mathbf{h}}_2\left((k-1)T\right) = \widehat{\mathbf{h}}_2\left((k-1)T\right) - \mathbf{h}_o$.

We assume that at the beginning of the test interval the coefficients $\widehat{\mathbf{h}}_2\left((k-1)T\right)$ are re-initialized with $\widehat{\mathbf{h}}_1\left((k-1)T\right)$, therefore we have:

$$v_{1_i}\left((k-1)T\right) = v_{2_i}\left((k-1)T\right) \tag{2.102}$$

Clearly, from (2.101) and (2.102) it follows that the MSE of the speed mode filter $\widehat{\mathbf{h}}_1(n)$ is smaller than the MSE of the adaptive filter $\widehat{\mathbf{h}}_2(n)$ at the end of the test interval[3].

The above analysis was made for a test interval (n from $(k-1)T$ to kT) with the assumption that the coefficients $\widehat{\mathbf{h}}_2(n)$ are initialized with $\widehat{\mathbf{h}}_1(n)$ at $n = (k-1)T$. If at the beginning of the test interval, the coefficients $\widehat{\mathbf{h}}_2(n)$ are not re-initialized, the same analysis can be extended for two or more consecutive intervals. Let us consider that at $n = (k-1)T$ the coefficients $\widehat{\mathbf{h}}_2(n)$ are not re-initialized, but they are re-initialized at $n = (k-2)T$. The average of the coefficients errors at time instant $n = kT$ can be expressed as follows:

$$\begin{aligned} E\left[v_{1_i}(kT)\right] &= \left[1 - \mu_1 \lambda_i\right]^{2T} E\left[v_{1_i}((k-2)T)\right], \quad i = \overline{1, N} \\ E\left[v_{2_i}(kT)\right] &= \left[1 - \mu_2((k-2)T)\lambda_i\right]^{T} \left[1 - \mu_2((k-1)T)\lambda_i\right]^{T} E\left[v_{2_i}((k-2)T)\right], \\ & \quad i = \overline{1, N}. \end{aligned} \tag{2.103}$$

[3] In Section 2.1.1, we have concluded that the steady-state MSE decreases when the step-sizes is decreased. Here we discuss the behavior of the MSE after a number of T consecutive iterations during the transient period of the adaptive filters. The conclusion is that a larger step-size will give smaller transient MSE, which is intuitively correct since larger step-size means faster convergence to the optimum solution.

Since $E\left[v_{1_i}((k-2)T)\right] = E\left[v_{2_i}((k-2)T)\right]$ and $\mu_2\left((k-2)T\right) < \mu_2\left((k-1)T\right) < \mu_1$, it follows that $E\left[v_{1_i}(kT)\right] < E\left[v_{2_i}(kT)\right]$, for every coefficients of the adaptive filters.

The transient MSE of both adaptive filters, at time instant $n = kT$ can be approximated by:

$$\begin{aligned} J_1(kT) &\approx J_{min} + \sum_{i=1}^{N} \lambda_i \left(1 - \mu_1 \lambda_i\right)^{2T} v_{1_i}\left((k-2)T\right) \\ J_2(kT) &\approx J_{min} + \sum_{i=1}^{N} \lambda_i \left(1 - \mu_2\left((k-1)T\right)\lambda_i\right)^{T} \left(1 - \mu_2\left((k-2)T\right)\lambda_i\right)^{T} v_{2_i}\left((k-2)T\right) \end{aligned} \tag{2.104}$$

As a consequence, the speed mode filter has a smaller transient MSE than the accuracy mode filter $\widehat{\mathbf{h}}_2(n)$.

From the above analytical results we can conclude that the transient MSE of the speed mode filter is smaller than the MSE of $\widehat{\mathbf{h}}_2(n)$ at the end of every test interval. This conclusion justifies the method for step-size update in (2.93) and the re-initialization of the accuracy mode filter coefficients from (2.92).

Of course, when the speed mode filter is at steady-state, its MSE can be approximated by:

$$J_{1_{st}} = J_{min}\left(1 + \frac{\mu_1}{2} tr\left[\mathbf{R}\right]\right). \tag{2.105}$$

Since the speed mode filter, converges faster than $\widehat{\mathbf{h}}_2(n)$, the step-size $\mu_2(n)$ will only be decreased after $\widehat{\mathbf{h}}_1(n)$ has converged. Finally, the accuracy mode filter converges to the following level of the MSE:

$$J_{2_{st}} = J_{min}\left(1 + \frac{\mu_3}{2} tr\left[\mathbf{R}\right]\right), \tag{2.106}$$

obtained when $\mu_2(n)$ attains its minimum bound.

2.2.3 Noise Constrained Variable Step-Size LMS algorithm

The inconvenience of the CP-VSLMS algorithm described in the previous section is its increased computational complexity, as compared with the VSSLMS and RVSLMS algorithms of [1] and [48], due to the use of two adaptive filters which operate in parallel. In this section we introduce a variable step-size LMS algorithm that exploits the information about noise in order to obtain a fast convergence and a small steady-state error. The proposed filtering structure is introduced mainly for system identification applications and it uses just one adaptive filter such that the computational complexity is highly decreased. The block diagram of an FIR adaptive filter implementing the proposed Noise Constrained Variable Step-Size LMS (NCVSLMS) algorithm for system identification is depicted in Fig. 2.2. In this figure, $\mathbf{h}$ is the unknown system modelled as an FIR filter of length N, $\widehat{\mathbf{h}}(n)$ is the adaptive FIR filter, $x(n)$ is the input sequence, $y(n)$ and $\widehat{y}(n)$ are the outputs of the unknown system and the adaptive filter respectively.

The coefficients of the adaptive filter are updated by the following equation:

$$\widehat{\mathbf{h}}(n+1) = \widehat{\mathbf{h}}(n) + \mu(n)e(n)\mathbf{x}(n). \tag{2.107}$$

where $e(n) = y(n) - \widehat{y}(n) + v(n)$ is the output error, $y(n) = \mathbf{h}^T(n)\mathbf{x}(n)$, $\widehat{y}(n) = \widehat{\mathbf{h}}^T(n)\mathbf{x}(n)$ and $v(n)$ is the output noise.

To update the variable step-size $\mu(n)$ we propose the following formula:

$$\mu(n+1) = \begin{cases} \dfrac{\mu_{max} + \mu(n)}{2}, & \text{if } \begin{cases} med_T\left[e^2(n)\right] > C, \\ \text{and} \\ n = T, 2T, \ldots \end{cases} \\ \max\left\{\alpha\mu(n), \mu_{\min}\right\} & \text{if } \begin{cases} med\left[e^2(n)\right] \leq C, \\ \text{and} \\ n = T, 2T, \ldots \end{cases} \\ \mu(n), & \text{otherwise} \end{cases} \tag{2.108}$$

where μ_{min} and μ_{max} are the minimum and the maximum bounds of the step-size and $med_T\left[e^2(n)\right]$ is the median of $\left[e^2(n), e^2(n-1), \ldots, e^2(n-T+1)\right]$.

The behavior of the step-size of the proposed algorithm can be described as follows. For a number of T consecutive iterations (the test interval of length T) the step-size $\mu(n)$ is constant and the coefficients of the adaptive filter are updated as in the case of a fixed step-size[4]. At the end of the test interval, the median of the square output error is computed and compared with a threshold C. If the median is larger than this threshold, this means the algorithm is on the transient period therefore, the step-size is increased to obtain a faster convergence. On the contrary, when the median is comparable with the threshold, it means that the algorithm is already at the steady-state and the step-size is decreased in order to obtain a smaller level of the misadjustment. The minimum value of the step-size is μ_{min} and it is attained when the adaptive filter is at the steady-state.

As we can see from the above description of the algorithm, the main idea here is to increase the step-size $\mu(n)$ during the transient period and to decrease its value at the steady-state. As a consequence, the threshold C must contain information which allows us to test when the adaptive filter is at the steady-state. To this end we propose to use (2.24) which approximates the steady-state MSE in the case of a fixed step-size and the

[4]Since $\mu(n)$ is constant during a test interval, (2.107) is equivalent with (2.12) for the standard LMS algorithm for T consecutive iterations.

threshold is given by:

$$C(n) = \sigma_v^2 \left(1 + \frac{\mu(n)}{2} N \sigma_x^2\right) \tag{2.109}$$

where σ_v^2 is the variance of the output noise and σ_x^2 is the variance of the input sequence $x(n)$.

The threshold $C(n)$ as described by (2.109) is not constant during the adaptation process but it is computed all the time the step-size is changed. This is because the steady-state MSE depends on the value of the step-size.

For the values of the step-sizes μ_{min} and $\mu_{\max}$ we can choose any value as long as the convergence is obtained. In order to increase the convergence speed, the step-size μ_{max} must be chosen close to the stability boundary $\frac{2}{3N\sigma_x^2}$. The convergence speed depends also on the length T of the test interval. If T is too large, the algorithm converges inside the test interval and the step-size is not enough times updated. The value of α must be in the interval $(0, 1)$ such that, when the algorithm goes to the steady-state, the step-size is decreased.

At the steady-state, the step-size $\mu(n)$ converges to μ_{min}, therefore the misadjustment of the algorithm can be approximated as follows:

$$\mathcal{M} = \frac{\mu_{min} N \sigma_x^2}{2}. \tag{2.110}$$

In the step-size adaptation (see (2.108) and (2.109)), an accurate estimation of σ_v^2 and σ_x^2 are necessary. In some practical applications, information about noise variance is available by modeling or measuring the noise [52], [77]. The variance of the input sequence can also be estimated. An equivalent of this algorithm in which there is no necessity to compute the input variance is described in the next chapter.

2.2.4 Simulations and results

In this section, the above mentioned algorithms are implemented in system identification framework depicted in Fig. 2.2. The unknown system has $N = 10$ coefficients and all the tested adaptive filters have equal lengths. The parameters of the algorithms were chosen to give comparable levels of the misadjustment. More than that, the selection of the parameters was done using the guidelines from the corresponding papers and they are shown in Table 2.1. For benchmark purposes, in addition to the variable step-size LMS algorithms, in our simulations we have also included two LMS algorithms with fixed step-size denoted as LMS1 and LMS2. The first algorithm LMS1 has a large step-size whereas LMS2 has a small step-size chosen to obtain the same level of the misadjustment

LMS1	$\mu = 0.05$
LMS2	$\mu = 0.002$
VSSLMS	$\mu_{max} = 0.05, \quad \mu_{min} = 0.002, \quad \alpha = 0.97, \quad \gamma = 0.057$
RVSLMS	$\mu_{max} = 0.05, \quad \mu_{min} = 0.002, \quad \alpha = 0.97, \quad \beta = 0.99, \quad \gamma = 1$
CP-LMS	$\mu_1 = 0.05, \quad \mu_2 = 0.002, \quad T = 100, \quad L = 1$
CP-VSLMS	$\mu_1 = 0.05, \quad \mu_{min} = 0.002, \quad T = 100, \quad \alpha = 0.6, \quad L = 1$
NCVSLMS	$\mu_{max} = 0.05, \quad \mu_{min} = 0.002, \quad T = 100, \quad \alpha = 0.6$

Table 2.1: The parameters of the compared algorithms.

as the VSSLMS algorithms. The output noise $v(n)$ was zero mean Gaussian-distributed with variance $\sigma_v^2 = 10^{-3}$. The input sequence was also zero mean Gaussian-distributed with unity variance. Results are obtained by averaging over 100 independent runs, each run containing 8×10^3 iterations. The learning curves for the compared algorithms are shown in Fig. 2.17 and Fig. 2.18 where we have plotted the mean square coefficient error defined as:

$$MSC(n) = \sum_{i=1}^{N} \left(\widehat{h}_i(n) - h_i \right)^2 \tag{2.111}$$

where h_i is the i^{th} coefficient of the unknown filter $\mathbf{h}$ in Fig. 2.2 and $\widehat{h}_i(n)$ is the i^{th} coefficient of the adaptive filter $\widehat{\mathbf{h}}(n)$ at time instant n.

In the case of CP-LMS and CP-VSLMS, there are two filters working in parallel, the speed mode filter $\widehat{\mathbf{h}}_1(n)$ and the accuracy mode filter $\widehat{\mathbf{h}}_2(n)$ and in both cases the filter of interest is $\widehat{\mathbf{h}}_2(n)$. Therefore, for these two filters (2.111) was evaluated using $\widehat{\mathbf{h}}_2(n)$ instead of $\widehat{\mathbf{h}}(n)$.

From the learning curves shown in Fig. 2.17 and Fig. 2.18 we can see that CP-VSLMS has the best performance among the compared algorithms. Of course, LMS1, which uses a large step-size has the faster convergence but at the expense of an increased steady-state MSC. The CP-LMS has the slowest convergence among the variable step-size LMS algorithms as we can see from Fig. 2.18. Anyway, when the step-size of the accuracy mode filter is made time varying, the speed of the new CP-VSLMS algorithm is highly improved. The performances of both proposed algorithms, the CP-VSLMS and NCVSLMS are comparable.

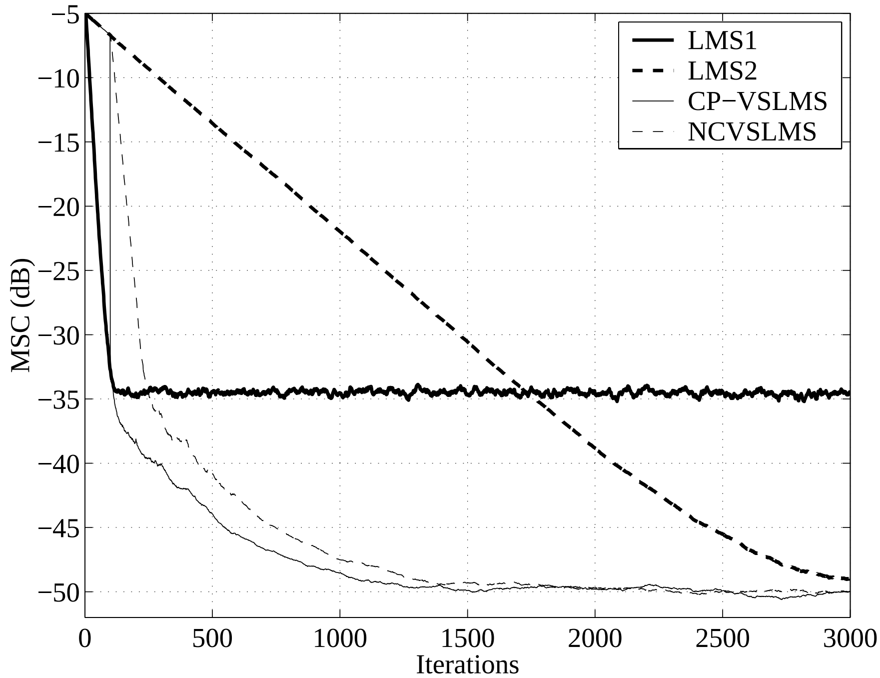

Figure 2.17: The mean square coefficient error for LMS1, LMS2, CP-VSLMS and NCVSLMS.

2.2.5 Comparison of the variable step-size LMS algorithms

In this section, we compare different Variable Step-Size LMS algorithms in terms of computational complexity, memory load and simplicity of implementation. Also the advantages and the drawbacks of the compared algorithms are discussed together with some guidelines for practical implementation.

To this end, in Table 2.2 the memory load and computational complexity of the VSSLMS, RVSLMS, CP-LMS, CP-VSLMS and NCVSLMS are shown. Among all algorithms, the CP-LMS and CP-VSLMS have the largest complexity. However the setup of their parameters is very simple (the minimum bound of the step-size μ_{min} can be obtained from (2.106) for a desired level of the misadjustment).

We note that, in the case of NCVSLMS, in the formula for step-size update, the noise variance σ_v^2 is needed. As a consequence, this algorithm is more suitable in applications where σ_v^2 can be approximated. Actually, the same discussion is valid also for VSSLMS and RVSLMS algorithms if we look at (2.84) and (2.86). These two equations are used in order to setup the value of the parameter γ for both algorithms and in both equations the value of the minimum MSE is included. As a consequence, the parameter γ computed

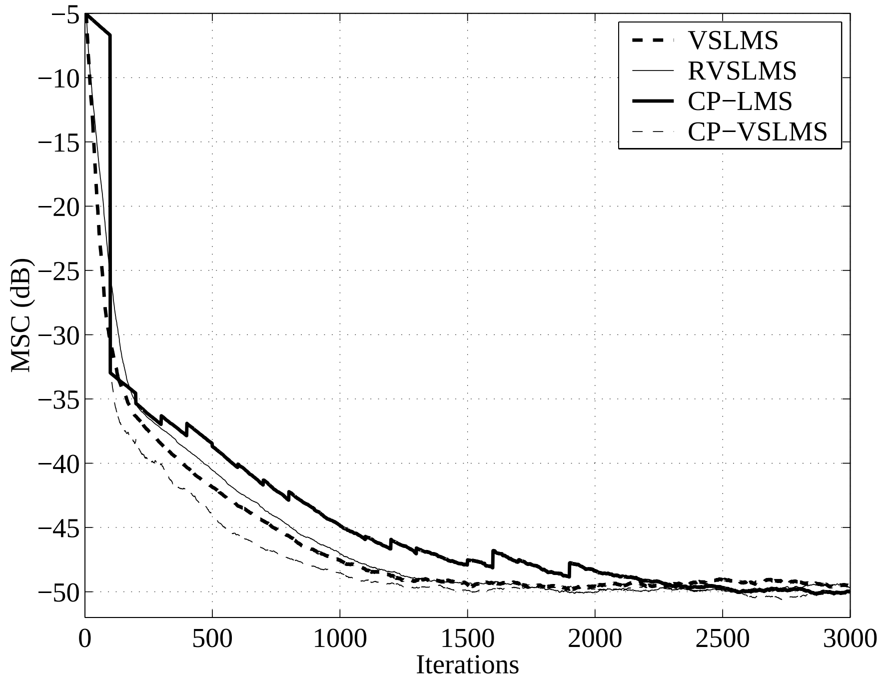

Figure 2.18: The mean square coefficient error for VSSLMS, RVSLMS, CP-LMS and CP-VSLMS.

to obtain a desired misadjustment depends upon J_{min}. If J_{min} increases, the value of γ should be decreased to maintain the same output error. This problem can be avoided in the case of VSSLMS and RVSLMS if the γ is chosen to have a very small value, such that the same misadjustment is obtained for different noise levels[5].

The advantages and disadvantages of the compared algorithms are synthesized in Table 2.3.

	VSSLMS	RVSLMS	CP-LMS	CP-VSLMS	NCVSLMS
Memory load	2N+9	2N+12	3N+11	3N+14	2N+10
Add. and sub.	2N+1	2N+3	4N+2	4N+3	2N+3
Multip. and div.	2N+4	2N+7	4N+5	4N+6	2N+6

Table 2.2: The complexity of the compared algorithms.

[5]For small output SNR and very small value of γ, the steady-state value of the step-size equals the minimum bound μ_{min}. As a consequence, the misadjustment can be computed similar to (2.104). If J_{min} increases, the misadjustment is computed with (2.84) and (2.86) respectively.

	VSSLMS	RVSLMS	CP-LMS	CP-VSLMS	NCVSLMS
Complexity	small	small	large	large	small
Speed	fast	fast	slower	faster	fast
Setup	needs J_{min}	needs J_{min}	simple	simple	needs J_{min}

Table 2.3: Advantages and disadvantages of the compared algorithms.

2.3 Variable length LMS algorithms

In some practical applications, such as, system identification, the goal is to obtain good approximations of the coefficients of an unknown system. To this end, the LMS algorithm or some modifications, as those discussed in the previous section, can be very easily applied with excellent results. However, in the existing implementations almost all authors use an adaptive filter that has length equal to the length of the unknown system. Therefore, the optimum length has to be a priori known or it is truncated to some predefined value. There are just few implementations of the variable length LMS adaptive filters [69], [80], but these implementations do not really find the optimum filter length. For instance, in [69], the authors propose a Variable Length LMS in which the length of the filter is increased as the algorithm goes to its steady-state. The same behavior of the adaptive filter length is presented in the algorithm proposed in [80], although the modification of the filter length is based on some other formulas. In the above mentioned papers, some maximum length N_{max} is imposed for the adaptive filter and this length is obtained at the steady-state. Actually, by implementing a variable length for the LMS algorithm the authors in [69] and [80] wanted to improve the speed of convergence of the adaptive algorithm. Here we refer to the length adaptation from another point of view, namely we are interested to approximate also the correct length of the adaptive system. In Section 2.1.2, the analytical MSE was obtained as a function of the adaptive filter length. Based on that analysis, an algorithm that finds the optimum coefficients together with the optimum filter length is derived here.

2.3.1 The proposed algorithm

We addres the problem of system identification where we are interested not only in finding the correct values of the filter coefficients but also to approximate its correct length. Our proposed Variable Length LMS (VLLMS) algorithm is based on the analytical results from Section 2.1.2 where we have shown that, the steady-state MSE is smaller when the length N_{ad} of the adaptive filter is close to the length of the unknown system N

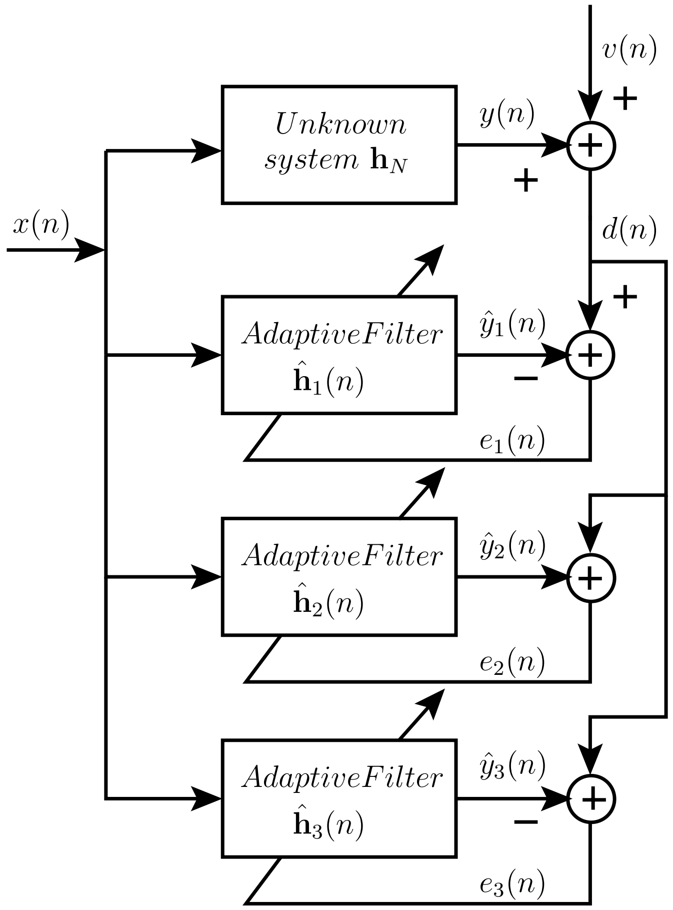

Figure 2.19: Block diagram of the VLLMS for system identification.

and $N_{ad} < N$. On the other hand, if $N_{ad} \geq N$ then the steady-state MSE does not depend on the filter length. This conclusion is valid for an uncorrelated input sequence $x(n)$ and it is generally not valid for correlated input signals. The block diagram of the proposed VLLMS algorithm is presented in Fig. 2.19 where $\widehat{\mathbf{h}}_1(n)$, $\widehat{\mathbf{h}}_2(n)$ and $\widehat{\mathbf{h}}_3(n)$ are three adaptive filters working in parallel. The lengths of these filters are N_1, N_2 and N_3 respectively with $N_1 < N_2 < N_3$. The proposed algorithm can be summarized as follows:

1. Initialization: $N_1(0) = N_0$, $N_2(0) = N_0 + 1$, $N_3(0) = N_0 + 2$, $\mu_1(0) = \dfrac{\mu N_0}{N_1(0)}$, $\mu_2(0) = \dfrac{\mu N_0}{N_2(0)}$, $\mu_3(0) = \dfrac{\mu N_0}{N_3(0)}$.

2. At every time instant n, compute the output errors $e_l(n) = d(n) - \widehat{\mathbf{h}}_l^t(n)\mathbf{x}_l(n)$,

$\forall l = 1,2,3$ and update the coefficients of the adaptive filters:

$$\widehat{\mathbf{h}}_l(n+1) = \widehat{\mathbf{h}}_l(n) + \mu_l(n)\mathbf{x}_l(n)e_l(n), l = 1,2,3.$$

where $\mathbf{x}_l(n) = [x(n), x(n-1), \ldots, x(n-N_l+1)]^t$.

3. For every L^{th} iteration (where L is an integer parameter) do:

   - compute the following averages: $m_l = \frac{1}{L}\sum_{j=n-L+1}^{n} e_l^2(j), \forall l = 1,2,3.$
   - update the lengths:

$$N_1(n+1) = \begin{cases} N_1(n)+1, & \text{if} \quad m_1 > m_2 > m_3, \\ N_1(n), & \text{if} \quad m_1 > m_2 \leq m_3, \\ N_1(n)-1, & \text{otherwise.} \end{cases} \qquad \begin{array}{l} N_2(n+1) = N_1(n+1)+1, \\ N_3(n+1) = N_1(n+1)+2. \end{array} \tag{2.112}$$

   - update the step-sizes: $\mu_l(n+1) = \frac{\mu N_0}{N_l(n+1)}, \forall l = 1,2,3.$

The parameter L in the above algorithm has to be large enough, such that the MSE can be approximated with the average of past L square errors, and also it has to be small in order to have a sufficient number of updates. Actually, in our simulations, we have obtained good results using a variable parameter $L(k)$ chosen to be a multiple of the time constant τ:

$$L(n) = \left[P\frac{N_2}{2\mu_2(n)tr\left[\mathbf{R}_{N_2}\right]}\right] = \left[\frac{P}{2\mu_2(n)\sigma_x^2}\right], \tag{2.113}$$

where $[A]$ represents the integer part of A and P is an integer parameter. Note that, the value of $L(n)$ is changed together with $\mu_2(n)$.

The specifics of the new algorithm are: the lengths $N_1(k)$, $N_2(k)$ and $N_3(k)$ are not changed at each iteration, but are constant for a number of $L(k)$ iterations and after that, based on the estimated mean square errors, the lengths are changed according to (2.112). When the lengths are decreased, the last coefficient from each of the coefficients vector is simply eliminated. When the lengths are increased, the new coefficient added to each vector $\widehat{\mathbf{h}}_1(k)$, $\widehat{\mathbf{h}}_2(k)$ and $\widehat{\mathbf{h}}_3(k)$ is initialized with zero. When all the lengths are smaller than the optimum length, then $m_1 > m_2 > m_3$ and all the lengths are increased by one as shown in (2.112). If the length N_1 is smaller than the optimum length and the others are equal or larger than the optimum length we have just $m_1 > m_2$. In this case, the lengths are left unchanged. Finally, if $m_1 \leq m_2 \leq m_3$ (ideally they are all equals), it means that all the lengths are larger than the optimum length and they are decreased. The second adaptive filter is the filter of interest in the sense that its coefficients vector will be the closest one to the optimum Wiener filter and its length is closer to the optimum.

We emphasize, that the step-sizes of all three adaptive filters satisfies the condition $\mu_1(n)N_1(n) = \mu_2(n)N_2(n) = \mu_3(n)N_3(n)$ at every time instant which ensures the same misadjustments of the algorithms. Also due to this condition, the length update can be done using (2.112).

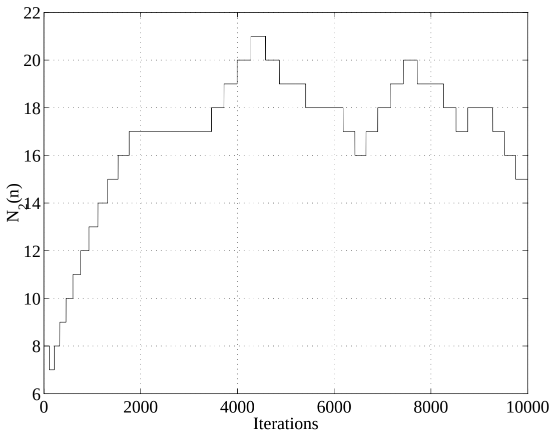

Figure 2.20: The length for the second adaptive filter ($N_2(n)$) during one run.

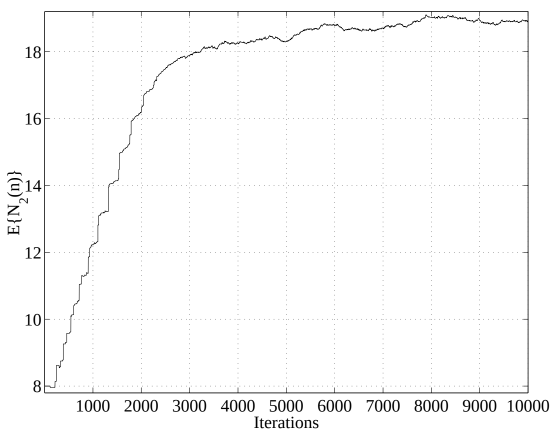

Figure 2.21: The average length for the second adaptive filter ($E\{N_2(n)\}$).

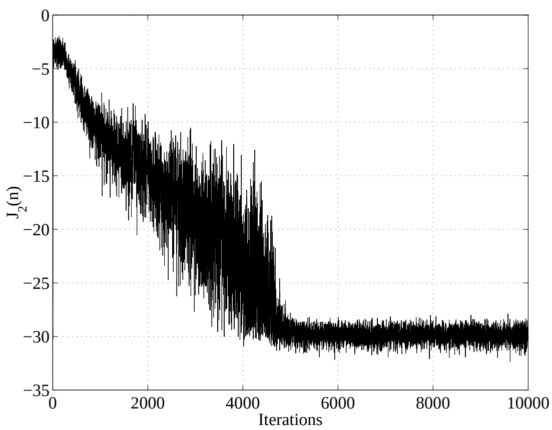

Figure 2.22: The MSE of the second adaptive filter.

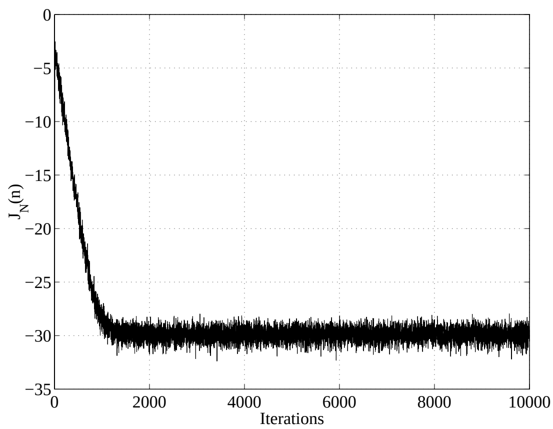

Figure 2.23: The MSE of an adaptive filter with length $N = 19$.

2.3.2 Simulations and results

The new algorithm was tested in the system identification framework. The length of the unknown system was $N = 19$. The lengths of the adaptive filters were initialized with: $N_1(0) = 7$, $N_2(0) = 8$ and $N_3(n) = 9$ respectively. The step-size μ was initialized with 10^{-2} which satisfy the condition for convergence in the mean square sense and the number of time constants in (2.113) was $P = 2$. We note that due to the fact that the step-sizes of all adaptive filters satisfy the condition $\mu_1(n)N_1(n) = \mu_2(n)N_2(n) = \mu_3(n)N_3(n)$ at each iteration, the convergence of all adaptive filters is ensured when $\mu = 10^{-2}$.

The input signal $x(n)$ was white Gaussian random with zero mean and variance $\sigma_x^2 = 1$. The output noise was random Gaussian distributed with zero mean and variance $\sigma_v^2 = 10^{-3}$.

The length behavior of the second adaptive filter during one run is depicted in Fig. 2.20 and the expected value of the $N_2(n)$ during the adaptation is depicted in Fig. 2.21. The expected value of $N_2(n)$ in Fig. 2.21, was computed by averaging the results of 100 independent runs. We can see from these figures, that the length of the second adaptive filter converges close to the length of the unknown filter.

In Fig. 2.22 and Fig. 2.23 the MSE of the second adaptive filter and the MSE of an adaptive filter with length equal to $N = 19$ are depicted. The convergence of the filter with adaptive length is slower than the convergence of the adaptive filter which has length equal to the optimum length, due to the length adaptation. At early stages of the length adaptation, the MSE is larger due to the fact that there are many coefficients of the unknown filter that does not have correspondent in the adaptive filter. When the length is close to the optimum the MSE decreases.

2.4 Step-size adaptation in time-varying environment

It is well known that, in the case of tracking a time-varying system, the steady-state MSE is a nonlinear function of the algorithm step-size. Moreover, there is an optimum step-size which minimizes the steady-state MSE, in a time-varying environment, as detailed in Section 2.1.3. There are many papers in the open literature, that study the behavior of different LMS based algorithms for tracking time-varying systems. However, to the best of our knowledge in the existing literature the computation of the optimum step-size is done making some assumptions about the parameters of the time-varying systems and the theoretical formulas (2.80) and (2.81) are used. Here, we introduce a simple adaptive algorithm which iteratively adjusts the value of the step-size toward the optimum, such that the steady-state output MSE is minimized. The proposed algorithm uses the parameterization of the nonlinear function that gives the dependence between the steady-state MSE and the step-size. During the adaptation, the parameters of this nonlinear function are computed therefore, some estimates of the optimum step-size can be easily obtained without any prior information about the system parameters. The step-size of the proposed algorithm converges near to μ_{opt}^{mse} as it can be seen from the simulations shown at the end of this section.

To introduce our algorithm, we refere to (2.79) from Section 2.1.3 and we make the following notations:

$$A = J_{min} = \sigma_v^2, \quad B = \tfrac{1}{2}\sigma_v^2 tr\left[\mathbf{R}\right], \quad C = \frac{1}{2}tr\left[\mathbf{Q}\right] \tag{2.114}$$

With these notations, (2.79) can be written as follows:

$$J_{st} = A + \mu B + \frac{C}{\mu}. \tag{2.115}$$

In (2.115) A, B and C are unknown. We only know the step-size μ and the steady-state MSE[6]. To estimate J_{st} during the adaptation process, we use the following iterative method:

$$P(n) = \alpha P(n-1) + (1-\alpha)e^2(i), \quad J(n) = \tfrac{1}{L}\sum_{i=n-L+1}^{n} P(i) \tag{2.116}$$

$\alpha \in (0,1)$ being a constant parameter and L is the number of consecutive iterations on which $J(n)$ is computed.

At this point, we have to make a very important remark: in (2.116) the MSE is estimated during the adaptation process while J_{st} in (2.115) is the MSE at the steady-state. This is why we have used the notation $J(n)$ instead of J_{st} in (2.116). We shall emphasize here that using $J(n)$ instead of the steady-state MSE in (2.115), the optimum value of the step-size computed when the algorithm is far from the steady-state ($n << \infty$) are erroneous. For this reason, we do not compute the optimum step-size just once during the adaptation process but we compute it many times. When the algorithm goes close to its steady-state, the MSE estimate from (2.116) converges close to J_{st} and the value of the estimated optimum step-size based on (2.115) is close to μ_{opt}^{mse} given by (2.80).

The parameters A, B and C do not depend on the adaptive filter but only on the statistics of the input signal, output noise and the unknown filter statistics. For three

[6]Actually we estimate the steady state J_{st} during the adaptation.

adaptive LMS filters with same length and different step-sizes μ_1, μ_2 and μ_3, at the steady-state, three different MSE's (J_{st_1}, J_{st_2} and J_{st_3}) are obtained. The nonlinear functions that give the dependence between these three steady-state MSE's and the corresponding step-sizes are expressed as follows:

$$\begin{cases} J_{st_1} = A + \mu_1 B + \dfrac{C}{\mu_1} \\ J_{st_2} = A + \mu_2 B + \dfrac{C}{\mu_2} \\ J_{st_3} = A + \mu_3 B + \dfrac{C}{\mu_3} \end{cases} \tag{2.117}$$

The system of equations in (2.117) is linear in A, B and C. Its solution can be easily obtained as follows:

$$C = \frac{(J_{12}\mu_{23} - J_{23}\mu_{12})\,\mu_1\mu_2\mu_3}{\mu_{12}\mu_{23}\mu_{13}}, \quad B = \frac{J_{23}}{\mu_{23}} + \frac{C}{\mu_2\mu_3} \tag{2.118}$$

where $J_{12} = J_{st_1} - J_{st_2}$, $J_{23} = J_{st_2} - J_{st_3}$, $\mu_{12} = \mu_1 - \mu_2$, $\mu_{13} = \mu_1 - \mu_3$ and $\mu_{23} = \mu_2 - \mu_3$.

Finally, the optimum step-size μ_{opt}^{mse} is estimated as follows:

$$\mu_{opt}^{mse} = \sqrt{\frac{C}{B}}. \tag{2.119}$$

Equations (2.117)-(2.119) are valid only when all three adaptive filters are at the steady-state. If they are used during the transient period of the adaptive filters, the estimated step-size in (2.119) is far from the optimum. Since the computation of the parameters A, B and C is done many times during the adaptation process when the three algorithms converges, also the value of the step-size computed using (2.119) converge to μ_{opt}^{mse}.

The same block diagram as the one in Fig. 2.19 is used for the proposed algorithm. The difference is that the three FIR adaptive filters have the same lengths N and different step-sizes. Their coefficients are updated as follows:

$$\begin{aligned} \hat{\mathbf{h}}_1(n+1) &= \hat{\mathbf{h}}_1(n) + \mu_1 e_1(n)\mathbf{x}(n), \\ \hat{\mathbf{h}}_2(n+1) &= \hat{\mathbf{h}}_2(n) + \mu_2(n) e_2(n)\mathbf{x}(n), \\ \hat{\mathbf{h}}_3(n+1) &= \hat{\mathbf{h}}_3(n) + \mu_3(n) e_3(n)\mathbf{x}(n) \end{aligned} \tag{2.120}$$

Note that in (2.120) two of the adaptive filters have time-varying step-sizes updated as follows:

$$\mu_2(n+1) = \begin{cases} \sqrt{\frac{C}{B}}, & \text{if} \quad n = L, 2L, \ldots \\ \mu_2(n), & \text{otherwise} \end{cases}, \quad \mu_3(n+1) = \frac{3}{4}\mu_2(n+1), \tag{2.121}$$

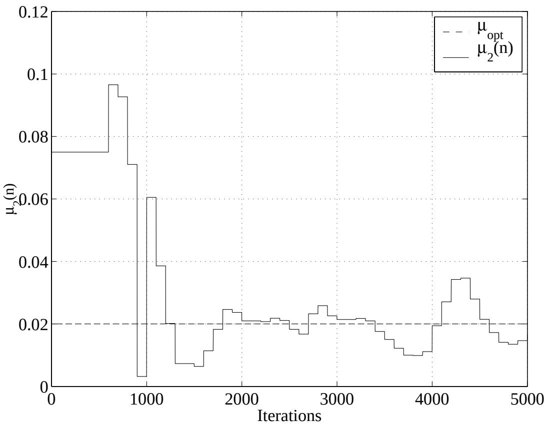

Figure 2.24: Step-size behavior for the new algorithm in the first case $\sigma^2_{\epsilon_1} = \sigma^2_{\epsilon_2} = \cdots = \sigma^2_{\epsilon_8} = 10^{-6}$

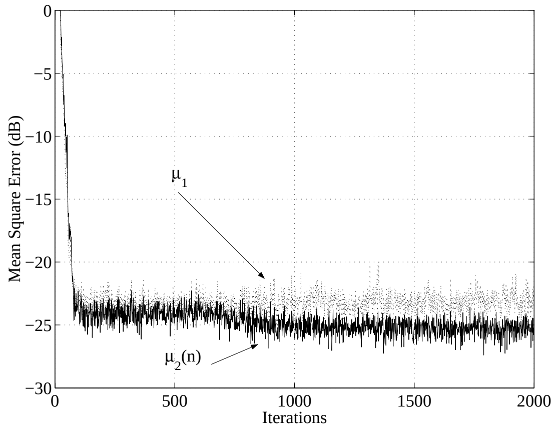

Figure 2.25: MSE of two adaptive filters with $\mu_1 = 0.1$ and $\mu_2(n)$ in the first case $\sigma^2_{\epsilon_1} = \sigma^2_{\epsilon_2} = \cdots = \sigma^2_{\epsilon_8} = 10^{-6}$

where C and B are computed as in (2.118).

The proposed algorithm can be described as follows: three adaptive LMS filters with different step-sizes are used. All the filters perform independently and they have fixed step-sizes for a number of L consecutive iterations (test interval of length L). At the end of the test interval some estimates of the MSE for each filter are computed using (2.116) and based on a linear system of equations (2.117)-(2.119) an intermediate value of the optimum step-size is estimated. The step-size $\mu_2(n)$ of the second adaptive filter is updated with the value of this estimated optimum step-size. The adaptations continue with the middle filter having the new step-size and after each test interval a new optimum step-size is computed and $\mu_2(n)$ is changed accordingly. The step-size $\mu_3(n)$ of the third adaptive filter is modified by (2.121). Choosing this formula to modify the step-size $\mu_3(n)$ we ensure that always $\mu_2(n) > \mu_3(n)$. Imposing also, that $\mu_1 > \mu_2(n)$ at each time instant, by setting μ_1 close to the stability limit, the three algorithms will have different step-sizes and the system of equations (2.117) will always have a unique solution.

2.4.1 Simulations and results

The proposed algorithm was tested in the channel estimation framework as depicted in Fig. 2.19. The model of the channel used in our simulations is given by (2.67). The output noise $v(n)$ was white Gaussian with zero mean and variance $\sigma_v^2 = 25 \times 10^{-4}$. The length of the time-varying channel and the lengths of all adaptive filters were chosen equal to $N = 8$. The input sequence $x(n)$ was white Gaussian with zero mean and variance

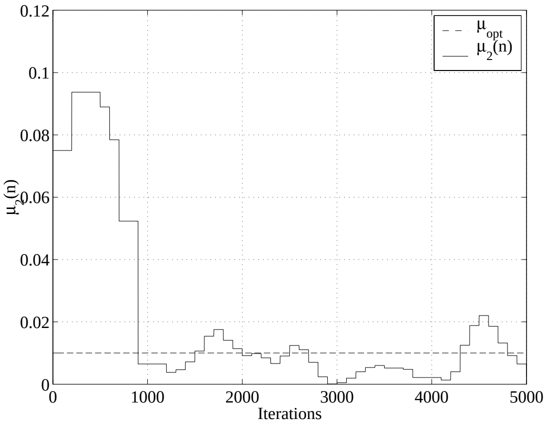

Figure 2.26: Step-size behavior for the new algorithm in the second case $\sigma^2_{\epsilon_1} = \sigma^2_{\epsilon_8} = 10^{-6}$ and $\sigma^2_{\epsilon_2} = \cdots = \sigma^2_{\epsilon_7} = 0$

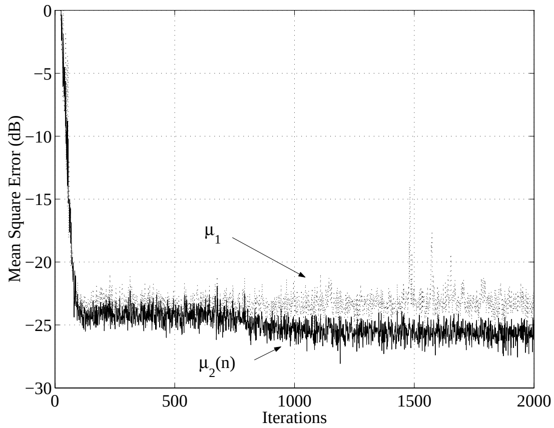

Figure 2.27: MSE of two adaptive filters with $\mu_1 = 0.1$ and with $\mu_2(n)$ in the second case $\sigma^2_{\epsilon_1} = \sigma^2_{\epsilon_8} = 10^{-6}$ and $\sigma^2_{\epsilon_2} = \cdots = \sigma^2_{\epsilon_7} = 0$

$\sigma_x^2 = 1$. For the first adaptive filter with fixed step-size we have chosen $\mu_1 = 0.1$ and for the two adaptive filters with variable step-sizes they were initialized with $\mu_3(0) = 0.05$ and $\mu_2(0) = 0.075$. The length of the test interval was chosen $L = 100$ iterations and the smoothing coefficient in (2.116) was $\alpha = 0.99$. The simulation results are presented for two different cases.

In the first case, every element of the vector $\epsilon(n)$ in (2.67) was chosen from a random zero mean sequence with variance $\sigma_\epsilon^2 = 10^{-6}$. In this case, $\mathbf{Q} = \sigma_\epsilon^2 \mathbf{I}$, with $\mathbf{I}$ being the identity matrix. The direct computation of the optimum step-size using (2.80) gives:

$$\mu_{opt}^{mse} = \sqrt{\frac{tr\,(\mathbf{Q})}{\sigma_v^2 tr\,[\mathbf{R}]}} = 0.02$$

where $tr\,[\mathbf{Q}] = N\sigma_\epsilon^2 = 8 \times 10^{-6}$, $tr\,[\mathbf{R}] = N\sigma_x^2 = 8$ and $\sigma_v^2 = 25 \times 10^{-4}$.

The behavior of the step-size $\mu_2(n)$ during the adaptation is depicted in Fig. 2.24. The MSE's of the adaptive filter with fixed step-size μ_1 and of the adaptive filter with variable step-size $\mu_2(n)$ are depicted in Fig. 2.25. As we can see from Fig. 2.24, during the transient period of the algorithms, the value of the step-size $\mu_2(n)$ is far from the optimum. As the adaptive filters $\hat{\mathbf{h}}_1(n)$, $\hat{\mathbf{h}}_2(n)$ and $\hat{\mathbf{h}}_3(n)$ go to their steady-state the value of $\mu_2(n)$ converges to approximately $\mu_{opt} = 0.02$. From Fig. 2.25 we can conclude that an adaptive step-size gives better performances in terms of lower steady-state MSE.

In the second case, just the first and the last coefficients of the channel were time-varying ($\sigma^2_{\epsilon_1} = \sigma^2_{\epsilon_8} = 10^{-6}$) and the rest of the coefficients were left unchanged ($\sigma^2_{\epsilon_2} =$

$\cdots = \sigma_{\epsilon_7}^2 = 0)$. The optimum step-size obtained by direct application of (2.77) was:

$$\mu_{opt} = \sqrt{\frac{tr\,(\mathbf{Q})}{\sigma_v^2 tr\,[\mathbf{R}]}} = 0.01$$

where $tr\,[\mathbf{Q}] = 2\sigma_\epsilon^2 = 2 \times 10^{-6}$.

The behavior of the step-size $\mu_2(n)$ is depicted in Fig. 2.26 and the MSE's of the adaptive filter with step-size μ_1 and of the adaptive filter with variable step-size $\mu_2(n)$ are shown in Fig. 2.27. From Fig. 2.26 and 2.27 we can see that the step-size $\mu_2(n)$ converges to the optimum μ_{opt}. The adaptive filter with variable step-size $\mu_2(n)$ gives lower steady-state MSE than the adaptive filter with fixed step-size μ_1.

2.5 Order Statistics Least Mean Squared algorithms

The LMS suffers serious performance degradation and may fail when the input and/or the desired signals are corrupted by an impulsive noise. To overcome this difficulty, the class of Order Statistic LMS (OSLMS) adaptive filters was introduced (see [23], [32], [33], [39] and the references therein). In the case of OSLMS algorithms the coefficients of the adaptive filter are updated as in the following formula:

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu O\left\{\mathbf{g}(n)\right\}\mathbf{a}, \qquad (2.122)$$

where $\hat{\mathbf{h}}(n)$ is the vector of the adaptive filter coefficients, μ is the step-size, $\mathbf{a} = [a_1 ... a_L]^T$ is a vector of weighting coefficients for smoothing the gradient, $O\left\{\mathbf{g}(n)\right\}$ is the ordering operation applied to each row of the matrix $\mathbf{g}(n)$, which is given by:

$$\mathbf{g}(n) = \begin{bmatrix} e(n)x(n) & \ldots & e(n-L+1)x(n-L+1) \\ e(n)x(n-1) & \ldots & e(n-L-1)x(n-L+2) \\ & \ldots & \\ e(n)x(n-N+1) & \ldots & e(n-L-1)x(n-N-L+2) \end{bmatrix}, \qquad (2.123)$$

$x(n)$ and $e(n)$ are the input sequence and the output error, respectively. The class of OSLMS filters includes the following adaptive filters as particular cases:

- the Average LMS (ALMS) with $\mathbf{a} = \frac{1}{L}\left[1, 1, ..., 1\right]^T$;
- the Median LMS (MLMS) with $\mathbf{a} = \left[0, \ldots, 1, \ldots, 0\right]^T$;
- the Trimmed Mean LMS (MxLMS) with $\mathbf{a} = \left[0, \ldots, \frac{1}{L-2M}, ..., \frac{1}{L-2M}, \ldots, 0\right]^T$, where L is the length of the weighting vector and M is the number of samples eliminated from the left and right side of the ordered input sequence;

- the Outer Mean LMS (OxLMS) with $\mathbf{a} = [1/2, 0, \ldots 0, 1/2]^T$).

As shown in [32], [33], [39], the OSLMS algorithms can reduce the variance of the gradient estimate if the weighting coefficients are chosen properly. This leads to a reduction of the steady-state excess MSE. It was already established in [33] that for impulsive environments the MLMS and MxLMS algorithms have better behavior comparing to OxLMS and ALMS, whereas for Gaussian and uniform noise environments the optimum choices are the ALMS and OxLMS algorithms, respectively. However, selection of each of these algorithms has to be based on a priori knowledge of the noise distribution or, more generally, on the knowledge of the gradient distribution. Without this knowledge, an arbitrarily chosen filter may have poor performance. In this paper we propose a new AOSLMS algorithm that uses adaptive weighting coefficients $\mathbf{a}(n)$ for smoothing the gradient. A novelty of the new algorithm is the fact that no prior information about the gradient distribution is necessary. Some approaches that use the adaptation of the weighting coefficients $\mathbf{a}(n)$ based on some statistic measurements of the gradient have been reported in the literature (see, e.g., [32]), but they are limited to the modification of the trimming coefficient, such that the OS filter is modified between mean and median.

2.5.1 The proposed algorithm

The update equation (2.122) of the OSLMS filter is modified as:

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu O\left\{\mathbf{g}(n)\right\}\mathbf{a}(n); \tag{2.124}$$

where the notations are those from (2.122).

Note that in (2.124) the values of the weighting coefficients $\mathbf{a}(n)$ are not constants during the adaptation, but are adapted to the gradient distribution. In order to adapt the coefficients $\mathbf{a}(n)$ we have implemented an L-LMS filter (see, e.g., [23] and [64]). It was already proved that these filters possess the ability to adapt their coefficients to the distribution of the input sequence. Due to the topology of the matrix $\mathbf{g}(n)$, the distribution of its rows will be similar therefore, the adaptation of $\mathbf{a}(n)$ can be done using samples of the gradient contained in the first row of $\mathbf{g}(n)$. The block diagram of the new AOSLMS algorithm for system identification is presented in Fig. 2.28 and the block diagram of the L-LMS filter for the gradient is depicted in Fig. 2.29.

The new algorithm consists of the following steps.

- Compute the output $\hat{y}(n)$ and the error $e(n)$ of the AOSLMS filter (see Fig. 2.28):

$$\hat{y}(n) = \hat{\mathbf{h}}(n)^T\mathbf{x}(n), \quad e(n) = y(n) + v(n) - \hat{y}(n).$$

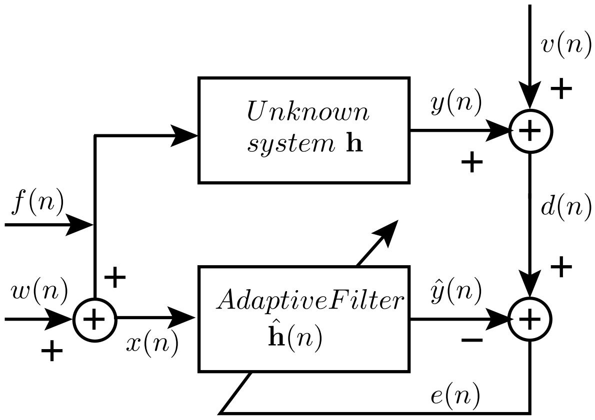

Figure 2.28: The block diagram of the OSLMS for system identification when the noise appears at the input and output of the filters.

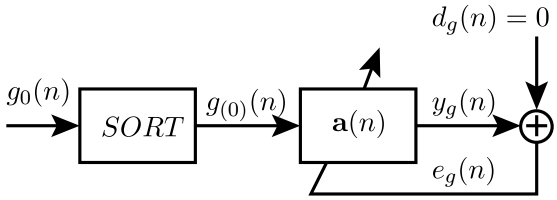

Figure 2.29: The block diagram of the L-LMS filter used for adaptation of the weighting coefficients $\mathbf{a}(n)$

- Update the weighting coefficients $\mathbf{a}(n)$ (see Fig. 2.29):

$$\mathbf{a}(n+1) = \mathbf{a}(n) + \alpha \mathbf{g}_{(0)}(n) e_g(n). \tag{2.125}$$

where $\mathbf{g}_{(0)}(n)$ is the ordered version of the first row of $\mathbf{g}(n)$, and $e_g(n)$ is the error of the L-LMS filter applied to the gradient.

- Update the coefficients of the OSLMS filter:

$$\hat{\mathbf{h}}(n+1) = \hat{\mathbf{h}}(n) + \mu O\left\{\mathbf{g}(n)\right\} \mathbf{a}(n). \tag{2.126}$$

In (2.125), we have used the error $e_g(n)$ for updating the weighting coefficients. Since in this case there is no desired signal available for filtering the gradient ($d_g(n) = 0$ in Fig. 2.29), we have chosen the constrained L-LMS described in [79], and (2.125) becomes:

$$\mathbf{a}(n+1) = \mathbf{P}\left[\mathbf{a}(n) + \alpha \mathbf{g}_{(0)}^T(n)\left(-y_g(n)\right)\right] + \mathbf{F}. \tag{2.127}$$

where $y_g(n) = \mathbf{g}_{(0)}(n)\mathbf{a}(n)$ is the output from the L-LMS filter, and the matrix $\mathbf{P}$ and the vector $\mathbf{F}$ are respectively given by (see [79] and the references therein):

$$\mathbf{P} = \mathbf{I} - \mathbf{C}\left(\mathbf{C}^T\mathbf{C}\right)^{-1}\mathbf{C}^T; \qquad \mathbf{F} = \mathbf{C}\left(\mathbf{C}^T\mathbf{C}\right)^{-1}\mathfrak{F} \tag{2.128}$$

$$\mathbf{C} = \begin{cases} \begin{bmatrix} & \mathbf{I}_{\frac{L-1}{2}} \\ \mathbf{1} & 0\ldots0 \\ & -\mathbf{Q}_{\frac{L-1}{2}} \end{bmatrix}, & \text{L odd} \\ \\ \begin{bmatrix} & \mathbf{I}_{\frac{L}{2}} \\ \mathbf{1} & 0\ldots0 \\ & -\mathbf{Q}_{\frac{L}{2}} \end{bmatrix}, & \text{L even} \end{cases} \qquad \mathfrak{F} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \tag{2.129}$$

$\mathbf{I}$ and $\mathbf{Q}$ are the identity and the opposite identity matrices, respectively.

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ddots & \ldots \\ 0 & 0 & \ldots & 1 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 0 & \ldots & 0 & 1 \\ 0 & \ldots & 1 & 0 \\ \ldots & \ddots & \ldots & \ldots \\ 1 & \ldots & 0 & 0 \end{bmatrix}.$$

In the case of the new algorithm, there are basically two adaptive algorithms: one is used for adaptation of the coefficients $\mathbf{a}(n)$, and the second one is the AOSLMS algorithm. Therefore, there are two step-sizes that have to be chosen for the convergence of the algorithm. The most sensible step-size is that of the L-LMS filter employed for the gradient. If this coefficient is not appropriately chosen and the L-LMS filter diverges, then all the algorithm diverges. The main problem is to find a robust condition for this step-size, that ensures stability of the algorithm for a wide range of gradient distributions. To this aim, we have employed a normalized L-LMS and the value of α in (2.127) is replaced with:

$$\alpha = \frac{\tilde{\mu}}{\gamma + \left\|\mathbf{g}_{(0)}(n)\right\|^2} \tag{2.130}$$

Finally, the new algorithm is described by formulas (2.125)-(2.130).

Steady-state study

The equation for updating the coefficients of the L-LMS filter is given by (2.127). Denoting $\mathbf{z}(n) = E\left\{\mathbf{a}(n) - \mathbf{a}_o\right\}$ ($\mathbf{a}_o$ are the optimal coefficients of the L-LMS filter), and using the development of Frost in [31], one obtains:

$$\mathbf{z}(n+1) = \left[\mathbf{I} - \alpha\mathbf{P}\mathbf{R}_{(g_0)}\mathbf{P}\right]\mathbf{z}(n) = \left[\mathbf{I} - \alpha\mathbf{P}\mathbf{R}_{(g_0)}\mathbf{P}\right]^{n+1}\mathbf{z}(0) \tag{2.131}$$

where $\mathbf{R}_{(g_0)}$ is the autocorrelation matrix of the ordered first row of matrix $\mathbf{g}(n)$ from (2.123). The matrix $\mathbf{P}\mathbf{R}_{(g_0)}\mathbf{P}$ determines both the speed of convergence and also the steady-state variance of the weighting coefficients $\mathbf{a}(n)$. If $0 < \alpha < 1/\lambda_{max}$ (λ_{max} is the maximum eigenvalue of matrix $\mathbf{P}\mathbf{R}_{(g_0)}\mathbf{P}$), then the convergence in the mean of the weighting vector $\mathbf{a}(n)$ is ensured. A more restrictive condition for the step-size, that ensures also convergence of the average MSE was given in [31]:

$$0 < \alpha < \frac{1}{\sigma_{max} + (1/2)tr(\mathbf{P}\mathbf{R}_{(g_0)}\mathbf{P})}, \tag{2.132}$$

where $tr[\mathbf{A}]$ represents the trace of the matrix $\mathbf{A}$.

If α satisfies

$$0 < \alpha < \frac{2}{3tr\left(\mathbf{R}_{(g_0)}\right)}, \tag{2.133}$$

then it is guaranteed to satisfy also (2.132) (see [41] and the referred papers). The matrix $\mathbf{R}_{(g_0)}$ is different for various gradient distributions and a value of α chosen based on (2.133) for a certain gradient distribution may not be suitable for another gradient distribution. Since the convergence of the algorithm has to be ensured for any distribution without an a priori knowledge, the step-size in the proposed algorithm is chosen as $\alpha = \dfrac{\tilde{\mu}}{\gamma + \|\mathbf{g}_0(n)\|^2}$ thus, (2.133) becomes:

$$0 < \tilde{\mu} < \frac{2}{3}. \tag{2.134}$$

The asymptotic convergence point of (2.126), which we denote as $\hat{\mathbf{h}}_o$, is the point where $E\{O\{\mathbf{g}(k)\}\mathbf{a}(k)\} = 0$ when $\hat{\mathbf{h}}(k) = \hat{\mathbf{h}}_o$. In the case of absence of input noise ($w(n) = 0$, $f(n) = x(n)$), for i^{th} coefficient of $\hat{\mathbf{h}}(n)$ we would have:

$$E\{O\{\mathbf{g}_i(k)\}\mathbf{a}(k)\} = E\{O\{[g_{i,0}(k)\ldots g_{i,L-1}(k)]\}\mathbf{a}(k)\} \tag{2.135}$$

where $g_{i,j}(k) = e(k-j)f(k-i-j)$, $i = \overline{0, N-1}$, $j = \overline{0, L-1}$. The value of $g_{i,j}(k)$ can be written as $g_{i,j}(k) = \left[\left[\mathbf{h}^T - \hat{\mathbf{h}}^T(k-j)\right]\mathbf{f}(k-j) + v(k-j)\right] f(k-i-j)$.

If $\hat{\mathbf{h}}(k) = \mathbf{h}$ for a large value of k, then (2.135) becomes:

$$E\{O\{\mathbf{g}_i(k)\}\mathbf{a}(k)\} = E\{O\{v(k)f(k-i)\ldots v(k-L)f(k-i-L)\}\mathbf{a}(k)\}$$

Since $f(k)$ and $v(k)$ are both zero mean, i.i.d., with symmetric distributions, and independent on each other, and if we assume that the weighting coefficients $\mathbf{a}(k)$ are constants for large k, then $E\{O\{\mathbf{g}_i(k)\}\mathbf{a}(k)\} = 0$, $\forall i = \overline{0, N-1}$. This result means that the coefficients $\hat{\mathbf{h}}_o = \mathbf{h}$ represent the asymptotic convergence point of (2.126) for the case with only output noise.

For the case with only input noise (v(n)=0), the value of $g_{i,j}(k)$ in (2.135) is given by:

$$g_{i,j}(k) = \left[\mathbf{h}^T\mathbf{f}(k-j) - \hat{\mathbf{h}}^T(k-j)\mathbf{x}(k-j)\right] f(k-i-j)$$

	Multip.	Add.
OSLMS **a** fixed	$N \cdot L$	$N \cdot (L-1)$
AOSLMS **a** adaptive	$L \cdot (N+L)$ $+2(L+1)$	$N \cdot (L-1)$ $+L^2+2L-1$

Table 2.4: Extra computations for OSLMS filters.

Making the same assumptions, it can be shown that the coefficients $\hat{\mathbf{h}}(n)$ converge within a small ball around $\mathbf{h}$, (see [32], [33], [39]).

Thus we can conclude that the new algorithm converges within a small ball around $\mathbf{h}$, for input and also for the output noise cases provided that the L-LMS filter is convergent too.

Computational complexity analysis

The computational complexity of the new algorithm is increased comparing to the other OSLMS algorithms. For comparison purposes, in Table 2.4 we present the extra computations needed for the standard OSLMS and the new AOSLMS algorithms. In addition with these computations, each algorithm needs N sorting operations performed on L samples, where N is the length of the OSLMS filter and L is the length of the weighting vector.

2.5.2 Simulation results

The simulations were done in the system identification framework depicted in Fig. 2.28. The new AOSLMS algorithm was compared with the following OSLMS algorithms: MLMS, MxLMS, OxLMS and ALMS. The length of the filters was $N = 11$, the length of the weighting vector was $L = 7$. The step-sizes of the compared algorithms was chosen in such a way, that they would have the same convergence speed. The step-size $\tilde{\mu}$ has a fixed value chosen to satisfy (2.134). The input signal $f(n)$ was Gaussian random sequence with zero mean and unity variance. The noise, either $w(n)$ or $v(n)$ has a generalized exponential density of the form:

$$p(r) = k_1 e^{-k_2|r|^\beta}, \qquad |r| < \infty, \qquad \beta > 0$$
$$k_1 = \left(\beta k_2^{1/\beta} / 2\Gamma\left(\frac{1}{\beta}\right)\right), \quad k_2 = \left[\Gamma\left(\tfrac{3}{\beta}\right) / \Gamma\left(\tfrac{1}{\beta}\right)\right]^{\beta/2} \tag{2.136}$$

where Γ is the gamma function.

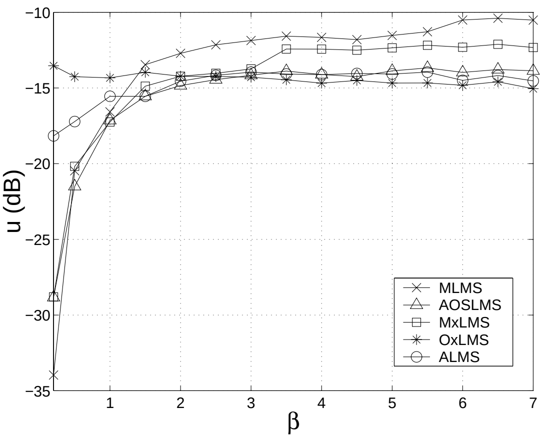

Figure 2.30: Steady-state sum of squared coefficient errors for output noise case $w(n) = 0$.

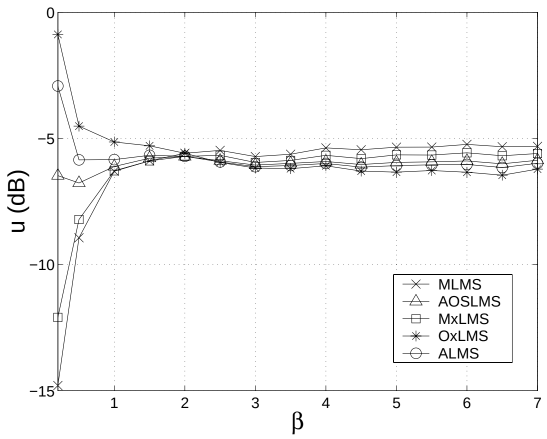

Figure 2.31: Steady-state sum of squared coefficient errors for input noise case $v(n) = 0$.

As β in (2.136) increases, the resulting noise density varies from highly impulsive ($\beta = 0.2$) to Gaussian ($\beta = 2$) and near uniform ($\beta = 7$). The algorithms were compared using the steady-state sum of squared coefficient errors:

$$u = 10 \cdot \log \left\{ \sum_{i=1}^{N} \left[h_i - \hat{h}_i(n) \right]^2 \right\}$$

For each algorithm, a number of 100 independent runs were performed and the results were averaged. The corresponding learning curves are given in Fig. 2.30 for the case of output noise ($w(n) = 0$) and in Fig. 2.31 for the input noise case ($v(n) = 0$). We can see from these figures that for impulsive environments ($\beta < 2$) the new algorithm has a performance similar to MxLMS and MLMS and for ($\beta > 2$) the new AOSLMS algorithm performs similar to OxLMS and ALMS.

Chapter 3

Transform domain implementations

This chapter considers the transform domain implementations of two classes of adaptive algorithms: the Transform Domain Variable Step-Size LMS (TDVSLMS) and the Transform Domain LMS algorithm for time-varying environments. The algorithms developed here represent the transform domain counterparts of the algorithms from Chapter 2 and they are introduced to improve the behavior of their time domain counterparts. For instance, the convergence of the time domain VSSLMS algorithms is still slow for correlated inputs and it was found that the TDLMS can improve the convergence using the decorrelation of the input sequence. More than that, if the same approach of time-varying step-size is used in the TDLMS, its speed of convergence is increased even further as we will see in the sequel. The time domain adaptation of the step-size toward the optimum, for time-varying environments, requires, in the proposed algorithm, three adaptive filters working in parallel as shown in Section 2.4. Doing the step-size adaptation in the transform domain the required number of adaptive filters reduces to two and also the convergence speed is increased when the input sequence is highly correlated. As a result, the transform domain implementation may be an alternative to the time domain adaptation of the step-size.

In Section 3.1, the plain Transform Domain Least Mean Squared (TDLMS) algorithm [20], [25] is reviewed and its theoretical analysis is briefly described. However, the theoretical analysis of the TDLMS is not presented in great details since the same derivations as those in Chapter 2, can be applied and also there are many papers in the open literature which study the analytical behavior of the TDLMS (see [4], [5], [20], [25], [30], [45], [51], [58], [61], [73] and the references therein). In the analysis of the TDLMS, we follow two main directions and the differences between the time domain and transform domain LMS algorithms are emphasized. First, the transient and steady-state behavior in terms of the MSE for stationary environments is discussed and then the analysis of the TDLMS algorithm for tracking a time-varying system with fixed length is presented. The analytical expressions of the optimum step-sizes which minimize the steady-state MSE and the steady-state mean squared coefficient error (MSC) are obtained next. At the end of this section, some simulations which demonstrate the validity of the analytical results are presented.

In Section 3.2, the class of Transform Domain Variable Step-size TDVSLMS algorithms is addressed. First, a variable step-size LMS in transform domain proposed by Kim in [45] is shortly presented and then, based on the analytical results outlined in the first section, three new TDVSLMS algorithms are introduced. However, the DCT-LMS algorithm is slightly different from our approach since it does not uses the output error to update the step-size as in our algorithms. Simulations results, showing the behavior of these new algorithms for the problem of system identification when the input sequence is highly correlated, are given in the sequel. In the simulations, we compare the behavior of the proposed algorithms with the plain TDLMS, DCT-LMS from [45] and the time domain VSSLMS algorithms described in Section 2.2 such that, the simulations shown here, complete the results from Section 2.2.

Section 3.3 is dedicated to the problem of tracking time-varying systems. From the theoretical results shown in the first section, we directly derive an adaptive algorithm in which the step-size is time-varying and converges near the optimum. The difference between the algorithm described here and the algorithm proposed in Section 2.4 relies in the fact that the transform domain implementation uses just two adaptive filters. The new algorithm is implemented in the time-varying system identification framework and the simulation results are shown at the end of this section.

At the end of the chapter, a very short presentation of the Scrambled LMS (SCLMS) algorithm is presented. Although, the SCLMS does not use an orthogonal transform at the input, we have chosen to include it here for two reasons. The first reason is that the SCLMS transforms the input sequence by means of a scrambling device and for correlated inputs this operation acts a whitening process. As a consequence, both SCLMS and TDLMS perform a decorrelation of the input prior to the adaptation of the coefficients. The second reason to include here the SCLMS is its increased interest for practical applications. In many applications a secure transmission of the data is necessary, which can be realized by scrambling the transmitted data. When adaptation must be done for scrambled data, the resulting algorithm is the SCLMS.

For these two reasons, in the last section of this chapter, a comparison between the TDLMS and SCLMS for the problem of local echo cancellation is presented. The specific application addressed is the digital data transmission over a telephone line and the comparison is made in terms of convergence speed, steady-state mean coefficient error and steady-state MSE.

3.1 The transform domain least mean squared algorithm

In the previous chapter, we have discussed about the time domain Least Mean Squared algorithm and some of its variants. From the analysis shown in Section 2.1, we have seen that the MSE of the LMS can be approximated with a sum of exponentials whose time constants are inversely proportional with the eigenvalues of the input autocorrelation matrix $\mathbf{R}$. As a consequence, if one of the eigenvalues of $\mathbf{R}$ is very small, the convergence of the adaptive filter in this direction will be slow. We can conclude that the convergence speed of the time domain LMS depends on the eigenvalue spread[1] of the input autocorrelation matrix. Some algorithms which try to improve the convergence of the LMS such as the VSSLMS can be implemented. However, the VSSLMS algorithms do not modify the input sequence and its autocorrelation matrix, therefore they are expected to have slow convergence for correlated inputs. One solution to this problem is to perform a decorrelation of the input sequence using an orthogonal transform and the resulting algorithm is called the Transform Domain LMS.

In this chapter, we consider the transform domain adaptive filter whose block diagram is depicted in Fig. 3.1, where $\mathbf{T}$ represents the orthogonal transformation applied to the input vector $\mathbf{x}(n) = [x(n), \ldots, x(n-N+1)]^t$, $\widehat{\mathbf{h}}(n) = \left[\widehat{h}_1(n), \ldots, \widehat{h}_N(n)\right]^t$ is the $N \times 1$ vector of the adaptive filter coefficients, $\mathbf{s}(n) = [s_1(n), \ldots, s_N(n)]^t$, $\widehat{y}(n)$, $e(n)$ and $d(n)$ are the transform coefficients, the output of the adaptive filter, the output error and the desired sequence respectively. Other transform domain structures such as subband adaptive filters [20] also exist in the literature but they are not the subject of this thesis.

With reference to Fig. 3.1, the transform coefficients are computed as follows:

$$\mathbf{s}(n) = \mathbf{T}\mathbf{x}(n), \tag{3.1}$$

and the output of the adaptive filter is obtained from:

$$\widehat{y}(n) = \widehat{\mathbf{h}}^t(n)\mathbf{s}(n) = \mathbf{s}^t(n)\widehat{\mathbf{h}}(n). \tag{3.2}$$

[1]The eigenvalue spread is defined as the ratio between the largest and the smallest eigenvalues of $\mathbf{R}$.

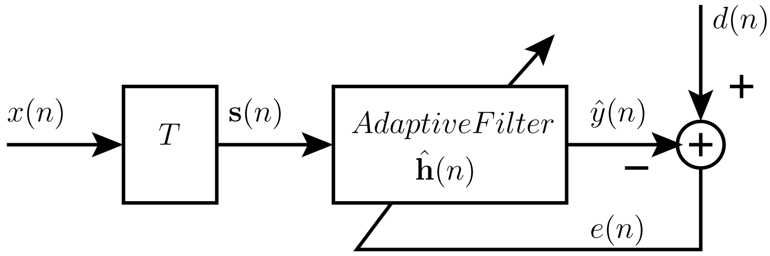

Figure 3.1: The block diagram of the transform domain adaptive FIR filter.

with $\mathbf{T}$ being the $N \times N$ orthogonal matrix with real valued elements satisfying the following relation:

$$\mathbf{T}^t\mathbf{T} = \mathbf{T}\mathbf{T}^t = \mathbf{I}. \tag{3.3}$$

where t represents the transposition operator and $\mathbf{I}$ is the $N \times N$ identity matrix.

Other transforms[2] with complex valued elements can equally be used for the orthogonalization of the input sequence $x(n)$. In this thesis the Discrete Cosine Transform is used because it gives real valued coefficients. However, the orthogonal transformation used in the implementations must be chosen based on the characteristics of the input signal $x(n)$. A very good discusion on the selection of the best transform can be find in [21] and [45]. Since the above referenced papers provide a well documented discusion, here we do not address the problem of transform selection.

The cost function used to optimize the coefficients of the transform domain adaptive filter is the MSE defined as:

$$J(n) = E\left[e^2(n)\right] \quad \text{where} \quad e(n) = d(n) - \widehat{y}(n). \tag{3.4}$$

From (3.4), the following Wiener-Hopf equation can be obtained for transform domain which gives the coefficients of the optimum filter (see Section 2.1 for more details):

$$\mathbf{h}_{T_o} = \mathbf{R}_s^{-1}\mathbf{p}_s. \tag{3.5}$$

where $\mathbf{R}_s = E\left[\mathbf{s}(n)\mathbf{s}^t(n)\right] = \mathbf{TRT}^t$ is the autocorrelation matrix of the transformed coefficients $\mathbf{s}(n)$, $\mathbf{R}$ is the autocorrelation matrix of $\mathbf{x}(n)$, $\mathbf{p}_s = \mathbf{Tp}$ is the cross-correlation vector between $\mathbf{s}(n)$ and $d(n)$ and $\mathbf{p}(n)$ is the cross-correlation vector between $\mathbf{x}(n)$ and $d(n)$.

It follows from (3.5), after some simple mathematical manipulations, that the optimum coefficients vector is given by:

$$\mathbf{h}_{T_o} = \mathbf{T}\mathbf{h}_o. \tag{3.6}$$

[2]Called unitary transforms which satisfy $\mathbf{T}^H\mathbf{T} = \mathbf{T}\mathbf{T}^H = \mathbf{I}$ and H is the hermitian transposition.

In (3.6), by $\mathbf{h}_{T_o}$ we denoted the optimum coefficients in transform domain, whereas $\mathbf{h}_o$ represents the Wiener solution in time domain expressed by (2.9). We can conclude that the optimum solution in the mean squared sense in transform domain is obtained simply by applying the orthogonal transformation to the optimum coefficients in time domain.

The input autocorrelation matrix which governs the convergence of the transform domain adaptive filter is given by $\mathbf{R}_s = \mathbf{T}\mathbf{R}\mathbf{T}^t$. In the ideal case, when the elements of $\mathbf{s}(n)$ are uncorrelated the matrix $\mathbf{R}_s$ is diagonal. It follows that, when the transformation matrix $\mathbf{T}$ is applied to the input sequence the input autocorrelation matrix is diagonalized. However, the diagonal elements of $\mathbf{R}_s$ are not equal and its eigenvalue spread is equal to the eigenvalue spread of $\mathbf{R}$. A solution to this problem is to normalize $\mathbf{R}_s$ with its diagonal elements $diag\left(\mathbf{R}_s\right)$. Specifically, this normalization is applied only in the update formula of the adaptive filter coefficients, which for the i^{th} filter coefficient can be written as follows:

$$\widehat{h}_i(n+1) = \widehat{h}_i(n) + \frac{\mu}{\epsilon + \sigma_{s_i}(n)^2} s_i(n)e(n). \tag{3.7}$$

where $\sigma^2_{s_i}(n)$ is the power estimate of the i^{th} transform coefficient $s_i(n)$ and ϵ is a small constant which eliminates the numerical instability when $\sigma^2_{s_i}(n)$ is close to zero.

The powers of the transform coefficients $s_i(n)$ can be estimated by the following simple formula[3]:

$$\sigma^2_{s_i}(n) = \alpha\sigma^2_{s_i}(n-1) + (1-\alpha)\, s_i^2(n), \quad \forall i = \overline{1, N} \tag{3.8}$$

which it is proven to converge close to the diagonal elements of $\mathbf{R}_s$.

From (3.7) and (3.8) we can conclude that each coefficient of the adaptive filter is updated by a different step-size $\mu_i(n) = \dfrac{\mu}{\epsilon + \sigma^2_{s_i}(n)}$ which is time-varying due to the normalization term. For stationary inputs, it can be shown that, (3.8) converges fast to the real powers of the transform coefficients, therefore to simplify the analysis, one can consider that the estimates $\sigma^2_{s_i}$ are constant. However, in some cases where other power estimators are used, the step-sizes are time-varying due to the normalization term. Such an example is the DCT-LMS algorithm suggested in [45], where other formula is used instead of (3.8).

In all our implementations, we have used (3.8) to estimate the powers of the transform coefficients, therefore we can assume that, after few iterations, the estimates in (3.8) are constant and close to the diagonal terms of $\mathbf{R}_s$. As a consequence, in the analytical derivations we use $\sigma^2_{s_i}$ instead of $\sigma^2_{s_i}(n)$.

Finally the TDLMS algorithm is described by the following six steps:

[3] Other approaches are also available in the open literature [45].

Transform Domain Least Mean Squared algorithm:

At each iteration n do:

1. Form the input vector $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^t$ from the input sequence $x(n)$;
2. Compute the coefficients $\mathbf{s}(n)$: $\mathbf{s}(n) = \mathbf{T}\mathbf{x}(n)$;
3. Compute the output of the adaptive filter: $\widehat{y}(n) = \mathbf{s}^t(n)\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}^t(n)\mathbf{s}(n)$;
4. Compute the output error: $e(n) = d(n) - \widehat{y}(n)$;
5. Update the power estimates of the coefficients $\mathbf{s}(n)$ using (3.8);
6. Update every coefficient of the adaptive filter: $\widehat{h}(n+1) = \widehat{h}(n) + \frac{\mu}{\epsilon+\sigma_{s_i}^2(n)} s_i(n)e(n)$;

3.1.1 Analysis of the TDLMS for stationary environments

We point out the important formulas for the transient and steady-state MSE and mean coefficient error for the TDLMS, when operating in a stationary environment. These analytical results will be used in subsequent sections to introduced the new class of variable step-size algorithms in transform domain. Since most of the equations derived for the time domain case in Section 2.1 can be applied here with small modifications, we will not give the entire derivations but we will emphasize the differences between the time domain approach and its transform domain counterpart. To make the analytical analysis mathematically tractable, similar assumptions as in time domain are made (see Section 2.1.1).

We start with the coefficient error vector defined as:

$$\Delta\mathbf{h}(n) = \widehat{\mathbf{h}}(n) - \mathbf{h}_{T_o}, \tag{3.9}$$

where $\mathbf{h}_{T_o}$ is the optimum solution defined by (3.6).

The update equation (3.7) can be written in a more compact form as follows:

$$\widehat{\mathbf{h}}(n+1) = \widehat{\mathbf{h}}(n) + \mu\mathbf{\Gamma}^{-1}\mathbf{s}(n)e(n) \tag{3.10}$$

where $\mathbf{\Gamma}$ is a diagonal matrix composed by the terms $\epsilon + \sigma_{s_i}^2(n)$.

Subtracting $\mathbf{h}_{T_o}$ from (3.10) and taking the expectation operator, after some simple mathematical manipulations we obtain:

$$E\left[\Delta\mathbf{h}(n+1)\right] = \left(\mathbf{I} - \mu E\left[\mathbf{\Gamma}^{-1}\right]\mathbf{R}_s\right) E\left[\Delta\mathbf{h}(n)\right], \tag{3.11}$$

where we have used the assumption that $\widehat{\mathbf{h}}(n+1)$ is independent on $\mathbf{s}(n)$ and the optimum error defined as $e_o(n) = d(n) - \mathbf{h}_{T_o}\mathbf{s}(n)$ is orthogonal to the input vector $\mathbf{s}(n)$[4].

Assuming that the diagonal elements of $\mathbf{\Gamma}$ are constant and equal to the powers of the elements of $\mathbf{s}(n)$, equation (3.11) can be simplified to:

$$\Delta\mathbf{h}(n+1) = (1-\mu)^n \, \Delta\mathbf{h}(0). \tag{3.12}$$

From (3.12), the condition for the convergence of the coefficients in the mean is:

$$0 < \mu < 2. \tag{3.13}$$

Comparing (3.13) with its counterpart from time domain (equation (2.19) from Section 2.1.1) we can see that in the case of TDLMS the condition for the convergence of the adaptive filter coefficients does not depend in the eigenvalues of the input autocorrelation matrix while in time domain the condition depend on λ_{max}. Intuitively, we can arrive at the same conclusion if we take into account the fact that the autocorrelation matrix is diagonalized by the orthogonal transform and the power normalization makes the autocorrelation matrix equal to identity[5]. As a result, the eigenvalues of the autocorrelation matrix are all equal to unity.

To obtain an analytical expression for the cross-correlation matrix of the coefficient error vector we subtract $\mathbf{h}_{T_o}$ from (3.10). Pre-multiplying the result by its transpose and taking the statistical expectation we arrive at the following expression:

$$\begin{aligned}\mathbf{C}(n+1) = \mathbf{C}(n) - \mathbf{C}(n)\mathbf{R}_s\mathbf{\Gamma}^{-1}\mu - \mu\mathbf{\Gamma}^{-1}\mathbf{R}_s\mathbf{C}(n) + 2\mu^2\mathbf{\Gamma}^{-1}\mathbf{R}_s\mathbf{C}(n)\mathbf{R}_s\mathbf{\Gamma}^{-1} +\\ \mu^2\mathbf{\Gamma}^{-1}\mathbf{R}_s\mathbf{\Gamma}^{-1}tr\left[\mathbf{R}_s\mathbf{C}(n)\right] + J_{min}\mu^2\mathbf{\Gamma}^{-1}\mathbf{R}_s\mathbf{\Gamma}^{-1}\end{aligned} \tag{3.14}$$

where $\mathbf{C}(n) = E\left\{\Delta\mathbf{h}(n)\Delta\mathbf{h}^t(n)\right\}$.

Making the assumption that $\mathbf{\Gamma}^{-1}\mathbf{R}_s \approx \mathbf{I}$, equation (3.14) can be simplified as follows:

$$\mathbf{C}(n+1) = \mathbf{C}(n) - \mu\mathbf{C}(n) - \mu\mathbf{C}(n) + 2\mu^2\mathbf{C}(n) + \mu^2\mathbf{\Gamma}^{-1}tr\left[\mathbf{R}_s\mathbf{C}(n)\right] + J_{min}\mu^2\mathbf{\Gamma}^{-1} \tag{3.15}$$

Following a development similar to the one in time domain, and starting from (3.4), the following expression can be obtained to describe the MSE:

$$J(n) = J_{min} + tr\left[\mathbf{R}_s\mathbf{C}(n)\right] \tag{3.16}$$

[4]The orthogonality of the optimum error to the input vector $\mathbf{s}(n)$ can be proven in a similar manner as in time domain from the fact that the gradient, when $\widehat{\mathbf{h}}(n)$ approaches $\mathbf{h}_{T_o}$, equal zero.

[5]Actually it is close to identity. The only transform which makes the autocorrelation matrix equal to the identity matrix is the KLT. However, the DCT was shown to be a good approximation of the KLT for many practical signals.

where $J_{min} = E\left\{d(n) - \mathbf{h}_{T_o}^t \mathbf{s}(n)\right\}$ is the minimum mean squared error obtained in the case of perfect adaptation and the term $tr\left[\mathbf{R}_s \mathbf{C}(n)\right]$ appears due to the imperfect adaptation of the coefficients.

The second term in (3.16) can be obtained by pre-multiplication of (3.15) with $\mathbf{R}_s$ and taking the trace of the result. At the steady-state we have $\mathbf{C}(n+1) \approx \mathbf{C}(n)$ and the MSE can be expressed as follows:

$$J_{st} = J_{min}\left(1 + \frac{\mu N}{2 - (N+2)\,\mu}\right). \tag{3.17}$$

which, for small values of the step-size μ, is simplified to:

$$J_{st} = J_{min}\left(1 + \frac{\mu N}{2}\right). \tag{3.18}$$

Intuitively (3.18) and (2.24) are equivalent, since in the TDLMS the normalization by the powers of the transform coefficients is used, therefore the matrix which gives the misadjustment of the algorithm is approximated by the identity matrix.

The condition for the convergence of the MSE can be obtained from (3.17) forcing the misadjustment to be bounded. The value of the step-size for which the misadjustment tends to infinity is $\mu_{max} = \dfrac{2}{N+2}$, therefore the step-size should satisfy the following stability condition:

$$0 < \mu < \frac{2}{N+2}. \tag{3.19}$$

3.1.2 Optimum step-size for time-varying environments

In this section, the behavior of the transform domain LMS algorithm for the problem of a time-varying system identification depicted in Fig. 3.2 is addressed. To this end, we derive the analytical equations which describe the steady-state MSE and the steady-state mean squared coefficient error in a similar manner as in Chapter 2. Due to the fact that many derivations are similar with those in the previous chapter, here we emphasize the differences between time domain and transform domain implementations. The same setup is followed as in time domain, where the unknown system is modeled as follows:

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \epsilon(n), \tag{3.20}$$

with $\epsilon(n)$ being the vector of the increments of the unknown system, at time instant n.

We consider the case when the unknown filter $\mathbf{h}(n)$ and the adaptive filter $\widehat{\mathbf{h}}(n)$ in Fig. 3.2 have the same length N. The notations in Fig. 3.2 are the following: $x(n)$ is

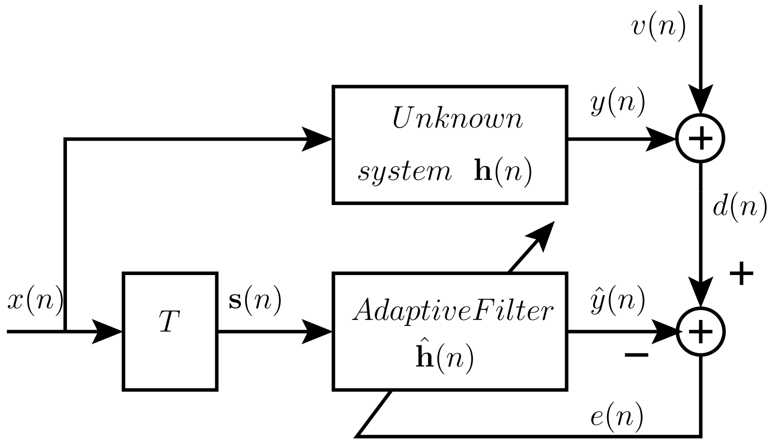

Figure 3.2: The block diagram of the transform domain adaptive FIR filter implemented for time-varying system identification.

the input sequence, the block denoted by T represents the transform layer which transforms the input vector $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^t$ into $\mathbf{s}(n) = \mathbf{T}\mathbf{x}(n) = [s_1(n), s_2(n), \ldots, s_N(n)]^t$, $v(n)$ is the output noise, $\widehat{y}(n)$, $y(n)$ and $e(n)$ are the output of the adaptive filter, the output of $\mathbf{h}(n)$ and the output error respectively.

To make the theoretical analysis more tractable, we make the following assumptions which are commonly used in the open literature [20], [21], [28], [41]:

1. *The sequences $\mathbf{x}(n)$, $v(n)$ and $\epsilon(n)$ are statistically independent of one another;*

2. *The sequences $\mathbf{x}(n)$ and $v(n)$ are zero mean, stationary, jointly normal and with finite moments;*

3. *The successive increments of the channel tap weights $\epsilon(n)$ are independent of one another. However, the elements of $\epsilon(n)$ for a given n, may be statistically dependent. The sequence $\epsilon(n)$ is zero mean and stationary with a constant covariance matrix $\mathbf{Q} = E\left\{\epsilon(n)\epsilon^T(n)\right\}$;*

4. *At time n, the vector $\hat{\mathbf{h}}(n)$ is statistically independent of $v(n)$ and $\mathbf{X}(n)$. This assumption is true when μ is small* [20], [41].

The Wiener-Hopf equations in transform domain can be written in a similar manner as in time domain by taking the derivative of the gradient with respect to the adaptive filters coefficients, equal to zero. Taking into account that the input of the adaptive filter is now $\mathbf{s}(n) = \mathbf{T}\mathbf{x}(n)$, the desired sequence is given by $d(n) = \mathbf{h}^t(n)\mathbf{x}(n) + v(n)$ and taking into account the above assumptions, the optimum Wiener solution is given by:

$$\mathbf{h}_{T_o}(n) = \mathbf{T}\mathbf{h}(n). \tag{3.21}$$

It follows from (3.21), that the optimum coefficients vector at time instant n equal the transformed coefficients vector of the unknown filter at the same time instant n.

Subtracting $\mathbf{h}_{T_o}(n) = \mathbf{T}\mathbf{h}(n+1)$ from both sides of (3.10) one obtains:

$$\Delta\widehat{\mathbf{h}}(n+1) = \widehat{\mathbf{h}}(n+1) - \mathbf{T}\mathbf{h}(n+1) = \widehat{\mathbf{h}}(n) - \mathbf{T}\mathbf{h}(n+1) + \mu e(n)\mathbf{\Gamma}^{-1}\mathbf{s}(n), \tag{3.22}$$

Using $e(n) = \mathbf{h}^t(n)\mathbf{x}(n) - \widehat{\mathbf{h}}^t(n)\mathbf{s}(n) + v(n)$, equation (3.22) can be written as follows:

$$\Delta\widehat{\mathbf{h}}(n+1) = \widehat{\mathbf{h}}(n) - \mathbf{T}\mathbf{h}(n+1) + \mu\mathbf{\Gamma}^{-1}\mathbf{s}(n)\left(\mathbf{x}^t(n)\mathbf{h}(n) - \mathbf{s}^t(n)\widehat{\mathbf{h}}(n)\right) + \mu\mathbf{\Gamma}^{-1}\mathbf{s}(n)v(n), \tag{3.23}$$

Making the notation $\epsilon_T(n) = \mathbf{T}\mathbf{h}(n+1) - \mathbf{T}\mathbf{h}(n) = \mathbf{T}\epsilon(n)$ and using the above assumptions and the fact that $\mathbf{s}^t(n) = (\mathbf{T}\mathbf{x}(n))^t = \mathbf{x}^t(n)\mathbf{T}^t$, the above equation can be written as follows:

$$\Delta\widehat{\mathbf{h}}(n+1) = \Delta\widehat{\mathbf{h}}(n) - \epsilon_T(n) - \mu\mathbf{\Gamma}^{-1}\mathbf{s}(n)\mathbf{s}^t(n)\Delta\widehat{\mathbf{h}}(n) + \mu\mathbf{\Gamma}^{-1}\mathbf{s}(n)v(n), \tag{3.24}$$

The mean coefficient error vector can be obtained taking the expectation operator on both sides of (3.24), and we obtain:

$$E\left\{\Delta\widehat{\mathbf{h}}(n+1)\right\} = E\left\{\Delta\widehat{\mathbf{h}}(n)\right\} - E\left\{\epsilon_T(n)\right\} - \mu E\left\{\mathbf{\Gamma}^{-1}\mathbf{s}(n)\mathbf{s}^t(n)\Delta\widehat{\mathbf{h}}(n)\right\} + \mu E\left\{\mathbf{\Gamma}^{-1}\mathbf{s}(n)v(n)\right\}, \tag{3.25}$$

Using the Assumptions 2, 3 and 4, one obtains[6]:

$$E\left\{\Delta\widehat{\mathbf{h}}(n+1)\right\} = \left(\mathbf{I} - \mu\mathbf{\Gamma}^{-1}\mathbf{R}_s\right)^{n+1} E\left\{\Delta\widehat{\mathbf{h}}(0)\right\} \tag{3.26}$$

where $\mathbf{I}$ is the $N \times N$ identity matrix, $\Delta\widehat{\mathbf{h}}(0) = \widehat{\mathbf{h}}(0) - \mathbf{h}_{T_o}$ is the initial coefficient error vector and $\mathbf{R}_s = E\left\{\mathbf{s}(n)\mathbf{s}^t(n)\right\}$ is the autocorrelation matrix of the transform coefficients.

Equation (3.26) can be simplified further if we take into account that the matrix $\mathbf{R}_s$ is close to diagonal and has on the main diagonal the powers of the transform coefficients such that the following approximation can be made:

$$\mathbf{R}_s\mathbf{\Gamma}^{-1} = \mathbf{\Gamma}^{-1}\mathbf{R}_s \approx \mathbf{I} \tag{3.27}$$

Using (3.27) in (3.26) one obtains:

$$E\left\{\Delta\widehat{\mathbf{h}}(n+1)\right\} = (1-\mu)^{n+1} E\left\{\Delta\widehat{\mathbf{h}}(0)\right\} \tag{3.28}$$

The convergence of the coefficients in the mean is ensured if:

$$-1 < 1 - \mu < 1, \quad \Rightarrow \quad 0 < \mu < 2, \tag{3.29}$$

[6] Assumption 3 also implies that $\epsilon_T(n)$ is zero mean and stationary.

We should emphasize here that (3.28) proves the convergence of the adaptive filter coefficients to $\mathbf{Th}(n)$ in view of the Assumption 3 which stipulates that the sequences ϵ and ϵ_T are zero mean. If the increments of the unknown filter are not zero mean, the coefficients of the adaptive filter will converge to a biased solution due to the non-zero term $E\left\{\epsilon_T(n)\right\}$ which does not vanish in (3.25).

Our main goal here is to derive the analytical equations for the steady-state MSE in the case of time-varying environment, which will be used further to introduce a new algorithm. To this end, we first derive the weight-error correlation matrix defined as $\mathbf{C}(n) = E\left\{\Delta\widehat{\mathbf{h}}(n)\Delta\widehat{\mathbf{h}}^t(n)\right\}$. Pre-multiplying (3.24) by its transpose and following a procedure similar to the one in [21], we obtain:

$$\mathbf{C}(n+1) = \mathbf{C}(n) - \mu\mathbf{\Gamma}^{-1}\mathbf{R}_s\mathbf{C}(n) + \mathbf{Q}_s - \mathbf{C}(n)\mathbf{R}_s\mathbf{\Gamma}^{-1}\mu + \mu\mathbf{\Gamma}^{-1}\mathbf{R}_s\mathbf{\Gamma}^{-1}\mu tr\left[\mathbf{R}_s\mathbf{C}(n)\right] + \\ +2\mu\mathbf{\Gamma}^{-1}\mathbf{R}_s\mathbf{C}(n)\mathbf{R}_s\mathbf{\Gamma}^{-1}\mu + \sigma_v^2\mu\mathbf{\Gamma}^{-1}\mathbf{R}_s\mathbf{\Gamma}^{-1}\mu \quad (3.30)$$

where $\mathbf{Q}_s = E\left\{\epsilon_s(n)\epsilon_s^t(n)\right\}$.

Using the approximation from (3.27) in (3.30) one obtains:

$$\mathbf{C}(n+1) = \mathbf{C}(n) - 2\mu\mathbf{C}(n) + 2\mu^2\mathbf{C}(n) + \mu^2\mathbf{\Gamma}^{-1}tr\left[\mathbf{R}_s\mathbf{C}(n)\right] + \sigma_v^2\mu^2\mathbf{\Gamma}^{-1} + \mathbf{Q}_s, \quad (3.31)$$

At the steady-state, for small values of the step-size μ, the third and the fourth terms in (3.31), can be neglected[7]. Also for $n \to \infty$, we have $\mathbf{C}(n+1) \approx \mathbf{C}(n)$ and the steady-state value of the mean squared coefficient error, defined as $\Theta(n+1) = tr\left[\mathbf{C}(n+1)\right]$ can be written in the following manner:

$$\Theta_{st} = \frac{1}{2}\left[\mu\sigma_v^2 tr\left[\mathbf{\Gamma}^{-1}\right] + \mu^{-1}tr\left[\mathbf{Q}_s\right]\right]. \quad (3.32)$$

The output MSE defined by $J(n) = E\left\{e^2(n)\right\}$, is obtained, after some mathematical manipulations as follows:

$$J(n) = J_{min} + tr\left[\mathbf{R}_s\mathbf{C}(n)\right] = \sigma_v^2 + tr\left[\mathbf{R}_s\mathbf{C}(n)\right] \quad (3.33)$$

where $J_{min} = E\left[d(n) - \mathbf{h}_{T_o}^t(n)\mathbf{s}(n)\right]$ is the minimum mean squared error.

Multiplying (3.30) by $\mathbf{R}_s$ and taking the trace, the analytical expression of the steady-state MSE can be obtained:

$$J_{st} = \sigma_v^2 + \frac{1}{2-\mu(N+2)}\left[\mu\sigma_v^2 N + \mu^{-1}tr\left[\mathbf{R}_s\mathbf{Q}_T\right]\right] \quad (3.34)$$

[7] This was justified in [22].

For small values of the step-size a simplified analytical result can be obtained which is more useful for practical implementations:

$$J_{st} = \sigma_v^2 + \frac{1}{2}\left[\mu\sigma_v^2 N + \mu^{-1} tr\left[\mathbf{R}_s\mathbf{Q}_s\right]\right] \tag{3.35}$$

From (3.32) and (3.35), we can see that Θ_{st} and J_{st} are nonlinear functions on the step-size μ. Both analytical results show that the steady-state values Θ_{st} and J_{st} contain two components. The first component is proportional with the step-size and it is due to the noisy estimates of the adaptive filters coefficients. The second component that is inversely proportional with μ appears due to the time variations in the unknown filter coefficients. Due to this fact both Θ_{st} and J_{st} possess a minimum for a certain value of the step-size. The optimum step-size μ_o^{Θ} which minimizes Θ_{st} and the step-size μ_o^{mse} which minimizes J_{st} are not equal in general and they can be expressed by the following equations:

$$\mu_o^{mse} = \sqrt{\frac{tr\left[\mathbf{R}_s\mathbf{Q}_s\right]}{\sigma_v^2 N}} \quad \text{and} \quad \mu_o^{\Theta} = \sqrt{\frac{tr\left[\mathbf{Q}_s\right]}{\sigma_v^2 tr\left[\mathbf{\Gamma}^{-1}\right]}} \tag{3.36}$$

Although (3.32) and (3.35) were obtained making some assumptions and approximations, we found that they provide good practical models for the behavior of Θ_{st} and J_{st}. We should emphasize here that these results were obtained for the case when the desired sequence is obtained as the output of a time-varying FIR filter and the input sequence $x(n)$ is stationary[8].

The reason to introduce this theoretical analysis was to have the basis for the introduction of a new algorithm with adaptive step-size for time-varying environments. We are interested to develop an algorithm in which the step-size is updated toward μ_o^{mse}, such that the steady-state MSE J_{st} is minimized. To this end, we analyze (3.35) and we make the notation $A = tr\left[\mathbf{R}_s\mathbf{Q}_s\right]$. With this notation, equation (3.35) can be written as follows:

$$J_{st} = \sigma_v^2 + \frac{1}{2}\sigma_v^2 N\mu + \frac{1}{2\mu}A. \tag{3.37}$$

In the above equation there are just two unknown quantities σ_v^2 and A as opposed with the time domain counterpart (2.79) where the unknowns are σ_v^2, $tr\left[\mathbf{R}\right]$ and $tr\left[\mathbf{Q}\right]$. As a consequence, in order to derive an algorithm for step-size adaptation which does not need the knowledge of the statistics of $\mathbf{h}(n)$, only these two values must be estimated during the adaptation. For the transform domain, a system of two equations must be

[8]When the input $x(n)$ is non-stationary the input autocorrelation matrix $\mathbf{R}$ and its transform domain counterpart $\mathbf{R}_s$ are both non-stationary. Analysis of this case is left beyond the scope of this thesis.

solved in order to compute μ_o^{mse}, therefore just two adaptive filters working in parallel are necessary[9].

The reduction of the number of adaptive filters was the main reason to implement the adaptive step-size in the transform domain. However, the reduction in computational complexity is shadowed by the fact that the input vector $\mathbf{x}(n)$ must be transformed to $\mathbf{s}(n)$ by means of an orthogonal transformation $\mathbf{T}$, which introduces some extra computations. The second reason of equal interest, to introduce the adaptation of the step-size in transform domain, was to increase the convergence speed for highly correlated input sequences. In practice the user must choose one of the two alternatives which is more suitable for the application at hand. If the time domain implementation provides enough convergence speed (i.e. for small filter lengths), then it might be a good choice. Otherwise, the transform domain is the alternative which ensures an increased speed of convergence.

3.1.3 Simulations and results

In this section, we show the simulations results conducted with the aim to verify the analytical results from the previous section (we verify the validity of (3.32), (3.35) and (3.36)). To this end, we have implemented an adaptive FIR filter in a time-varying environment as depicted in Fig. 3.2. The model of the time-varying coefficients of the unknown system is expressed by (3.20), where the elements of the vector $\epsilon(n)$ are chosen to be random independent zero mean Gaussian-distributed sequences. The variances of the elements of $\epsilon(n)$ were all equal to $\sigma_c^2 = 10^{-6}$. The lengths of $\mathbf{h}(n)$ and $\widehat{\mathbf{h}}(n)$ were equal to $N = 4$. The output noise $v(n)$ was a random Gaussian-distributed sequence with zero mean and variance $\sigma_v^2 = 25 \times 10^{-4}$. The algorithm used to update the coefficients of the adaptive filter was the transform domain LMS and the orthogonal transformation $\mathbf{T}$ was the Discrete Cosine Transform. The coefficients of the adaptive filter were updated at each iteration using equation (3.10) and the elements of the diagonal matrix $\mathbf{\Gamma}^{-1}$ were iteratively estimated by (3.8). The coefficient α in (3.8) was chosen to be equal to 0.9 which shows a good trade-off between the accuracy of estimation and the convergence of the diagonal elements of $\mathbf{\Gamma}$. The model used to generate the input sequence was the Markov(1) model:

$$x(n+1) = \beta x(n) + \eta(n). \tag{3.38}$$

where $\beta = 0.75$ and $\eta(n)$ was a random zero mean Gaussian-distributed sequence with the variance chosen, such that the variance of $x(n)$ was unity.

[9]In time domain due to the fact that J_{st} in (2.79) was parametrized with three parameters $A = \sigma_v^2$, $B = \frac{\sigma_v^2 tr\,[\mathbf{R}]}{2}$ and $C = \frac{tr\,[\mathbf{Q}]}{2}$, three adaptive filters were needed.

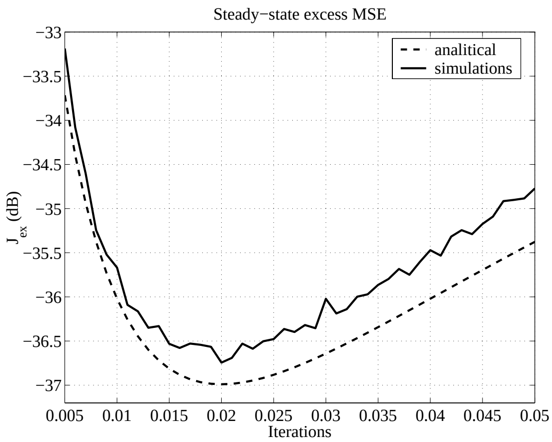

Figure 3.3: Steady-state excess MSE vs. the step-size for a time-varying system identification in transform domain: experimental results (continuous line) and analytical results (dashed line)

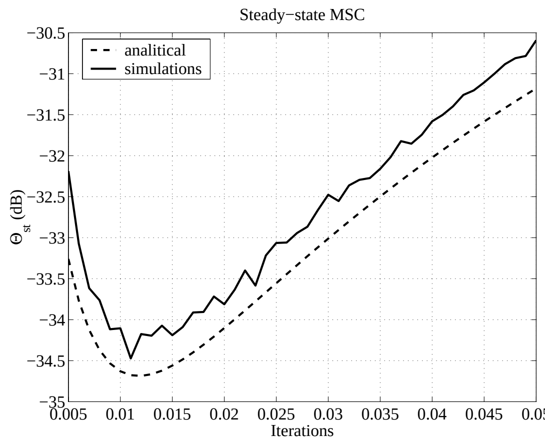

Figure 3.4: Steady-state mean squared coefficient error vs. the step-size for a time-varying system identification in transform domain: experimental results (continuous line) and analytical results (dashed line)

With this system setup, the optimum step-sizes which minimizes the steady-state MSE and the steady-state mean squared coefficient error were found to be:

$$\mu_0^{mse} = 0.02 \quad \text{and} \quad \mu_o^{\Theta} = 0.0118. \tag{3.39}$$

where we have used (3.36) to compute μ_o^{mse} and μ_o^{Θ}.

To obtain experimentally the dependence between J_{st} and the step-size and also the dependence between Θ_{st} and μ, we have conducted a set of different simulations. During one simulation, the step-size of the TDLMS was constant. However, the step-sizes used in different simulations were not equal. We start with $\mu = 5 \times 10^{-3}$ for the first simulation and continue until $\mu = 0.05$ for the last simulation. The increment of the step-size between two experiments was 10^{-3}. All the experiments contained a number of 100 independent runs of 5×10^3 iterations and the results were averaged. In order to have a more clear representation, instead of plotting the MSE, we have chosen to plot the excess MSE as a function of the step-size. Therefore, in every experiment we have computed the steady-state excess MSE and the steady-state mean squared coefficient error and they are plotted versus the corresponding step-size in Fig. 3.3 and Fig. 3.4 respectively. In Fig. 3.4 and Fig. 3.3 we have also plotted the value of Θ_{st} obtained by estimation of (3.32) and the steady-state excess MSE obtained from the evaluation of the following analytical expression:

$$J_{st} = \frac{1}{2}\left\{\mu\sigma_v^2 N + \mu^{-1} tr\left[\mathbf{R}_s\mathbf{Q}_s\right]\right\} \tag{3.40}$$

From Fig. 3.3 and Fig. 3.4 we can see that there are small differences between the experimental and analytical results. These differences are due to the fact that some assumptions and approximations were done in the derivation of (3.32) and (3.35). For instance, to obtain (3.32) we have assumed that the third and fourth terms in (3.31) can be neglected. Moreover, the term $2-\mu(N+2)$ was approximated with 2 in order to simplify both (3.32) and (3.35). Anyway, these differences between the theory and practice are small and the important issue is that in the experimental results and analytical results the optimum step-size is preserved (for both steady-state MSE and Θ_{st} the analytical and experimental curves have minimum around the same point corresponding to the optimum step-size).

In conclusion, we can state that (3.35) represents a good basis for the derivation of step-size adaptation as we will see in Section 3.3.

3.2 Transform domain variable step-size LMS algorithms

In this section we address the problem of step-size adaptation for transform domain LMS algorithm. We note that our discussion here is emphasized for the stationary environments whereas implementations for time-varying environments are addressed in the next section. It is well known that, for highly correlated input signals the speed of convergence of the time domain LMS algorithm degrades dramatically. As an alternative solution, different modifications of the LMS algorithms with variable step-size as well as transform domain LMS (TDLMS) algorithms have been developed in the open literature (see, e.g., [1], [4], [7], [9], [11], [18], [19], [20], [25], [27], [29], [30], [38], [58], [61], [62]).

As we have seen in Section 3.1, in the case of TDLMS an input signal is transformed using an orthogonal transform and the filter coefficients are updated by independent step-sizes as shown in (3.7). In the existing approaches, the step-sizes are often considered time-varying due to the power estimates of the transform coefficients[10] in (3.7). When the power estimates become constants, the different step-sizes corresponding to each filter coefficients are also constants. As a consequence, when the input signal is stationary, the

[10]Some authors consider the step-size corresponding to the i^{th} adaptive filter coefficient in (3.7) to be $\mu_i(n) = \dfrac{\mu}{\epsilon + \sigma_i^2(n)}$ and it is time-varying due to the normalization with $\sigma_{s_i}^2(n)$.

step-size of each filter tap is time-varying just during the early stages of the adaptation and after that it is constant. However, there are no TDLMS algorithms known so far that uses in the update of the step-size the output error. Here we introduce some modifications of the TDLMS algorithm which have the following feature: we define for each step-size a local component depending on the power normalization and a global component that is the same for each filter tap. As opposed to the existing approaches, the global component is also time-variable, and depends on the output error, such that the speed of convergence of the new algorithms is significantly improved.

In our discussion first we briefly describe the existing approaches in the open literature and then three new transform domain algorithms with variable step-size are introduced.

3.2.1 Existing approaches

When the TDLMS is used to update the coefficients of an adaptive filter, equation (3.7) is usually implemented and the power estimates are obtained in various ways. One way is to use the average method of (3.8), whereas in other publications some other methods are proposed, such as the Gram-Schmidt normalization in [60] or other averaging method as in [45]. Also, analysis of different orthogonal transforms for various types of input sequences are published [4], [21] and many performance indexes which express their decorelation properties are defined. It is well known that the optimum transform is the KLT which diagonalizes the autocorrelation matrix $\mathbf{R}_s$, but it requires the knowledge of the input signal statistics in order to be implemented. Other transforms, such as the DCT are shown to be close to the KLT for different signal distributions.

A very interesting approach was proposed by Kim and Wilde in [45], where the coefficients of the adaptive filter have different time-varying step-sizes. In the case of the DCT-LMS introduced in [45], the step-sizes are changed based on the following formula:

$$\mu_i(n+1) = \beta\mu_i(n) + \gamma\,(1-\beta)\left(\frac{1}{\epsilon + \frac{1}{M}\mathbf{s}_i(n)^t\mathbf{s}_i(n)}\right). \tag{3.41}$$

where $\mathbf{s}_i(n) = [s_i(n), s_i(n-1), ..., s_i(n-M+1)]^T$ is the vector of the past M values of the i^{th} transform coefficient $s_i(n)$, $\beta \in [0,1]$, $\gamma \in [0,1]$, and $0 < \epsilon \ll 1$ are some constant parameters.

In [45], the theoretical analysis of the DCT-LMS was provided together with the simulations showing the performances of the proposed algorithm for system identification and channel equalization applications. It was shown that the DCT-LMS performs better in terms of convergence speed than the TDLMS which uses (3.7) and (3.8) for the coefficients adaptation. The theoretical analysis for the DCT-LMS was done in a similar manner with that of the time domain LMS and the value of the steady-state misadjustment was found to be:

$$\mathcal{M} = \frac{1}{2}\sum_{i=1}^{N} y_i \lambda_i \tag{3.42}$$

where $y_i = \dfrac{E\left[\mu_i^2(\infty)\right]}{E\left[\mu_i(\infty)\right]}$ with $E\left[\mu_i(\infty)\right]$ and $E\left[\mu_i^2(\infty)\right]$ being the steady-state mean and respectively mean squared value of the step-size. The analytical expression for the mean and mean squared value of the step-size were also given in [45].

Since the DCT-LMS was the only transform domain algorithm with variable step-size, which we have found in the open literature, for benchmark purposes, our new algorithms are compared with it. However, as it can be seen in (3.41), the step-size of every coefficient of the DCT-LMS is not updated based on the evolution of the output error, therefore one can include this algorithm in the class of algorithms which uses an improved normalization method.

3.2.2 Transform domain LMS adaptive filter with variable step-size

Here, we propose a new algorithm that uses the output error in order to update the step-size of each filter tap resulting in a significant improvement of the convergence speed. To develop our new algorithm we start from the well known method, in which the step-size of the i^{th} coefficient is computed as follows:

$$\mu_i(n) = \frac{\mu}{\epsilon + \sigma_{s_i}^2(n)}. \tag{3.43}$$

In (3.43), the numerator μ can be viewed as the global component of the step-size, since it is the same for each coefficient and the denominator can be viewed as the local component of the step-size, and it depends on the power estimate $\sigma_{s_i}^2(n)$ of the corresponding transform coefficient.

In the approaches proposed so far, only the local component is variable whereas in our new algorithm also the global component is time-varying and depends on the output error as follows:

$$\mu'(n) = \alpha\mu(n) + \frac{\gamma}{L}\sum_{i=n-L+1}^{n} e^2(i) \tag{3.44}$$

and

$$\mu(n+1) = \begin{cases} \mu'(n) & \text{if} \quad n = L, 2L, \ldots \quad \text{and} \quad \mu'(n) \in (\mu_{min}, \mu_{max}) \\ \mu_{max} & \text{if} \quad n = L, 2L, \ldots \quad \text{and} \quad \mu'(n) \geq \mu_{max}, \\ \mu_{min} & \text{if} \quad n = L, 2L, \ldots \quad \text{and} \quad \mu'(n) \leq \mu_{min}, \\ \mu(n) & \text{if} \quad n \neq L, 2L, \ldots . \end{cases} \tag{3.45}$$

In the case of the new TDVSLMS algorithm, the update of each filter coefficient is given by:

$$\widehat{h}_i(n+1) = \widehat{h}_i(n) + \frac{\mu(n)}{\epsilon + \sigma^2_{s_i}(n)} e(n) s_i(n). \tag{3.46}$$

where the notations are the same as for the standard TDLMS algorithm, $\mu(n)$ is given by (3.44) and (3.45) and we have used (3.8) for power estimation.

The behavior of the new TDVSLMS algorithm can be described as follows: for a number of L consecutive iterations (the test interval of length L), the global component $\mu(n)$ is constant, and the new algorithm behaves as a standard TDLMS. At the end of the test interval, the average of the past L squared values of the error is computed, and $\mu(n)$ is updated according to (3.44) and (3.45). In this way, when the output error is large the step-size is increased, such that the convergence time is shortened. When the adaptive filter goes toward the steady-state, the error decreases which decrease also the global component of every step-size. In all our simulations, we have used $L = 10$ which shows good performances. Usually, the parameter γ in (3.44) has a small value, and it may be chosen to meet the misadjustment requirements.

The steady-state mean squared error analysis of the new TDVSLMS algorithm

The steady-state analysis of the TDVSLMS algorithms is done in order to find the relation between the misadjustment of the algorithm and its parameters. Based on the analytical expression of the steady-state missadjustment, we will discuss how to set the parameters of the algorithm in order to obtain the desired performances in terms of convergence speed and small steady-state error. To start the analysis, we first rewrite (3.10) in the following form:

$$\widehat{\mathbf{h}}(n+1) = \widehat{\mathbf{h}}(n) + \tilde{\mu}(n)\mathbf{s}(n)e(n), \tag{3.47}$$

where $\tilde{\mu}(n) = \mu(n)\mathbf{\Gamma}^{-1}(n)$ is an $N \times N$ diagonal matrix with the diagonal elements given by $\tilde{\mu}_i(n) = \mu(n)\Gamma_i^{-1}(n)$ and $\mu(n)$ is given by (3.45).

To make the convergence analysis of the TDVSLMS algorithm more tractable, besides the usual assumptions, we introduce the following ones:

Assumption 1: After few iterations the power estimates of the transform coefficients $s_i(n)$ become constant therefore, the step-sizes $\tilde{\mu}_i(n)$ are independent to $\mathbf{s}(n)$.

Assumption 2: The step-sizes $\tilde{\mu}_i(n)$ in (3.47) and the output error $e(n)$ are independent. This can be justified by the update method of the step-size which is done using just some past L values of the error as shown in (3.45).

The mean squared error (MSE) is defined by (3.16) which we rewrite here for convenience:

$$J(n) = J_{min} + tr\left[\mathbf{R}_s\mathbf{C}(n)\right], \tag{3.48}$$

where J_{min} is the minimum MSE obtained in the case of perfect adaptation, $\mathbf{C}(n)$ is the covariance matrix of the coefficient error vector, $\mathbf{R}_s$ is the autocorrelation matrix of the transform coefficients.

The input autocorrelation matrix can be expressed as $\mathbf{R}_s = Q\mathbf{\Lambda}_s Q^t$, where $\mathbf{\Lambda}_s = diag\left[\lambda_0, \ldots, \lambda_{N-1}\right]$ is the diagonal matrix having on the main diagonal the eigenvalues of $\mathbf{R}_s$, Q is the modal matrix of $\mathbf{R}_s$, $QQ^t = \mathbf{I}$ and $Q^{-1} = Q^t$. Denoting $\mathbf{C}'(n) = Q\mathbf{C}(n)Q^t$, (3.48) can be rewritten as follows:

$$E\left\{e^2(n)\right\} = J_{min} + tr\left[\mathbf{\Lambda}_{ss}\mathbf{C}'(n)\right] = J_{min} + \sum_{i=0}^{N-1}\lambda_i c'_{ii}(n) \tag{3.49}$$

where $c'_{ii}(n)$ are the diagonal elements of $\mathbf{C}'(n)$.

The coefficient error vector is defined as the difference between the adaptive filter coefficients and the optimum Wiener solution and it can be expressed as in the following equation:

$$\widehat{\mathbf{h}}(n+1) - \mathbf{h}_{T_o} = \widehat{\mathbf{h}}(n) - \mathbf{h}_{T_o} + \tilde{\mu}(n)\mathbf{s}(n)e(n) = \left[\mathbf{I} - \tilde{\mu}(n)\mathbf{s}^t(n)\right]\Delta\mathbf{h}(n) + \tilde{\mu}(n)e_0(n)\mathbf{s}(n) \tag{3.50}$$

Computing the outer product of (3.50) by itself, taking the expectations on both sides and using the fact that $\mathbf{C}'(n)$ was obtained from the relation $\mathbf{C}'(n) = Q\mathbf{C}(n)Q^t$, one obtains:

$$\mathbf{C}'(n+1) = \mathbf{C}'(n) - E\left\{\tilde{\mu}(n)\right\}\left[\mathbf{C}'(n)\mathbf{\Lambda}_{ss} + \mathbf{\Lambda}_{ss}\mathbf{C}'(n)\right] + E\left\{\tilde{\mu}^2(n)\right\}\Big[J_{min}\mathbf{\Lambda}_{ss} + 2\mathbf{\Lambda}_{ss}\mathbf{C}'(n)\mathbf{\Lambda}_{ss} + tr\left[\mathbf{\Lambda}_{ss}\mathbf{C}'(n)\right]\mathbf{\Lambda}_{ss}\Big] \tag{3.51}$$

Thus the diagonal elements $c'_{ii}(n)$ of the matrix $\mathbf{C}'(n)$ are obtained from (3.51) as follows:

$$c'_{ii}(n+1) = \left[1 - 2E\left\{\tilde{\mu}_i(n)\right\}\lambda_i\right]c'_{ii}(n) + 2E\left\{\tilde{\mu}_i^2\right\}\lambda_i^2 c'_{ii}(n) + E\left\{\tilde{\mu}_i^2(n)\right\}J_{min} + +E\left\{\tilde{\mu}_i^2(n)\right\}\sum_{m=0}^{N-1}\lambda_m c'_{mm}(n) \tag{3.52}$$

The mean value of the variable step-size $\tilde{\mu}_i(n)$ is given by:

$$E\left\{\tilde{\mu}_i(n)\right\} = E\left\{\mu(n)\Gamma_i^{-1}(n)\right\} = \frac{E\left\{\mu(n)\right\}}{E\left\{\Gamma_i(n)\right\}}, \tag{3.53}$$

with $E\left\{\mu(n)\right\}$ given by:

$$E\left\{\mu(n)\right\} = \begin{cases} E\left\{\mu'(n)\right\}, & \text{if } \; n-1 = L, 2L, \ldots \; \text{ and } \; \mu'(n) \in (\mu_{min}, \mu_{max}) \\ \mu_{min}, & \text{if } \; n-1 = L, 2L, \ldots \; \text{ and } \; \mu'(n) < \mu_{min} \\ \mu_{max}, & \text{if } \; n-1 = L, 2L, \ldots \; \text{ and } \; \mu'(n) > \mu_{max} \\ E\left\{\mu(n-1)\right\}, & \text{otherwise} \end{cases} \tag{3.54}$$

Taking the expectation operator in (3.44) the mean value of the step-size $E\left\{\mu'(n)\right\}$ can be computed as follows:

$$E\left\{\mu'(n)\right\} = \beta E\left\{\mu(n-1)\right\} + E\left\{\frac{\gamma}{L}\sum_{k=n-L}^{n-1} e^2(k)\right\} = \beta E\left\{\mu(n-1)\right\} + \frac{\gamma}{L}\sum_{k=n-L}^{n-1} J(k). \tag{3.55}$$

where $E\left\{e^2(k)\right\} = J(k)$ is the MSE at time instant k.

The mean squared value of the step-size $\tilde{\mu}_i(n)$ corresponding to the i^{th} coefficient, is given by:

$$E\left\{\tilde{\mu}_i^2(n)\right\} = \frac{E\left\{\mu^2(n)\right\}}{E\left\{\Gamma_i^2(n)\right\}} \tag{3.56}$$

and the numerator in (3.56) can be expressed as follows:

$$E\left\{\mu^2(n)\right\} = \begin{cases} E\left\{\mu'^2(n)\right\} & \text{if } \; n-1 = L, 2L, \; \text{ and } \; \mu'(n) \in [\mu_{min}, \mu_{max}] \\ \mu_{min}^2 & \text{if } \; n-1 = L, 2L, \; \text{ and } \; \mu'(n) < \mu_{min} \\ \mu_{max}^2 & \text{if } \; n-1 = L, 2L, \; \text{ and } \; \mu'(n) > \mu_{max} \\ E\left\{\mu^2(n-1)\right\} & \text{otherwise} \end{cases} \tag{3.57}$$

The mean squared value of $\mu'(n)$ is obtained from (3.44) as follows:

$$E\left\{\mu'^2(n)\right\} = \beta^2 E\left\{\mu^2(n-1)\right\} + \frac{2\beta\gamma}{L} E\left\{\mu(n-1)\sum_{k=n-L}^{n-1} e^2(k)\right\} + \frac{\gamma^2}{L^2} E\left\{\left(\sum_{k=n-L}^{n-1} e^2(k)\right)^2\right\} \tag{3.58}$$

For small values of γ the term $\frac{\gamma^2}{L^2} E\left\{\left(\sum\limits_{k=n-L}^{n-1} e^2(k)\right)^2\right\}$ have negligible values at the steady-state and it can be discarded from (3.58). More than that, if we use **Assumption 2**, the following expression is obtained to express the mean squared value of $\mu'(n)$:

$$E\left\{\mu'^2(\infty)\right\} = \beta^2 E\left\{\mu^2(\infty)\right\} + 2\beta\gamma E\left\{\mu(\infty)\right\} J_{st} \tag{3.59}$$

To obtain the steady-state MSE we need to compute the mean and mean squared values of the step-sizes $\mu_{i_{st}}$ at the steady-state. Combining (3.53) with (3.54) and (3.55), and combining (3.56) with (3.57) and (3.58), the mean and mean squared values of $\tilde{\mu}_{i_{st}}$ are obtained as follows:

$$E\left\{\tilde{\mu}_{i_{st}}\right\} = \frac{E\left\{\mu'(\infty)\right\}}{\Gamma_i} = \frac{\gamma J_{st}}{\Gamma_i\left(1-\beta\right)}, \quad E\left\{\tilde{\mu}^2_{i_{st}}\right\} = \frac{E\left\{\mu'^2(\infty)\right\}}{\Gamma_i^2} = \frac{2\beta\gamma^2 J_{st}^2}{\Gamma_i^2\left(1-\beta^2\right)\left(1-\beta\right)} \tag{3.60}$$

We note that, in the derivation of (3.60) we have assumed that the step-size is between μ_{min} and μ_{max}. The steady-state misadjustment can be obtained from (3.49), (3.52) and (3.60) following a derivation similar to the one in [48] and [45]:

$$\mathcal{M} = \frac{J_{st} - J_{min}}{J_{min}} \approx \frac{1}{2}\sum_{i=0}^{N-1} y_i \lambda_i \approx \frac{\beta\gamma J_{st}}{(1-\beta^2)} \sum_{i=0}^{N-1} \frac{\lambda_i}{\Gamma_i} \tag{3.61}$$

Since the matrix $\mathbf{R}_s$ is near diagonal and the power estimates Γ_i are close to the real powers, then the summation in (3.61) can be approximated by N (due to the fact that $\Gamma_i \approx \lambda_i$). Thus finally, the steady-state misadjustment can be written as follows:

$$\mathcal{M} \approx \frac{\frac{\beta\gamma N}{1-\beta^2} J_{min}}{1 - \frac{\beta\gamma N}{1-\beta^2} J_{min}} \tag{3.62}$$

which is similar to the results derived in [48].

We note that for a constant step-size, say $\mu(n) = \mu$, equation (3.62) can be simplified to:

$$\mathcal{M} \approx \frac{1}{2}\mu N \tag{3.63}$$

that is the well known approximation for the misadjustment of the TDLMS with fixed step-size.

The value of the parameter L in (3.45) is not critical for the algorithm. Actually we have seen in our simulations that L has to be smaller than the convergence time of the algorithm in order to have enough step-size updates. Any values $L \leq 10$ seems to be good choices for a wide range of applications.

For the parameter β we have found that a value $\beta = 0.9$ shows good performances in all our simulations. Setting $\beta = 0.9$, the value of the parameter γ can be obtained from (3.62), such that, a desired level of the misadjustment is obtained.

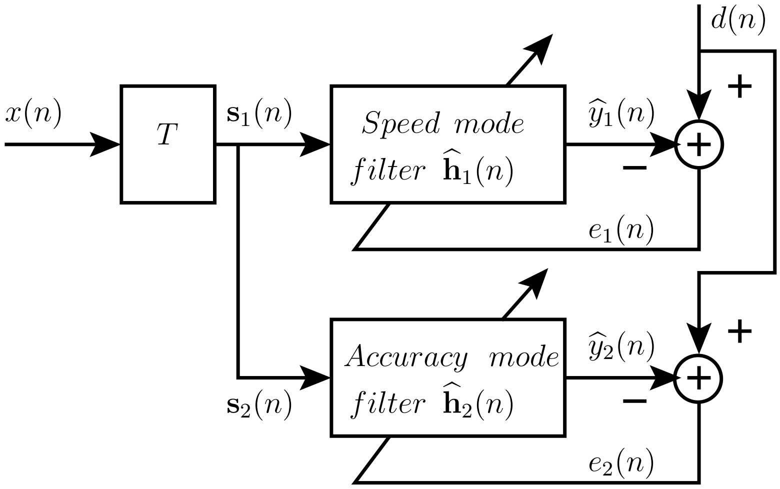

Figure 3.5: The block diagram of the transform domain complementary pair LMS with variable step-size.

3.2.3 The transform domain complementary pair variable step-size LMS algorithm

In the development of the transform domain complementary pair LMS algorithm with variable step-size (TDCPVSLMS), we consider the equation (3.43) which describes the independent step-size computation. Again μ in (3.43) is viewed as an overall component of $\mu_i(n)$ since it is the same for each filter tap, whereas $\gamma+\sigma_i^2(n)$ is the local component of $\mu_i(n)$ (it is different for each filter tap). In the new implementation, the overall component is also time-variable as in the algorithm introduced in the previous section. However, the expression used to update $\mu(n)$ is different. Actually, the TDCPVSLMS algorithm, depicted in Fig. 3.5, is the transform domain implementation of the CP-VSLMS from Section 2.2. In Fig. 3.5, the block denoted by T represents the transform applied to the input signal $\mathbf{x}(n)$, and $\mathbf{s}(n)$ is the vector of the transform coefficients. As in the case of CP-VSLMS, there are two adaptive algorithms that work in parallel. The *Speed mode filter* $\widehat{\mathbf{h}}_1(n)$, and the *Accuracy mode filter* $\widehat{\mathbf{h}}_2(n)$. The adaptive filter filter $\widehat{\mathbf{h}}_1(n)$ is an additional filter which uses a large and fixed overall component μ_1 and is implemented just to increase the convergence speed of the algorithm, whereas $\widehat{\mathbf{h}}_2(n)$ represents the filter of interest.

The TDCPVSLMS algorithm is described as follows:

- Compute the outputs and the output errors of the two adaptive filters:

$$\widehat{y}_1(n) = \widehat{\mathbf{h}}_1^t(n)\mathbf{s}(n), \quad \widehat{y}_2(n) = \widehat{\mathbf{h}}_2^t(n)\mathbf{s}(n), \quad e_1(n) = d(n) - \widehat{y}_1(n), \quad e_2(n) = d(n) - \widehat{y}_2(n). \tag{3.64}$$

- Update the coefficients of the speed mode filter:

$$\widehat{\mathbf{h}}_1(n+1) = \widehat{\mathbf{h}}_1(n) + \mu_1 \mathbf{\Gamma}^{-1} e_1(n)\mathbf{s}(n). \tag{3.65}$$

- Update the coefficients of the accuracy mode filter:

$$\widehat{\mathbf{h}}_2(n+1) = \begin{cases} \widehat{\mathbf{h}}_1(n+1), & \text{if} \quad n = L, 2L, \ldots, \quad \text{and} \quad \prod_{i=1}^{M} Q(i) = 1 \\ \widehat{\mathbf{h}}_2(n) + \mu_2(n)\mathbf{\Gamma}^{-1}(n) e_2(n)\mathbf{s}(n), & \text{otherwise} \end{cases}. \tag{3.66}$$

- Update the overall component of the step-size for the accuracy mode filter:

$$\mu_2(n+1) = \begin{cases} \dfrac{\mu_2(n) + \mu_1}{2}, & \text{if} \quad n = L, 2L, \ldots, \quad \text{and} \quad \prod\limits_{i=1}^{M} Q(i) = 1 \\ \max\{\alpha\mu_2(n), \mu_{min}\}, & \text{if} \quad n = L, 2L, \ldots, \quad \text{and} \quad \prod\limits_{i=1}^{M} Q(i) = 0 \\ \mu_2(n), & \text{otherwise} \end{cases} \tag{3.67}$$

The matrix $\mathbf{\Gamma}$ in (3.65) and (3.66) is a diagonal matrix whose elements are iteratively computed using (3.8). The value of Q in (3.66) and (3.67) is computed as:

$$Q(i) = \begin{cases} 1 & \text{if} \sum\limits_{k=n-iM}^{n-(i-1)M} e_2^2(k) > \sum\limits_{k=n-iM}^{n-(i-1)M} e_1^2(k), \\ 0 & \text{otherwise} \end{cases} \tag{3.68}$$

According to the above equations, the behavior of the new algorithm can be described as follows: for L consecutive iterations, which represents the test interval, the value of the overall component $\mu_2(n)$ is constant and the adaptive filters $\widehat{\mathbf{h}}_1(n)$ and $\widehat{\mathbf{h}}_2(n)$ perform independently as plain TDLMS filters. Also on this interval, the local averages of $e_1^2(n)$ and $e_2^2(n)$ are computed. If the local average of $e_2^2(n)$ is larger than the local average of $e_1^2(n)$ for M consecutive test intervals then the coefficients $\widehat{\mathbf{h}}_2(n)$ are updated with the values $\widehat{\mathbf{h}}_1(n)$. This is due to the fact that the filter $\widehat{\mathbf{h}}_1(n)$ performs faster than $\widehat{\mathbf{h}}_2(n)$ due to the larger value of μ_1. When the speed mode filter performs better (in terms of output mean squared error) than the accuracy mode filter, it also means that a larger step-size would be more beneficial to be used. This is the reason why the value of $\mu_2(n)$ is increased in order to get a faster convergence. When the two algorithms approximate the steady-state, the local average of $e_2^2(n)$ becomes smaller than the local average of $e_1^2(n)$ and the value of $\mu_2(n)$ is decreased in order to obtain the desired steady-state missadjustment[11].

[11] The speed mode filter reaches its steady-state faster than the accuracy mode filter due to the fact that $\mu_2(n) \leq \mu_1$ at each iteration.

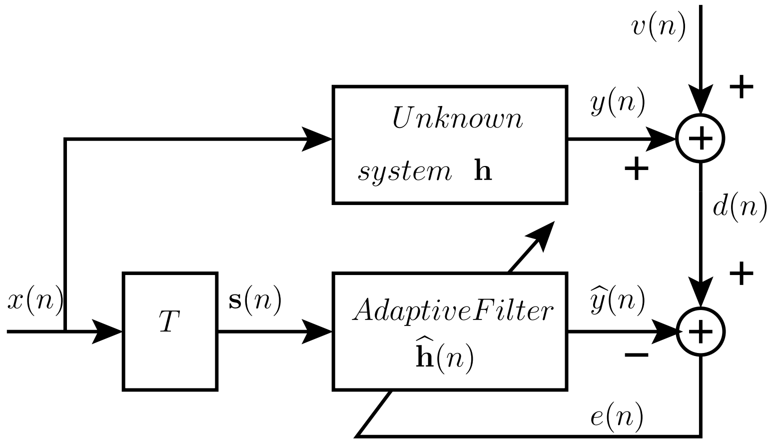

Figure 3.6: The block diagram of the transform domain noise constrained LMS with variable step-size, for system identification.

The maximum and minimum values of $\mu_2(n)$ are μ_1, and μ_{min} respectively. At the steady-state the value of the overall component of the step-size $\mu_2(n)$ is constant and equal to μ_{min}, as a consequence the steady-state misadjustment of the new algorithm equals the misadjustment of a plain TDLMS algorithm with step-size μ_{min}.

The size of parameter L (the length of the test interval) in (3.68) should be sufficiently large $L >> 1$, such that the statistical average of $e_1^2(n)$ and $e_2^2(n)$ can be obtained. Also, L should be much smaller than the length P of the training input $x(n)$, such that, a sufficient number of re-initializations will be possible. The parameter M in (3.66) and (3.67), must be chosen according to the training input length and the noise level. The total comparison length $M \times L$ should be much smaller than P in order to ensure a prompt update of the variable step and filter coefficients. The value of M has to be also large in order to avoid the mistaken re-initializations due to the noise level. In our simulations we have used $L = 10$, and $M = 3$ and we have obtained good results.

3.2.4 Noise constrained LMS algorithm in transform domain

As in the time domain implementation from Chapter 2, the main reason to introduce the Transform Domain Noise Constrained Variable Step-size LMS (TDNCVSLMS) algorithm was to reduce the complexity of the algorithm from the previous section which uses two adaptive filters working in parallel to increase the convergence speed. The algorithm from Section 3.2.2 uses just one adaptive filter and its computational complexity is much lower than the computational complexity of the TDCPVSLMS. However, the setup of its parameters is more complicated since in the analytical expression of the misadjustment (3.62), the value of the minimum MSE is included. Here, we propose a new algorithm that uses some information about the noise variance σ_v^2 for updating the step-size in order to increase the convergence speed[12]. The block diagram of the new algorithm for system identification is shown in Fig. 3.6. The algorithm is described by the same steps as in the case of TDLMS, the only difference is that the global component of the step-size is time-varying.

The coefficients of the adaptive filter are updated by the following equation:

$$\widehat{\mathbf{h}}(n+1) = \widehat{\mathbf{h}}(n) + \mu(n)\mathbf{\Gamma}^{-1}\mathbf{s}(n)e(n) \tag{3.69}$$

and the step-size is changed according to:

$$\mu(n+1) = \begin{cases} \dfrac{\mu_{max} + \mu(n)}{2} & \text{if } \dfrac{1}{L}\displaystyle\sum_{i=n-L+1}^{n} e(i)^2 > C(n) \quad \text{and} \quad n = L, 2L, \ldots \\ max\{\mu_{min}, \alpha\mu(n)\} & \text{if } \dfrac{1}{L}\displaystyle\sum_{i=n-L+1}^{n} e(i)^2 \leq C(n) \quad \text{and} \quad n = L, 2L, \ldots \\ \mu(n) & \text{otherwise} \end{cases} \tag{3.70}$$

The behavior of the new TDNCVSLMS algorithms is as follows: for L consecutive iterations, the global component $\mu(n)$ of the step-size is kept constant and the algorithm performs as a standard TDLMS algorithm with fixed step-size. With this constant value of $\mu(n)$ the algorithm would have a certain steady-state MSE denoted by $C(n)$. At the end of the test interval (after L iterations), the average of the squared error is computed. If the average of the squared error is larger than $C(n)$, then the step-size $\mu(n)$ is increased. This means that the algorithm is far from the intermediate steady-state and in order to speed-up the convergence, the global component of the step-size has to be increased. Otherwise, when the algorithm reaches the intermediate steady-state within the test interval of length L, the step-size is decreased in order to obtain a desired steady-state level. As we can see, the value of $C(n)$ is changed each time when the step-size is modified and represents the MSE obtained if the algorithm would have a constant step-size until convergence.

The minimum value allowed for the global component $\mu(n)$ is μ_{min} and at the steady-state, the algorithm performs as a TDLMS with constant step-size $\mu(n) = \mu_{min}$. There-

[12] This is the transform domain counterpart of the algorithm introduced in Section 2.2.3. We emphasize the difference between the NCVSLMS and the transform domain implementation which relies in the fact that in time domain we have used the median operation to test when the algorithm is at steady-state whereas in transform domain the average of the square error is implemented.

fore, the final level of the misadjustment is given by μ_{min}. The maximum value of $\mu(n)$ can be very close to μ_{max} but is always smaller than μ_{max}.

Analysis and the setup of the coefficients

Here, we discuss about the selection of the coefficients of the TDNCVSLMS algorithm. We start with the condition that ensures the stability of the algorithm. As we can see from (3.70), the global component $\mu(n)$ of the new algorithm is bounded by $\mu_{min} \leq \mu(n) < \mu_{max}$. Since the new algorithm performs as the standard TDLMS algorithm inside of each test interval, the stability analysis can be done using well known methods (see [45], [58] and the references therein). Following the derivations in [45] and making the usual assumptions, the TDNCVSLMS algorithm converges when $0 < \mu_{max} < \frac{2}{N}$.

In order to compute the value of $C(n)$, we will consider L consecutive iterations on which the overall step-size is constant. The value of $C(n)$ is set to be the steady-state MSE obtained at the output of the adaptive filter if the step-size is kept constant. In Section 3.1 a simplified analytical expression for the steady-state MSE was derived therefore $C(n)$ can be approximated by:

$$C(n) = \left(1 + \frac{1}{2}N\mu(n)\right)\sigma_v^2 \tag{3.71}$$

If the noise variance cannot be accurately estimated then a penalty term Φ can be introduced in (3.71) as follows:

$$C(n) = \Phi\left(1 + \frac{1}{2}N\mu(n)\right)\sigma_v^2 \tag{3.72}$$

where $\Phi \geq 1$ and close to 1.

The value of L in (3.70) has to be large enough such that the MSE can be approximated and also it has to be smaller than the convergence time for an TDLMS with fixed step-size μ_{min}, such that a sufficient number of step-size updates occur. In a large number of simulations, for different signal to noise ratio, the following selection gives good performance $L = 10$ and $\alpha = 0.9$.

3.2.5 Simulations and results

In this section the results obtained for the problem of system identification are presented. The compared algorithms, in this framework, are: the plain time domain LMS with constant step-size, the VSSLMS from [48], the RVSLMS introduced in [1], the CP-LMS from [61], the CP-VSLMS and the NCVSLMS described in Section 2.2, the plain TDLMS with fixed step-size, the DCT-LMS introduced in [45] and the TDVSLMS, TDCPVSLMS and TDNCVSLMS described in this chapter. For the time-domain implementations the block diagrams used in the simulations are those from Section 2.2 whereas for the time domain implementations we have used the block diagrams shown in the previous sections.

The unknown system has $N = 16$ constant coefficients and all adaptive filters were chosen of the same length. The transform used in all transform domain implementations was the DCT which gives real coefficients and also shows good decorrelation properties for many input sequences found in practice.

The input sequence, was generated by the following autoregressive model:

$$x(n) = 1.79x(n-1) - 1.85x(n-2) + 1.27x(n-3) - 0.41x(n-4) + \nu(n)$$

where $\nu(n)$ is a white Gaussian random signal with zero mean and variance $\sigma_\nu^2 = 0.14817$.

The eigenvalue spread ratio of the autocorrelation matrix, for this highly correlated input sequence was found to be 944.67. The signal to noise ratio at the output of the unknown system was 50 dB, and all the simulations were obtained by averaging 100 independent runs of the algorithms. The parameters of all the compared algorithms are given in Table 3.1, and they were chosen such that the steady-state missadjustments are comparable (around 0.04). The selection of the parameters was done following the guidelines from the corresponding papers and the levels of the misadjustments, obtained experimentally, are shown in Table 3.2.

The learning curves (the output excess MSE) for the compared algorithms are shown in Fig. 3.7 for VSSLMS, CP-VSLMS and NCVSLMS, in Fig. 3.8 for LMS, RVSLMS and CP-LMS, in Fig. 3.9 for TDLMS, DCT-LMS and TDCPVSLMS and in Fig. 3.10 for TDVSLMS, TDCPVSLMS and TDNCVSLMS.

As expected, the time domain implementations do not perform well for such highly correlated input signal. Indeed if we compare the results shown in Fig. 3.7 and Fig. 3.8 with those shown in Chapter 2 we can see that the convergence speed of the time domain adaptive algorithms is much smaller for highly correlated input sequence. We note that the learning curves shown in chapter 2 represent the mean squared coefficient error and not the excess MSE of the algorithms. The comparison can be made between this figure and the ones presented here if we take into account that the simulations performed in Chapter 2 were done with a random zero mean Gaussian-distributed input sequence and in that case the excess MSE and the mean squared coefficient error are proportional[13].

[13]Also the lengths of the filters in Chapter 2 were smaller which means that the speed of convergence would be larger. However the difference between the convergence speed of the time domain implementations in the framework of Chapter 2 and in the framework of this chapter are much larger.

Table 3.1: The parameters of the compared algorithms (LMS, VSSLMS, RVSLMS, CP-LMS, CPVSLMS, NCVSLMS, TDLMS, TDVSLMS, TDCPVSLMS, TDNCVSLMS).

LMS	$\mu = 5 \cdot 10^{-3}$
VSSLMS	$\gamma = 1, \quad \mu_{min} = 5 \cdot 10^{-3}, \quad \alpha = 0.97, \quad \mu_{max} = 3 \cdot 10^{-2}$
RVSLMS	$\mu_{max} = 3 \cdot 10^{-2}, \quad \alpha = 0.97, \quad \mu_{min} = 5 \cdot 10^{-3}, \quad \beta = 0.99, \quad \gamma = 1$
CP-LMS	$\mu_1 = 3 \cdot 10^{-2}, \quad L = 1, \quad \mu_2 = 5 \cdot 10^{-3}, \quad T = 100$
CPVSLMS	$\mu_1 = 3 \cdot 10^{-2}, \quad L = 1, \quad \mu_{min} = 5 \cdot 10^{-3}, \quad T = 100, \quad \alpha = 0.6$
NCVSLMS	$\mu_{min} = 5 \cdot 10^{-3}, \quad \alpha = 0.6, \quad \mu_{max} = 3 \cdot 10^{-2}, \quad T = 100$
TDLMS	$\mu = 5 \cdot 10^{-3}, \quad \beta = 0.9, \quad \epsilon = 2.5 \cdot 10^{-2}$
DCT-LMS	$M = 10, \quad \gamma = 2 \cdot 10^{-3}, \quad \beta = 0.9985, \quad \epsilon = 8 \cdot 10^{-4}$
TDVSLMS	$\mu_{max} = 5 \cdot 10^{-2}, \quad \alpha = 0.9, \quad \mu_{min} = 5 \cdot 10^{-3}, \quad \beta = 0.9,$ $\epsilon = 2.5 \cdot 10^{-2}, \quad L = 10, \quad \gamma = 10^{-3}$
TDCPVSLMS	$\mu_{min} = 5 \cdot 10^{-3}, \quad M = 3, \quad \mu_{max} = 5 \cdot 10^{-2}, \quad L = 10,$ $\epsilon = 2.5 \cdot 10^{-2}, \quad \alpha = 0.6, \quad \beta = 0.9,$
TDNCVSLMS	$\mu_{min} = 5 \cdot 10^{-3}, \quad L = 10, \quad \mu_{max} = 5 \cdot 10^{-2}, \quad \alpha = 0.6, \quad \epsilon = 2.5 \cdot 10^{-2}$

Table 3.2: The misadjustments of the compared algorithms.

	LMS	VSSLMS	RVSLMS	CP-LMS	CPVSLMS	NCVSLMS
$\mathcal{M}$	4.18%	3.98%	4.35%	4.20%	4.35%	4.23%
	TDLMS	DCT-LMS	TDVSLMS	TDCPVSLMS	TDNCVSLMS	
$\mathcal{M}$	3.91%	3.46%	4.64%	3.90%	3.96%	

In Fig. 3.9 and Fig. 3.10, the learning curves of the transform domain implementations are presented. One can see from these plots that using the output error to adjust the global component of the step-size, the convergence speed of the transform domain implementations can be significantly increased.

Of course other techniques which can improve the convergence may be included in addition to the step-size adaptation. One method is the selection of the orthogonal transform which must be chosen based on the properties of the input sequence. Another way to improve the convergence speed is to implement other expressions for power estimation instead of (3.8). Also, when (3.8) is implemented, initialization of the power estimates is important. In our simulations we have observed that better performances are obtained

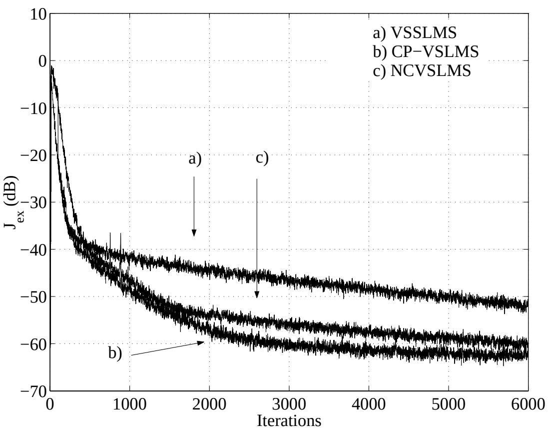

Figure 3.7: Comparison between VSSLMS, CP-VSLMS, and NCVSLMS

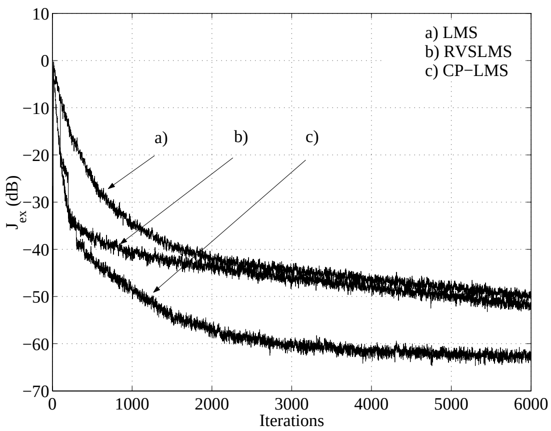

Figure 3.8: Comparison between LMS, RVSLMS and CP-LMS

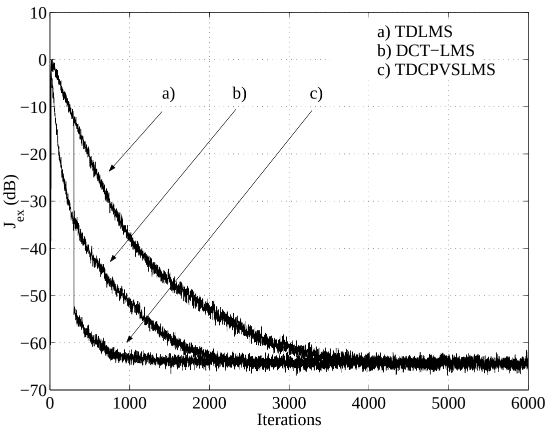

Figure 3.9: Comparison of convergence curves for TDLMS, DCT-LMS and TD-CPVSLMS

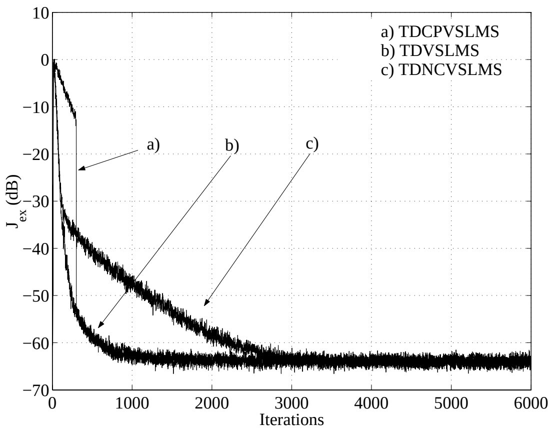

Figure 3.10: Comparison of convergence curves for TDVSLMS, TDCPVSLMS, and TDNCVSLMS

when the power estimates are initialized with values close to the adaptive filter length[14]. All these techniques are well described in the referenced publications, therefore they are not detailed here.

[14]Actually the power estimates must be initialized close to the real powers of the transform coefficients for faster convergence. Due to the fact that in our simulations we have used the DCT and the power of the input sequence was unity, this initialization seems to be a good choice.

Algorithm	TDLMS	DCT-LMS	TDCPVSLMS	TDNCVSLMS	TDVSLMS
Mult./Div.	$6N+1$	$5N+MN=$ $=15N$ $(M=10)$	$12N+5$	$6N+8$	$6N+4$
Add./Sub.	$3N$	$3N+MN=$ $=13N$ $(M=10)$	$6N+3$	$3N+3$	$3N+2$

Table 3.3: Computational complexity of the TDLMS, DCT-LMS, TDCPVSLMS, TDNCVSLMS and TDVSLMS algorithms.

3.2.6 Comparison of the transform domain variable step-size LMS algorithms

Here, we compare the transform domain implementations, described above, in terms of their computational complexity, memory load and simplicity of the parameter setup. To this end, in Table 3.3 the memory load and computational complexity of the TDLMS, TDCPVSLMS, TDNCVSLMS, DCT-LMS and TDVSLMS are shown.

As expected, among all the algorithms, the TDCPVSLMS have a large complexity due to the use of two adaptive filters that work in parallel. Despite this fact, it possesses the faster convergence and the setup of its parameters is very simple. A single parameter μ_{min} must be chosen in order to obtain a desired level of the misadjustment and the other parameters influence the convergence speed. We shall note that the expression $\dfrac{\mu(n)+\mu_1}{2}$ is probably not the best choice to increase the value of the step-size. Other expressions that provide better results may be used instead. We emphasize here that the complexity of the DCT-LMS is also large, due to the calculation of the power estimates. Moreover the complexity of the DCT-LMS depends on the parameter M.

In the case of TDNCVSLMS, in the formula for step-size update, the noise variance σ_v^2 is needed[15]. Therefore it can be implemented in applications where σ_v^2 is known or it can be estimated. Actually, the same discussion is valid also for TDVSLMS and DCT-LMS algorithm if we take into account the analytical expression to estimate their misadjustment (see (3.61) and (3.42) respectively). These two equations are used in order to setup the values of the parameters of both algorithms and in both equations the value of the minimum MSE is included. In system identification applications, the minimum MSE equals the noise variance therefore, also these two algorithms require some information

[15] As in the case of its time domain counterpart.

about σ_v^2. If we refer to the time domain implementations, VSSLMS and RVSLMS the same discusion is valid for their parameter's setup. The TDCPVSLMS and CP-VSLMS does not have this problem since their misadjustments depends only on the minimum bounds of their step-sizes.

Finally, the advantages and disadvantages of the transform domain implementations are synthesized in Table 3.4.

	TDLMS	DCT-LMS	TDCPVSLMS	TDNCVSLMS	TDVSLMS
Complexity	small	small	large	small	small
Speed	slowest	fast	fast	fast	fast
Setup	simple	needs J_{min}	simple	needs J_{min}	needs J_{min}

Table 3.4: Advantages and disadvantages of the transform domain algorithms.

3.3 Transform domain LMS algorithm with optimum step-size

In Section 3.1.2, we have given a brief theoretical analysis of the TDLMS for tracking time-varying channels. Using some common assumptions, we have obtained a simplified formula (3.35) which describes the steady-state MSE of the algorithm. From (3.35), it follows that the dependence between the steady-state MSE and the step-size is nonlinear and the steady-state MSE has three components. One component σ_v^2 which is the minimum level of the MSE achieved in the case of perfect adaptation, the second component $\frac{1}{2}\sigma_v^2 N$, proportional with the step-size μ, and represents the component due to the imperfect adaptation of the coefficients. Finally, the third part $\mu^{-1} tr\left[\mathbf{R}_D \mathbf{Q}_D\right]$ is due to the time variations of the channel coefficients and is inversely proportional with the step-size.

The primary goal of the algorithm introduced here is to adaptively adjust the step-size μ toward the optimum μ_{opt}^{mse} which minimizes the steady-state MSE. We emphasize that the steady-state MSE and the steady-state mean squared coefficient error are minimized by different step-sizes which are expressed in (3.36). In order to compute μ_{opt}^{mse}, one needs to know the trace of the matrix $\mathbf{R}_D \mathbf{Q}_D$ and the variance of the output noise σ_v^2. When these two values are available, the computation of μ_{opt}^{mse} is trivial. Here we assume that this information is not available, and we propose an iterative method for computing the optimum value of the step-size μ_{opt}^{mse}.

For the simplicity of the exposition we make the following notations regarding (3.35):

$$A = \frac{1}{2}\sigma_v^2, \quad B = \tfrac{1}{2}tr\left[\mathbf{R}_s\mathbf{Q}_s\right] \tag{3.73}$$

With these notations, equation (3.35) can be written as $J_{st} = 2A + \mu NA + \dfrac{B}{\mu}$. If we consider two adaptive TDLMS filters with equal lengths N and different step-sizes μ_1 and μ_2, their steady-state mean squared errors can be approximated by the following expressions:

$$J_{st_1} = 2A + \mu_1 NA + \frac{B}{\mu_1} \quad \text{and} \quad J_{st_2} = 2A + \mu_2 NA + \frac{B}{\mu_2}. \tag{3.74}$$

In (3.74), the length N and the step-sizes μ_1 and μ_2 of the adaptive filters are known. Also some estimates of J_{st_1} and J_{st_2} can be obtained for instance by averaging the output squared errors. As a consequence, the system of equations in (3.74) can be easily solved in order to compute the unknowns A and B. The solution is given by the following expression:

$$A = \frac{J_{st_1}\mu_1 - J_{st_2}\mu_2}{(\mu_1 - \mu_2)\left[2 + N(\mu_1 + \mu_2)\right]}, \quad B = \mu_1\left(J_{st_1}(\infty) - 2A - \mu_1 NA\right) \tag{3.75}$$

The optimum step-size, which minimizes the steady-state MSE can be computed by the following formula:

$$\mu_{opt}^{mse} = \sqrt{\frac{B}{NA}} \tag{3.76}$$

The proposed approach for step-size adaptation introduced in the sequel is based on this concept, which uses two adaptive filters with equal lengths and different step-sizes.

3.3.1 The proposed implementation

Based on the above derivations, we propose the following TDLMS with adaptive step-size whose block diagram is depicted in Fig. 3.11. The new algorithm contains two adaptive TDLMS filters with equal lengths N that operate in parallel. The first adaptive filter with coefficients $\hat{\mathbf{h}}_1(n)$ has a fixed step-size μ_1 while the second adaptive filter $\hat{\mathbf{h}}_2(n)$ has a variable step-size $\mu_2(n)$ which is adapted using the following formula:

$$\mu_2(n+1) = \begin{cases} \sqrt{\frac{B}{NA}}, & \text{if, } \quad n = L, 2L, 3L, ... \\ \mu_2(n), & \qquad \text{otherwise} \end{cases} \tag{3.77}$$

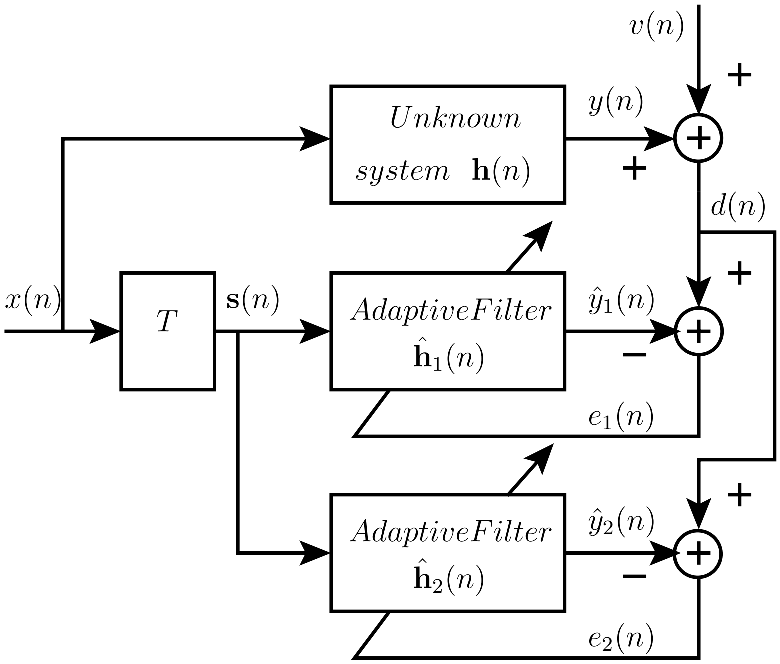

Figure 3.11: The block diagram of the proposed transform domain adaptation of the step-size.

The parameters A and B are computed using (3.75). We note that A and B have constant values due to the fact that the input $x(n)$ is stationary and the covariance matrix $\mathbf{Q}_s$ has constant elements.

As we can see from (3.77), the behavior of the proposed algorithm can be described as follows: for a number of L consecutive iterations, called the test interval, the step-sizes of both adaptive TDLMS filters are constant. At the end of the test interval, the output MSE for both adaptive filters (J_{st_1} and respectively J_{st_2}) are computed and the step-size $\mu_2(n)$ is updated according with (3.77). To compute the output MSE of the two adaptive filters we propose the following analytical expression[16]:

$$\begin{aligned} J_1(n) &= \delta J_1(n-1) + (1-\delta)\,\tfrac{1}{L}\sum_{i=n-L+1}^{n} e_1^2(i), \\ J_2(n) &= \delta J_2(n-1) + (1-\delta)\,\tfrac{1}{L}\sum_{i=n-L+1}^{n} e_2^2(i) \end{aligned} \tag{3.78}$$

with $0 < \delta < 1$ being a constant parameter.

We should emphasize that in (3.78) the values of the output MSE at time instant n are computed whereas in (3.75) the steady-state MSE values are necessary. Therefore, at

[16]Other expressions which give better approximation of the output MSE can be used as well.

the beginning of the adaptation, the value of the step-size $\mu_2(n)$ is far from the optimum due to the use of the transient mean squared error. For this reason we do not update the step-size $\mu_2(n)$ just once during the adaptation but we update it many times using (3.77). When both adaptive filters go near the steady-state, the estimates in (3.78) are close to the steady-state values of the MSE and the step-size $\mu_2(n)$ converges to the optimum.

In all our experiments we have used a fixed value of the parameter L, which is the length of the test interval. Another possibility, which is not addressed here, is to use a time-varying test interval, for instance proportional with the time constant of the algorithm.

We have tested the proposed algorithm in system identification framework depicted in Fig. 3.11. The output noise $v(n)$ was a Gaussian zero mean sequence with variance $\sigma_v^2 = 25 \times 10^{-4}$. The input sequence $x(n)$ was given by the model:

$$x(n) = \gamma x(n-1) + \theta(n)$$

where $\gamma = 0.9$ and $\theta(n)$ is a random Gaussian-distributed sequence with zero mean and variance chosen, such that the variance of $x(n)$ was unity.

The lengths of the unknown system and of the adaptive filters were equal to $N = 10$. The step-size of the first adaptive filter was chosen $\mu_1 = 5 \times 10^{-3}$ while the step-size of the second adaptive filter was initialized with $\mu_2(0) = 10^{-3}$.

To update the coefficients of both adaptive filters we have used (3.10) and the powers of the transform coefficients were estimated by (3.8). The parameter used to estimate the powers of the transform coefficients was $\alpha = 0.9$ and the coefficients used to estimate the MSE in (3.78) was $\delta = 0.9$. The length of the test interval in (3.77) was chosen $L = 50$. The time-varying unknown system was modelled by (3.20) and the increments $\epsilon(n)$ of the time-varying channel coefficients were random zero mean sequences with variance 10^{-6}.

The plotted results are obtained by averaging a number of 100 independent runs each run containing a number of 5×10^3 iterations.

3.3.2 Simulations and results

The behavior of the expected value of the step-size $E[\mu_2(n)]$, during the adaptation, is depicted in Fig. 3.12 together with the value of the optimum step-size which was computed from (3.36). We can see from this figure that the step-size of the proposed algorithm converge close to the optimum μ_{opt}^{mse}. The transient period in Fig. 3.12 is due to the transient periods of both adaptive filters.

The steady-state MSE at the output of the first adaptive filter with fixed step-size $\mu_1 = 5 \times 10^{-3}$ was $J_{st_1} = -24.2760$ dB while for the second adaptive filter with adaptive

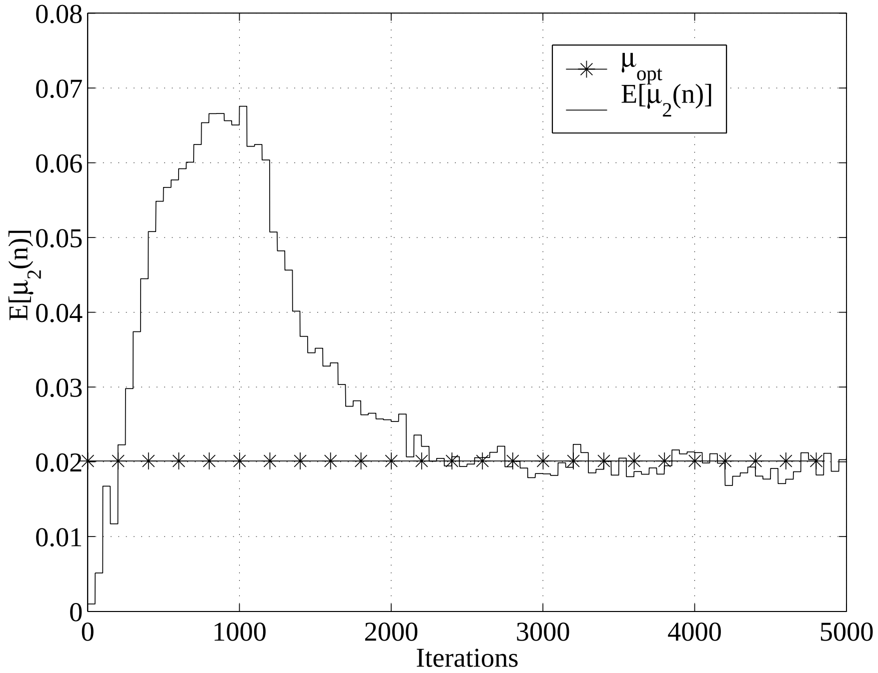

Figure 3.12: Step-size behavior during adaptation.

step-size $\mu_2(n)$ the steady-state MSE was found to be $J_{st_2}(n) = -25.0490$ dB. Clearly there is a reduction on the steady-state MSE when a step-size close to the optimum is used.

3.4 The Scrambled Least Mean Squared algorithm

The LMS algorithm is widely utilized in adaptive filtering for several reasons. Its principal characteristics responsible for attracting the users are low computational complexity, clear convergence analysis in stationary environment, unbiased convergence in the mean to the Wiener solution, stable behavior when is implemented with finite-precision arithmetic, it is straightforward to setup and there is a single parameter to be pre-defined [25], [41]. However, the convergence speed of the LMS algorithm depends on the eigenvalue spread of the input autocorrelation matrix $\mathbf{R}$, as we have seen in the first chapter of this thesis. For highly correlated inputs, the eigenvalue spread of $\mathbf{R}$ is high leading to a slow convergence of the LMS. During last decades there has been a large interest in adaptive filtering community to improve LMS convergence properties without affecting too much computational complexity. One important class of LMS like adaptive algorithms is the class of variable step-size LMS algorithms and some of its components were described also in the first chapter of the thesis. The main idea of the VSSLMS algorithms is to use a time-variable step-size that has large values when the algorithm is far from the optimum, to increase the convergence speed, and smaller values near the steady-state to obtain a small missadjustment. Although, the VSSLMS algorithms have shown improved speed of convergence for uncorrelated input signals their behavior for highly correlated signals is still poor as we have illustrated in the simulations shown in Chapter 3. Another attempt to improve the convergence speed and the steady-state misadjustment of the LMS is the cost function adaptation [71], [72], which does not make the subject of this thesis.

An alternative way to increase the convergence speed of the LMS is to modify not only the coefficients update equation (by implementing a variable step-size or cost function adaptation), but to change also the statistics of the input signal, such that the input autocorrelation matrix will be as close as possible to the identity matrix. When the input autocorrelation matrix $\mathbf{R}$ is the identity matrix (or near identity matrix) all the convergence modes of the adaptive filter are equally excited and the convergence speed is improved. The algorithms that uses an orthogonal transform to diagonalize the matrix $\mathbf{R}$ belongs to the class of the Transform Domain LMS adaptive algorithms, such as those presented earlier in this chapter.

In the communication community the technique of scrambled transmission is well known. When a secure communication is needed, the transmitted signal is transformed in such a way that its information content is unintelligible to a third part and this transformation can be made by means of a scrambling device. The main classes of scrambling devices are the digital scrambler and the analog scrambler. The advantage of the digital scrambler is its higher degree of security comparing with the analog counterpart. Although the digital scrambler offers a higher degree of security its main drawback is the fact that the resulting waveform (the scrambled signal) occupies a much higher bandwidth than the baseband unscrambled signal.

While scrambling was initially introduced for the reason of securing the data transmission, in digital communication systems it also provides a source of pseudodata for adjustment of the timing and Automatic Gain Control (AGC). Consequently, various subsystems in data communication systems, such as equalizers and echo canceler, work better with uncorrelated input sequences. Scrambling is also a way for whitening a correlated signal, such that the convergence speed of the adaptive filters operating with scrambled signals is improved [3]. The scrambled sequence must be unscrambled at the receiver in order to preserve overall bit sequence transparency.

From the above considerations one can say that the transformation of the input signal

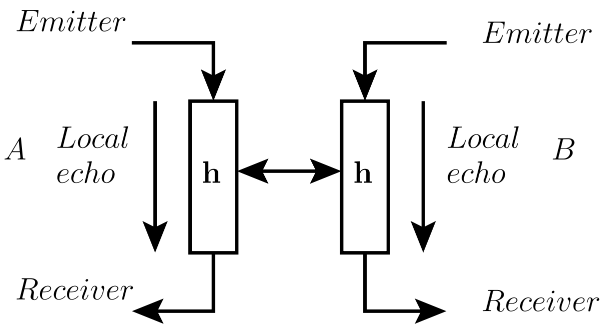

Figure 3.13: A full duplex speech communication channel.

by means of an orthogonal transform and by means of a scrambling device represents two ways to improve the convergence speed of the adaptive filters. The question is which technique is of most interest in practical implementations. In this section, we focus on the application of Transform Domain LMS (TDLMS) and Scrambled LMS (SCLMS) for the problem of transmission of digital data through a telephone channel. Although, the Scrambled LMS does not use an orthogonal transformation at the input, we can consider that it is based on the same principle of whitening the input sequence as the transform domain LMS, therefore we have include this discussion in the present chapter.

3.4.1 Problem formulation and theoretical background

In this section, we study the problem of full-duplex digital transmission over a telephone line as depicted in Fig. 3.13. Ideally, all the energy of the transmitted signal from the transmitter A has to be received by the receiver B and vice-versa. Since the hybrid terminations are not ideal, a small part of the transmitted data signal goes to the local receiver (through the local echo path) and disturb the communication. This signal is called the local echo. Another source of an echo signal represents the reflected signal due to the imperfect impedance adaptation at the ends of the telephone line. Here we concentrate only on the local echo cancellation problem. The block diagram for echo cancellation at emitter A is depicted in Fig. 3.14.

The main problem in the case of local echo cancellation is to approximate the transfer function of the local echo path $\mathbf{h}$ using an adaptive FIR filter $\widehat{\mathbf{h}}(n)$ and then to subtract the estimated echo $\widehat{y}(n)$ from the returning echo $y(n)$, such that the resulting echo is minimized.

When the LMS algorithm is used to modify the coefficients of the adaptive filter $\widehat{\mathbf{h}}(n)$,

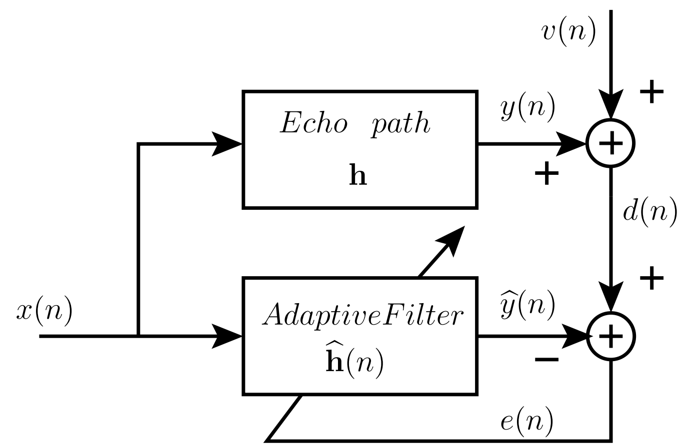

Figure 3.14: Adaptive local echo cancellation block diagram.

the update equation is given by:

$$\widehat{\mathbf{h}}(n+1) = \widehat{\mathbf{h}}(n) + \mu e(n)\mathbf{x}(n), \tag{3.79}$$

where $\widehat{\mathbf{h}}(n) = \left[\widehat{h}_1(n), \widehat{h}_2(n), \ldots, \widehat{h}_N(n)\right]^t$ is the $N \times 1$ vector of the adaptive filter coefficients, $\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^t$ is the $N \times 1$ vector containing the past N samples of the input sequence $x(n)$, μ is a constant step-size that controls the convergence speed and ensures the stability of the algorithm and $e(n)$ is the output error which is computed as follows:

$$e(n) = y(n) + v(n) - \widehat{y}(n), \tag{3.80}$$

with $y(n) = \mathbf{h}^t\mathbf{x}(n)$, $\widehat{y}(n) = \widehat{\mathbf{h}}^t(n)\mathbf{x}(n)$ and $v(n)$ being the output of the echo path (local echo), the output of the adaptive filter (estimated local echo) and the output noise, respectively. Usually, the sequence $v(n)$ contains the transmitted signal from emitter B plus the reflected echo and the channel noise. For the simplicity of exposition and without loss of generality we consider here that the reflected signal and the transmitted signal from emitter B are zero. This corresponds to the situation when the transmission is done just from user A to user B and the hybrid connections at the receiver are perfect.

The convergence speed of the LMS algorithm is governed by the eigenvalues of the input autocorrelation matrix $\mathbf{R} = E\{\mathbf{x}(n)\mathbf{x}^t(n)\}$. If some of the eigenvalues of $\mathbf{R}$ are very small then the convergence of the LMS in the direction of these eigenvalues will be slow resulting in a slow overall convergence of the algorithm. A good measure of the convergence speed of the LMS is the eigenvalue spread $\gamma(\mathbf{R})$ of the input autocorrelation matrix $\mathbf{R}$, computed as the ratio between the largest eigenvalue to the minimum eigenvalue of $\mathbf{R}$. For high values of $\gamma(\mathbf{R})$, the LMS has slow convergence. If the input signal $x(n)$ is uncorrelated then the matrix $\mathbf{R}$ equal the identity matrix and has equal eigenvalues resulting in $\gamma(\mathbf{R}) = 1$. For correlated input sequences the eigenvalues of $\mathbf{R}$ are not equal and $\gamma(\mathbf{R})$ might be much larger than unity. In order to speed up the convergence of the LMS when operating with correlated signals the class of Transform Domain LMS was introduced.

3.4.2 Analysis of the LMS algorithm for digital data transmission

We study here the behavior of the LMS adaptive algorithm in the special case when the transmitted sequence is constant with all its samples equal to +1. The input autocorrelation matrix for this special case can be written as follows:

$$\mathbf{R} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 \end{bmatrix} \tag{3.81}$$

which does not admit an inverse.

The system of equations used to compute the Wiener solution can be written as follows:

$$\mathbf{R}\mathbf{h}_o = \mathbf{R}\mathbf{h}, \tag{3.82}$$

where $\mathbf{h}_o$ is the vector of the optimum coefficients and $\mathbf{h}$ is the vector of the echo path.

Since the autocorrelation matrix does not admit an inverse it is not possible to premultiply (3.82) by $\mathbf{R}^{-1}$ to obtain the optimum coefficients. Moreover, taking into account the structure of matrix $\mathbf{R}$ shown in (3.81), the system of N equations in (3.82) reduces to a single equation:

$$h_{o_1} + h_{o_2} + \dots + h_{o_N} = h_1 + h_2 + \dots + h_N \tag{3.83}$$

Clearly equation (3.83) has multiple solutions which suggest that the LMS does not have a unique convergence point. If we take into account the update equation of the adaptive filter coefficients (3.79) we can see that all coefficients are updated by the same quantity since the elements of the vector $\mathbf{x}(n)$ are all equal. It follows from this fact that the coefficients of the adaptive filter are all equal at every time instant, provided that they are initialized with the same value[17]. Using this observation and (3.83) we can find the convergence point of the LMS algorithm as follows:

$$h_{o_1} = h_{o_2} = \cdots = h_{o_N} = \frac{1}{N}\sum_{i=1}^{N}(h_1 + h_2 + \cdots + h_N) \tag{3.84}$$

The steady-state MSE can be approximated by the well known expression:

$$J_{st} = J_{min} + \frac{\mu J_{min}}{2} tr\left[\mathbf{R}\right] = J_{min}\left(1 + \frac{\mu N}{2}\right), \tag{3.85}$$

For this special case, the matrix $\mathbf{R}$ has $N-1$ eigenvalues equal to zero and one eigenvalue equal to N and the eigenvalue spread of $\mathbf{R}$ equal infinity. This fact suggests that the algorithm might not be convergent. Actually, this is not true due to the fact that for this case of equal inputs, the system is equivalent to the one in which the echo path have just one coefficient equal to the sum $H = \sum_{i=1}^{N} h_i$ and also the adaptive filter have just one coefficient, which converges to H. Due to this fact the convergence speed is expected to be N times faster when the inputs are equal than in the case of non-equal inputs.

The minimum MSE can be computed as in the sequel:

$$J_{min} = E\left\{e_o^2(n)\right\} = E\left\{(y(n) - \widehat{y}_o(n) + v(n))^2\right\} = \sigma_v^2 + E\left\{(y(n) - \widehat{y}_o(n))^2\right\} \tag{3.86}$$

where $\widehat{y}_o(n)$ is the optimum output obtained in the case of perfect adaptation and it can be written as:

$$\widehat{y}_o(n) = \sum_{i=1}^{N} x(n-i+1)h_{o_i} = \sum_{i=1}^{N} h_{o_i} \tag{3.87}$$

where in (3.87) we have taken into account that $x(n) = 1$ at each time instant n.

Replacing (3.87) in (3.86) and using (3.84), the minimum MSE at the output of the adaptive filter equals $J_{min} = \sigma_v^2$ and the steady-state MSE is expressed by the following analytical result:

$$J_{st} = \sigma_v^2\left(1 + \frac{\mu N}{2}\right). \tag{3.88}$$

In conclusion, the convergence point of the LMS when operating with constant input sequences is different than the convergence point when the inputs are not equal. This situation is actually equivalent with the one in which the echo path and the adaptive filter have just one coefficient. However, the minimum and the steady-state MSE for equal and non-equal inputs are the same.

[17] If the coefficients are initialized with different values the convergence point is different.

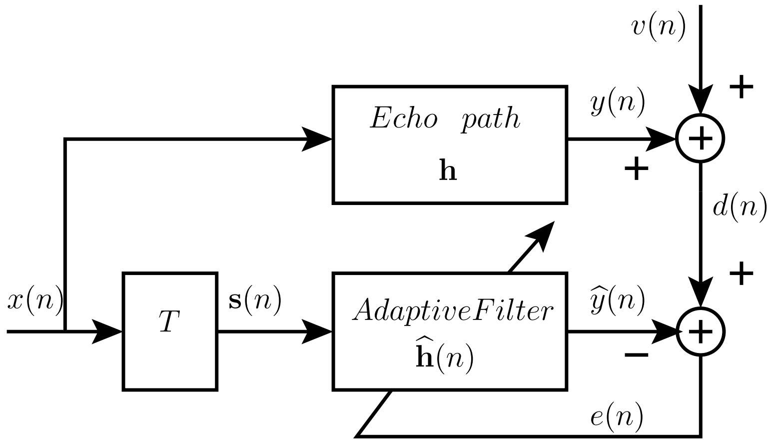

Figure 3.15: The block diagram of the adaptive echo cancellation in transform domain.

3.4.3 Analysis of the Transform Domain LMS algorithm for digital data transmission

To implement the Transform Domain LMS, the block diagram from Fig. 3.14 is modified as shown in Fig. 3.15, by the introduction of the block denoted as $\mathbf{T}$ which represents the orthogonal transformation applied to the input signal $x(n)$. The coefficients update formula becomes:

$$\widehat{\mathbf{h}}(n+1) = \widehat{\mathbf{h}}(n) + \mu\mathbf{\Gamma}^{-1}(n)e(n)\mathbf{s}(n), \tag{3.89}$$

where $\mathbf{s}(n) = \mathbf{T}^t\mathbf{x}(n)$ is the vector of the transform coefficients, $\mathbf{T}$ represents the matrix of the orthogonal transformation, $\mathbf{\Gamma}(n)$ is a diagonal matrix having on the main diagonal the power estimates of the transform coefficients and $e(n)$ is the output error computed as follows:

$$e(n) = y(n) + v(n) - \widehat{y}(n), \tag{3.90}$$

and $\widehat{y}(n) = \widehat{\mathbf{h}}^t(n)\mathbf{s}(n)$.

The equation which gives the optimum coefficients of the adaptive filter in transform domain was derived in the previous sections at it was found to be:

$$\mathbf{R}_s\mathbf{h}_{T_o} = \mathbf{R}_s\mathbf{T}\mathbf{h}. \tag{3.91}$$

Usually, the equation (3.91) is pre-multiplied with $\mathbf{R}_s^{-1}$ and the optimum solution is $\mathbf{h}_{opt} = \mathbf{T}\mathbf{h}$. In the special case, when the digital data transmitted through the communication channel represents a long string of constant values (for instance when the transmitter sends just +1s), the matrix $\mathbf{R}_s$ has a special structure and it does not admit an inverse. If the input sequence $x(n)$ consists of a long string of +1's, then the vector $\mathbf{x}(n)$ will have all its elements equal to +1s and the transformed vector $\mathbf{s}(n)$ will have the first element non-zero and the other elements equal to zero. Therefore, in this special case the structure of the matrix $\mathbf{R}_s$ is expressed by:

$$\mathbf{R}_s = \begin{bmatrix} r_{11} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{bmatrix} \tag{3.92}$$

Using (3.92) in (3.91) we can conclude that the first element of the vector $\widehat{\mathbf{h}}_{T_o}$ is equal to the first element of the vector $\mathbf{Th}$. Moreover, from the update equation (3.89) we can see that just the first element of the vector $\mathbf{s}(n)$ is non-zero which implies that just first element of $\widehat{\mathbf{h}}(n)$ is adapted and the others remain unchanged. If the vector $\widehat{\mathbf{h}}(n)$ is initialized by zeros, then the optimum vector will be:

$$\widehat{\mathbf{h}}_{T_o} = [\mathbf{Th}_1, 0, \dots, 0]^t \,. \tag{3.93}$$

where by $\mathbf{Th}_1$ we denoted the first element of the vector $\mathbf{Th}$.

The steady-state MSE can be obtained following a procedure similar to the one in Section 3.2 of this thesis. Finally, the following analytical expression gives the value of the steady-state MSE:

$$J_{st_{td}} = J_{min} + tr\left[\mathbf{R}_s\mathbf{C}(\infty)\right] \tag{3.94}$$

where $\mathbf{C}(\infty) = \lim_{n\to\infty} E\left\{\Delta\widehat{\mathbf{h}}(n)\Delta\widehat{\mathbf{h}}^t(n)\right\}$, $\Delta\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n) - \mathbf{h}_{T_o}$, $\mathbf{R}_s$ is given in (3.92) and J_{min} is the minimum MSE obtained in the case of perfect adaptation when the coefficients of the adaptive filter equals $\mathbf{h}_{T_o}$.

Taking into account (3.92), we can write (3.94) in the following form:

$$J_{st_{td}} = J_{min} + r_{11}C_{11}(\infty), \tag{3.95}$$

where $C_{11}(n) = \widehat{h}_1(n) - h_{o_1}$ and h_{o_1} is the first element of $\mathbf{h}_{T_o}$.

From the above equations, the analytical expression for the steady-state MSE can be obtained after some simple mathematical manipulations, as follows:

$$J_{st_{td}} = J_{min}\left(1 + \frac{1}{2}\mu\right). \tag{3.96}$$

From (3.96) we can see that when the input sequence is constant, the steady-state MSE does not depend on the length N of the adaptive filter. This is expected since in the adaptation process just one coefficient of the adaptive filter is modified. Actually, this situation is similar with the case in which the adaptive filter have just one coefficient.

The minimum MSE J_{min} can be obtained in the following way:

$$J_{min} = E\left\{e_o^2(n)\right\} = E\left\{[y(n) + v(n) - y_o(n)]^2\right\} \tag{3.97}$$

where $y(n) = \sum_{i=1}^{N} x(n-i+1)h_i$ and the optimum output is given by $y_o(n) = \sum_{i=1}^{N} s_i(n)h_{o_i}$ with $s_i(n)$ being the i^{th} element of the transformed input vector $\mathbf{s}(n)$ and h_{o_i} is the i^{th} element of the optimum vector $\mathbf{h}_{T_o}$.

For a constant input sequence (say $x(n) = 1$) just the first element of $\mathbf{s}(n)$ is non-zero and the others are equal to zero. In this case equation (3.97) simplifies to:

$$J_{min} = E\left\{\left(\sum_{i=1}^{N} h_i - s_1(n)h_{o_1} + v(n)\right)^2\right\} \tag{3.98}$$

When the DCT transform is used, for the decorrelation of the sequence $x(n)$, the input vector $\mathbf{s}(n)$ and the optimum vector $\mathbf{h}_{T_o}$ can be written in the following manner[18]:

$$\mathbf{s}(n) = \begin{bmatrix} \frac{N}{\sqrt{N}} & 0 & \dots & 0 \end{bmatrix}^t \quad \text{and} \quad \mathbf{h}_{T_o} = \begin{bmatrix} \frac{1}{\sqrt{N}}\sum_{i=1}^{N} h_i & 0 & \dots & 0 \end{bmatrix}^t \tag{3.99}$$

Using (3.99) in (3.97), the minimum MSE is expressed by:

$$J_{min} = E\left\{\left(\sum_{i=1}^{N} h_i - \frac{N}{\sqrt{N}}\frac{1}{\sqrt{N}}\sum_{i=1}^{N} h_i + v(n)\right)^2\right\} = E\left\{v^2(n)\right\} = \sigma_v^2 \tag{3.100}$$

Finally, the steady-state MSE is obtained combining (3.96) and (3.100), and it is expressed by:

$$J_{st_{td}} = \sigma_v^2\left(1 + \frac{1}{2}\mu\right). \tag{3.101}$$

It follows from the above analytical result that the transform domain LMS converges to an optimum solution, which is not equal to the coefficients of the echo path when the inputs are all equal. Moreover, the steady-state excess MSE is N times smaller than the steady-state excess MSE of the time domain LMS which uses the same step-size.

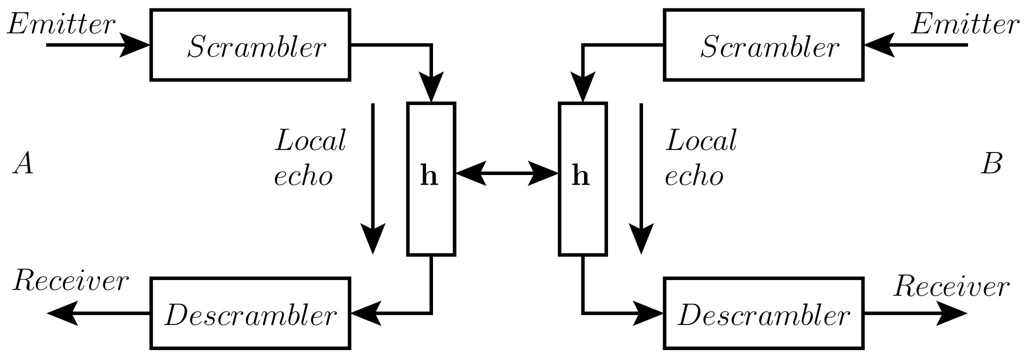

Figure 3.16: A full duplex scrambled communication channel.

3.4.4 The Scrambled LMS

A block diagram of a full duplex scrambled transmission over a telephone line is depicted in Fig. 3.16. We can see that in the case of a scrambled transmission the blocks denoted by *Scrambler* and *Descrambler* are introduced. Their main goal is to secure the data transmission. When a message has to be sent from the user A to the user B, the signal is scrambled prior to the transmission. The user B receives the scrambled signal and using a corresponding descrambler, decodes the transmitted data. Here, we focus on the application of the digital scrambler/descrambler.

From the adaptive filtering point of view an important fact is that the scrambler acts to "whitten" the data to be sent. For modem components, such as the echo canceler and the equalizer, to function properly, a common assumption has to be made that the data is random and i.i.d. (independent and identically distributed). This assumption can be easily violated since long sequences of equal samples can be sent. The scrambler tries to ensure this pretext by making the bit sequence to look random and the input symbol data $x_{sc}(n)$ are uncorrelated.

The block diagram of the local echo cancellation in the case of scrambled transmission is depicted in Fig. 3.17. The difference between Fig. 3.17 and Fig. 3.14 is that the input sequence $x(n)$ is transformed by the block denoted *Scrambler* and the resulting sequence $x_{sc}(n)$ is transmitted over the telephone line. This is why the scrambler device appears also at the input of the echo path in Fig. 3.17. We note that the orthogonal transformation $\mathbf{T}$ appears in Fig. 3.15 just at the input of the adaptive filter.

When the LMS algorithm is used to update the coefficients of the adaptive filter the update equation is exactly the same as (3.79) with the only difference that the input signal

[18]For the case of a constant input sequence $x(n) = 1$.

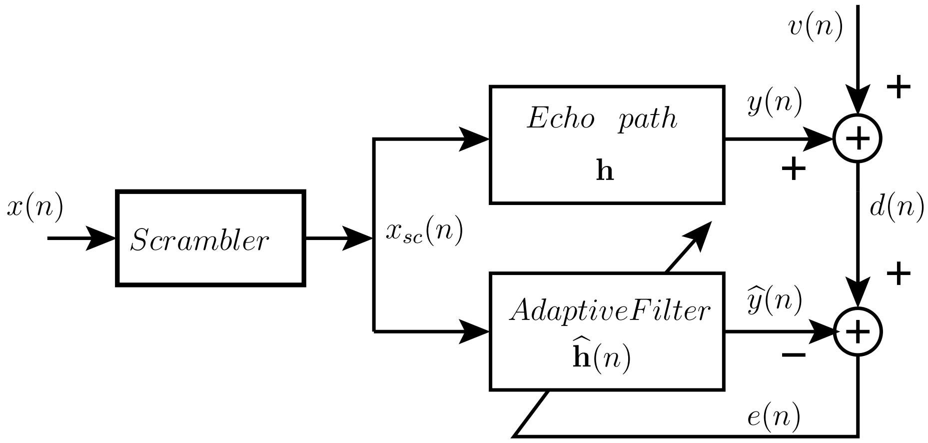

Figure 3.17: Scrambled adaptive local echo cancellation.

into the adaptive filter and into the echo path is now the scrambled sequence $x_{sc}(n)$.

It is easy to show that the optimum solution when the Scrambled LMS algorithm is used to update the coefficients of the adaptive filter is $\mathbf{h}_{sc_o} = \mathbf{h}$ and the steady-state MSE can be expressed by[19]:

$$J_{st_{sc}} = \sigma_v^2 + \frac{1}{2}\mu tr\left[\mathbf{R}_{sc}\right]\sigma_v^2. \tag{3.102}$$

where $\mathbf{R}_{sc}$ is the autocorrelation matrix of the scrambled sequence $x_{sc}(n)$.

Actually, when the Scrambled LMS is implemented the scrambler does not only permute the input samples but it inserts some -1's between the samples of the input sequence $x(n)$ to obtain the scrambled sequence $x_{sc}(n)$. In this way, the autocorrelation matrix $\mathbf{R}_{sc}$ has the diagonal elements equal to 1 and very small off-diagonal elements. Due to this fact, the analysis of the Scrambled LMS can be made in a similar manner as in Chapter 2.

3.4.5 Comparison between scrambled LMS, transform domain LMS and time domain LMS for echo cancellation

In the system identification applications, such as the echo cancellation, the common way to measure the performance of an adaptive filter is the Normalized Estimation Error

[19]The derivation of the steady-state mean squared error can be made as in Chapter 2 using the assumption that $x_{sc}(n)$ is i.i.d.

(NEE) defined as (see [71], [72]):

$$NEE(n) = 10log_{10}\frac{||\mathbf{h} - \widehat{\mathbf{h}}(n)||^2}{||\mathbf{h}||^2}, \quad \text{for LMS and SCLMS} \tag{3.103}$$

$$NEE(n) = 10log_{10}\frac{||\mathbf{Th} - \widehat{\mathbf{h}}(n)||^2}{||\mathbf{Th}||^2}, \quad \text{for TDLMS} \tag{3.104}$$

At the steady-state, taking into account that $\widehat{\mathbf{h}}(\infty)$ has null elements except the first one, equation (3.104), for the transform domain LMS, becomes:

$$NEE_{td}(\infty) = 10log_{10}\frac{\left(\mathbf{Th}_1 - \widehat{h}_1(\infty)\right)^2 + \sum_{i=2}^{N}\mathbf{Th}_i}{\sum_{i=1}^{N}\mathbf{Th}_i^2}, \tag{3.105}$$

It follows from (3.105) that, even in the case of a perfect adaptation when $\widehat{h}_1(\infty) = \mathbf{Th}_1$, the steady-state level of the NEE cannot be made very small due to the terms $\mathbf{Th}_i$, $i = \overline{2, N}$ which are not vanished by the corresponding adaptive filter coefficients. On the other hand, the steady-state MSE in (3.101) can be reduced to a desired level by decreasing the value of the step-size μ.

A similar discussion can be made for the LMS algorithm since the coefficients of the adaptive filter do not converge to the coefficients of the echo path but they are all equal to the average of h_i[20]. As a consequence, the steady-state NEE cannot be made very small decreasing the step-size μ. The steady-state MSE on the other hand is proportional with the step-size and the length of the adaptive filter as seen in (3.88).

Due to the fact that the optimum coefficients of the Scrambled LMS equal the coefficients of the echo path, the steady-state NEE and the steady-state MSE decreases when the step-size is decreased.

Since in the case of echo cancellation the interest is to minimize the output error between the echo $y(n)$ and its estimate $\widehat{y}(n)$, the MSE is the best choice to measure the performances of the TDLMS algorithm. Actually, in echo cancellation application the interest is not to identify the echo path but to reduce the local echo. As a consequence, the best measure to compare two algorithms (for instance LMS, transform domain LMS and scrambled LMS) is to compare their output MSEs.

[20]This is true when the coefficients of the LMS are initialized with zeros.

3.4.6 Simulations and results

Here we study the performances, in terms of MSE and mean squared coefficient error of the time domain LMS, scrambled LMS and the transform domain LMS in the echo cancellation framework for digital data transmission. The length of the echo path and of the adaptive filters were $N = 9$ in all experiments. In order to illustrate the analytical results from the previous section, two different experiments were performed. In the first experiment, we have studied an extreme case where the input sequence was a constant signal, whereas in the second case we tested the three algorithms when the input sequence was a binary sequence with elements from the set $\{-1; 1\}$. However the samples of the input sequence in the second experiment were correlated in the sense that long strings of -1's and $+1$'s were included in $x(n)$. Although a constant input sequence as in the first experiment is unlikely to appear in practice, we have chosen to include here this example in order to get a more clear insight about the adaptation mechanism for these three algorithms.

First experiment: The block diagram used to implement the LMS, SCLMS and TDLMS algorithms are those depicted in Fig. 3.14, Fig. 3.15 and Fig. 3.17 respectively. The input sequence $x(n)$ have a constant level of $+1$. The step-sizes used in the LMS and SCLMS were made equal whereas for the TDLMS we have used a step-size that is 9 times larger[21].

The values of the first coefficients of the TDLMS and **Th**, during the adaptation, are shown in Fig. 3.18. We can see that $\widehat{h}_1(n)$ converges to the first element of the vector **Th** and the other coefficients of the TDLMS were zero, which is in agreement with the theoretical results.

The value of the first coefficient during the adaptation together with the average of the coefficients of the echo path are shown in Fig. 3.19 for the LMS algorithm. The plotted learning curves show a good agreement with the analytical result of (3.84) (the other coefficients of the adaptive filter were equal to the first coefficient).

In Fig. 3.20 and Fig. 3.21 the second and the fifth coefficients of the adaptive filter are plotted for the scrambled LMS together with the corresponding coefficients of the echo path model. Also, these figures have shown a good agreement with the theory (we see that the coefficients of the SCLMS approximate the coefficients of the echo path $\mathbf{h}$).

In Fig. 3.22, the Normalized Estimation Error (see (3.103) and (3.104)) is plotted for the TDLMS and SCLMS. We can see from this figure, that the steady-state level of the NEE in the case of TDLMS is higher than in the case of SCLMS which is due to

[21] This was suggested by the expression of the misadjustment in (3.96).

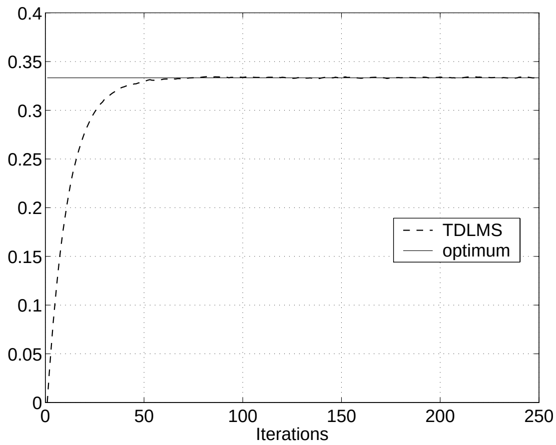

Figure 3.18: The behavior of the first coefficient of the TDLMS and the first coefficient of **Th** during the adaptation in the first experiment.

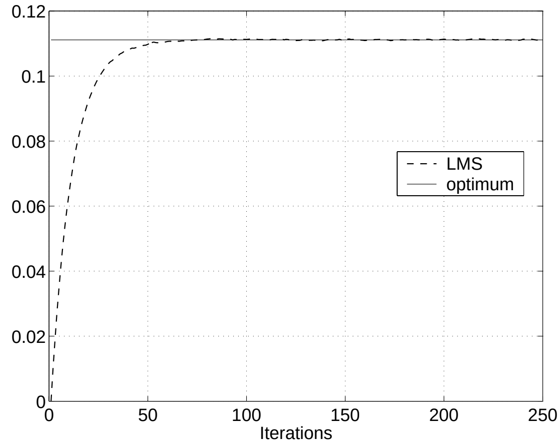

Figure 3.19: The behavior of the first coefficient of the LMS and the average of the coefficients of **h** during the adaptation in the first experiment.

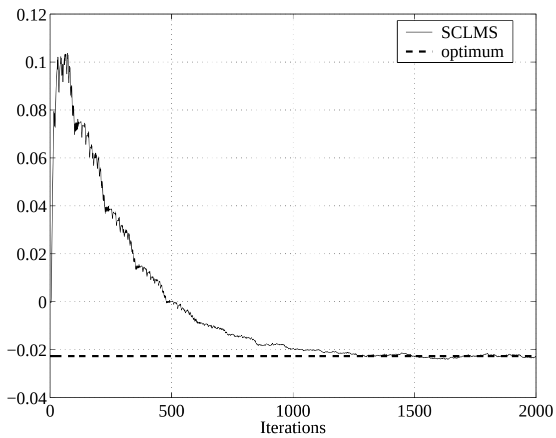

Figure 3.20: The behavior of the second coefficient of the SCLMS and the second coefficient of the echo path model in the first experiment.

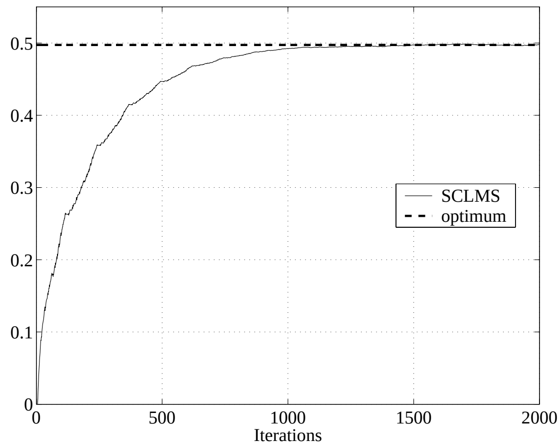

Figure 3.21: The behavior of the fifth coefficient of the SCLMS and the fifth coefficient of the echo path model in the first experiment.

the fact that some coefficients of the adaptive filter are not updated. The reason of this phenomenon is the fact that the transformed input vector $\mathbf{s}(n)$ have just one non-zero element.

Finally, the excess MSE for the LMS, SCLMS and TDLMS are shown in Fig. 3.23,

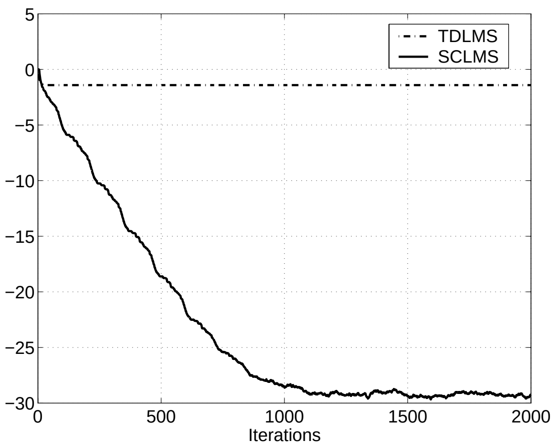

Figure 3.22: Normalized estimation error for the SCLMS and TDLMS in the first experiment.

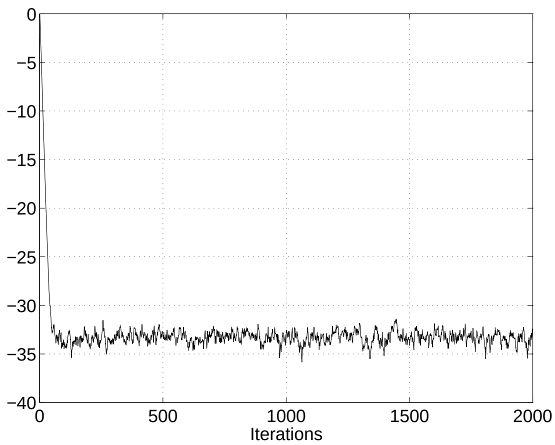

Figure 3.23: The excess MSE for the LMS in the first experiment.

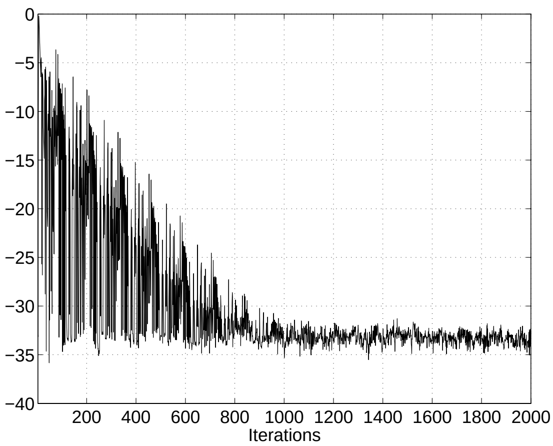

Figure 3.24: The excess MSE for the SCLMS in the first experiment.

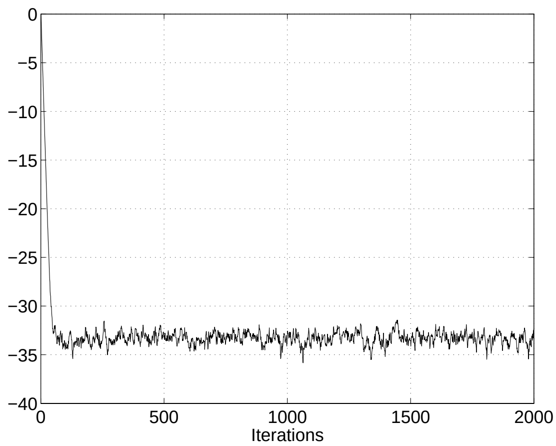

Figure 3.25: The excess MSE of the TDLMS for the first experiment.

Fig. 3.24 and Fig. 3.25 respectively. We can see that the TDLMS performs better than the SCLMS in terms of MSE, for this particular input sequence. The step-size of the TDLMS was 9 times larger than the step-sizes of the LMS and SCLMS as suggested by (3.96) and (3.102), in order to obtain the same level of the steady-state missadjustment. We can see that both filters converge to the same level of the MSE although they have different step-sizes, which proves the validity of (3.96).

Second experiment: The same block diagrams were used to implement the three compared algorithms. The input sequence was bipolar with elements in $\{-1, +1\}$ and it

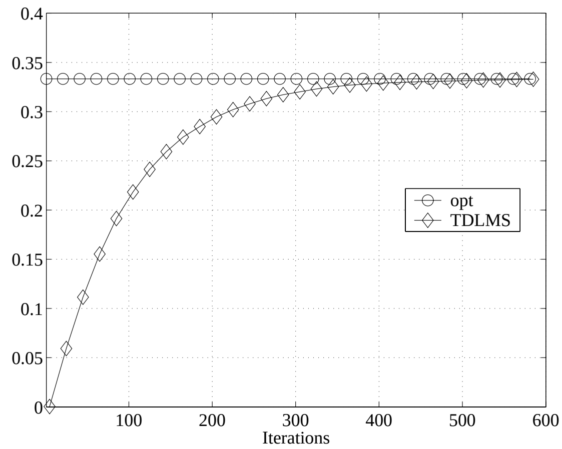

Figure 3.26: The first coefficient of the TDLMS and its optimum value during the adaptation (second experiment).

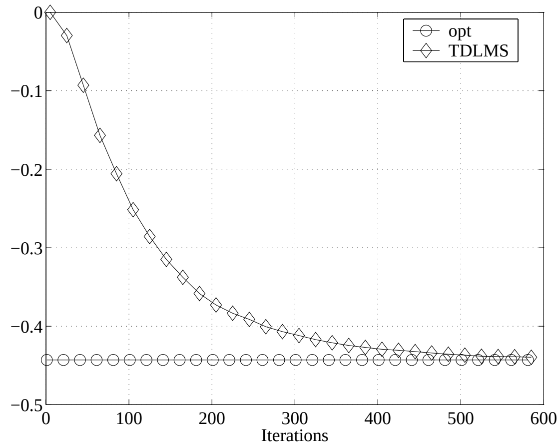

Figure 3.27: The third coefficient of the TDLMS and its optimum value during the adaptation (second experiment).

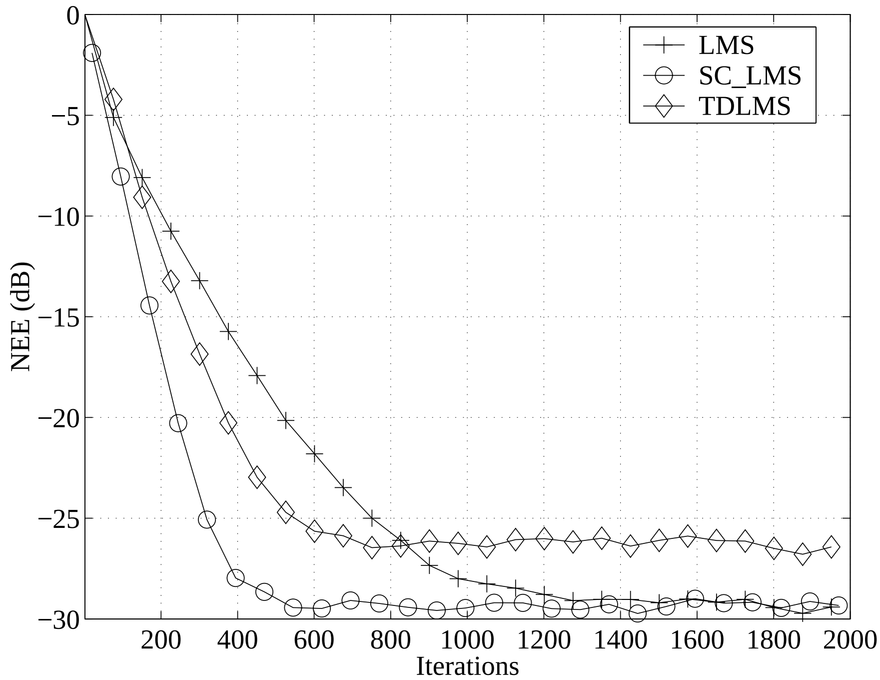

Figure 3.28: Normalized estimation error for the LMS, SCLMS and TDLMS in the second experiment.

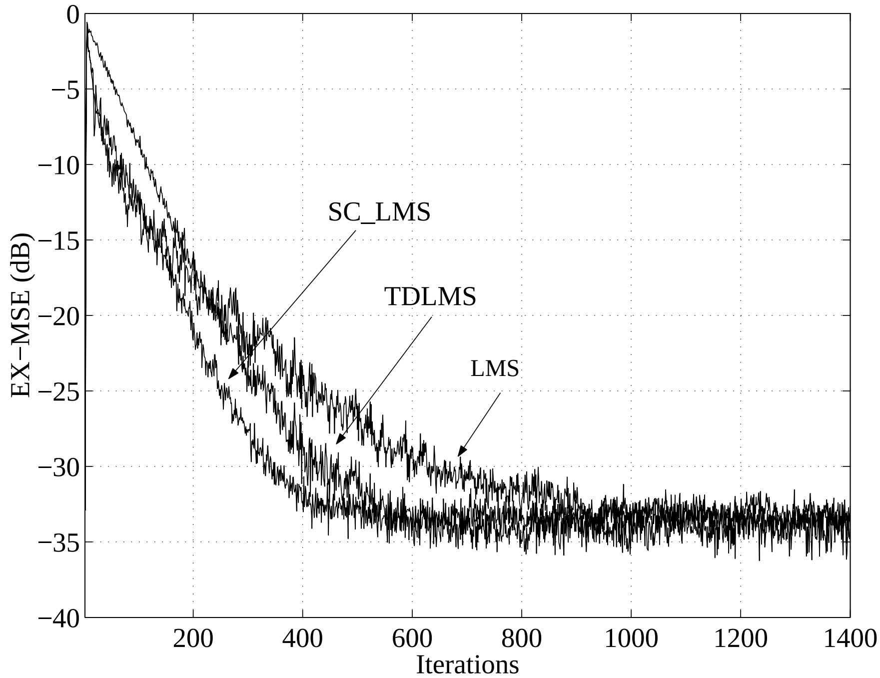

Figure 3.29: The excess MSE for the LMS, SCLMS and TDLMS in the second experiment.

contains long strings of consecutive −1's and +1's. In this case, the input autocorrelation matrix is not diagonal and the eigenvalue spread was larger than unity.

The behavior of the first and the third coefficient of the TDLMS together with the first and the third coefficient of **Th** are plotted in Fig. 3.26 and Fig. 3.27 respectively. We can see that in this case the coefficients of the TDLMS converge near **Th**. However, from the learning curves shown in Fig. 3.28 we can see that also in this case the TDLMS converges to a higher level of the NEE comparing to the LMS and SCLMS. This is due to the fact that during the adaptation the long strings of input samples with constant values are transformed by the orthogonal transform into vectors having just one non-zero coefficient and the other coefficients are zero. Because of this phenomenon, the coefficients of the adaptive filter are not enough times updated resulting into a higher level of the steady-state NEE.

The excess MSEs obtained with the LMS, SCLMS and TDLMS are plotted in Fig. 3.29. We can see that TDLMS and SCLMS converges faster than the LMS for correlated input sequence. This faster convergence was expected since both algorithms uses the decorrelation of the input sequence which increases the convergence speed.

Here we have analyzed the problem of local echo cancellation for digital transmission over a telephone. For this practical application we have studied the behavior of three different adaptive algorithms, the Least Mean Squared (LMS) algorithm, the Scrambled LMS (SCLMS) and the Transform Domain LMS (TDLMS) algorithm.

We have found that in the case of TDLMS the level of the steady-state NEE is higher compared to LMS and SCLMS, when the input sequence contains long strings of consecutive equals samples and it cannot be decreased due to the fact that there are some coefficients that are not enough times updated. However, all algorithms converge to the same level of the steady-state MSE which is of interest in practical applications, when the step-sizes are appropriately chosen.

For input sequences which contains long string of equal samples, the convergence speed of both TDLMS and SCLMS are comparable and higher than the convergence speed of the LMS. This result is expected since it is well known that the LMS converges slower for correlated inputs than other algorithms which perform the decorrelation at the input of the adaptive filter.

Chapter 4

Applications

In this chapter, we discuss about the implementation of various adaptive algorithms introduced in the previous chapters for three practical applications. First, we address the problem of channel equalization for Gaussian and non-Gaussian noise environments. Second the behavior of different adaptive algorithms for the problem of Code-Division Multiple-Access using the Direct-Sequence (DS) spread spectrum signaling is discussed. Finally, the problem of echo cancellation for digital data transmission is addressed. Computer experiments showing the results obtained with the compared algorithms are presented and the advantages and the disadvantages of the implementations are discussed.

Section 4.1 is dedicated to the problem of channel equalization and the algorithms which are implemented for this problem are the order statistics LMS, the plain time domain LMS, the variable step-size LMS in time domain and the transform domain implementations (the plain transform domain LMS and transform domain LMS algorithms with variable step-size). All these algorithms were described in the previous sections of the thesis. We first briefly describe the problem of channel equalization and the block diagrams of the time domain and transform domain implementations are presented. In Section 4.1.1 the order statistic LMS algorithms are implemented in channel equalization framework for non-Gaussian channel noise whereas in Section 4.1.2 the time domain and transform domain algorithms are compared for the situation in which the channel noise is Gaussian distributed.

In Section 4.2 the problem of CDMA multiuser detection is addressed. We shortly describe first the framework in which the adaptive filters are implemented and then the simulations and results obtained with the time domain adaptive algorithms with variable step-size are shown. For this application we have implemented the following adaptive algorithms: the plain LMS, the variable step LMS proposed in [48], the robust variable step

LMS proposed in [1] and the complementary pair variable step LMS algorithm described in Section 2.2.

In the last part of this chapter, an extension of the Variable Length LMS for correlated input sequence is introduced. The proposed algorithm is a combination of the VLLMS algorithm introduced in Chapter 2 and SCLMS presented in Chapter 3. The performances of the new implementation are studied in echo cancellation framework.

4.1 Channel equalization

For the problem of channel equalization, we study the performances of two main classes of adaptive filters namely the time domain implementations and the transform domain implementations. The block diagram implemented for channel equalization in time domain is depicted in Fig. 4.1, where $\mathbf{h}$ represents the time invariant linear communication channel, $v(n)$ represents the channel noise, $\widehat{\mathbf{h}}(n)$ is the vector containing the coefficients of the adaptive filter, $x_{in}(n)$ is the sequence transmitted through the channel and $x_{out}(n)$ the output sequence from the channel.

The noise channel is added at the output sequence $x_{out}(n)$ and the result $x(n)$ represents the input into the adaptive filter. The coefficients of the adaptive filter are updated, such that the MSE at the output of $\widehat{\mathbf{h}}(n)$ is minimized. The error is computed as the difference between the desired sequence $d(n)$ and the output of the adaptive filter and the sequence $d(n)$ is obtained as a delayed version of the transmitted signal $x_{in}(n)$. Actually, in practice the sequence $d(n)$ is stored in the receiver and during the adaptation period the same sequence is transmitted through the channel. During the training period, the sequence $x_{in}(n)$ and its delayed version $d(n)$ does not contain useful information. After the training period the coefficients of the adaptive filter are maintained constant and the sequence which contains the useful information is transmitted. This method is known as the training based channel equalization since the transmition of the useful information is interrupted from time to time in order to perform the equalization of the channel during the training period.

For transform domain implementations the same scheme of training-based channel equalization is implemented as shown in Fig. 4.2. The difference between the two figures consists in the block denoted as $\mathbf{T}$ which appears in Fig. 4.2. This block represents the orthogonal transformation applied at the input of the adaptive filter, such that the output from the channel plus the channel noise $x(n) = x_{out}(n) + v(n)$ is first transformed to $\mathbf{s}(n) = \mathbf{T}\mathbf{x}(n)$ and then applied to the adaptive filter.

In all simulations presented here the transmitted sequence is bipolar with values ran-

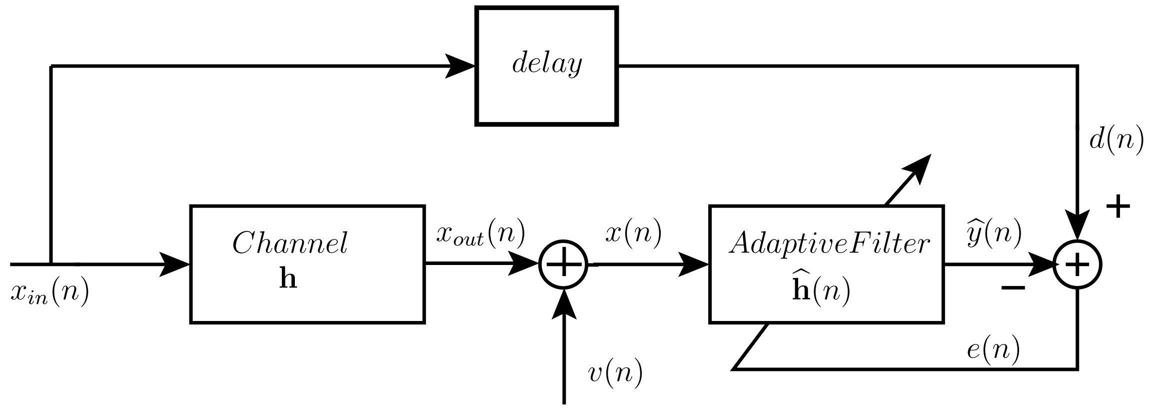

Figure 4.1: The block diagram for channel equalization in time domain.

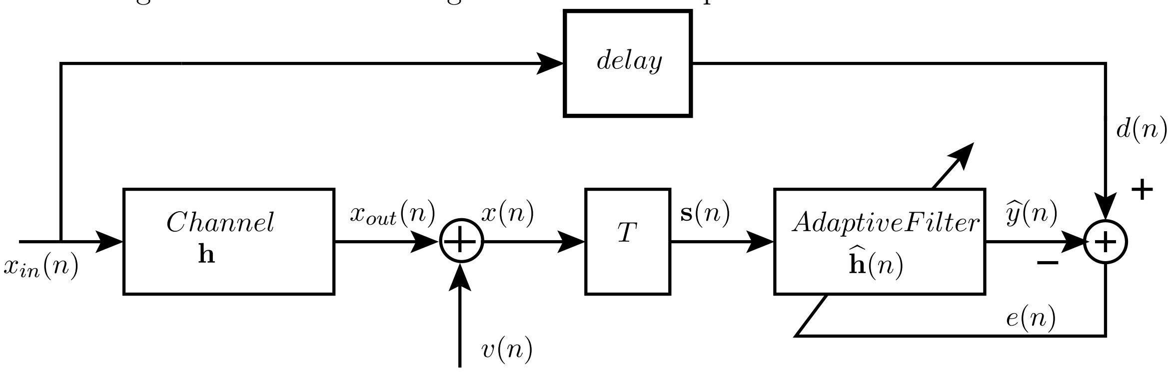

Figure 4.2: The block diagram for channel equalization in transform domain.

domly chosen from $\{-1; +1\}$. Although the transmitted signal $x_{in}(n)$ represents a random sequence, the samples $x_{out}(n)$ at the output of the channel are correlated due to the channel coefficients. Due to this fact the autocorrelation matrix of the sequence $x(n)$ can be in some cases very high and the convergence speed of the time domain implementations is low. This is the reason why we have implemented also the transform domain adaptive filters for this application.

Another situation, which can occur in practice, is when the channel noise $v(n)$ have non-Gaussian distribution. For instance, when the noise distribution is impulsive, the LMS have stability problems. This is due to the fact that the impulses contained in $v(n)$ influences the adaptation of the coefficients through the term $\mathbf{x}(n)e(n)$[1]. For such

[1]We note that in this application, the noise appears just at the input of the adaptive filter. However due to the fact that the desired sequence $d(n)$ is bipolar, a certain degree of impulsivity exists at the output of the adaptive filter as well.

situations the Order Statistics LMS algorithms might be an alternative solution.

Computer experiments, showing the performances of the OSLMS algorithms described in Chapter 2, are presented in the next section, for various noise distributions. For the case of Gaussian noise distribution in Section 4.1.2 the results obtained with the variable step-size LMS algorithms in time and transform domain are presented.

4.1.1 Channel equalization in non-Gaussian noise environments

The block diagram used in the experiments is depicted in Fig. 4.1 and the compared algorithms were the order statistic LMS from Section 2.5. A delayed version of the transmitted sequence $x_{in}(n)$, with an appropriate delay D, is used as the desired signal $d(n) = x_{in}(n-D)$ for the adaptive filters. The length of the channel was $N_{ch} = 11$, the lengths of all the compared adaptive filters were $N = 11$ and the length of the weighting vector ($\mathbf{a}$ and $\mathbf{a}(n)$) for the gradient was $L = 7$ (see Section 2.5).

The distribution of the channel noise $v(n)$ has a generalized exponential density given by:

$$p(r) = k_1 e^{-k_2|r|^\beta}, \quad |r| < \infty, \quad 0 < \beta \leq \infty, \tag{4.1}$$

where k_1 and k_2 are given by:

$$k_1 = \left(\beta k_2^{1/\beta}\right) / 2\Gamma\left(\frac{1}{\beta}\right), k_2 = \left[\frac{\Gamma\left(\frac{3}{\beta}\right)}{\Gamma\left(\frac{1}{\beta}\right)}\right]^{\beta/2} \sigma_v^{-\beta}, \tag{4.2}$$

with Γ being the ordinary gamma function and σ_v^{-2} the standard deviation.

In order to have a fair comparison the step-sizes of all the algorithms were chosen to give comparable convergence speeds and the step-size for the L-LMS algorithm was chosen to be $\tilde{\mu} = 0.1$ which satisfies the stability condition from Section 2.5.

As β in (4.1) increases from a value close to zero, the resulting density varies from highly impulsive to Gaussian and to uniform. However, the gradient has a certain degree of impulsivity also in the case of Gaussian and uniform channel noise due to the desired signal $d(n)$, which is bipolar. Therefore, we expect that the Outer Mean LMS does not give satisfactory results for any considered noise distribution. More than that, the gradient distribution is also influenced by the distribution of the channel noise and input sequence $x(n)$ and the performance of the Median LMS is expected to be also poor. With these observations we can expect that among OSLMS algorithms with fixed weighting coefficients the Trimmed Mean LMS would have the best performance. Note, that an

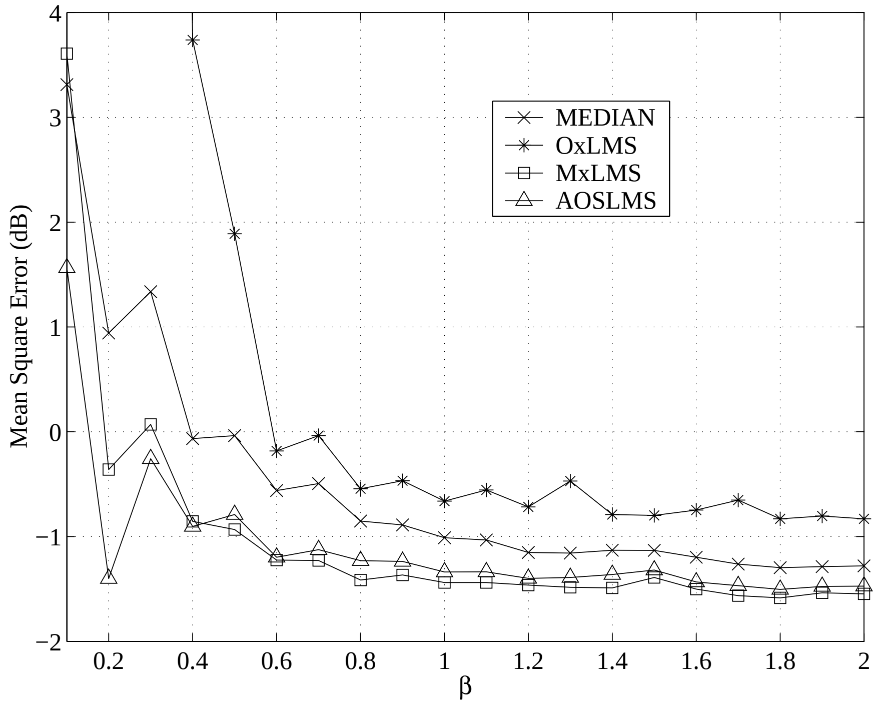

Figure 4.3: Steady-state mean squared error for median LMS, MxLMS, OxLMS and AOSLMS for $SNR = 0$ dB.

algorithm similar to the proposed AOSLMS in which not only the trimming coefficient is adapted but also the envelope of the weighting coefficients would give even better results.

The simulations showing the performance of the AOSLMS filter compared with other OSLMS algorithms are given in Fig. 4.3 and Fig. 4.4 for different noise distributions and different signal-to-noise ratio at the input of the adaptive filters. In these figures, the steady-state MSE for each compared algorithms are plotted. The results shown in Fig. 4.3 were obtained for a signal-to-noise ratio of $SNR = 0$ dB, at the output of the channel, whereas in Fig. 4.4 the signal to noise ratio was $SNR = 10$ dB. From these figures, we can see that the algorithm which uses an adaptive filter to smooth the gradient gives better results for almost all considered noise distributions and this is in agreement with the theoretical considerations from Section 2.5.

Here, we have applied a new AOSLMS adaptive filter to the problem of channel equalization for non-Gaussian noise environments. The approach of channel equalization differs from that of the system identification, in which the impulsive nature of the gradient is mainly given by the noise present in the system. Usually, in the case of channel equal-

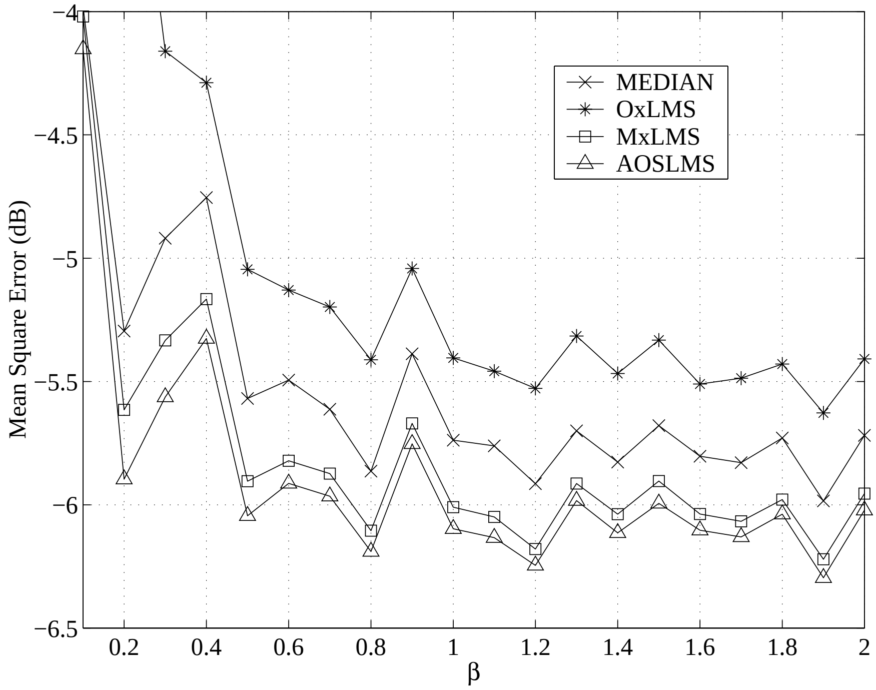

Figure 4.4: Steady-state mean squared error for median LMS, MxLMS, OxLMS and AOSLMS for $SNR = 10$ dB.

ization it is difficult to predict the distribution of the gradient and hence the optimal weighting vector to smooth the gradient. In such cases, the proposed AOSLMS algorithm would give better results due to its ability to adapt the weighting coefficients to the unknown gradient distribution.

4.1.2 Channel equalization using variable step-size adaptive algorithms

The algorithms compared here, in channel equalization framework, are the time domain and transform domain algorithms with fixed and variable step-sizes that were described in Chapter 2 and Chapter 3. The block diagram for the transform domain implementations is depicted in Fig 4.2 and for the time domain implementations in Fig. 4.1. The compared algorithms were: the plain LMS, the Variable Step-Size LMS proposed in [48], the robust variable step LMS (RVSLMS) from [1], the plain TDLMS using the DCT transform, the DCT-LMS using the modified power estimator proposed in [45] and the TDVSLMS from Section 3.2 of this thesis.

The transform used at the input of all transform domain implementations was the

Algorithm	LMS	VSSLMS	RVSLMS	TDLMS	DCT-LMS	TDVSLMS
MSE	0.0155	0.0155	0.0157	0.0153	0.0161	0.0153

Table 4.1: The steady-state mean squared error for the compared algorithms in the channel equalization framework.

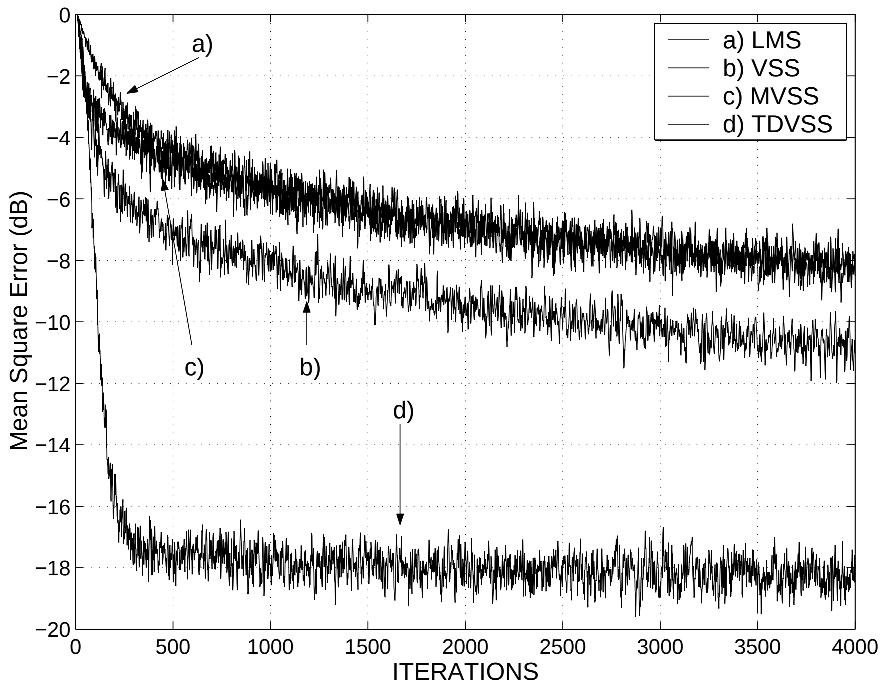

Figure 4.5: Mean squared error of the LMS, VSLMS, RVSLMS and TDVSLMS implemented for channel equalization.

DCT. The signal to noise ratio at the output of the channel was $SNR = 30$ dB and the parameters of all the compared algorithms were chosen, such that they have comparable steady-state MSE. In order to setup the parameters of the implemented algorithms, we have followed the guidelines presented in Section 3.2.

The plotted learning curves were obtained by averaging the squared errors of 200 independent runs each run containing a number of 15×10^4 iterations. The steady-state mean squared errors obtained experimentally are given in Table 4.1 and these values were computed averaging the last 1000 values from the corresponding MSE's.

All the adaptive filters have the same number of coefficients $N = 17$ and the trans-

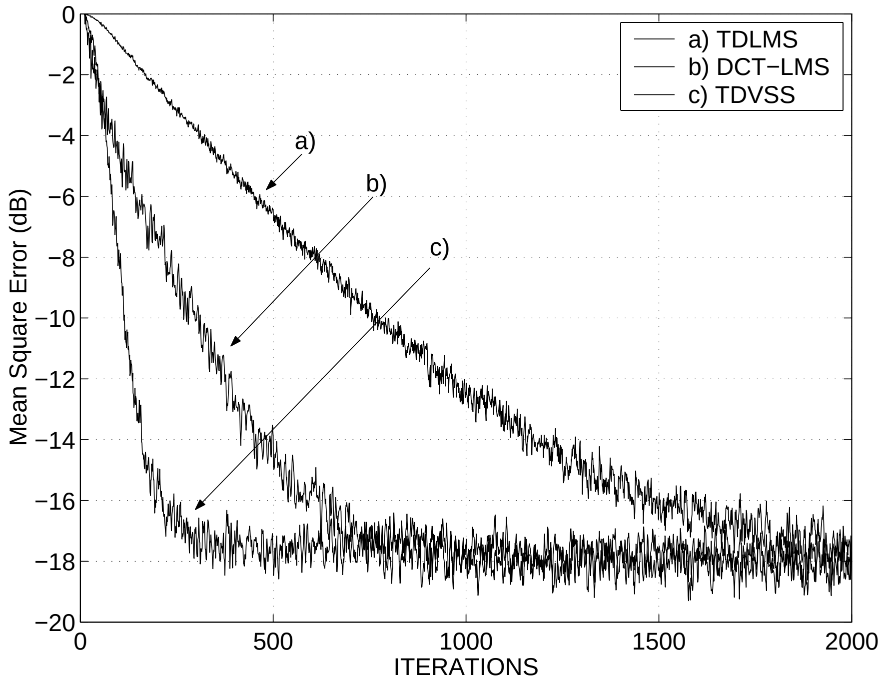

Figure 4.6: Mean squared error of the TDLMS, DCT-LMS and TDVSLMS implemented for channel equalization.

mission channel has three coefficients given by the following model (see [45]):

$$h_i = \begin{cases} \frac{1}{2}\left[1 + \cos\left\{\frac{2\pi}{W}(i-2)\right\}\right], & \text{if} \quad i = 1, 2, 3, \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

In Fig. 4.5, the learning curves obtained for the LMS, VSSLMS, RVSLMS and TD-VSLMS algorithms are depicted. In order to have a more clear representation, just the first 4000 samples of each learning curve are plotted. We can see from this figure that the TDVSLMS clearly has higher convergence speed than the time domain implementations. This is expected since the input signal into the adaptive filter was highly correlated due to the coefficient $W = 3.75$ in (4.3).

A more interesting result is presented in Fig. 4.6, where the TDVSLMS algorithm is compared with the plain TDLMS and the DCT-LMS using the modified power estimator. We can see from this figure that the TDVSLMS is the fastest algorithm among these three transform domain implementations.

The transform domain variable step-size LMS algorithm described in Section 3.2, applied to the problem of channel equalization, shows better convergence speed compared to other well known time domain and also transform domain algorithms. We have seen in Chapter 3 that the computational complexity of the TDVSLMS algorithm is comparable with that of the plain TDLMS, which makes it a very good candidate for practical implementations. Another transform domain algorithm which can be implemented in this framework is the TDCPVSLMS from Section 3.2 of this thesis. Although it is more computational expensive, the setup of its parameters is simpler.

4.2 CDMA multiuser detection

Code Division Multiple Access (CDMA) using the direct sequence (DS) spread-spectrum signaling has been implemented with success in telecommunication applications. Some of the main advantages of the DS/CDMA technique are: the ability of asynchronous operation, a better channel usage compared with other techniques that allows a single user to be transmitted over the channel at a certain time and its ability to operate in the presence of narrow band communication systems. When a given user is demodulated in a DS/CDMA system, two types of interferences must be minimized, namely the wide band Multiple Access Interference (MAI) and the Narrow Band Interference (NBI), as well as the channel noise. The MAI is caused by other spread spectrum users into the channel while the NBI interference is caused by other conventional communication systems.

Among other demodulation techniques, the adaptive methods have been successfully applied to reduce both the MAI and NBI interferences in DS/CDMA systems. When the spreading code and the channel parameters of the desired user are known or can be estimated, the blind adaptive detectors can be easily used [36], [37], [42], [47], [67], [68], [81], [75], whereas in absence of these pieces of information the trained based implementations are preferred [50], [53], [57].

In the trained based systems a known training sequence is transmitted which is used to tune the coefficients of the adaptive filter before the actual data is sent. The well known adaptive algorithm used in both blind and training based demodulators is the LMS algorithm which has the advantage of having a simple implementation and low computational complexity. However, the main disadvantages of the LMS algorithm are its slow convergence when operating with highly correlated input signals and the trade-off between the convergence speed and the output error, as we have pointed out during this thesis [41]. In order to reduce these disadvantages many of its variants where introduced in the open literature, such as the class of Variable Step-Size LMS algorithms.

In this section, we analyze the behavior of different VSSLMS adaptive algorithms for the problem of multiuser detection in a synchronous CDMA system. We show, by means of simulations, that the Complementary Pair Variable Step-Size LMS (CP-VSLMS)

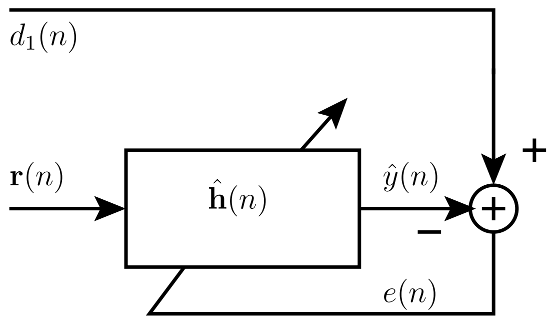

Figure 4.7: Block diagram of an adaptive detector using the LMS algorithm.

adaptive algorithm introduced in Section 2.2 possess a faster convergence speed than other known algorithms, while reducing the trade-off between convergence speed and steady-state output error.

4.2.1 Problem formulation and theoretical background

For the sake of simplicity, we consider a synchronous CDMA system in which a number of K users transmit over a single-path time-invariant channel. The processing gain is denoted by N, the attenuation of each user data are denoted by a_k and the data symbols transmitted by all users are aligned in time.

The received signal, sampled at chip rate, can be written in vector form as follows:

$$\mathbf{r}(n) = \mathbf{SAd}(n) + \mathbf{v}(n), \tag{4.4}$$

where the j^{th} column of $\mathbf{S}$ represents the received spreading code of the j^{th} user, the vector $\mathbf{d}(n) = [d_1(n), \ldots d_K(n)]^T$ contains the data symbols transmitted by all users at the time instant n, the $N \times 1$ vector $\mathbf{v}$ is the sampled channel noise and the $K \times K$ matrix $\mathbf{A}$ is given by:

$$\mathbf{A} = \begin{bmatrix} a_1 & 0 & 0 & \ldots & 0 \\ 0 & a_2 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & a_K \end{bmatrix}$$

Assuming that the desired user is user 1, a block diagram of a trained based detector using the standard LMS adaptive algorithm is depicted in Fig. 4.7, where $\mathbf{r}(n)$ is the input vector described in (4.4), $\hat{\mathbf{h}}(n) = \left[\hat{h}_1(n), \ldots, \hat{h}_N(n)\right]^t$ is the $N \times 1$ vector containing the coefficients of the demodulator, $d_1(n)$ is the known desired sequence that is the same as the data sequence transmitted by the user 1 and $e(n)$ is the output error.

The LMS adaptive algorithm used to train the coefficients of the adaptive filter $\hat{\mathbf{h}}(n)$ can be described by the following steps (see also Section 2.2 of this thesis):

1. Compute the output of the adaptive filter $\hat{\mathbf{h}}(n)$:

$$\widehat{y}(n) = \hat{\mathbf{h}}^t(n)\mathbf{r}(n) = \sum_{i=1}^{N} \hat{h}_i(n) r_i(n), \tag{4.5}$$

   where $r_i(n)$ is the i^{th} element of the vector $\mathbf{r}(n)$ in (4.4).

2. Compute the output error:

$$e(n) = d_1(n) - \widehat{y}(n), \tag{4.6}$$

3. Update the coefficients of the adaptive demodulator:

$$\widehat{\mathbf{h}}(n+1) = \widehat{\mathbf{h}}(n) + \mu e(n)\mathbf{r}(n). \tag{4.7}$$

   where μ is a constant parameter called step-size, which controlls the steady-state error and the convergence speed.

For the training based detector the convergence speed is governed by the eigenvalue spread of the input autocorrelation matrix which is defined as follows:

$$\mathbf{R} = E\left[\mathbf{r}(n)\mathbf{r}^t(n)\right] \;\; = E\left[\mathbf{S}\mathbf{A}\mathbf{d}(n)\mathbf{d}^t(n)\mathbf{A}^t\mathbf{S}^t\right] + E\left[\mathbf{v}(n)\mathbf{v}^t(n)\right], \tag{4.8}$$

where we have assumed that the elements of the vector $\mathbf{v}$ are random zero-mean and independent from $\mathbf{S}$, $\mathbf{A}$ and $\mathbf{d}(n)$.

It is clear from (4.8) that the eigenvalue spread of the input autocorrelation matrix $\mathbf{R}$ can be far from unity and an adaptive demodulator using the standard LMS algorithm will have a very slow convergence. Since in the case of training based detectors, during the adaptation period no data sequences can be transmitted, a slow convergence will decrease also the transmission rate. Therefore, in practical applications, the convergence speed of the detector has to be increased while maintaining a small steady-state error.

Besides a slow convergence, the plain LMS algorithm has also the disadvantage of a trade-off between speed and steady-state output error. Indeed from (4.7) we can see, that in order to obtain a small steady-state error, one has to choose a small step-size, but a small value of μ decreases the speed of convergence of the algorithm.

In the sequel we study, by means of computer experiments, the behavior of the adaptive algorithms described in Section 2.2 of this thesis.

Table 4.2: Steady-state MSE and the parameters for the compared algorithms for CDMA multiuser detection.

Algorithm	Parameters	Steady-State MSE (dB)
LMS	$\mu_{min} = 3 \times 10^{-4}$	-17.4256
LMS	$\mu_{max} = 3 \times 10^{-3}$	-14.0252
CP-VSLMS	$\mu_{min} = 3 \times 10^{-4}$, $\mu_2 = 3 \times 10^{-3}$, $\alpha = 0.9$, $T = 50$	-17.4251
VSSLMS	$\mu_{min} = 3 \times 10^{-4}$, $\mu_{max} = 3 \times 10^{-3}$, $\alpha = 0.9$, $\gamma = 0.002$	-17.2241
RVSSLMS	$\mu_{min} = 3 \times 10^{-4}$, $\mu_{max} = 3 \times 10^{-3}$, $\alpha = 0.97, \beta = 0.99$ $\gamma = 2$	-17.2643

4.2.2 Simulations and results

We compare the performances, in terms of convergence speed and steady-state MSE, of the CP-VSLMS, plain LMS, VSSLMS, and RVSLMS algorithms, in the CDMA multiuser detection framework. The signal model is given in (4.4) and the number of users was $K = 4$ with the first user being the user of interest. The attenuation of the first user was 10 dB below the attenuation of the other three users. The spreading codes were chosen from a set of Gold sequences of length $N = 31$ and the channel noise $v(n)$ was white Gaussian with zero mean and variance $\sigma_v^2 = 10^{-2}$. The transmitted data (the elements of the vector $d(n)$ in (4.4)) were equiprobable bipolar sequences with values in $\{-1, +1\}$.

The parameters of all the tested algorithms are presented in Table 4.2 together with the corresponding values of theirs steady-state MSE. These parameters were chosen to give comparable steady-state MSE for all adaptive filters. One exception is the LMS, implemented with fixed step-size $\mu_{max} = 3 \times 10^{-3}$, which was included for benchmark purposes. The learning curves (the output MSE during the adaptation) for all algorithms are shown in Fig. 4.8, Fig. 4.9, Fig. 4.10, Fig. 4.11 and Fig. 4.12. These results were obtained by averaging a number of 100 runs of length 4×10^4 iterations. From these figures, we can see, that the CP-VSLMS has faster convergence compared with the VSSLMS, RVSLMS and the LMS having a small step-size while their steady-state MSE are comparable.

In order to have a more clear insight of the behavior of the compared algorithms in Fig. 4.13, Fig. 4.14 and Fig. 4.15 the expected value of the step-size during the adaptation for the CP-VSLMS, VSSLMS and RVSLMS respectively are plotted. From these figures, we can see, that the step-size of the CP-VSLMS algorithm has the smallest variations at the steady-state and also its value is very close to $\mu_{min} = 3 \times 10^{-4}$. These results are

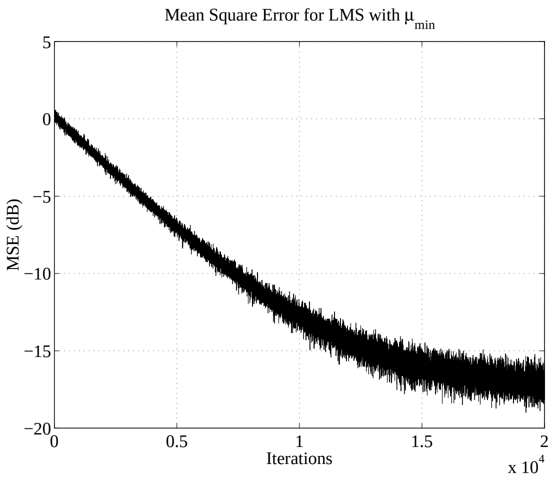

Figure 4.8: Output mean squared error for the LMS with $\mu_{min} = 3 \times 10^{-4}$.

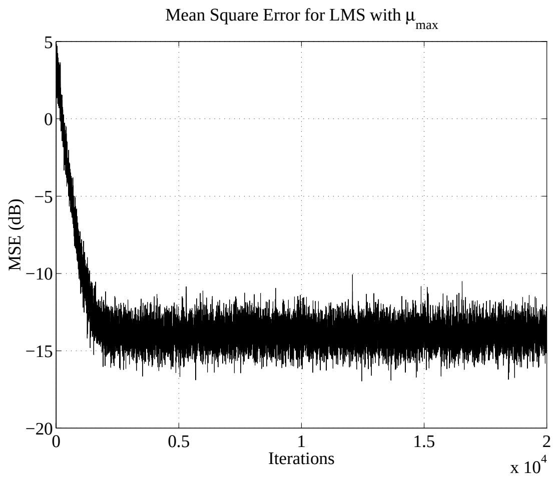

Figure 4.9: Output mean squared error for the LMS with $\mu_{max} = 3 \times 10^{-3}$.

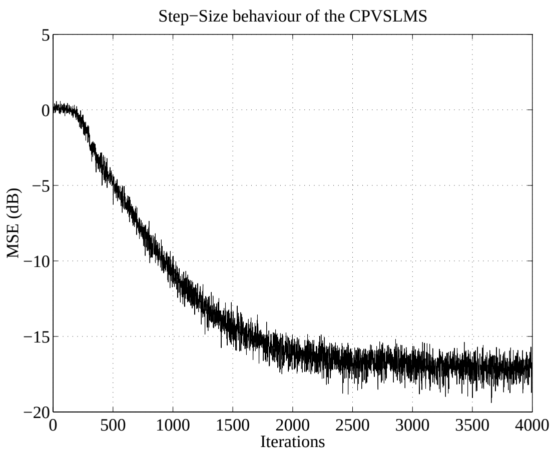

Figure 4.10: Output mean squared error for CP-VSLMS.

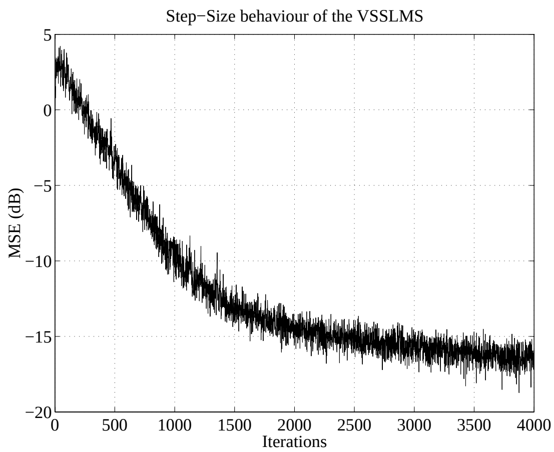

Figure 4.11: Output mean squared error for VSSLMS.

in agreement with our theoretical considerations from section 2.2 that the steady-state misadjustment of the CP-VSLMS is given by μ_{min}.

In Chapter 2, the computational complexity and memory load[2] of the variable step-size algorithms were compared and we have seen that the computational complexity and memory load of the CP-VSLMS algorithm are almost double compared with the other algorithms. However, the benefit of the proposed algorithm is the increased convergence speed and the fact that the dependence between the speed of convergence and the steady-

[2]The number of memory locations necessary to store the variables and the parameters.

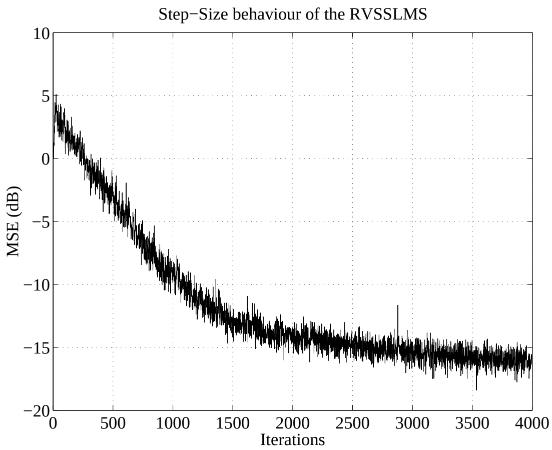

Figure 4.12: Output mean squared error for RVSLMS.

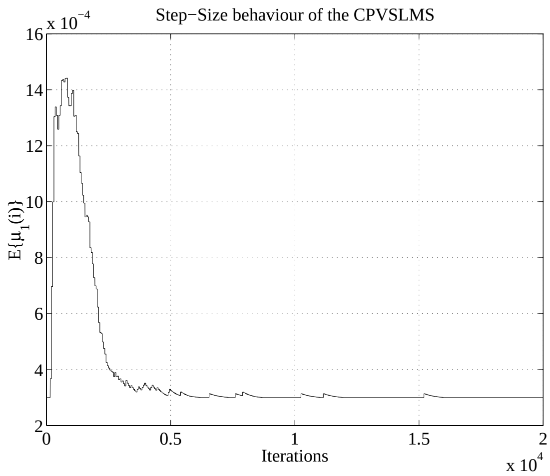

Figure 4.13: Step-size behavior for CP-VSLMS.

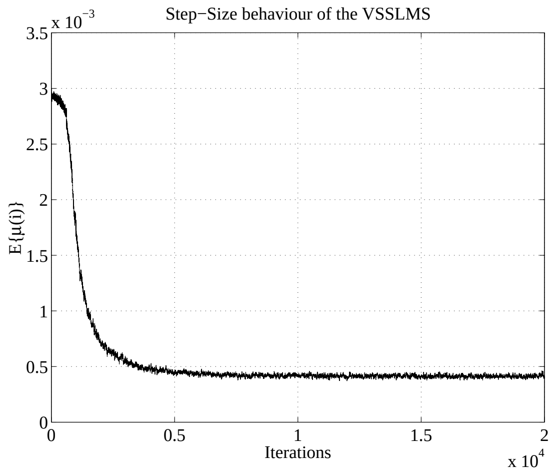

Figure 4.14: Step-size behavior for VSSLMS.

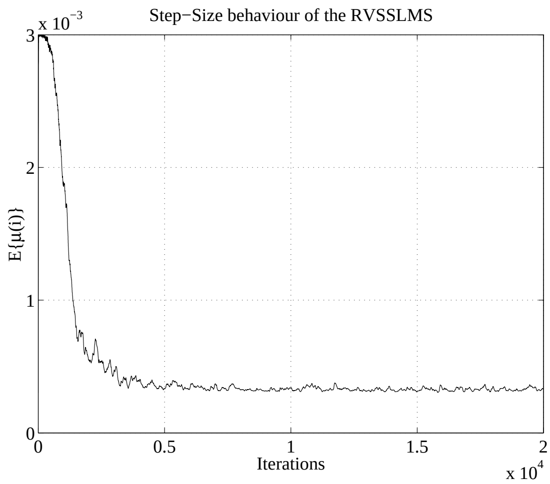

Figure 4.15: Step-size behavior for RVSLMS.

state error is eliminated. Indeed the steady-state misadjustment of the CP-VSLMS is given by its steady-state step-size which is $\mu_1(\infty) = \mu_{min}$, whereas the speed of convergence can be tuned selecting the other parameters, such as, α, μ_{max} and T. In the case of VSSLMS and RVSSLMS the equations that gives the values of the parameters, provided in [48] and [1], are sometimes difficult to be used, due to the fact that they depend on the minimum MSE.

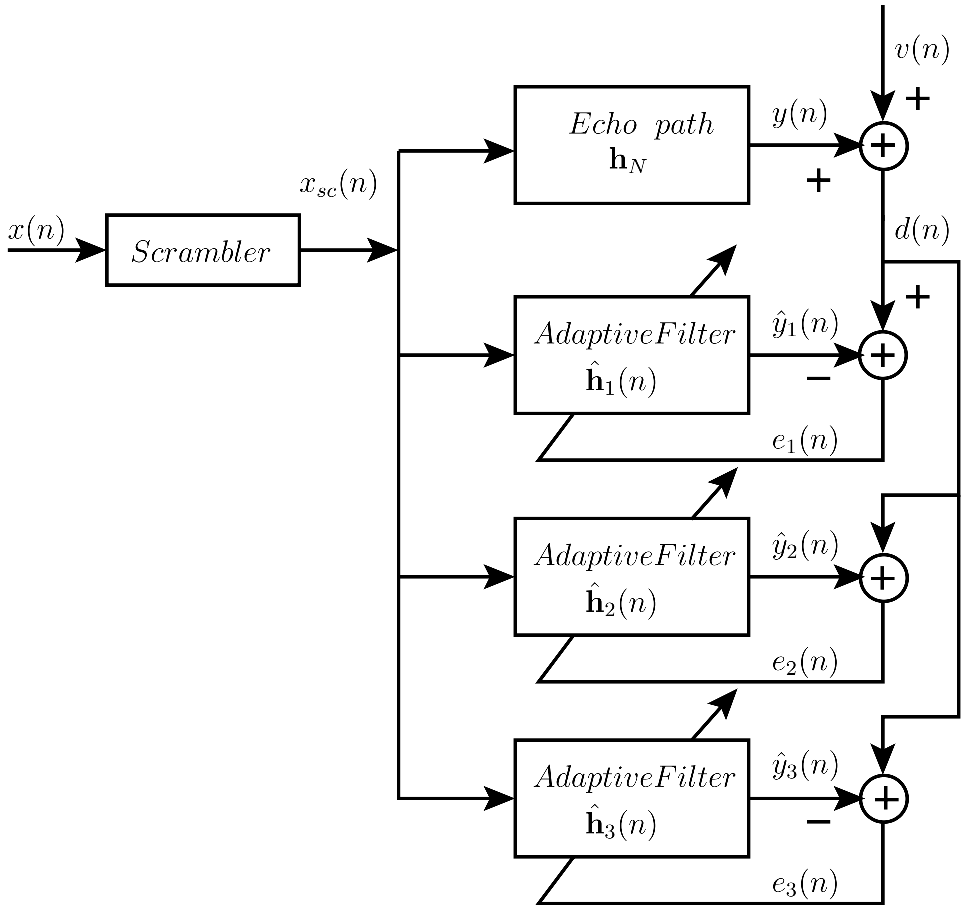

Figure 4.16: Block diagram of the local echo cancellation using the Scrambled LMS with adaptive length.

4.3 Scrambled LMS with adaptive length for echo cancellation

In this section, we address the problem of local echo cancellation for digital data transmition over a telephone line. Moreover, the transmition is secured by the use of a scrambling device at both users as depicted in Fig. 3.16. In our discussion we assume the length of the echo path unknown and the adaptive filter implemented is a combination of the scrambled LMS described in the previous chapter and the variable length introduced in Section 2.3. The aim of this implementation is to reduce the bias which appears in the steady-state MSE when the length of the echo path and the length of the adaptive filter are not equal.

A detailed block diagram implemented in our simulations is depicted in Fig. 4.16 where we have used three adaptive filters with different lengths as in the approach of Section 2.3. The transmitted digital data $x(n)$ has samples chosen from $\{-1; +1\}$ and it contains long strings of consecutive -1's and $+1$'s. The sequence $x(n)$ is passed through a scrambling device, which generate the scrambled sequence $x_{sc}(n)$. As we have seen in the previous chapter, the scrambling device decorrelate the transmitted data, therefore the analytical results for length adaptation from Chapter 2 can be used here[3].

The far-end sequence $v(n)$ is obtained by adding a random zero mean Gaussian-distributed sequence and a bipolar sequence with samples from $\{-a; +a\}$. The Gaussian component of $v(n)$ simulate the channel noise whereas the bipolar component of $v(n)$ is due to the transmitted data from user B to user A (we have assumed that the hybrid connections of the telephone line are ideal).

The transfer function of the echo path used in the simulations was [72]:

$$H(z) = \sum_{i=0}^{N-1} p^i z^{-i}, \tag{4.9}$$

where $p = 0.80025$ and the impulse response of the echo path is shown in Fig. 4.17.

The attenuation of the transmitted data from user B to user A was chosen $a = 0.1$ and the variance of the channel noise was $\sigma_v^2 = 10^{-3}$. The length of the echo path model in (4.9) is $N = 19$ and the lengths of the adaptive filters were initialized with $N_1(0) = 7$, $N_2(0) = 8$ and $N_3(0) = 9$ respectively. A step-size $\mu = 10^{-2}$ was used to initialize the step-sizes of all adaptive algorithms. This value of μ satisfies the stability condition and also ensures equal misadjustments. A time-varying test interval was used for length adaptation in (2.113) and the parameter P was chosen as $P = 2$.

All the results were obtained by averaging a number of 100 independent runs each of them containing 10^4 iterations. The same algorithm as the one described in Section 2.3 was implemented for length adaptation and the adaptive filter of interest is $\widehat{\mathbf{h}}_2(n)$. The average of the length $N_2(n)$ of the second adaptive filter $\widehat{\mathbf{h}}_2(n)$, during the adaptation, is shown in Fig. 4.18. The value of $N_2(n)$ during one run is shown in Fig. 4.19. We can see, that the length of the second adaptive filter converges close to the optimum length which is $N = 19$. However, there is a small difference between the steady-state value of $E\{N_2(n)\}$ and $N = 19$ due to the fact that the autocorrelation matrix of the scrambled sequence $x_{sc}(n)$ is not diagonal and the off-diagonal terms influence the minimum MSE as explained in Chapter 2.

[3] Actually the autocorrelation matrix of the scrambled sequence $x_{sc}(n)$ is not perfectly diagonal, therefore a certain deviation from the optimum length is expected to occur.

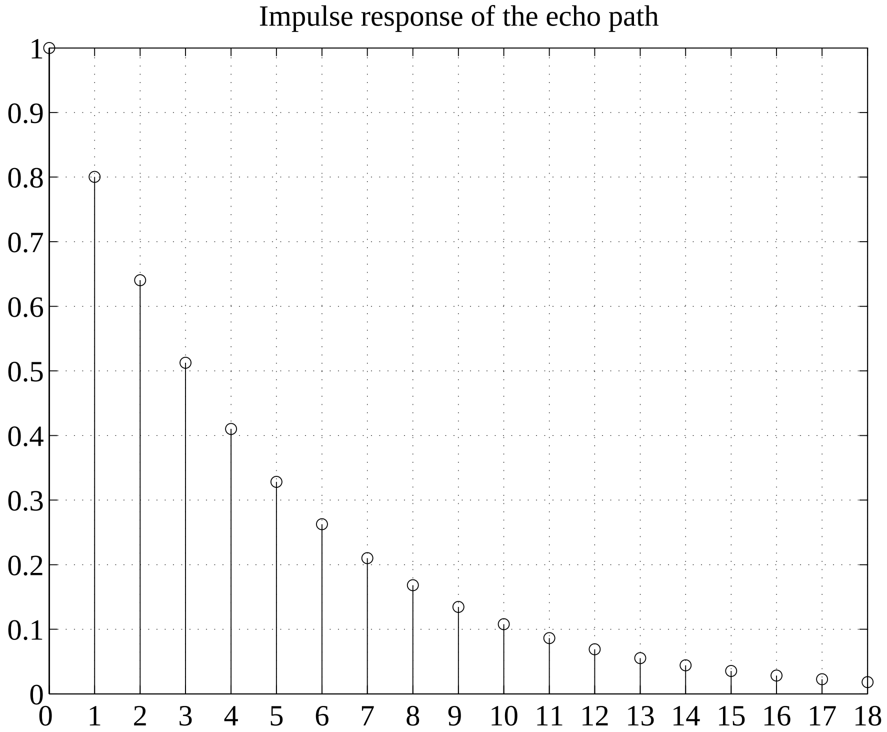

Figure 4.17: Impulse response of the echo path.

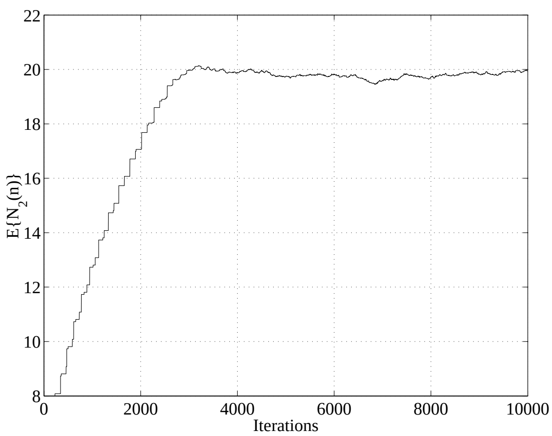

Figure 4.18: The average of the length $N_2(n)$.

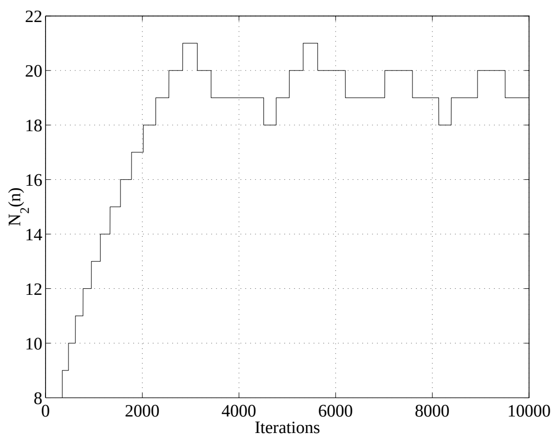

Figure 4.19: The length of the second adaptive filter during one run.

The MSE of the second adaptive filter is depicted in Fig. 4.20, whereas the MSE of an adaptive filter with constant length $N = 19$ is shown in Fig. 4.21. The step-sizes used in the adaptation were chosen to obtain the same misadjustments for both adaptive filters.

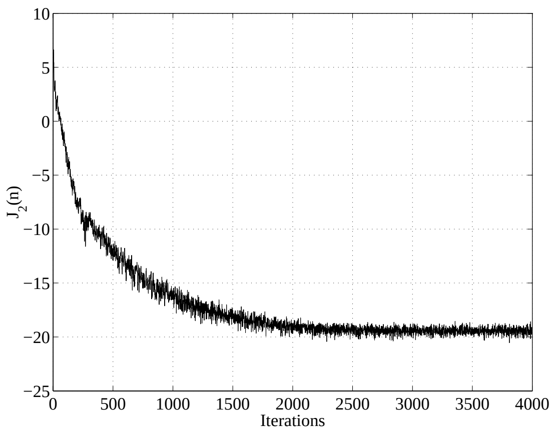

Figure 4.20: Output mean squared error of the second adaptive filter.

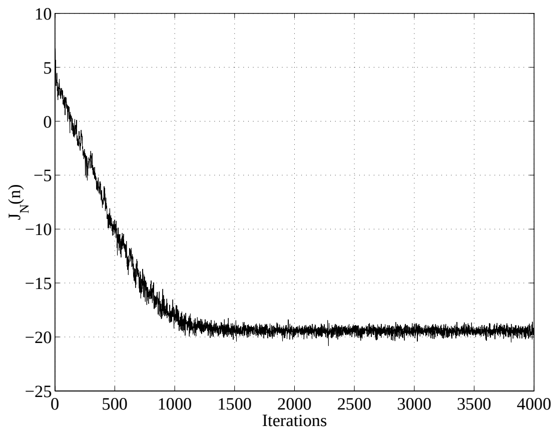

Figure 4.21: Output mean squared error of an adaptive filter with optimum length.

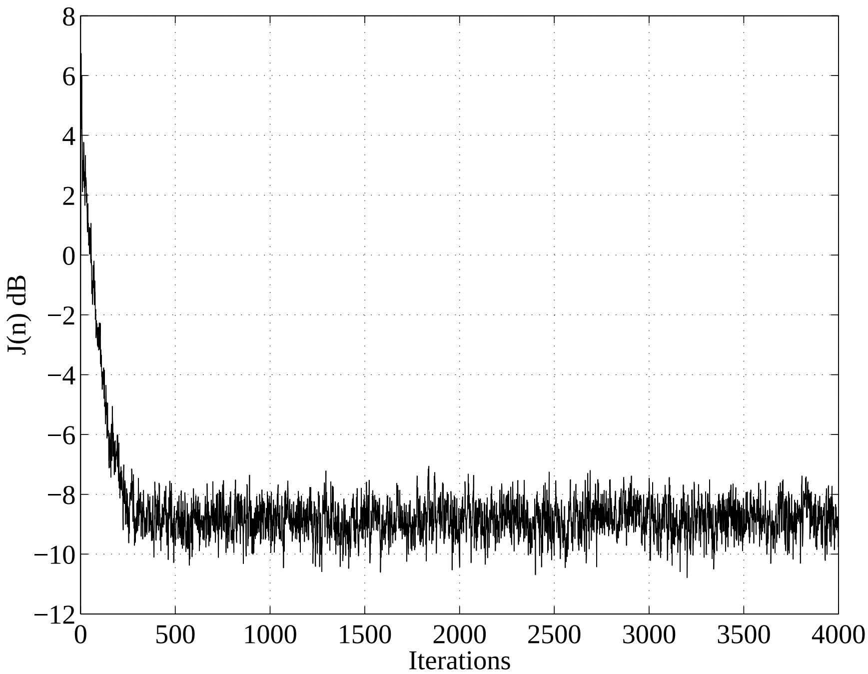

Figure 4.22: Mean squared error for the case of a constant length smaller than the length of the echo path.

For comparison purposes in Fig. 4.22 the MSE of an adaptive filter with fixed length $N_{ad} = 7$ is depicted. The learning curve is obtained in the same framework as the ones shown in Fig. 4.20 and Fig. 4.21. Clearly the filter with adaptive length converges to a smaller steady-state MSE than the filter with fixed length $N_{ad} < N$. Also, due to the transient time of the length adaptation, the filter $\widehat{\mathbf{h}}_2(n)$ converges in a slower form than the filter with optimum length.

From the results shown in this section, we can conclude that in echo cancellation applications, a very important issue is the length adaptation. If a smaller adaptive filter is used, the steady-state MSE is larger as compared with the situation when the length of the echo path is known. When a scrambling device is used to secure the data transmission, this also performs a decorrelation of the input sequence into the adaptive filter and an algorithm as the one introduced in Chapter 2 can be implemented for length adaptation. However, due to the imperfect decorrelating properties of the scrambling device, the off-diagonal terms of the matrix $\mathbf{R}_{sc}$ are non-zero and the estimation of the length is not perfect.

Chapter 5

Conclusions

This thesis has introduced several new algorithms for adaptive filtering, all of them derived from the well known Least Mean Squared adaptive algorithm, which is widely used in many practical applications due to its simplicity. Despite its simplicity, the LMS has some major drawbacks which are mentioned and discussed during this work. One goal of this thesis has been to analyze each of these inconveniences of the LMS and to provide solutions to improve its performance in terms of convergence speed, adaptation error, tracking capabilities and stability in Gaussian and non-Gaussian noise environments.

The new developed techniques differ by addressed framework, selection of parameters and application. We have derived two new adaptive algorithms with variable step-size for time domain which provide high convergence speed while maintaining a small steady-state error. The proposed algorithms use the output error in the adaptation of the step-size and this concept was utilized by many researchers. The difference between the new methods and the existing approaches is that the output error is not directly included in the expression of the step-size, although the step-size adaptation is still based on the mean squared error. As a consequence, the analytical expression of the steady-state misadjustment is simplified[1] and the setup of the parameters is very easy. Even more the parameters of the proposed algorithms are not sensitive to the level of the signals involved.

It is well known that the transform domain LMS increases its convergence speed for highly correlated inputs in comparison with the time domain implementations. We have shown here that using the same concept of step-size adaptation, but in transform domain, the convergence speed can be even more increased. As a consequence, a new class of transform domain adaptive algorithms with variable step-size has been introduced in this

[1] Actually it is shown to be the same as for plain LMS with fixed step-size.

thesis.

The estimation of the model length in system identification applications might be of great interest. Also when there is a mismatch between the length of the model and adaptive filter, the output mean squared error is increased. To address this problem, we have introduced a variable length LMS algorithm in which not only the coefficients of the adaptive filter are adapted but also its length.

Tracking capability for identification applications is also addressed in this thesis and two new algorithms with adaptive step-size are introduced. In a time-varying environment, the steady-state mean squared error possess a minimum for a certain value of the step-size. In the existing approaches the optimum step-size is computed based on some prior information about the statistics of the model. Our proposed algorithms do not use any information for step-size adaptation. The aim of step-size adaptation for a time-varying environment was not to increase the convergence speed, but to adapt their step-size toward the optimum. However one of the two proposed algorithms is derived for the transform domain which provides also an improved convergence speed.

Non-Gaussian noise environments are known to be difficult tasks for the LMS algorithm due to the use of the instantaneous gradient to update the coefficients of the adaptive filter. Due to this fact the LMS may have stability problems for impulsive distributions of the signals. An answer to this problem is to use a smoothed version of the gradient in the update formula. The gradient can be smoothed using different nonlinear filters and the resulting algorithms are available in the open literature. The nonlinear filter has to be chosen based on the distribution of the gradient. In our new approach we use an nonlinear filter with adaptive coefficients for smoothing the gradient and the prior information about the gradient distribution is not any more necessary.

The scrambled LMS algorithm was primarily introduced for the applications where there is a necessity to secure the data transmition. However scrambling was shown to be a good decorelation technique, which can increase the convergence speed. The question which arises is how the scrambled LMS performs comparing with the transform domain LMS. To have an insight to this problem, the analytical expressions of the steady-state mean squared error and mean squared coefficient error for LMS, TDLMS and scrambled LMS are derived in this thesis. These expressions are obtained for the special case when the input sequence has equals samples, in the framework of digital transmission over a telephone line. The optimum solution of the compared algorithms is also derived and the simulations results supporting the theoretical considerations are presented.

As a final conclusion we can state that the contributions of this thesis have both theoretical and practical importance succeeding to introduce several solutions to improve the behavior of the Least Mean Squared adaptive algorithm. All the proposed algorithms were developed based on analytical expressions derived for different situations which can appear in practice. Moreover, this thesis work can also provide many other possible topics of future research.

Bibliography

[1] T. Abounasr and K. Mayas, "A Robust Variable Step-Size LMS-type Algorithm: Analysis and Simulations," *in IEEE Transactions on Signal Processing*, vol. 45, pp. 631–639, Mar. 1997.

[2] N. Ahmed and K. R. Rao, *Orthogonal Transforms for Digital Signal Processing.* Germany: Springer-Verlag, 1975.

[3] Z. D. Banjac, B. D. Kovacevic, M. M. Milosavljevic, and M. D. Veinovic, "Local Echo Canceller with Optimal Input for True Full-Duplex Speech Scrambling System," *in IEEE Transactions on Signal Processing*, vol. 50, pp. 1877–1882, Aug. 2002.

[4] F. Beaufays, "Transform-Domain Adaptive Filters: An Analytical Approach," *in IEEE Transactions on Signal Processing*, vol. 43, pp. 422–431, Feb. 1995.

[5] N. J. Bershad and P. L. Feintuch, "A Normalized Frequency Domain LMS Adaptive Algorithm," *in IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, pp. 452–461, Jan. 1986.

[6] R. C. Bilcu, K. Egiazarian, P. Kuosmanen, and C. Rusu, "Comparative Study Between Scrambling and Transform Domain Adaptive Filtering Techniques," in *Proceedings of the 3rd International TICSP Workshop on Spectral Methods and Multirate Signal Processing, SMMSP 2003*, (Barcelona, Spain), pp. 167–172, Sept. 2003.

[7] R. C. Bilcu, P. Kuosmanen, and K. Egiazarian, "A New Variable Step LMS Algorithm for Transform Domain," in *Proceedings of the 8th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2001*, (Malta), pp. 167–172, Sept. 2001.

[8] R. C. Bilcu, P. Kuosmanen, and K. Egiazarian, "A New Variable Length LMS Algorithm: Theoretical Analysis and Implementations," in *Proceedings of the 9th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2002*, (Dubrovnik, Croatia), pp. 167–172, Sept. 2002.

[9] R. C. Bilcu, P. Kuosmanen, and K. Egiazarian, "A Transform Domain Adaptive Filter with Variable Step-Size," *in IEEE Signal Processing Letters*, vol. 9, pp. 51–53, Feb. 2002.

[10] R. C. Bilcu, P. Kuosmanen, and K. Egiazarian, "Channel Equalization using a New Transform Domain LMS Algorithm with Variable Step-Size," *in WSEAS Transactions on Circuits*, vol. 1, no. 1, pp. 113–119, 2002.

[11] R. C. Bilcu, P. Kuosmanen, and K. Egiazarian, "Noise Constrained LMS Algorithm in Transform Domain," in *Proceedings of the XI European Signal Processing Conference, EUSIPCO 2002*, (Toulouse, France), pp. 167–172, Sept. 2002.

[12] R. C. Bilcu, P. Kuosmanen, and K. Egiazarian, "On Order Statistic Least Mean Square Algorithms," in *Proceedings of the IEEE International Conference on Audio, Speech and Signal Processing, ICASSP 2002*, (Orlando, Florida, USA), pp. 167–172, May 2002.

[13] R. C. Bilcu, P. Kuosmanen, and K. Egiazarian, "Adaptive Channel Equalization in Non-Gaussian Noise Environments," in *Proceedings of the 5th IEEE International Conference and Exhibition Digital Signal Processing and its Applications, DSPA 2003*, (Moscow, Russia), pp. 1–1, Mar. 2003.

[14] R. C. Bilcu, P. Kuosmanen, and K. Egiazarian, "LMS Adaptive Filter with Optimum Step-Size for Tracking Time-Varying Channels," in *Proceedings of the 3rd IEEE International Symposium on Image and Signal Processing and Analysis ISPA 2003*, (Rome, Italy), pp. 883–887, Sept. 2003.

[15] R. C. Bilcu, P. Kuosmanen, and K. Egiazarian, "Tracking Time-Varying Channels with Adaptive Step-Size Transform Domain LMS Algorithm," in *Proceedings of the WSEAS Recent Advances in Intelligent Systems and Signal Processing*, (Corfu, Greece), pp. 104–109, July 2003.

[16] R. C. Bilcu, P. Kuosmanen, and K. Egiazarian, "On Adaptive Interpolated FIR Filters," in *Proceedings of the IEEE International Conference on Audio, Speech and Signal Processing, ICASSP 2004*, (Montreal, Canada), pp. –, May 2004.

[17] R. C. Bilcu, P. Kuosmanen, and K. Egiazarian, "On Length Adaptation for the Least Mean Square Adaptive Filters," *IEEE Transactions on Circuits and Systems I*, submitted for publication.

[18] R. C. Bilcu, P. Kuosmanen, and C. Rusu, "A Noise Constrained VS-LMS Algorithm," in *Proceedings of the IEEE Wireless Communications and Networking Conference, EUROCOMM 2000*, (Munich, Germany), pp. 159–165, May 2000.

[19] R. C. Bilcu, P. Kuosmanen, and C. Rusu, "Improving Performances of Complementary Pair LMS Algorithm," in *Proceedings of the X European Signal Processing Conference, EUSIPCO 2000*, (Tampere, Finland), pp. 159–165, Sept. 2000.

[20] B. F. Boroujeny, *Adaptive Filters: Theory and Applications.* Chichester-New York-Weinheim-Brisbane-Singapore-Toronto: John Wiley & Sons, 1999.

[21] B. F. Boroujeny and S. Gazor, "Selection of Orthonormal Transforms for Improving the Performance of the Transform Domain Normalized LMS Algorithm," *in IEE Proceedings-F*, vol. 139, pp. 327–335, Oct. 1992.

[22] B. F. Boroujeny and S. Gazor, "Performance of LMS-Based Adaptive Filters in Tracking a Time Varying Plant," *in IEEE Transactions on Signal Processing*, vol. 44, pp. 2868–2871, Nov. 1996.

[23] J. A. Chambers, "Normalized Order Statistic LMS Adaptive Filtering," in *Proceedings of the IEE Colloquium on Nonlinear Filters*, pp. 3/1–3/8, May 1994.

[24] W. D. Davenport and W. L. Root, *An Introduction to the Theory of Random Signals and Noise.* New-York: IEEE Press, 1987.

[25] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation.* USA: Kluwer Academic Publishers, 1997.

[26] K. Egiazarian, P. Kuosmanen, and R. C. Bilcu, "Variable Step-Size LMS Adaptive Filters for CDMA Multiuser Detection," in *Proceedings of the 6th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services*, (Nis, Serbia and Montenegro), pp. 1–1, Oct. 2003.

[27] J. B. Evans, P. Xue, and B. Liu, "Analysis and Implementation of Variable Step Size Adaptive Algorithms," *in IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 41, pp. 2517–2535, Aug. 1993.

[28] E. Eweda, "Comparison of RLS, LMS and Sign Algorithms for Tracking Randomly Time-Varying Channels," *in IEEE Transactions on Signal Processing*, vol. 42, pp. 2937–2944, Nov. 1994.

[29] A. Feuer and E. Weinstein, "Convergence Analysis of LMS Filters with Uncorrelated Gaussian Data," *in IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, pp. 222–230, Feb. 1985.

[30] S. Florian and N. J. Bershad, "A Weighted Normalized Frequency Domain LMS Adaptive Algorithm," *in IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, pp. 1002–1007, July 1988.

[31] O. L. Frost, "An Algorithm for Linearly Constrained Adaptive Array Processing," *in Proceedings of the IEEE*, vol. 60, pp. 926–935, Aug. 1972.

[32] Y. Fu and G. A. Williamson, "Convergence of Adaptive Algorithms with Order Statistic Based Gradient Estimates," in *Proceedings of the International Symposium on Circuits and Systems*, (London, UK), pp. 397–400, June 1994.

[33] Y. Fu, G. A. Williamson, and P. M. Clarkson, "Adaptive Algorithms for Non-Gaussian Noise Environments: The Order Statistic Least Mean Square Algorithms," *in IEEE Transactions on Signal Processing*, vol. 42, pp. 2945–2954, Nov. 1994.

[34] J. F. Galdino and E. L. Pinto, "Performance of the LMS Algorithm on the Estimation of Time-Varying Channels," in *Proceedings of the IEEE International Conference on Communications, ICC 2002*, (New York, USA), pp. 94–98, May 2002.

[35] S. Gazor, "Prediction in LMS-Type Adaptive Algorithms for Smoothly Time-Varying Environments," *in IEEE Transactions on Signal Processing*, vol. 47, pp. 1735–1739, June 1999.

[36] Y. Gong, B. F. Boroujeny, and T. J. Lim, "Variable Step-Size LMS Blind CDMA Multiuser Detector," in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, (Salt Lake City, USA), pp. 2285–2288, May 2001.

[37] Y. Gong, T. J. Lim, and B. F. Boroujeny, "Performance Analysis of the LMS Blind Minimum-Output-Energy CDMA Detector," in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, (Istanbul, Turkey), pp. 2473–2476, June 2000.

[38] R. W. Harris and D. M. Chambries, "A Variable Step (VS) Adaptive Filtering Algorithm," *in IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-34, pp. 309–316, Apr. 1986.

[39] T. I. Hawel and P. M. Clarkson, "A Class of Order Statistic LMS Algorithms," *in IEEE Transactions on Signal Processing*, vol. 40, pp. 44–53, Jan. 1992.

[40] S. Haykin, A. H. Sayed, J. R. Zeidler, P. Yee, and P. C. Wei, "Adaptive Tracking of Linear Time-Variant Systems by Extended RLS Algorithm," *in IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 45, pp. 1118–1128, May 1997.

[41] S. S. Haykin, *Adaptive Filter Theory.* NJ: Prentice-Hall, Englewood Cliffs, 1991.

[42] M. Honig, U. Madhow, and S. Verdu, "Blind Adaptive Multiuser Detection," *in IEEE Transactions on Information Theory*, vol. 41, pp. 944–960, July 1995.

[43] N. Kalouptsidis and S. Theodoridis, *Adaptive System Identification and Signal Processing Algorithms.* NJ: Prentice-Hall, Englewood Cliffs, 1993.

[44] D. W. Kim, J. H. Choi, Y. S. Choi, C. H. Jeon, and H. Y. Ko, "A VS-LMS Algorithm Using Normalized Absolute Estimation Error," in *Proceedings of the IEEE TENCON-Digital Signal Processing Applications*, (Perth Australia), pp. 692–697, Nov. 1996.

[45] D. I. Kim and P. D. Wilde, "Performance Analysis of the DCT-LMS Adaptive Filtering Algorithm," *in Signal Processing*, vol. 80, pp. 1629–1654, Aug. 2000.

[46] S. Koike, "A Class of Adaptive Step-Size Control Algorithms for Adaptive Filters," *in IEEE Transactions on Signal Processing*, vol. 50, pp. 1315–1326, June 2002.

[47] V. Krishnamurthy and G. Yin, "Variable Step-Size Algorithms for Blind CDMA Multiuser Detector," in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, (Salt-Lake City), pp. 2285–2288, May 2001.

[48] R. H. Kwong and E. W. Johnston, "A Variable Step Size LMS Algorithm," *in IEEE Transactions on Signal Processing*, vol. 40, pp. 1633–1642, July 1992.

[49] B. P. Lahti, *Modern Digital and Analog Communication Systems.* Oxford: Oxford University Press, 1998.

[50] M. Latva-aho and M. J. Juntti, "LMMSE Detection for DS-CDMA Systems in Fading Channels," *in IEEE Transactions on Communications*, vol. 48, pp. 194–199, Feb. 2000.

[51] J. C. Lee and C. K. Un, "Performance of Transform-Domain LMS Adaptive Digital Filters," *in IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. ASSP-34, pp. 499–510, June 1986.

[52] W. C. Lee, *Mobile Communications Engeneering*. New York: McGraw-Hill, 1982.

[53] T. J. Lim, Y. Gong, and B. F. Boroujeny, “Convergence Analysis of LMS Multiuser CDMA Detectors,” in *Proceedings of the Wirelles Communications and Networking Conference*, (New Orleans, USA), pp. 75–79, Sept. 1999.

[54] L. Lindbom, M. Sternard, and A. Ahlen, “Tracking of Time-Varying Mobile Radio Channels-Part I: The Wiener LMS Algorithm,” *in IEEE Transactions on Communications*, vol. 49, pp. 2207–2217, Dec. 2001.

[55] L. Lindbom, M. Sternard, and A. Ahlen, “Tracking of Time-Varying Mobile Radio Channels-Part II: A Case Study,” *in IEEE Transactions on Communications*, vol. 50, pp. 156–167, Dec. 2002.

[56] D. F. Marshall, W. K. Jenkins, and J. J. Murphy, “The Use of Orthogonal Transforms for Improving Performance of Adaptive Filters,” *in IEEE Transactions on Circuits and Systems*, vol. 36, pp. 474–484, Apr. 1989.

[57] S. L. Miller, “Training Analysis of Adaptive Interference Suppression for Direct-Sequence Code-Division Multiple-Access Systems,” *in IEEE Transactions on Communications*, vol. 44, pp. 488–495, Apr. 1996.

[58] S. Narayan, A. Peterson, and M. Narasimha, “Transform Domain LMS Algorithm,” *in IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 31, pp. 609–615, June 1983.

[59] M. Niedzwiecki, *Identification of Time-Varying Processes*. Chichester England: John Wiley & Sons, LTD, 2000.

[60] V. N. Parikh and A. Z. Baraniecki, “The Use of the Modified Escalador Algorithm to Improve the Performance of Transform-Domain LMS Adaptive Filters,” *in IEEE Transactions on Signal Processing*, vol. 46, pp. 625–635, Mar. 1998.

[61] M.-S. Park and W.-J. Song, “A Complementary Pair LMS Algorithm for Adaptive Filtering,” *in IEICE Transactions Fundamentals*, vol. E81-A, pp. 1493–1496, July 1998.

[62] S.-C. Pei and C.-C. Tseng, “Least Mean P-Power Error Criterion for Adaptive FIR Filter,” *in IEEE Journal on Selected Areas in Communications*, vol. 12, pp. 1540–1547, Dec. 1994.

[63] R. Pintelon and J. Schoukens, *System Identification: A Frequency Domain Approach.* New-York: IEEE press, 2001.

[64] I. Pitas and A. N. Venetsanopoulos, “Adaptive L-filters,” in *Proceedings of the European Conference on Circuit Theory and Design*, (Brighton UK), pp. 580–584, Sept. 1989.

[65] I. Pitas and A. N. Venetsanopoulos, “Adaptive Filters Based on Order Statistics,” *in IEEE Transactions on Signal Processing*, vol. 39, pp. 518–522, Feb. 1990.

[66] I. Pitas and A. N. Venetsanopoulos, “Application of Adaptive Order Statistic Filters in Digital Image/Image Sequence Filtering,” in *Proceedings of the International Symposium on Circuits and Systems*, (Chicago, IL, USA), pp. 327–330, May 1993.

[67] H. V. Poor and X. Wang, “Code-Aided Interference Suppression for DS/CDMA Communications-Part I: Interference Suppression Capability,” *in IEEE Transactions on Communications*, vol. 45, pp. 1101–1111, Sept. 1997.

[68] H. V. Poor and X. Wang, “Code-Aided Interference Suppression for DS/CDMA Communications-Part II: Parallel Blind Adaptive Implementations,” *in IEEE Transactions on Communications*, vol. 45, pp. 1112–1122, Sept. 1997.

[69] Z. Pritzker and A. Feuer, “Variable Length Stochastic Gradient Algorithm,” *in IEEE Transactions on Signal Processing*, vol. 39, pp. 997–1001, Apr. 1991.

[70] J. G. Proakis and D. G. Manolakis, *Introduction to Digital Signal Processing.* New-York: Macmillan, 1988.

[71] C. Rusu and C. F. N. Cowan, “Adaptive Data Echo Cancellation Using Cost Function Adaptation,” *in Signal Processing*, vol. 80, pp. 2457–2473, Nov. 2000.

[72] C. Rusu, *Several Approaches to Cost Function Adaptation and Phase Approximation.* PhD thesis, Tampere University of Technology, Tampere, Finland, 2000.

[73] J. J. Shynk, “Frequency-Domain and Multirate Adaptive Filtering,” *in IEEE Signal Processing Magazine*, vol. 9, pp. 14–37, Jan. 1992.

[74] P. C. Sommen, P. J. VanGerwen, H. J. Kotmans, and A. J. E. M. Janssen, “Convergence Analysis of a Frequency-Domain Adaptive Filter with Exponential Power Averaging and Generalized Window Function,” *in IEEE Transactions on Circuits and Systems*, vol. CAS-34, pp. 788–798, July 1987.

[75] S. Verdu, *Multiuser Detection.* New-York: Cambridge University Press, 1998.

[76] E. Walach and B. Widrow, "The Least Mean Fourth (LMF) Adaptive Algorithm," *in IEEE Transactions on Information Theory*, vol. 30, pp. 275–283, Mar. 1984.

[77] Y. Wet, S. B. Gelfand, and J. V. Krogmeier, "Noise-Constrained Least Mean Squares Algorithm," *in IEEE Transactions on Signal Processing*, vol. 49, pp. 1961–1970, Sept. 2001.

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

[79] G. A. Williamson and P. M. Clarkson, "On Signal Recovery with Adaptive Order Statistic Filters," *in IEEE Transactions on Signal Processing*, vol. 40, pp. 2622–2626, Oct. 1992.

[80] Y. K. Won, R.-H. Park, J. H. Park, and B.-U. Lee, "Variable LMS Algorithms Using the Time Constant Concept," *in IEEE Transactions on Consumer Electronics*, vol. 40, pp. 655–661, Aug. 1994.

[81] K. W. Wong and T. O'Farell, "Blind Adaptive Single User Detection with Adjustable Step Size for DS/CDMA Communication over a Rayleight Fading Mobile Channel," in *Proceedings of the IEE Third International Conference on 3G Mobile Communication Technologies*, (Leeds, UK), pp. 180–184, May 2002.

[82] N. R. Yousef and A. H. Sayed, "A Unified Approach to the Steady-State and Tracking Analyses of Adaptive Filters," *in IEEE Transactions on Signal Processing*, vol. 49, pp. 314–324, Feb. 2001.

[83] G. Zelniker and F. J. Taylor, *Advance Digital Signal Processing - Theory and Applications.* New-York: Marcel Dekker Inc., 1993.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O. Box 527
FIN-33101 Tampere, Finland