SPARSE ADAPTIVE FILTERING TECHNIQUES FOR ACOUSTIC ECHO CANCELLATION

by

Cemil Turan

Submitted to the Institute of Graduate Studies in Science and Engineering

in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Mevlana (Rumi) University

2016

SPARSE ADAPTIVE FILTERING TECHNIQUES FOR ACOUSTIC ECHO CANCELLATION

submitted by **Cemil Turan** in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical-Electronics Engineering at Mevlana (Rumi) University.

APPROVED BY:

Examining Committee Members:

Yrd. Doç. Dr. M. Shukri Salman …………………………

Prof. Dr. Yüksel Özbay …………………………

Prof. Dr. Bekir Karlık …………………………

Doç. Dr. Halis Altun …………………………

Yrd. Doç. Dr. İbrahim Onaran …………………………

Yrd. Doç. Dr. Mehmet Argın ………………………..

Head of Department, Electrical and Computer Engineering

Prof. Dr. Ali Sebetci …………………………

Director, Institute of Graduate Studies in Science and Engineering

DATE OF APPROVAL

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Cemil Turan

………………………

# ABSTRACT

## SPARSE ADAPTIVE FILTERING TECHNIQUES FOR ACOUSTIC ECHO CANCELLATION

Cemil Turan

Ph.D. Thesis, 2016

Thesis supervisor: Assist. Prof. Dr. Mohammad Shukri SALMAN

Co-supervisor: Assist. Prof. Dr. Alaa Eleyan

**Keywords**: Adaptive algorithms, system identification, sparse systems, echo cancellation.

With the increasing demand for mobile and wireless communication, wireless telephony became very popular and indispensable in recent years, because of its ease of use and flexibility. However in telecommunications, we often hear about echo problem which degrade the speech quality during conversation. To overcome this issue, numerous echo cancellers have been modeled in the field of digital signal processing. Because the acoustic echo signals vary due to the several conditions, the adaptive filters are one of the best solutions for acoustic echo cancellation (AEC) systems. Here the main goal is to generate the replica of echo signal via an adaptive filter and subtract it from the original signal. If we consider that an echo path is produced by an unknown system, then it can be assumed as a system identification (SI) problem.

The least-mean-square (LMS) adaptive algorithm is a well-known adaptive algorithm and very successful for SI problems. It has a constant step-size parameter that controls the convergence behavior of the recursive algorithm. When the number of coefficients of the system is relatively large as in many applications such as echo cancellation, the performance of the LMS-type algorithms fairly deteriorate. However, the impulse response in an echo canceller can be modeled as a sparse system that has only a few non-zero coefficients. For a better performance, the simplicity and robustness of the LMS algorithm can be combined with the advantages of variable step-size and the sparsity of the system.

In this thesis, we propose a new algorithm based on the function controlled variable step-size LMS (FC-VSSLMS) algorithm for sparse system identification setting. Firstly, the proposed algorithm is derived in time domain to check the performance with white Gaussian signals for the input and noise. After that, a transform domain version is proposed in order to overcome the correlated signal problems. Finally, a block implementation of the proposed algorithm is derived to decrease the computational time for a long filter-tap. For all these three versions of the algorithm, convergence and the stability analysis are presented and the computational complexity are derived.

Experiments are performed in the MATLAB package. The performances of the proposed algorithms are compared to those of the LMS, FC-VSSLMS, ZA-LMS and RZA-LMS algorithms in terms of convergence rates and MSD's. Simulations show that the proposed algorithms have superiority over the others under different conditions.

# ÖZET

## Akustik Yankı Gidericiler İçin Seyrek Adaptif Filtre Teknikleri

Cemil Turan

Doktora Tezi, 2016

Tez Danışmanı: Yrd. Doç. Dr. Mohammad Shukri Salman

Ek Danışman: Yrd. Doç. Dr. Alaa Eleyan

**Anahtar Kelimeler**: Adaptif algoritmalar, sistem tanımlama, seyrek sistemler, yankı giderme.

Son yıllarda hızla artan mobil iletişim talebi, kablosuz telefonları kullanım kolaylığı ve esnekliği nedeniyle çok popüler ve vazgeçilmez hale getirmiş durumdadır. Ancak bu tür haberleşme sistemlerinde, sohbet anında konuşma kalitesini düşüren yankı problemlerini sıklıkla duyarız. Bu problemi çözmek için dijital sinyal işleme alanında birçok yankı gidericiler modellenmiştir. Akustik yankı sinyalleri birçok farklı nedenlere bağlı olarak değişim gösterdiğinden dolayı, adaptif filtre kullanımının yankı giderme hususunda en iyi çözüm olduğu söylenebilir. Buradaki amaç, bir adaptif filtre yardımıyla yankı sinyalinin bir benzerini üreterek orjinal sinyalden çıkarmaktır. Eğer oluşan yankının bilinmeyen bir sistem tarafından üretildiğini varsayarsak, bunun bir sistem tanımlama (SI) problemi olduğunu söyleyebiliriz.

En küçük ortalama kare (LMS) adaptif algoritması SI problemlerinde iyi bilinen ve çok başarılı sonuçlar veren bir algoritmadır. Bu algoritma yinelemeli hesaplarda yakınsama hızını kontrol eden bir adım-boyut parametresine sahiptir. Yankı gidermenin de dahil olduğu birçok uygulamada, eğer sistem katsayıları göreceli bir fazlalığa sahipse LMS-sınıfı algoritmaların performansı oldukça zayıflama gösterir. Ancak yankı giderici sistemlerde dürtü yanıtı, çok az sıfır olmayan

katsayılara sahip seyrek sistemler oarak tanımlanabilir. Çok daha iyi bir performans için LMS algoritmasının basit ve güçlü yapısı değişken adım-boyut ve seyrek sistem avantajlarıyla birleştirilebilir. Bizler bu çalışmada seyrek sistem tanımlamalarında kullanılmak üzere fonksiyon kontrollü değişken adım-boyutlu LMS (FC-VSSLMS) algoritmasına dayanan yeni bir algoritma geliştirdik. Bu algoritma, öncelikle giriş ve gürültü sinyalleri beyaz Gaussian sinyalleri olarak varsayılarak, performans değerlendirmesi için zaman domeninde çıkarılmıştır. Daha sonra aynı algoritma yüksek korelasyonlu sinyal problemlerini çözmek amacıyla dönüştürülmüş domende tanımlanmıştır. Son olarak ta uzun filre katsayılarına sahip sistemlerde hesaplama süresini düşürmek amacıyla algoritmanın blok versiyonu çıkarılmıştır. Önerilen algoritmanın her üç versiyonu için de yakınsama ve kararlılık analizi çalışmalarıyla beraber hesaplama karmaşıklığı karşılaştırmalı olarak gösterilmektedir.
Bütün deneyler MATLAB bilgisayar programı yardımıyla gerçekleştirilmiş ve simulasyonlar elde edilmiştir. Önerdiğimiz algoritmaların performansları yakınsama hızı ve ortalama kare sapma (MSD) kriterleri boyunca LMS, FC-VSSLMS, ZA-LMS ve RZA-LMS algoritmalarıyla karşılaştırılmaktadır. Simulasyonlarda elde edilen sonuçlara göre önerdiğimiz algoritma farklı durum ve koşullara rağmen diğerlerine büyük bir üstünlük göstermiştir.

Dedicated to my family

# ACKNOWLEDGEMENTS

I would like to express my special appreciation and thanks to my advisor Assist. Prof. Dr. Mohammad Shukri Salman. He has been a tremendous mentor for me. I would like to thank him for encouraging my research and for allowing me to grow as a research scientist. His advice on both research as well as on my career have been priceless. I would also like to thank my committee members, Prof. Dr. Yuksel Ozbay, Assist. Prof. Dr. Alaa Eleyan and Assoc. Prof. Dr. Halis Altun for serving as my committee members even at hardship. I also want to thank them for letting my defense be an enjoyable moment, and for their brilliant comments and suggestions, thanks to them. I would, especially, like to thank assistants and secretary unit at Engineering Faculty in Mevlana University. All of them have been there to support me when I recruited patience and study for my Ph.D. thesis.

A special thanks to my family. Words cannot express how grateful I am to my wife and sons for all of the sacrifices and patience that they have made on my behalf. Their prayer for me was what sustained me thus far. I would also like to thank all of my friends who supported me in writing, and incentivized me to strive towards my goal. At the end, I would like express appreciation to the administration of Mevlana University for convenience they provided to me.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF SYMBOLS/ABBREVIATIONS

| | |
|---|---|
| $\gamma$ | Control parameter |
| $\lambda$ | Eigenvalue |
| $\mu$ | Step-size |
| $\sigma^2$ | Variance |
| $\tau$ | Time constant |
| $\mathbf{I}$ | Identity matrix |
| $\mathbf{\Lambda}$ | Diagonal matrix |
| $\mathbf{Q}$ | control matrix |
| $N$ | Filter length |
| $\mathbf{R}$ | Autocorrelation matrix |
| $\mathbf{w}$ | Tap weights vector |
| $\boldsymbol{\theta}$ | Misalignment vector |
| $\mathbf{x}$ | Tap input vector |
| $\mathrm{E}\{\cdot\}$ | Expected value |
| $\mathrm{J}\{\cdot\}$ | Cost function |
| ACGN | Additive Correlated Gaussian Noise |
| AEC | Acoustic Echo Cancellation |
| AWGN | Additive White Gaussian Noise |
| BSFC-VSSLMS | Block Sparse Function Controlled VSSLMS |
| CS | Compressive Sensing |
| DCT | Discrete Cosine Transform |
| DSP | Digital Signal Processing |
| FC-VSSLMS | Function Controlled Variable Step-Size LMS |
| FIR | Finite Impulse Response |
| IIR | Infinite Impulse Response |

| | |
|---|---|
| LMS | Least Mean Square |
| MSD | Mean Square Deviation |
| MSE | Mean Square Error |
| RZA-LMS | Reweighted Zero-Attracting LMS |
| SFC-VSSLMS | Sparse Function Controlled VSSLMS |
| $\mathrm{sgn}(\cdot)$ | Signum function |
| SNR | Signal-to-Noise Ratio |
| TDLMS | Transform Domain LMS |
| TDSFC-VSSLMS | Transform Domain Sparse Function Controlled VSSLMS |
| VSSLMS | Variable Step-size LMS |
| ZA-LMS | Zero-Attracting LMS |

# CHAPTER 1

# INTRODUCTION

## 1.1. Overview

Electrical signals are everywhere in our life. They appear in measurement devices, in communications, in audio and video instruments, in control systems or in computers. They convey information that can be extracted by signal processing techniques.

In general, signals can be classified in two forms: Continuous time (analog) or discrete time (digital) [1]. In an analog form of the signal, a quantity is represented by a voltage or current in a continuous time, where in a digital one, it is represented by a combination of ON/OFF pulses by means of binary numbers 1 or 0 [2]. Although the analog signals are massively used in electronic devices earlier, over the last few decades, digital signal technology became a major engineering discipline [3]. Because of availability of application in many different technical instruments and commercial demands, digital signal processing algorithms are used in various applications such as; telecommunication, radar, sonar, video and audio processing, pattern recognition, noise reduction, geophysics exploration, data forecasting, the processing of large database for the identification extraction and organization of unknown underlying structures and patterns [4]. Today, digital signal processing (DSP) techniques are used intensively in modern electronics [5].

A signal can be processed in a digital manner on a computer or DSP chip performing the acquisition, representation, manipulation, transformation and extraction of information from signals [2] with the processing types of digital filtering [6], digital integration or digital correlation [7]. We shall deal with digital filtering in general.

A signal can be described in time domain represented by amplitude vs. time graph or in frequency domain with amplitude vs. frequency graph as shown in Fig. 1.1 [8]. Based on the frequency domain, a filter is classified into four categories [9] (Fig. 1.2):

- Low-pass filter: The filter that allows to pass the signals below a predetermined cut-off frequency and attenuates signals over that cut-off frequency.
- High-pass filter: The filter that allows to pass the signals higher than the cut-off frequency and doesn't pass signals below that cut-off frequency.
- Band-pass filter: The filter that passes the signals inside a predefined frequency range but attenuates the frequencies outside that range.
- Band-stop filter: The filter that allows all frequencies except in a specific interval.

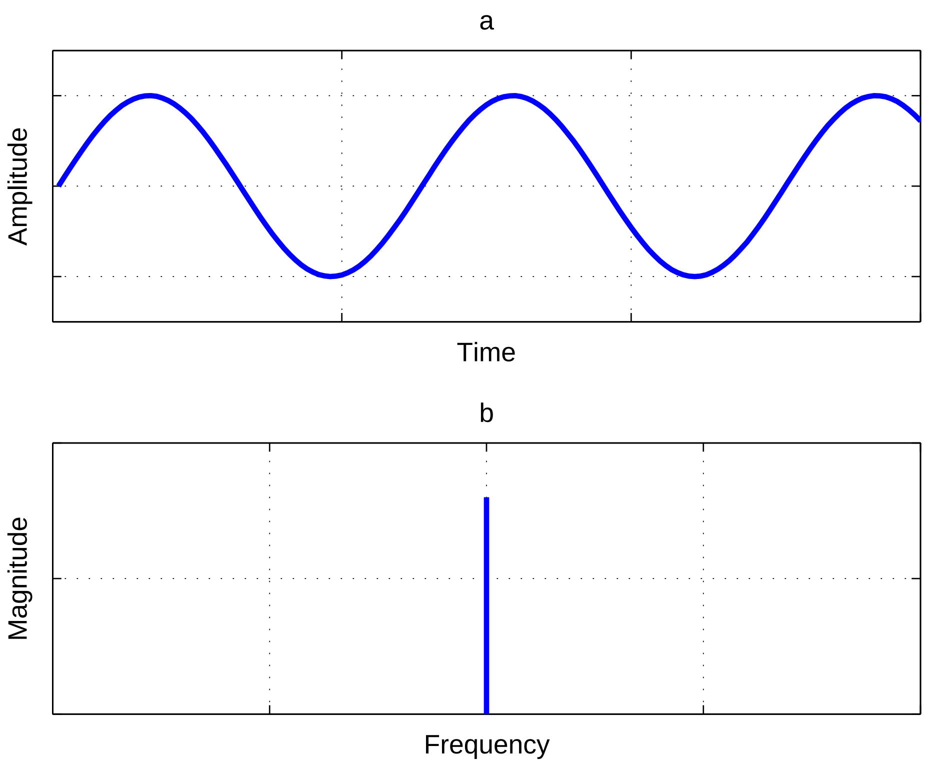


Figure 1.1. a) A signal in time domain, b) A signal in frequency domain.

A second type of classification of digital filters is based on the length of their impulse responses: An infinite impulse response (IIR) filter has an infinite length of **N** where a finite impulse response (FIR) filter is defined as a filter whose length **N** is finite [10].

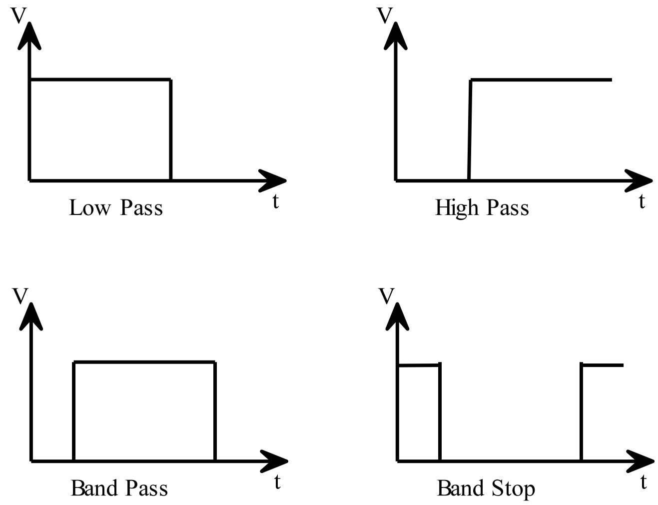


Figure 1.2. Ideal filter types according to Frequency response .

In general, a filter can be describe as a device that performs filtering, smoothing, prediction or deconvolution over a signal [11] by using signal processing techniques to manipulate the information carried by the signal [12]. Such a filter can be designed based on the following theoretical approaches [3]:

- Conventional approach.
- Optimal filtering (Wiener or Kalman filtering).
- Self-adjusting filters (adaptive filters).

However, we shall restrict our attention to adaptive filters.

## 1.2. Adaptive Filters

A digital filter with fixed coefficients can be designed by using well defined prescribed specifications. However, in some situations, where the specifications are not available or are time varying, a filter that adjusts the coefficients with time is required [3]. This type of filters is called an adaptive filter. In a brief definition, an adaptive filter is a self-designing and time varying system that adjusts its tap weights for operation in an unknown environment [13].

The adaptive filtering problem is described mathematically and then the filter coefficients are adjusted using optimization procedure that minimizes the error criterion which is defined as a cost function of the algorithm [14]. In Fig. 1.3, a general adaptive filter configuration is presented by a block diagram. An adaptive filter can be described by the following aspects [14]:

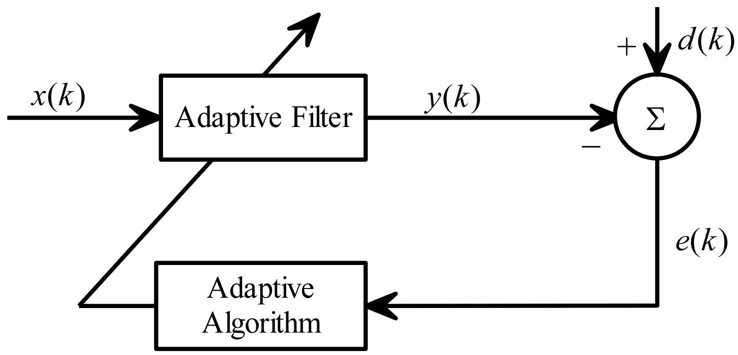


Figure 1.3. General Adaptive Filter Configuration.

- the signals which are processed by that adaptive filter,
- the structure of the filter which defines the computation method of output signal by processing the input signal,
- the adaptive algorithm in which the parameter adjustment method is described from a time to another,
- the parameters which are possible to change iteratively in the structure used for the alteration of the input and output signals relationship.

In adaptive filtering techniques, the transversal structure given in Fig. 1.4 is a commonly used configuration having a single input $x(k)$ and a single output $y(k)$; which is defined as a linear combination of delayed input signal samples and the filter tap as,

$$y(k) = \sum_{i=0}^{N-1} w_i(k)x(k-i) \tag{1.1}$$

where $w_i(k)$ are filter tap coefficients, $N$ is the length of the filter and $i = 0, 1, ..., N-1$. According to (1.1), the output depends only on the delayed input sequence $x(k-i)$, that is; it is not affected by the output sequence. It means that, there is no feedback mechanism to design. Because it has a finite duration and is non-recursive we refer to that filter type as an FIR filter [15]. On the contrary,

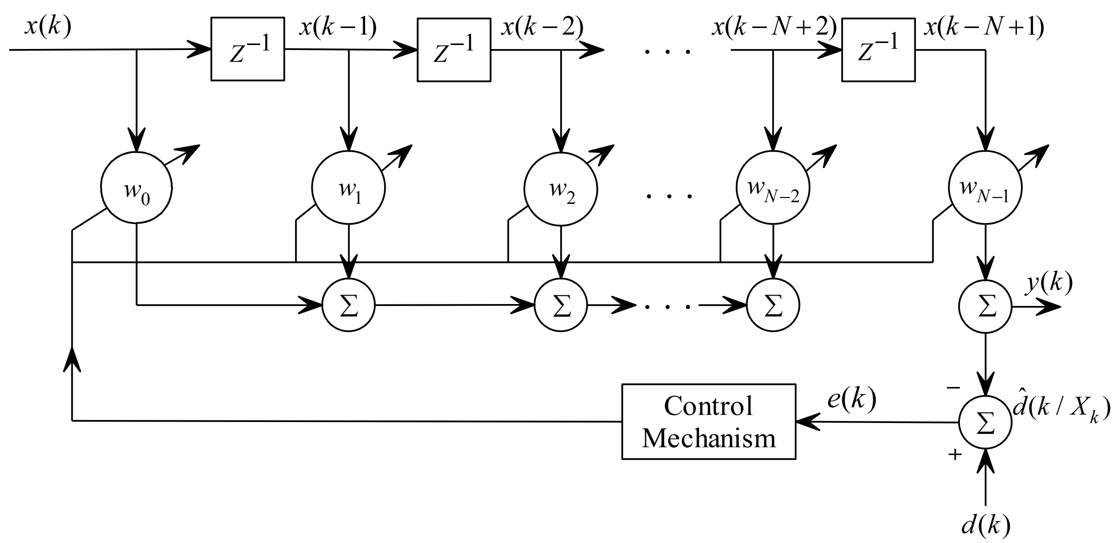


Figure 1.4. Structure of Adaptive Transversal Filter, [16].

in a recursive adaptive filter, the output varies depending on both the input and previous output with a feedback mechanism. This type of filters is called an IIR filter (Fig. 1.5) having an input-output relation as [3]:

$$y(k) = \sum_{i=0}^{M-1} b_i(k)x(k-i) + \sum_{j=1}^{N-1} a_j(k)y(k-j) \tag{1.2}$$

where $a_j(k)$ and $b_i(k)$ are the feedback and forward coefficients, respectively; $\mathbf{M}$ is the number of coefficients of the numerator and $\mathbf{N}$ is the number of coefficients of the denominator .

Adaptive filters have been used in diverse field of signal processing applications successfully. System identification, equalization for communication systems, active noise cancellation, speech processing, radar, sonar, seismology, beamforming, mechanical design, navigation systems and

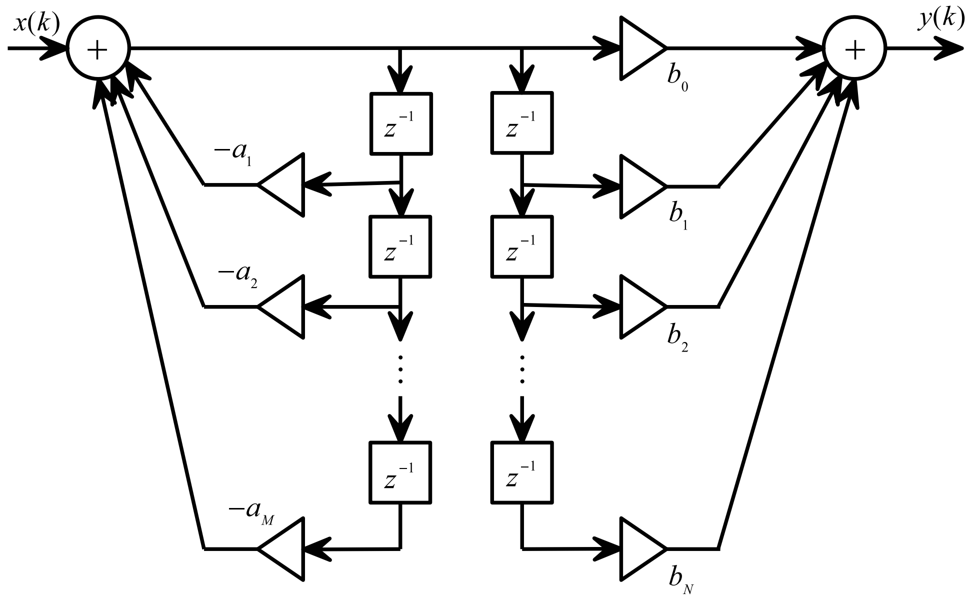


Figure 1.5. Structure of an IIR filter.

biomedical electronics [17, 18] are some examples where the adaptive filtering techniques are used.

## 1.3. Application of Adaptive filters

As we explained above, due to its essential and principal property of time varying and self-adjusting performance, adaptive filters are powerful devices used in numerous applications. In this section a brief introduction will be given about the typical classes of applications where adaptive filtering techniques are used [15, 17, 19].

### 1.3.1. Adaptive Modeling (System Identification)

In this model (Fig. 1.6), the adaptive filter estimates the unknown system's parameters. An input signal $x(k)$ is applied to the unknown system and adaptive filter simultaneously. The filter output $y(k)$ is compared to the desired signal, $d(k)$, that is, the output of the unknown system. Then the error signal $e(k)$ is produced by the difference of output signals and then goes to the adaptive filter as a feedback. The filter adjusts the coefficients until the error signal will be minimum. The unknown system is said to modeled when the error signal is minimized. Adaptive modeling and system identification is used in electrical and mechanical design and network or acoustic echo cancellation.

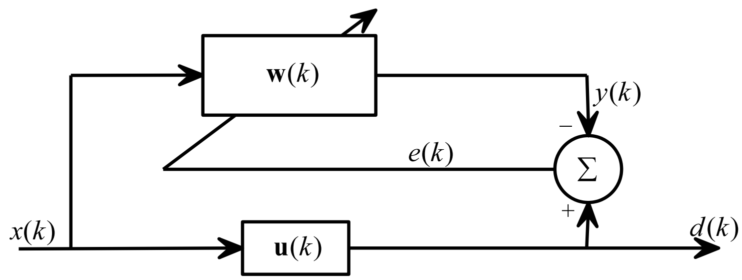


Figure 1.6. Adaptive System Identification Configuration, [16].

### 1.3.2. Adaptive Inverse Modeling (Deconvolution or Equalization)

Inverse modeling refers to the process of removing the unwanted effects of some device or medium on a signal. As shown in the model of channel equalization (Fig. 1.7), the input signal $x(k)$ goes to the adaptive filter after passing through the unknown system. Meanwhile, the desired signal $d(k)$ is obtained as a delayed version of the input signal and compared to the filter output $y(k)$ and then the error signal $e(k)$ is calculated. The main purpose is to minimize the error as in the previous model. This model is used to compensate the distortion of the channel as an equalizer having the inverse transfer function of the channel during data transmission with high speed over communication channels.

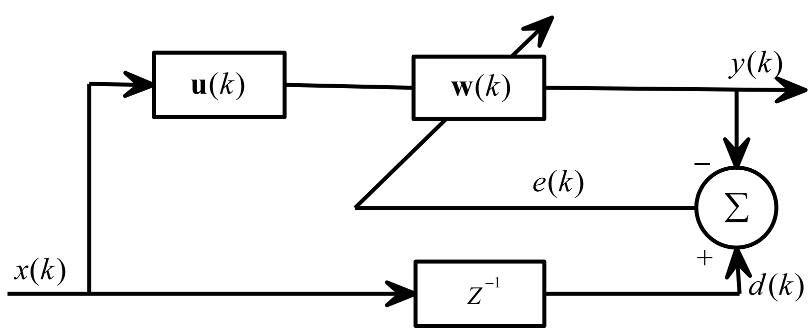


Figure 1.7. Adaptive Channel Equalization Configuration, [16].

### 1.3.3. Adaptive Linear Prediction

A linear predictor (Fig. 1.8) estimates the future values of a signal in a process that explained in the following: After being passed through a delay, the input signal $x(k)$ is sent to the filter. To obtain the error, the output signal $y(k)$ is compared to the desired signal $d(k)$ which is the same as the input signal. After that, the prediction error $e(k)$ is minimized and the input signal is said to be predicted by approaching to the desired one. This model is widely used in speech processing applications such as speech coding in cellular telephony, speech enhancement and speech recognition [4].

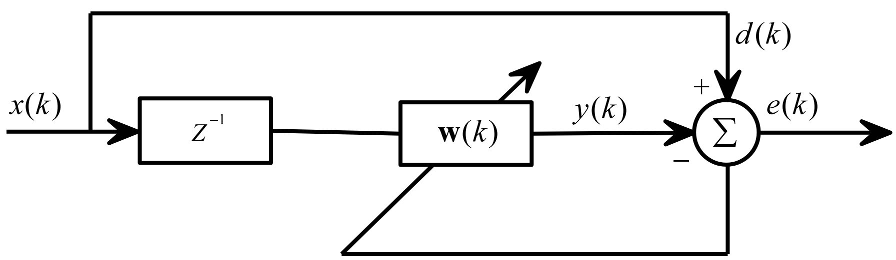


Figure 1.8. Adaptive Linear Prediction, [16].

### 1.3.4. Adaptive Interference Cancellation

In every interference cancelling process such as active noise cancellation, beamforming and vibration control, an interfering signal or noise is required to be cancelled from the desired signal which is corrupted by an uncorrelated interference. The main purpose of an interference cancellation is to estimate the interference and subtract that from the corrupted signal and recover the original one. First, an uncorrelated source of noise signal $N_1(k)$ is used for an input signal $x(k)$ and it is passed through the filter. The desired signal $d(k)$ that contains a signal $s(k)$ which is corrupted by another noise $N_0(k)$, located at a different point, is compared to the filter output (see (Fig. 1.9)). In this case, the adaptive filter provides an estimate $y(k)$ of the noise $N_0(k)$, by exploiting the correlation between $N_0(k)$ and $N_1(k)$ so that the error signal is minimized version of the target signal $s(k)$ [16].

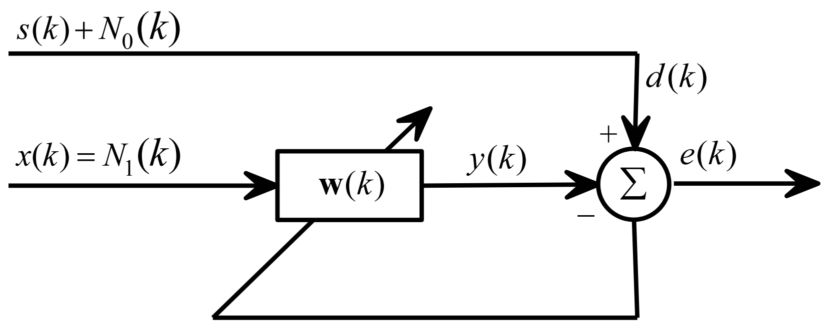


Figure 1.9. Adaptive Noise Cancellation Configuration, [16].

### 1.4. Echo Cancellation

Due to the rapid increase in mobility of human life, wireless communication became very popular and indispensable in the recent years because of its ease of use and flexibility. However, in telecommunications, we often hear about echo problems which degrade the speech quality during conversations. It is caused by two different reasons:

(i) due to the impedance mismatch in network elements,

(ii) due to the reverberation of audio signal during hands-free telephony or teleconferencing.

The solution to the first one is network echo canceller, and to the second one is acoustic echo canceller. In this thesis, we will focus on acoustic echo cancelling (AEC) using adaptive filtering techniques.

Acoustic echo is produced by the interference of transmitted and received signal during conversation because of the existence of microphone and speaker in the same environment [20]. That causes poor quality of communication. The problem is to reduce the influence of the noise produced by the echo on the conversation and increase the quality of speech in telecommunication. Because the acoustic echo signals vary due to several conditions such as the room dimensions or distance between the speaker and the microphone, the adaptive filters are the best solutions for AEC systems. Here the main goal is to generate a replica of the echo signal via an adaptive filter and subtract it from the original signal [21, 22]. If we consider that an echo path is produced by an unknown system, then it is assumed to be as a system identification problem.

In Fig. 1.10 an AEC diagram is shown. The speech signal $\mathbf{x}(k)$ coming from the far-end reaches the near-end speaker in the acoustic room which has the unknown system coefficients $\mathbf{f}(k)$. The output signal $\mathbf{u}(k)$ with background noise is sent to the combiner. Meanwhile, the output signal $y(k)$ is obtained by passing the signal $\mathbf{x}(k)$ through the adaptive filter and it is subtracted from $s(k)$ in the combiner. Finally, the clear speech signal $d(k)$ is obtained and sent to the far-end.

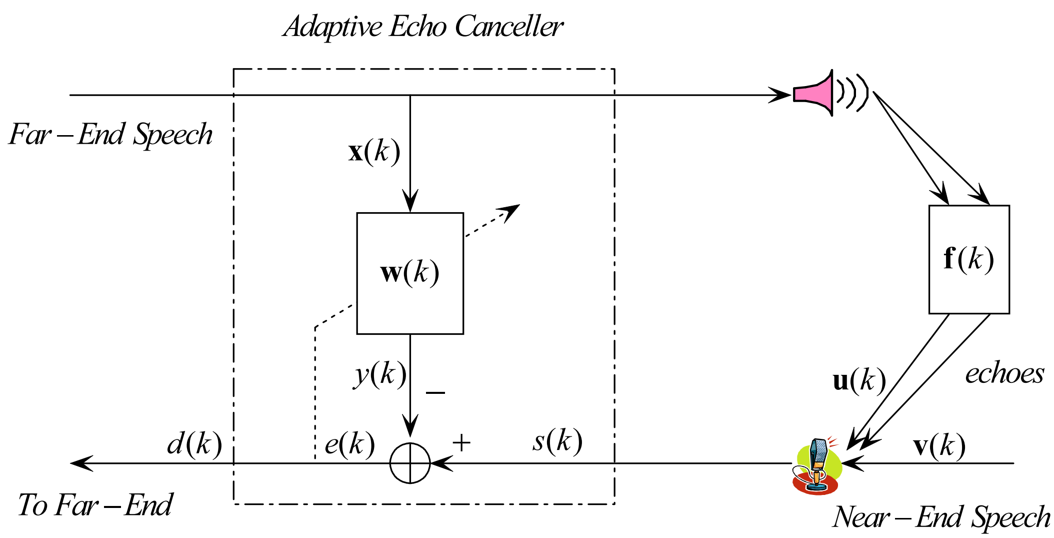


Figure 1.10. Acoustic echo cancellation configuration.

Actually, acoustic echo and reverberation control is one of the most challenging problem in DSP . For instance, in a typical teleconferencing room, the length of the acoustic echo response is in a range of 100 to 400 ms and hence adaptive filters that have long filter length (1024 taps or more) are required to achieve convenient level of echo cancellation [23]. In this work, we propose different adaptive algorithms to obtain the best performance for the AEC.

## 1.5. Sparsity and Sparse Systems

Sparsity is a measurable special feature of a vector or matrix. If most of the entries of a vector are zeros but only a few ones have significant values, the vector is said to be sparse. In the last decades, this property has been very popular for researchers in a wide area of signal processing applications such as adaptive filtering, image processing and statistical estimation [24].

Working on sparse vectors offers great advantages due to the ease of calculation of the most zero entries. Besides, by indicating only the position and value of the non-zero entries, one can store the sparse vector with less space in a digital media. The majority of the work in this area was done to obtain the sparsest vector for signal restoration with less cost. The main interest is to find the vector that has the minimum number of non-zero entries which is defined by $l_0$-norm in the following optimization problem [25]:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ subject to } \mathbf{A}\mathbf{x} = \mathbf{b}. \tag{1.3}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^m$.

In many problems, including sparsity, $l_1$-norm has been used instead of $l_0$-norm because of its simple mathematical formulation. Although, $l_0$-norm has the advantage of more accuracy than $l_1$-norm, it is very difficult to obtain its mathematical expression.

Sparsity is used in adaptive filtering in different manners and offers us many advantages. Actually, in adaptive filtering, many systems are generally assumed to be linear. But in some cases, like in digital TV transmissions channels [26] and echo paths, a few components of the impulse response are significant while the rest is zero or near-zero value [22]. For example, a network echo path has an active region only in a narrow interval with significant values and the rest of the impulse response coefficients is zero or negligible. An acoustic echo path also has similar characteristics as that of the network echo with a little more complicated structure depending on the movement and distance between the microphone and loudspeaker. Eventually, such systems are said to be sparse systems.

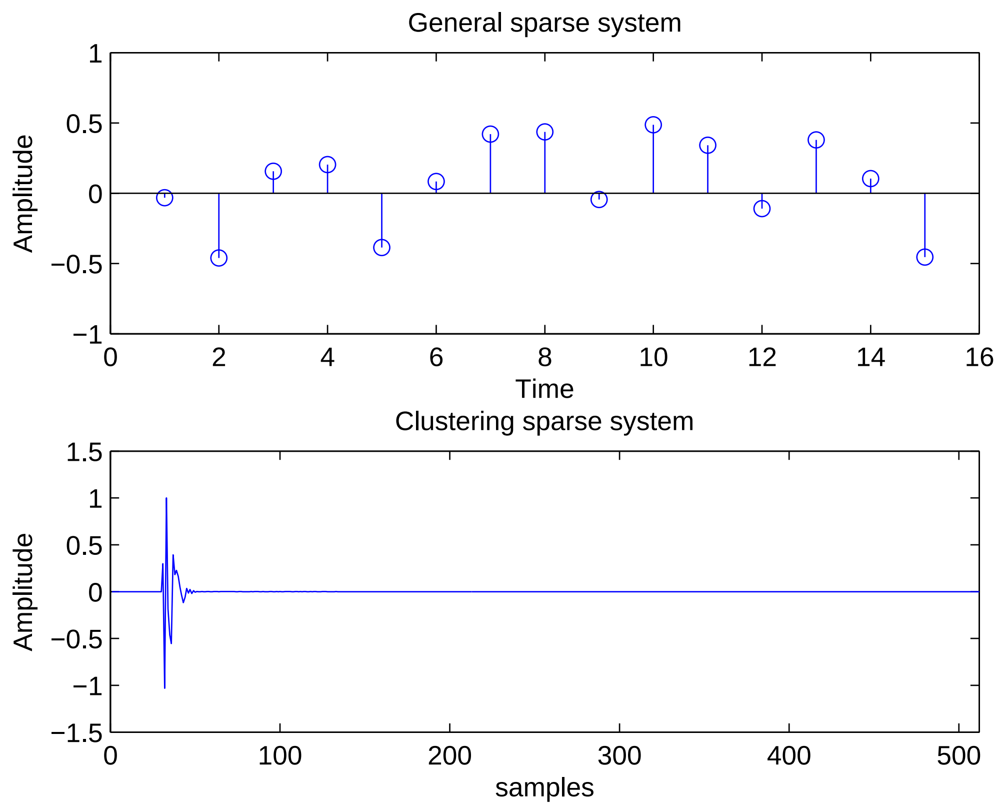

Figure 1.11. Typical sparse systems.

There are two types of sparse systems according to their non-zero components distributions [27]:

(i) general sparse systems have scattered non-zero distribution throughout the system response (Fig. 1.11.a),

(ii) clustering sparse systems consist of one or more clusters of non-zero coefficients along the entire system response (Fig. 1.11.b). A typical example of this type is said to be an echo path.

## 1.6. Adaptive Algorithms

In adaptive filtering theory, a wide variety of adaptive algorithms have been developed. The reason to choose one of them depends on the following factors [28]:

(i) **Rate of convergence** is the number of iteration required to approach the optimum Wiener solution in the mean-square sense for the algorithm. Less number of iterations means better performance.

(ii) **Misadjustment** is the difference between the mean-square-error (MSE) or mean-square-deviation (MSD) of the adaptive algorithm and the Wiener filter. The purpose is to decrease this value to near zero for a better performance.

(iii) **Tracking** is the capability of adaptation to the non-stationary environment to track the statistical variations. However, the tracking performance of the algorithm is affected by the rate of convergence and the steady-state fluctuations because of the noise.

(iv) **Robustness:** A small disturbance due to several internal or external factors can only result in a small estimation error which indicates the stability performance of the algorithm.

(v) **Computational requirements** is the number of mathematical operations which are important for a memory size consequently the cost of the system required to operate the algorithm. The aim is to decrease computational complexity keeping the other criteria at optimum level.

We can compare the performance of different types of adaptive algorithms according to aforementioned criteria. In general, adaptive algorithms being used to estimate a desired signal, are divided into three categories:

(i) Stochastic gradient method.

(ii) Least squares estimation.

(iii) Self-orthogonalizing algorithms.

Before examining these categories, we need to solve the typical estimation problem which is interested in linear optimum filtering.

### 1.6.1. Optimum Filter Design

An optimum filter aims to find an optimal solution according to the pre-described criterion by using optimization theory. Generally speaking, this criterion is to minimize the mean-square of the error which is defined as the difference between the real output of the filter and a desired signal. This type of filter which is roughly defined above, has been developed by Norbert Wiener and it is known as Wiener filter in adaptive filtering theory [3]. It can be described by mathematical notations according to the diagram given in Fig. 1.12. In the figure, the output of the filter is

$$y(k) = \mathbf{w}^T(k)\mathbf{x}(k) \tag{1.4}$$

where $x(k)$ is the input signal and $\mathbf{w}(k)$ is the filter coefficient. So the error signal is the difference between the desired signal and filter output as described below,

$$e(k) = d(k) - y(k). \tag{1.5}$$

#### 1.6.1.1. Wiener-Hopf Equation

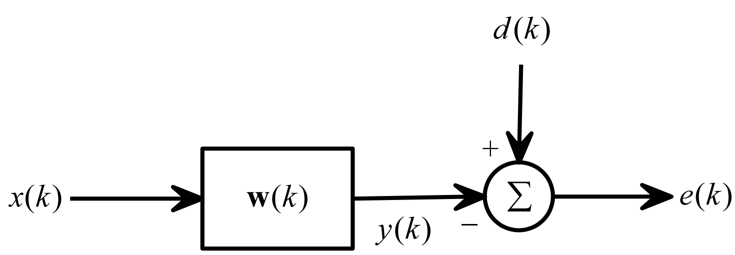


Figure 1.12. Block diagram of Wiener filter.

The MSE can be used as a cost function to find the optimal solution, as stated below,

$$J_{MSE} = E\{[e(k)]^2\}, \tag{1.6}$$

where $\mathrm{E}\{\cdot\}$ represents the expected value [29]. Substituting (1.4) and (1.5) in (1.6) we get,

$$J_{MSE} = \mathrm{E}\{[d^2(k) - 2d(k)\mathbf{x}^T(k)\mathbf{w}(k) + \mathbf{w}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k)]\}. \tag{1.7}$$

The ultimate purpose is to obtain as small as possible error for an optimum filter output that has the closest coefficients to that of the desired signal. This is actually an optimization problem as expressed in the following:

$$\mathbf{w}_{opt} = \arg\min_{\mathbf{w}} J_{MSE}(\mathbf{w}). \tag{1.8}$$

Equation (1.7) has a quadratic form and its optimal solution can be obtained when the cost function has the zero gradient [18], i.e.,

$$\nabla_{\mathbf{w}} J_{MSE}(\mathbf{w}) = \frac{\partial J_{MSE}}{\partial \mathbf{w}} = 0. \tag{1.9}$$

Substituting (1.7) in (1.9) and solving we obtain,

$$E[\mathbf{x}(k)\mathbf{x}^T(k)]\mathbf{w}_{opt} = E[\mathbf{x}(k)d(k)]. \tag{1.10}$$

Letting $E[\mathbf{x}(k)\mathbf{x}^T(k)] = \mathbf{R_x}$ and $E[\mathbf{x}(k)d(k)] = \mathbf{r}_{\mathbf{x}d}$, where $\mathbf{R_x}$ is the input autocorrelation matrix and $\mathbf{r_{xd}}$ is the cross-correlation vector between the input tap vector and the desired response; we can write the equation that is well known as Wiener-Hopf equation as following:

$$\mathbf{R_x w}_{opt} = \mathbf{r_x}_d. \tag{1.11}$$

Note that, the discrete-time stochastic process is wide-sense-stationary (WSS) and the input autocorrelation matrix $\mathbf{R_x}$ is symmetric, toeplitz and positive definite. Solving (1.11), we get the optimal solution for filter output.

$$\mathbf{w}_{opt} = \mathbf{R_x}^{-1}\mathbf{r_x}_d. \tag{1.12}$$

Now we may introduce some of the well-known adaptive methods:

### 1.6.2. Stochastic Gradient Estimation

Finding the ensemble average of the input data is almost impossible for real-time problems. To adjust the filter coefficients, the gradient descent estimation can be applied to the finite measured signal [37]. Since it is applied to a stochastic process then it called as 'stochastic gradient method'.

The steepest descent method uses the gradients of the performance surface to find the approximate solution in an iterative manner by seeking the minimum of the surface. The gradient at any point on the performance surface maybe obtained by differentiating the cost function with respect to the filter coefficient vector. We can find the local minimum by scrolling through the direction of the negative gradient for every iteration as in the following:

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \alpha \frac{\partial J(k)}{\partial \mathbf{w}(k)}, \tag{1.13}$$

where $\alpha$ is a proportionality constant and $J(k)$ is MSE cost function. Taking the gradient of the cost function,

$$\begin{aligned}\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}(k)} &= E\left\{\frac{\partial e^2(k)}{\partial \mathbf{w}(k)}\right\}\\ &= E\left\{2e(k)\frac{\partial e(k)}{\partial \mathbf{w}(k)}\right\}\\ &= E\left\{2e(k)\frac{\partial [d(k)-\mathbf{w}^T(k)\mathbf{x}(k)]}{\partial \mathbf{w}(k)}\right\}\\ &= -2E\{e(k)\mathbf{x}(k)\},\end{aligned} \tag{1.14}$$

then we get,

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu E\{e(k)\mathbf{x}(k)\}. \tag{1.15}$$

where $\mu = 2\alpha$ and is called the step-size of the algorithm. Using the above expectation we can rewrite (1.15) as:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu[\mathbf{r}_{\mathbf{x}d}(k) - \mathbf{R}_{\mathbf{xx}}(k)\mathbf{w}(k)]. \tag{1.16}$$

#### 1.6.2.1. Least-Mean-Square (LMS) Algorithm

Conventional LMS algorithm is a well known algorithm that is widely used in various applications of adaptive filtering due to its simplicity and ease of implementation [17, 38]. It is a stochastic gradient-based algorithm that uses the instantaneous gradient instead of the expected value to find the filter coefficient as follows:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\mathbf{x}(k). \tag{1.17}$$

As shown in (1.17), it has a constant step-size $\mu$. This step-size parameter has a critical effect on the performance of the LMS algorithm [39, 41, 42, 43]. A relatively large step-size means fast convergence but high MSE and a relatively small step-size means slow convergence but low MSE. So that, to improve the performance of the LMS algorithm, many different variable step-size LMS-type algorithms have been developed [44, 45, 46]. LMS-type algorithms will be discussed and analysed deeply in Chapter 3.

### 1.6.3. Least Squares Estimation

In Wiener-Hopf equation, for a random process, we use the expectation operator $E\{\cdot\}$ to find the optimal solution in terms of the ensemble average of autocorrelation matrix or cross-correlation vector. What to do when only finite data sets are available? There is a practical solution although we can not obtain the optimal solution as in the Wiener filter. The summation ($\sum$) can be replaced instead of the expectation operator, $E\{\cdot\}$, in the cost function of optimal filter to attain the cost function of the least squares algorithm [30].

$$J(\mathbf{w}) = \sum_{i=0}^{k} |e(k)|^2. \tag{1.18}$$

where $k = 1, 2, 3, ....$ Substituting (1.4) and (1.5) into (1.18) and minimizing with respect to $\mathbf{w}$, the equation of the least square estimation (LSE) can be derived. There are several solutions to LSE as explained in [30, 31, 32].

#### 1.6.3.1. Recursive Least Square (RLS) Algorithm

Conventional RLS method aims to solve the LSE problem recursively for every sample of the input signals. The cost function in RLS algorithm is given by:

$$J(\mathbf{w}) = \sum_{i=0}^{k} \lambda^{k-i} \varepsilon^2(i), \tag{1.19}$$

where $\varepsilon(i)$ is the a posteriori output error described as

$$\varepsilon(i) = d(i) - y(i)$$

at time $i$ and $\lambda$ is called the forgetting factor because the distinct past information has an increasingly negligible effect on the coefficient updating [12].

Minimizing (1.19) with respect to $\mathbf{w}$, we obtain the optimal vector $\mathbf{w}(k)$ which minimizes the cost function of the RLS error as following:

$$\frac{\partial J_{MSE}}{\partial \mathbf{w}} = -2 \sum_{i=0}^{k} \lambda^{k-i} \varepsilon^2(i) \mathbf{x}(i) [d(i) - \mathbf{x}^T(i)\mathbf{w}(k)]. \tag{1.20}$$

Equating (1.20) to zero and solving we get,

$$\mathbf{w}(k) = \left[ \sum_{i=0}^{k} \lambda^{k-i} \mathbf{x}(i) \mathbf{x}^T(i) \right]^{-1} \sum_{i=0}^{k} \lambda^{k-i} \mathbf{x}(i) d(i). \tag{1.21}$$

Taking the first part of the above equation as the deterministic input correlation matrix $\mathbf{R}_D(k)$ and the second part as the deterministic cross-correlation vector $\mathbf{r}_D(k)$, we get the equation of filter coefficient as

$$\mathbf{w}(k) = \mathbf{R}_D^{-1}(k) \mathbf{r}_D(k). \tag{1.22}$$

The RLS algorithm has a fast convergence even if the eigenvalue spread of the input autocorrelation matrix is large. However, it has a high computational complexity of order $O(N^2)$, because of the calculation of the inverse matrix $\mathbf{R}_D(k)$.

So far, several RLS based algorithms are proposed as in [33, 34, 35, 36]. However, we shall not go more deep for RLS since they are out of the scope of this work.

### 1.6.4. Self-Orthogonalizing Method

The performance of the time-domain adaptive algorithms deteriorate when the input signal is highly correlated. This is because it depends on the eigenvalue spread of the input covariance matrix [13]. The best convergence and consequently learning performance are obtained for equal eigenvalues those are possible only for white noise [47]. Several methods have been proposed to overcome this problem such as discrete Fourier transform (DFT) and discrete cosine transform (DCT) in which the input signal is decorrelated without adding much computations [14, 48]. These types of filters are known as the 'self orthogonalizing adaptive filters' or 'transform (or frequency) domain adaptive filters'.

In a transfer domain LMS (TDLMS) algorithm, the input vector $\mathbf{x}(k)$ is processed by a unitary transform such as DFT or DCT. Once the filter order $N$ is fixed, the transform is simply an $N \times N$ matrix $\mathbf{T}$, with orthonormal rows. And the transformed vector is obtained as

$$\mathbf{X}(k) = \mathbf{T}\mathbf{x}(k), \tag{1.23}$$

where $\mathbf{T}$ is a unitary matrix that is $\mathbf{T}^T\mathbf{T} = \mathbf{T}\mathbf{T}^T = \mathbf{I}$. The filter output is then

$$y(k) = \mathbf{W}^T(k)\mathbf{X}(k), \tag{1.24}$$

and the corresponding estimation error is

$$e\,(k) = d\,(k) - y(k), \tag{1.25}$$

where $\mathbf{W}(k)$ is transform domain filter coefficient vector. We may note that although $\mathbf{X}(k)$ and $\mathbf{W}(k)$ are in the transform domain, the filter output $y(k)$ and the estimation error $e(k)$ are both in time domain. The filter coefficients of TDLMS are then updated by

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \mu \mathbf{D}^{-1} e(k) \mathbf{X}(k), \tag{1.26}$$

where $\mathbf{D}$ is $N \times N$ diagonal matrix whose diagonal elements are the transform domain signal power component $E[|X_i|^2]$ [49] which can be calculated by a recursive equation as

$$P(k+1) = \beta P(k) + (1-\beta)|X(k)|^2. \tag{1.27}$$

The TDLMS algorithm will be discussed and analysed deeply in Chapter 3.

# CHAPTER 2

## PROBLEM FORMULATION AND PROPOSED SOLUTIONS

In this thesis, we deal with the solutions to acoustic echo problem over telecommunication; thereupon the solutions to system identification problem by using adaptive filtering. In this chapter, we state the problem in general, presenting our contributions and summarizing the structure of the thesis.

### 2.1. Problem Statement

It is a fact that with the increasing demand for mobile communications, very fast developments have been staged for the last two decades. Nowadays, hands-free telephony and teleconferencing over a network are extensively used applications for instant communication. However this revealed some extra problems to be solved such as acoustic echo cancellation. Actually, echo problem appeared with the use of the first telephone because of the existence of microphone and speaker in the same environment. Echo path is produced by an unknown system which depends on several conditions such as the noise in the environment, dimensions of the room, reverberation time, temperature and pressure, or distance between the speaker and microphone. Therefore, the adaptive filters are the best solution for AEC systems.

LMS-type filters are well-known types in adaptive filtering technology and they have been used successfully for system identification problems such as AEC. Numerous adaptive algorithms have been developed to obtain the best performance for a clear conversation over telecommunication systems. The main purpose in AEC is to produce a replica of the echo signal and subtract it from the noisy signal to provide a clear sound signal between both sides. This is a quite difficult task since the echo path is very sensitive to different situations as aforementioned and additionally it requires long length adaptive filters with hundreds or even thousands of coefficients [50]. In addition to the large filter length, highly correlated signals deteriorate the performance of the conventional LMS algorithm. Since the LMS algorithm is a gradient descent based algorithm, a constant step-size parameter is used to update the filter coefficients. This step-size parameter has a critical effect on the performance of the algorithm. A large step-size value provides a fast convergence but a high MSE where a small step-size causes slow convergence with low MSE. This trade-off can be set in the favor of both increase in the convergence speed and decrease in misadjustment for the best performance by using a variable step-size. Many different variable step-size LMS-type algorithms have been developed in the field of adaptive filtering [53, 54, 55]. One of them has been used in this thesis to derive our proposed algorithm. It has been proposed in [55] and called as “function controlled variable step-size LMS (FC-VSSLMS) algorithm". The algorithm based on selecting an appropriate function to control the step-size parameter.

Another feature of the echo path is its sparseness. That is, the majority of the coefficients of the acoustic impulse response are zero or near zero where only a few ones have significant values. If the LMS algorithm is modified to exploit the sparsity, a better performance can be obtained for a sparse system. By combining the instantaneous square error with the $l_1$-norm penalty of the coefficient vector in the cost function, a novel sparse LMS algorithm called “zero-attracting LMS (ZA-LMS) algorithm" has been proposed in [56]. A better performance has been obtained when the unknown system is highly sparse. However, by decreasing the sparsity of the system, the MSE got significantly worse than that of the LMS algorithm. To overcome this issue, another algorithm called “reweighted zero-attracting LMS (RZA-LMS) algorithm" has been proposed in the same article. In that algorithm, the cost function of the ZA-LMS algorithm has been modified by changing the $l_1$-norm with the log-sum penalty which behaves more similarly to the $l_0$-norm. Simulations showed that for all degree of sparseness, the MSE is always better than that of the LMS and ZA-LMS algorithms. Even if the system is non-sparse, the performance of the RZA-LMS is approximately the same as that of the LMS algorithm in addition to being better than that of the ZA-LMS algorithm.

## 2.2. Contributions

In this thesis, we proposed new solutions to the system identification problems such as sparseness, high correlation or long filter length for AEC. Initially, we proposed a new algorithm that combines the advantages of variable step-size and $l_0$-norm penalty for sparse system identification in time domain. We called this proposed algorithm as “sparse function controlled variable step-size LMS (SFC-VSSLMS) algorithm". Then we proposed another algorithm based on SFC-VSSLMS in transfer domain to overcome the high correlation problem and called that as “transfer domain sparse function controlled variable step-size LMS (TDSFC-VSSLMS) algorithm". Finally , we proposed the “block sparse function controlled variable step-size LMS (BSFC-VSSLMS) algorithm" to decrease the computation time for a long-length filter. For all these three versions of the algorithm, convergence and the stability analysis are presented and the computational complexities are derived.

All proposed algorithms have been compared to the pre-described algorithms proposed in the same area, based on the performance measure of convergence speed and MSD for different degrees of sparsity and length of the filter in a system identification settings.

It is predictable that, if the RZA-LMS with $l_0$-norm penalty has a better performance (the same as that of the LMS algorithm even if the system is non-sparse) than LMS, so there might be much better performance with the combination of the advantages of the zero-attraction and variable step-size. Indeed in our simulations, we saw that our proposed algorithms have much better performances than that of the others.

## 2.3. Outline of the Thesis

The rest of the thesis is organized as follows:

- In chapter 3, brief reviews of the adaptive algorithms to be compared with our proposed algorithms are introduced.

- In chapter 4, the proposed algorithm is presented in time domain along with its analysis and compared to the other algorithms. Experiments are performed in MATLAB.
- In chapter 5, a transform domain version of the algorithm is proposed, the convergence and stability analysis of the algorithm are derived. The performance of the algorithm is compared to the other algorithms especially for highly correlated input signal. Experiments and simulations are introduced at the end of the chapter.
- In chapter 6, a block implementation of the proposed algorithm is performed. The convergence and stability analysis are presented and experiments realized in MATLAB are simulated at the end of the chapter.
- In chapter 7, a summary and the conclusions are drawn for the overall thesis and possible future research directions on the same area are recommended.

# CHAPTER 3

## REVIEW OF THE RELATED ADAPTIVE ALGORITHMS

### 3.1. Introduction

In this chapter, the adaptive algorithms whose performances will be compared to our proposed algorithms are briefly reviewed.

Starting with the well-known LMS adaptive algorithm, we summarize the VSSLMS algorithm with the advantage of the variable step-size parameter. Thereafter, a VSSLMS-type algorithm, FC-VSSLMS, which is the basis of our proposed algorithm is introduced. Subsequently, the ZA-LMS and RZA-LMS algorithms which will be compared to our proposed algorithms for sparse system identification settings are presented.

### 3.2. Least Mean Square (LMS) Algorithm

The LMS is commonly used algorithm in adaptive filtering because of its ease of implementation, robustness and low computational complexity [15, 39]. It was firstly proposed by Widrow and Hoff in 1960. Basically, the LMS algorithm is a stochastic gradient based algorithm that provides a close approximation to the optimum Wiener-Hoff solution by using instantaneous estimates instead of ensemble averages [32]. Therefore, it is said to be very practical algorithm that calculates the filter coefficients iteratively.

### 3.2.1. Derivation of the LMS Algorithm

The standard LMS algorithm uses the instantaneous error in its cost function instead of the MSE, used in the cost function of the steepest descent algorithm, as described below,

$$J(k) = \frac{1}{2}|e(k)|^2, \tag{3.1}$$

where the instantaneous error $e(k)$ is given by,

$$e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k), \tag{3.2}$$

where $\mathbf{w}(k)$ is the tap-weight vector at instance k as $\mathbf{w}(k) = [w_0, w_1, ..., w_{N-1}]$, $\mathbf{x}(k)$ is the input tap vector as $\mathbf{x}(k) = [x_0, x_1, ..., x_{N-1}]$ and $d(k)$ is desired response related by

$$d\,(k) = \mathbf{h}^T\mathbf{x}\,(k) + n\,(k) \tag{3.3}$$

where $\mathbf{h} = [h_0, \ldots, h_{N-1}]^T$ is the unknown system coefficients with length $N$, $T$ is the transposition operator and $n(k)$ is the additive noise. The aim is to find the approximate filter coefficients by updating the tap-weight vector in the opposite direction of the steepest gradient for each input sample. Thereby, the update equation appears as that of the steepest descent method,

$$\mathbf{w}(k+1) \;\; = \;\; \mathbf{w}(k) - \mu\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}(k)}, \tag{3.4}$$

where $\mu$ is the step-size parameter that controls the convergence rate and the stability of the algorithm. Determining the gradient of the cost function as below,

$$\begin{aligned}\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}(k)} &= \frac{1}{2}\frac{\partial |e(k)|^2}{\partial \mathbf{w}(k)} \\ &= e(k)\frac{\partial [d(k) - \mathbf{w}^T(k)\mathbf{x}(k)]}{\partial \mathbf{w}(k)} \\ &= -e(k)\mathbf{x}(k) \qquad (3.5)\end{aligned}$$

and substituting in (3.4) we finally get the update equation of the tap-weight vector as follows,

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\mathbf{x}(k), \qquad (3.6)$$

where $\mathbf{w}(0)$ is commonly chosen as $\mathbf{w}(0) = 0$ as an initial step of the iteration. The block diagram

Table 3.1. Summary of the LMS algorithm.

| | |
|---|---|
| Filter Output | $y(k) = \mathbf{w}(k)^T\mathbf{x}(k)$ |
| Estimated Error | $e(k) = d(k) - y(k)$ |
| Tap-Weight Vector Adaptation | $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\mathbf{x}(k)$ |

of the LMS filter implementation is shown in Fig. 3.1. The accumulators (ACC) in the figure are used to memorize the previous coefficients [32]. The LMS algorithm is summarized in Table 3.1.

### 3.2.2. Stability Analysis of the LMS algorithm

The LMS algorithm uses the gradient descent to reach the minimum of the error surface in the opposite gradient direction by taking a predefined portion of the gradient of the cost function for every iteration [40]. This descent is controlled by a constant value called step-size. The choice of the step-size is very critical in the performance of the LMS algorithm. There are two stability

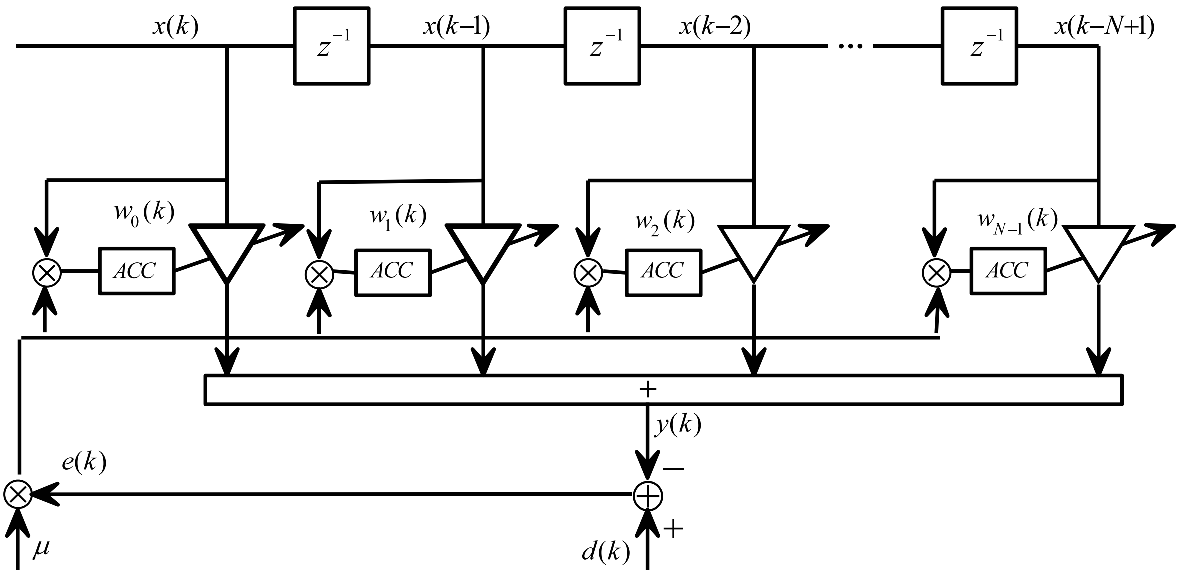


Figure 3.1. Block diagram of adaptive transversal filter for LMS algorithm, [32].

analysis criteria that show the conditions of convergence depending on the step-size value [11].

- Convergence in the mean sense in which the expected value of $\mathbf{w}(k)$ should reach to the Wiener solution while $k$ goes to infinity.
- Convergence in the mean-square sense in which the eventual steady-state value of the mean-square error is finite.

In the analysis, some assumptions listed below should be taken into account in order to handle the analysis smoothly and easily.

- The input signal vectors $\mathbf{x}(k)$ are assumed to be statistically independent.
- The desired signal $d(k)$ and the input tap vector $\mathbf{x}(k)$ are assumed to have Gaussian distributions and to be statistically independent from all previous desired responses.
- The filter-tap vectors at last time depend only on the previous input signal, the previous desired signal and the initial value of the filter-tap weights.
- As a result of the previous assumption, the last filter-tap vector is independent of both input and desired signals.

The misalignment vector $\boldsymbol{\theta}(k)$ of the LMS algorithm is generally defined as,

$$\boldsymbol{\theta}(k) = \mathbf{w}(k) - \mathbf{h}. \tag{3.7}$$

where $\mathbf{w}_0$ is the optimum solution of the filter. The aim of the first criterion is to find out the conditions for convergence in the mean sense. It is clear that the misalignment vector $\boldsymbol{\theta}(k)$ should approach to zero to find the optimum Wiener solution at infinity. It means that the expectation of the misalignment vector must go to zero as $E\{\boldsymbol{\theta}(k)\} = 0$. Subtracting $\mathbf{w}_0$ from both sides of (3.6) and using (3.7), we get the update equation of the misalignment vector as,

$$\begin{aligned} \mathbf{w}(k+1) - \mathbf{h} &= \mathbf{w}(k) - \mathbf{h} + \mu e(k)\mathbf{x}(k) \\ \boldsymbol{\theta}(k+1) &= \boldsymbol{\theta}(k) + \mu e(k)\mathbf{x}(k). \end{aligned} \tag{3.8}$$

Substituting (3.2) in (3.8) and manipulating we get,

$$\begin{aligned} \boldsymbol{\theta}(k+1) &= \boldsymbol{\theta}(k) + \mu\mathbf{x}(k)[\mathbf{x}^T(k)\mathbf{h} - \mathbf{x}^T(k)\mathbf{w}(k)] \\ &= \boldsymbol{\theta}(k) + \mu\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{h} - \mu\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{w}(k) \\ &= \boldsymbol{\theta}(k) - \mu\mathbf{x}(k)\mathbf{x}^T(k)[\mathbf{w}(k) - \mathbf{h}] \\ &= [\mathbf{I} - \mu\mathbf{x}(k)\mathbf{x}^T(k)]\boldsymbol{\theta}(k). \end{aligned} \tag{3.9}$$

If we take the expectation of (3.9) for both sides with the statistical independence we get,

$$E\{\boldsymbol{\theta}(k+1)\} = (\mathbf{I} - \mu\mathbf{R}_{\mathbf{x}})E\{\boldsymbol{\theta}(k)\}. \tag{3.10}$$

Assuming that the autocorrelation matrix, $\mathbf{R_x}$, is symmetric and positive definite, it has the unitary similarity decomposition as,

$$\mathbf{R_x} = \mathbf{Q\Lambda Q}^T \tag{3.11}$$

where $\mathbf{\Lambda}$ is a diagonal matrix that consists of the eigenvalues of $\mathbf{R_x}$ at its main diagonal and $\mathbf{Q}$ is the orthonormal matrix consisting of the eigenvectors of the correspondent eigenvalues [30]. Substituting (3.11) in (3.10) and multiplying both sides by $\mathbf{Q}^T$ we get,

$$\begin{aligned} E\{\mathbf{Q}^T\boldsymbol{\theta}(k+1)\} &= \mathbf{Q}^T(\mathbf{I} - \mu\mathbf{Q\Lambda Q}^T)E\{\boldsymbol{\theta}(k)\} \\ E\{\boldsymbol{\theta}'(k+1)\} &= (\mathbf{I} - \mu\mathbf{\Lambda})E\{\boldsymbol{\theta}'(k)\}, \end{aligned} \tag{3.12}$$

where $\boldsymbol{\theta}'(k) = \mathbf{Q}^T\boldsymbol{\theta}(k)$. Without expectation operator, (3.12) is a recursive equation which can be iterated from zero to $k$. Taking the initial misalignment vector as $\boldsymbol{\theta}(0)$ and writing it in a scalar form we get,

$$\theta_i'(k) = (1 - \mu\lambda_i)^k\theta_i'(0) \qquad \text{for} \quad i = 0, 1, ..., N-1. \tag{3.13}$$

In order the transformed misalignment vector to converge to zero, the step size parameter must be selected so that,

$$\begin{aligned} |1 - \mu\lambda_i| &< 1 \\ -1 < 1 - \mu\lambda_i &< 1 \\ 0 < \mu &< \frac{1}{\lambda_i}. \end{aligned} \tag{3.14}$$

Because $\mu$ is the same for all $i$, convergence is guaranteed only for,

$$0 < \mu < \frac{1}{\lambda_{max}}. \tag{3.15}$$

where $\lambda_{max}$ is the largest eigenvalue of the input autocorrelation matrix, $\mathbf{R_x}$.

The convergence time depends on the following time constant [12]:

$$\tau_i = \frac{1}{2\mu\lambda_i}, \tag{3.16}$$

the largest time constant can be obtained for the smallest eigenvalue as,

$$\tau_{max} = \frac{1}{2\mu\lambda_{min}}, \tag{3.17}$$

combining (3.15) and (3.17) we get,

$$\tau_{max} > \frac{\lambda_{max}}{\lambda_{min}}. \tag{3.18}$$

Now, it is clear that the convergence rate of the LMS algorithm is affected by the eigenvalue spread of the autocorrelation matrix which is given below as,

$$\chi(\mathbf{R}) = \frac{\lambda_{max}}{\lambda_{min}}. \tag{3.19}$$

That is, for a larger eigenvalue spread, the convergence of the LMS algorithm takes longer time.

Since the convergence in the mean-square sense will be performed partially in chapter 4, we don't mention it in this chapter in order for the content of this topic not to be large.

### 3.2.3. Computational Complexity of the LMS Algorithm

The update equation of the LMS algorithm has $O(N)$ complexity [51]. That can be calculated as follows:
For each iteration of $k$ , the LMS update in (3.6) requires one multiplication to compute $\mu e(k)$ and $N$ multiplications to compute $\mu e(k)x(k)$. The computation of $e(k)$ in (3.2) requires $N$ multiplications and $N-1$ additions for the filter output $y(k)$ and one addition for desired response $d(k)$. $N$ additions is required for update of $w(k)$. Then the overall computational complexity of the LMS algorithm is $2N+1$ multiplications and $2N$ additions at each iteration.

## 3.3. Variable Step-Size LMS (VSSLMS) Algorithm

A large step-size value provides a fast convergence but results in a high MSD. On the contrary, a small step-size value provides a low MSD with a slow convergence. In order to facilitate these conflicting requirements, the VSSLMS algorithm was introduced in [52]. If at the beginning of the iteration, the step-size parameter is taken as a large value then the convergence will be fast; however the step-size parameter should be decreased to reduce the misadjustment as the filter coefficient vector approaches the steady-state solution.

Using the above procedure, many different VSSLMS-type algorithms were proposed [45, 53, 54]. In [54], a widely used VSSLMS algorithm was proposed with a variable step-size parameter $\mu(k)$ that is adjusted by the following equation,

$$\mu'(k+1) = \alpha\mu(k) + \gamma e^2(k), \tag{3.20}$$

where $0 < \alpha < 1, \gamma > 0$ and

$$\mu(k) = \begin{cases} \mu_{max} & \text{if } \mu'(k+1) > \mu_{max} \\ \mu_{min} & \text{if } \mu'(k+1) < \mu_{min} \\ \mu'(k+1) & \text{otherwise,} \end{cases} \tag{3.21}$$

where $\mu_{min}$ and $\mu_{max}$ are arbitrary predefined values. It is seen in (3.20) that a large error produces a large step-size; whereas, as the instantaneous error decreases, the step-size is decreased in a suitable value for a lower misadjustment. Table 3.2 summarizes the VSSLMS algorithm.

Table 3.2. Summary of the VSSLMS algorithm.

| | |
|---|---|
| Filter Output | $y(k) = \mathbf{w}(k)^T\mathbf{x}(k)$ |
| Estimated Error | $e(k) = d(k) - y(k)$ |
| Tap-Weight Vector Adaptation | $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k)e(k)\mathbf{x}(k)$ |
| Step-size adaptation | $\mu(k+1) = \alpha\mu(k) + \gamma e^2(k)$ |

### 3.3.1. Function Controlled Variable Step-Size LMS (FC-VSSLMS) Algorithm

In (3.20), updating the constant parameter $\gamma$ plays an important role over the convergence rate and misadjustment of the VSSLMS algorithm. Mang Li et. al. [55] proposed a novel VSSLMS-type algorithm based on a function control which controls this parameter. They considered that if $\gamma$ were as a variable parameter, then it could be increased at the beginning of the iteration for a fast convergence and decreased at the later iterations for a lower misadjustment level. Additionally, it controls the noise power to avoid a large step-size that causes a larger error. The FC-VSSLMS algorithm uses an appropriate function to control $\gamma$ together with an estimated MSE for the step-size parameter to be less affected by the noise interference. The step-size update equation is expressed as follow,

$$\mu(k+1) = \alpha\mu(k) + \gamma_s f(k)\frac{|e(k)|^2}{\widehat{e}^2_{ms}}, \tag{3.22}$$

where $0 < \alpha < 1$, $\gamma_s > 0$ and $\widehat{e}^2_{ms}$ is the estimated MSE defined as,

$$\widehat{e}^2_{ms}(k) = \beta\widehat{e}^2_{ms}(k-1) + (1-\beta)|e(k)|^2, \tag{3.23}$$

where $0 < \beta < 1$, $\gamma_s > 0$. To achieve the best performance, a control function was introduced as a decreasing function to decrease the effect of the instantaneous error in (3.22) on the step-size value as follow,

$$f(k) = \begin{cases} 1/k, & k < L \\ 1/L, & k \geq L, \end{cases} \tag{3.24}$$

where $L$ is a relatively large constant like $L > 100$.

The computational complexity of the FC-VSSLMS algorithm can be calculated as follows:
We know that $2N + 1$ multiplications and $2N$ additions are required for update equation of the LMS algorithm. In the FC-VSSLMS update equation, additional 3 multiplications and 2 additions to compute the $\widehat{e}^2_{ms}(k)$ in (3.23). For update equation of $\mu(k)$ (3.22), we need 5 multiplications and one addition. So overall computational complexity of the algorithm is $2N + 9$ multiplications and $2N + 3$ additions. It is said to be $O(N)$ complexity.

The performance of the FC-VSSLMS algorithm was compared to those of the VSSLMS-type algorithms in terms of excess-MSE. Simulations of different performed experiments showed that it has a superiority over the others. A summary of the FC-VSSLMS algorithm is given in Table 3.3.

Table 3.3. Summary of the FC-VSSLMS algorithm.

| | |
|---|---|
| Filter Output | $y(k) = \mathbf{w}(k)^T\mathbf{x}(k)$ |
| Estimated Error | $e(k) = d(k) - y(k)$ |
| Tap-Weight Vector Adaptation | $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k)e(k)\mathbf{x}(k)$ |
| Step-Size Adaptation | $\mu(k+1) = \alpha\mu(k) + \gamma_s f(k)\frac{\lvert e(k)\rvert^2}{\widehat{e}^2_{ms}}$ |
| Control Function | $f(k) = 1/k$ if $k < L$ else $f(k) = 1/L$ |
| Estimated MSE | $\widehat{e}^2_{ms}(k) = \beta\widehat{e}^2_{ms}(k-1) + (1-\beta)\lvert e(k)\rvert^2$ |

### 3.4. Adaptive Algorithms for Sparse Systems

The performance of the adaptive algorithms deteriorate when the number of coefficients of the filter-tap vector is relatively large as in many applications such as echo cancellation [57]. However, the impulse response of the system in an echo canceller can be modeled as sparse; that has only a few non-zero coefficients [58]. This property is not exploited by the conventional LMS algorithm for the sparse system identification. To obtain a better performance, the simplicity and robustness of the LMS algorithm are combined with the advantage of the sparsity of the system and and hence many of sparse LMS-type algorithms were proposed in the literature [59, 60, 61].

A common method to exploit the sparsity of the system is modifying the cost function by adding $l_1$-norm penalty [62] or $l_0$-norm penalty to the square of the instantaneous error [63]. Even though the sparsity can be best exploited by the $l_0$-norm, it is difficult to express it mathematically compared to the $l_1$-norm. When the constraint on the cost function is introduced with the $l_1$-norm penalty, there will exist a better performance for sparse system identification depending on the sparseness level of the system. However, the performance decreases while the sparsity is decreased; even it may provide worse performance than that of the standard LMS algorithm at low sparsity levels. On the other hand, with the $l_0$-norm penalty, a better performance is obtained for all sparsity levels so that in the worst case the performance of the algorithm here is approximately the same as that of the LMS algorithm.

### 3.4.1. Zero Attracting LMS (ZA-LMS) Algorithm

The well-known algorithms with the cost function including $l_1$-norm penalty or $l_0$-norm penalty were first proposed by Chen et. al. [56] inspired by successes of the least absolute shrinkage and selection operator (LASSO) [64] and compressive sensing (CS) [65, 66, 67]. They performed two different experiments. The first one was provided by an algorithm named as “zero attracting LMS (ZA-LMS)” in which a new cost function defined by combining the instantaneous squared error with the $l_1$-norm penalty of the coefficient vector as follows,

$$J_1(k) = \frac{1}{2}e^2(k) + \gamma\|\mathbf{w}(k)\|_1, \tag{3.25}$$

where $\gamma$ is a constant that controls the effect of $l_1$-norm constraint. As in the standard LMS algorithm, the minimum of the cost function can be found iteratively by using the gradient descent method as,

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) - \mu\frac{\partial J_1(k)}{\partial \mathbf{w}(k)} \\ &= \mathbf{w}(k) - \rho \mathrm{sgn}[\mathbf{w}(k)] + \mu e(k)\mathbf{x}(k), \end{aligned} \tag{3.26}$$

where $\rho = \mu\gamma$ and $\mathrm{sgn}(\cdot)$ is the sign function defined as:

$$sgn(x) = \begin{cases} \frac{x}{|x|} & \text{if } x \neq 0 \\ 0 & \text{if } x = 0. \end{cases} \tag{3.27}$$

The extra term $-\rho\mathrm{sgn}(\mathbf{w}(n))$ in the update equation of the ZA-LMS algorithm is called the zero attractor. If the update equation in (3.26) is examined thoroughly, it is seen that the zero attractor always attracts the small-valued coefficients to zero under the effect of the $\rho$ parameter that controls the zero attractor strength. Also, the zero attractor will speed-up the convergence for a highly sparse system in which the most of the coefficients are zero. Table 3.4 gives the summary of the ZA-LMS algorithm.

Table 3.4. Summary of the ZA-LMS algorithm.

| | |
|---|---|
| Filter Output | $y(k) = \mathbf{w}(k)^T \mathbf{x}(k)$ |
| Estimated Error | $e(k) = d(k) - y(k)$ |
| Tap-Weight Vector Adaptation | $\mathbf{w}(k+1) = \mathbf{w}(k) - \rho \mathrm{sgn}(\mathbf{w}(k)) + \mu e(k)\mathbf{x}(k)$ |

In simulations of aforementioned work, the ZA-LMS algorithm demonstrated a better performance than that of the standard LMS algorithm when the the sparseness of the system is approximately 94%. However, when the system sparsity level is decreased to 50%, the performance degrades considerably compared to that of the LMS algorithm. That is why $l_1$-norm penalty is not so efficient for a low sparse system in a system identification settings.

### 3.4.2. Reweighted Zero Attracting LMS (RZA-LMS) Algorithm

By using the $l_0$-norm constraint instead of $l_1$-norm in the cost function of the LMS algorithm, a robust algorithm was derived for sparse system identification. Motivated by reweighting in compressive sampling, a new algorithm was proposed named as "reweighted zero attracting LMS (RZA-LMS)" which has a modified cost function introduced by combining the instantaneous squared error with the log-sum penalty that behaves more likely as the $l_0$-norm [65] as follow,

$$J_2(k) = \frac{1}{2}e^2(k) + \gamma' \sum_{i=0}^{N-1} \log\left(1 + \frac{|\mathbf{w}_i|}{\varepsilon'}\right), \tag{3.28}$$

where $\gamma'$ and $\varepsilon'$ are appropriate positive control parameters. By using the gradient descent method, the update equation of the RZA-LMS algorithm can be derived as,

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) - \mu \frac{\partial J_2(k)}{\partial \mathbf{w}(k)} \\ &= \mathbf{w}(k) - \frac{\rho \mathrm{sgn}(\mathbf{w}(k))}{1 + \varepsilon |\mathbf{w}(k)|} + \mu e(k)\mathbf{x}(k). \end{aligned} \tag{3.29}$$

where $\rho = \mu\gamma'/\varepsilon'$ and $\varepsilon = 1/\varepsilon'$. Unlike the zero attractor that attracts all coefficients uniformly to zero in the ZA-LMS, the RZA-LMS has different zero attractors for different coefficients at any sparsity level. As can be seen in (3.29), the reweighted zero attractor selectively shrinks the small-valued coefficients. As a result, its performance is high even at low sparsity levels. A summary of the RZA-LMS algorithm is given in Table 3.5.

Table 3.5. Summary of the RZA-LMS algorithm.

| | |
|---|---|
| Filter Output | $y(k) = \mathbf{w}(k)^T\mathbf{x}(k)$ |
| Estimation Error | $e(k) = d(k) - y(k)$ |
| Tap-Weight Vector Adaptation | $\mathbf{w}(k+1) = \mathbf{w}(k) - \frac{\rho \operatorname{sgn}(\mathbf{w}(k))}{1+\varepsilon\lvert\mathbf{w}(k)\rvert} + \mu e(k)\mathbf{x}(k)$ |

Simulations showed that, the RZA-LMS algorithm has a superiority over both the ZA-LMS and standard LMS algorithms even at low sparsity levels. When the system is non-sparse, the performance of the RZA-LMS algorithm is not worse than that of the LMS algorithm.

# CHAPTER 4

## PROPOSED ALGORITHM IN TIME DOMAIN

### 4.1. Introduction

As we mentioned in Chapter 3, one of the most popular algorithms in adaptive signal processing is LMS algorithm due to its simplicity [53]. Its basic characteristic is that the weights of coefficients are obtained by the stochastic gradient descent method using a constant step-size value. This step-size usually provides a high steady-state error if it is relatively large and a low convergence rate if it is relatively small. This trade-off is more prominent in a high-level measurement noise or if the input signal is highly correlated [55]. To overcome these problems, several adaptive algorithms have been developed [69, 70, 71, 72] in the recent years.

In [53], a VSSLMS algorithm is proposed. The algorithm regulates the step-size to allow the adaptive filter to track changes in the system as well as produce a small steady-state error. In [72], a more robust variable step-size LMS (R-VSSLMS) algorithm is proposed. The algorithm addresses the problem of the VSSLMS algorithm by proposing a new approach to adjust the VSSLMS based on a quotient of filtered quadratic error. However, as we noticed in our experiments, this algorithm converges very slowly when the system is sparse. In [55], a FC-VSSLMS algorithm is proposed. The algorithm is based on selecting an appropriate function to control the step size. The FC-VSSLMS algorithm is an enhanced version of the VSSLMS algorithm which guarantees faster convergence most of the time.

All of the aforementioned algorithms have been successfully implemented in system identification settings. However, in many scenarios (i.e., digital TV transmission channel [26] and echo paths [73]), impulse responses of unknown systems can be assumed to be sparse; that contains only a few non-zero coefficients.

Chen et al. [56] proposed the RZA-LMS algorithm in a system identification setting. Using such prior sparse information can improve the performance of the adaptive filter. Salman et. al. proposed a sparse adaptive filtering algorithm in [62]. The algorithm has shown high performance in highly sparse systems. However, the algorithm has poor performance if the system is relatively low sparse.

In this chapter, we propose a new algorithm that provides a high performance even if the system sparsity is relatively low. The proposed algorithm imposes an approximate penalty of the $l_0$-norm in the cost function of the FC-VSSLMS algorithm.

The chapter is organized as follows: In Section 4.2, a brief review of the FC-VSSLMS algorithm is provided and the proposed algorithm is derived. In Section 4.3, the convergence analysis of the proposed algorithm is presented and its stability condition is derived. In Section 4.4, simulation results that compare the performance of the proposed algorithm to other algorithms are provided and discussed.

## 4.2. The Proposed Algorithm

Before introducing the proposed algorithm, we may have a brief review of the FC-VSSLMS algorithm.

Consider a linear system with input-tap vector $\mathbf{x}\left(k\right) = [x_0, ..., x_{N-1}]^T$ and output $d(k)$ related by

$$d\left(k\right) = \mathbf{h}^T\mathbf{x}\left(k\right) + n\left(k\right) \tag{4.1}$$

where $\mathbf{h} = [h_0, \ldots, h_{N-1}]^T$ is the unknown system coefficients with length $N, T$ is the transposition operator and $n(k)$ is the additive noise. The noise samples $n(k)$ are assumed to be independent and identically distributed (i.i.d.) with zero mean and variance of $\sigma_n^2$. Also, the input data sequence

$x(k)$ and the additive noise samples $n(k)$ are assumed to be independent.

The cost function of the FC-VSSLMS algorithm is given by,

$$J\left(\mathbf{w}\left(k\right)\right) = \frac{1}{2}\left|e\left(k\right)\right|^{2}, \tag{4.2}$$

where $\mathbf{w}(k)$ is filter-tap vector at time $k$ and $e(k)$ is the instantaneous error and given by

$$e\left(k\right) = d\left(k\right) - \mathbf{w}^{T}\left(k\right)\mathbf{x}\left(k\right). \tag{4.3}$$

The update equation of the FC-VSSLMS algorithm can be written as

$$\begin{aligned} \mathbf{w}\left(k+1\right) &= \mathbf{w}\left(k\right) - \mu\left(k\right)\frac{\partial J\left(\mathbf{w}\left(k\right)\right)}{\partial \mathbf{w}\left(k\right)} \\ &= \mathbf{w}\left(k\right) + \mu\left(k\right)e\left(k\right)\mathbf{x}\left(k\right) \end{aligned} \tag{4.4}$$

where $\mu\left(k\right)$ is the variable step-size parameter and given by (3.22).

However, the performance of such algorithm can be further improved if the unknown system is assumed to be sparse. This improvement can be achieved by modifying the cost function in (4.2) to become

$$J\left(\mathbf{w}\left(k\right)\right) = \frac{1}{2}\left|e\left(k\right)\right|^{2} + \xi\|\mathbf{w}\left(k\right)\|_{0} \tag{4.5}$$

where $\|.\|_0$ denotes the $l_0$-norm of the weights vector and $\xi$ is a small positive constant. The main obstacle in the cost function given in (4.5), is deriving the $l_0$-norm term with respect to $\mathbf{w}\left(k\right)$.

Thereby, we may use the $l_0$-norm approximation [74] as given below:

$$\|\mathbf{w}\,(k)\,\|_0 \simeq \sum_{k=0}^{N-1} \left(1 - e^{-\lambda|\mathbf{w}(k)|}\right) \tag{4.6}$$

where $\lambda$ is a positive number. Thereby, the filter-tap update equation is appeared as that of the method of steepest descent,

$$\mathbf{w}(k+1) \;\; = \;\; \mathbf{w}(k) - \mu \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}(k)}. \tag{4.7}$$

Deriving (4.5) with respect to $\mathbf{w}\,(k)$ and substituting in (4.7) yields

$$\mathbf{w}\,(k+1) = \mathbf{w}\,(k) + \mu\,(k)\,e\,(k)\,\mathbf{x}\,(k) - \rho(k)sgn\left[\mathbf{w}\,(k)\right]e^{-\lambda|\mathbf{w}(k)|} \tag{4.8}$$

where $\rho(k) = \mu(k)\xi\lambda$ and it should be a small positive number in order to guarantee convergence to the global optima.

The term $-\rho(k)sgn\left[\mathbf{w}\,(k)\right]e^{-\lambda|\mathbf{w}(k)|}$ in (4.8) imposes an attraction to zero on small coefficients (zero or near to zero coefficients). Particularly, if the filter weight coefficient is positive, it will decrease and if it is negative, it will increase and converge to zero faster. On the other hand, during the update process, if the coefficient to be attracted to zero is relatively high, it goes faster than the same coefficient in another zero attracting algorithm by the virtue of exponential $\left(e^{-\lambda|x|} \geq x \text{ with a proper selection of } \lambda \text{ and relatively small values of } x\right)$. In (4.8), to reduce the computational complexity we may use the second order Taylor series expansion of the exponential function.

## 4.3. Computational Complexity of the Proposed Algorithm

The computational complexity of the zero attractor, $\rho(k)sgn\left[\mathbf{w}\left(k\right)\right]e^{-\lambda|\mathbf{w}(k)|}$, requires $N$ multiplications for $\lambda\left|\mathbf{w}\left(k\right)\right|$, $N$ additions for $e^{-\lambda|\mathbf{w}(k)|}$ (taking the first two terms of taylor series), $N$ multiplications for $\rho(k)sgn\left[\mathbf{w}\left(k\right)\right]$, $N$ multiplications for one by one element product of $\lambda\left|\mathbf{w}\left(k\right)\right|$ by $\rho(k)sgn\left[\mathbf{w}\left(k\right)\right]$ and $N$ comparisons for $sgn\left[\mathbf{w}\left(k\right)\right]$. So, overall complexity of the zero attractor is $3N$ multiplications, $N$ additions and $N$ comparisons. Taking into account the complexity of the FC-VSSLMS algorithm, the total computational complexity of the proposed algorithm requires $5N+9$ multiplications, $3N+3$ additions and $N$ comparisons, that is, $(O(N)$ complexity.

## 4.4. Convergence Analysis of the Proposed Algorithm

In this section we perform the mean-square convergence analysis of the proposed algorithm for independent and identically distributed (i.i.d) zero-mean Gaussian input signal $x(k)$ and a zero-mean white noise $n(k)$. The misalignment vector of the LMS algorithm [39] is usually defined as

$$\boldsymbol{\theta}\left(k\right)=\mathbf{w}\left(k\right)-\mathbf{h} \tag{4.9}$$

where $\mathbf{h}$ is the impulse response of the unknown system and the data vector $\mathbf{x}(k)$ is assumed to be independent of the error vector $\boldsymbol{\theta}(k)$. The mean and covariance of the misalignment vector are defined respectively, as

$$\boldsymbol{\varphi}(k)=E\left\{\boldsymbol{\theta}(k)\right\}, \tag{4.10}$$

$$\boldsymbol{\Gamma}(k)=E\left\{\mathbf{s}(k)\mathbf{s}^{T}(k)\right\} \tag{4.11}$$

where $\mathbf{s}(k)$ is the zero-mean misalignment vector defined as

$$\mathbf{s}(k) = \boldsymbol{\theta}(k) - E\left\{\boldsymbol{\theta}(k)\right\}. \tag{4.12}$$

In this analysis, the MSD is taken as a figure of merit and the instantaneous MSD is defined as

$$z(k) = E\left\{\|\boldsymbol{\theta}(k)\|_2^2\right\} = \sum_{i=0}^{N-1} \Psi_i(k) \tag{4.13}$$

where $\Psi_i(k)$ denotes the $i$-th tap MSD and is defined with respect to the $i$-th element of $\boldsymbol{\theta}(k)$ as,

$$\Psi_i(k) = E\{\theta_i^2(k)\} = \Gamma_{ii}(k) + \varphi_i^2(k), \quad i = 0, ..., N-1. \tag{4.14}$$

$\Gamma_{ii}(k)$ corresponds to the $i$-th diagonal element of the auto-covariance matrix $\boldsymbol{\Gamma}(k)$ and $\varphi_i(k)$ is the $i$-th element of $\boldsymbol{\varphi}(k)$. In the following, the MSD is evaluated using the derivation of $\varphi_i(k)$ and $\Gamma_{ii}(k)$.

Combining (4.1), (4.3), (4.8) and (4.9) and considering the independence assumption [39], the update equation of the misalignment vector becomes:

$$\begin{aligned}
\boldsymbol{\theta}(k+1) &= \mathbf{w}(k+1) - \mathbf{h} \\
&= \mathbf{w}(k) + \mu(k)e(k)\mathbf{x}(k) - \rho(k)sgn[\mathbf{w}(k)]e^{-\lambda|\mathbf{w}(k)|} - \mathbf{h} \\
&= \boldsymbol{\theta}(k) + \mu(k)[\mathbf{h}^T\mathbf{x}(k) + n(k) - \mathbf{w}^T(k)\mathbf{x}(k)]\mathbf{x}(k) \\
&\qquad\qquad - \rho(k)sgn[\mathbf{w}(k)]e^{-\lambda|\mathbf{w}(k)|}
\end{aligned}$$

and we get

$$\begin{aligned}\boldsymbol{\theta}(k+1) &= \boldsymbol{\theta}(k) - \mu(k)\,\mathbf{x}(k)\,\mathbf{x}^T(k)\,\boldsymbol{\theta}(k) + \mu(k)\,\mathbf{x}(k)\,n(k) \\ &- \rho(k) sgn\left[\mathbf{w}(k)\right] e^{-\lambda|\mathbf{w}(k)|}. \end{aligned} \tag{4.15}$$

Taking the expectation of (4.15)

$$\begin{aligned}E\{\boldsymbol{\theta}(k+1)\} &= E\{\boldsymbol{\theta}(k)\} - E\left\{\mu(k)\,\mathbf{x}(k)\,\mathbf{x}^T(k)\,\boldsymbol{\theta}(k)\right\} + E\{\mu(k)\,\mathbf{x}(k)\,n(k)\} \\ &- E\left\{\rho(k) sgn\left[\mathbf{w}(k)\right] e^{-\lambda|\mathbf{w}(k)|}\right\}\end{aligned}$$

then we get,

$$\boldsymbol{\varphi}(k+1) = \boldsymbol{\varphi}(k) - \mu(k)\,E\left\{\mathbf{x}(k)\,\mathbf{x}^T(k)\right\}\boldsymbol{\varphi}(k) - \rho(k) E\left\{sgn\left[\mathbf{w}(k)\right] e^{-\lambda|\mathbf{w}(k)|}\right\}. \tag{4.16}$$

Subtracting (4.16) from (4.15) and adding $\mu(k)\mathbf{x}(k)\mathbf{x}^T(k)\boldsymbol{\varphi}(k)$ to both sides yields

$$\begin{aligned}\boldsymbol{\theta}(k+1) - \boldsymbol{\varphi}(k+1) &= \boldsymbol{\theta}(k) - \boldsymbol{\varphi}(k) + \mu(k)\,\mathbf{x}(k)\,\mathbf{x}^T(k)\left[-\boldsymbol{\theta}(k) + \boldsymbol{\varphi}(k)\right] \\ &+ \mu(k)\left[E\left\{\mathbf{x}(k)\,\mathbf{x}^T(k)\right\} - \mathbf{x}(k)\,\mathbf{x}^T(k)\right]\boldsymbol{\varphi}(k) \\ &+ \rho(k)\left[E\left\{sgn[\mathbf{w}(k)] e^{-\lambda|\mathbf{w}(k)|}\right\} - sgn[\mathbf{w}(k)] e^{-\lambda|\mathbf{w}(k)|}\right] \\ &+ \mu(k)\,\mathbf{x}(k)\,n(k),\end{aligned}$$

then we obtain,

$$\mathbf{s}(k+1) = \mathbf{A}(k)\mathbf{s}(k) + \mu(k)\mathbf{B}(k)\boldsymbol{\varphi}(k) + \rho(k)\mathbf{c}(k) + \mu(k)\mathbf{x}(k)n(k), \tag{4.17}$$

where

$\mathbf{A}(k) = \mathbf{I} - \mu\left(k\right)\mathbf{x}\left(k\right)\mathbf{x}^T\left(k\right)$,

$\mathbf{B}(k) = E\left\{\mathbf{x}\left(k\right)\mathbf{x}^T\left(k\right)\right\} - \mathbf{x}\left(k\right)\mathbf{x}^T\left(k\right)$,

$\mathbf{c}(k) = E\{sgn[\mathbf{w}(k)]e^{-\lambda|\mathbf{w}(k)|}\} - sgn[\mathbf{w}(k)]e^{-\lambda|\mathbf{w}(k)|}$

where $\boldsymbol{\theta}\left(k\right)$, $\mathbf{x}\left(k\right)$ and $n(k)$ are independent.

Now, we can calculate $\boldsymbol{\Gamma}(k+1)$ as follows:

$$\begin{aligned}
\boldsymbol{\Gamma}(k+1) &= E\left\{\mathbf{s}(k+1)\mathbf{s}^T(k+1)\right\} \\
&= E\left\{\mathbf{A}(k)\mathbf{s}(k)\mathbf{s}^T(k)\mathbf{A}^T(k)\right\} + E\left\{\mu^2(k)\mathbf{B}(k)\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k)\mathbf{B}^T(k)\right\} \\
&+ E\left\{\mu^2(k)\mathbf{x}(k)n(k)n^T(k)\mathbf{x}^T(k)\right\} + \rho(k)E\left\{\mathbf{A}(k)\mathbf{s}(k)\mathbf{c}^T(k)\right\} \\
&+ \rho(k)E\left\{\mathbf{c}(k)\mathbf{s}(k)\mathbf{A}^T(k)\right\} + \rho^2(k)E\left\{\mathbf{c}(k)\mathbf{c}^T(k)\right\} \qquad (4.18)
\end{aligned}$$

Calculating the expectation of each part we get,

$$\begin{aligned}
\boldsymbol{\Gamma}(k+1) &= (1 - 2E\left\{\mu(k)\right\}\sigma_x^2 + 2E\left\{\mu^2(k)\right\}\sigma_x^4)\boldsymbol{\Gamma}(k) + E\left\{\mu^2(k)\right\}\sigma_x^4 tr[\boldsymbol{\Gamma}(k)]I_N \\
&+ E\left\{\mu^2(k)\right\}\sigma_x^4\left[\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k) + tr[\boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k)]I_N\right] \\
&+ 2\left[1 - E\left\{\mu(k)\right\}\sigma_x^2\right]E\left\{\mathbf{w}(k)\mathbf{c}^T(k)\right\} \\
&+ E\left\{\mu^2(k)\right\}\sigma_x^2\sigma_n^2 + \rho^2(k)E\left\{\mathbf{c}(k)\mathbf{c}^T(k)\right\}. \qquad (4.19)
\end{aligned}$$

The calculation (4.19) is obtained by using the fourth moment of input [75] as well as using the symmetric behavior of the covariance matrix $\boldsymbol{\Gamma}(k)$. Finding the trace of (4.19),

$$\begin{aligned}
tr\left\{\boldsymbol{\Gamma}(k+1)\right\} &= [1 - 2E\left\{\mu(k)\right\}\sigma_x^2 + (N+2)E\{\mu^2(k)\} \times \sigma_x^4]tr\{\boldsymbol{\Gamma}(k)\} + (N+1)E\{\mu^2(k)\}\sigma_x^4 \\
&\times \boldsymbol{\varphi}(k)\boldsymbol{\varphi}^T(k) + N \cdot E\left\{\mu^2(k)\right\}\sigma_x^2\sigma_n^2 + \rho^2(k)E\left\{\mathbf{c}^T(k)\mathbf{c}(k)\right\} + 2E\{\rho(k)\} \\
&\times \left[1 - E\left\{\mu(k)\right\}\sigma_x^2\right]E\left\{\mathbf{w}(k)\mathbf{c}^T(k)\right\}. \qquad (4.20)
\end{aligned}$$

In (4.20), $\boldsymbol{\varphi}(k), \mathbf{c}(k)$, $E\{\mathbf{w}(k)\}$ and thus $E\{\mathbf{w}(k)\mathbf{c}^T(k)\}$ are all bounded [78]. Therefore, the adaptive algorithm will be stable if,

$$\left|1 - 2E\left\{\mu(k)\right\}\sigma_x^2 + (N+2)E\left\{\mu^2(k)\right\}\sigma_x^4\right| < 1, \tag{4.21}$$

as $k \rightarrow \infty$, $E\{\mu^2(k)\} \approx [\mu(k)]^2$, then (4.21) is simplified to,

$$0 < \mu(\infty) < \frac{2}{(N+2)\sigma_x^2}. \tag{4.22}$$

Equation (4.22) shows that the convergence of the proposed algorithm is guaranteed and its stability criteria is similar to that of the algorithm in [79]. Hence, the performance gain of the proposed algorithm is due to the control function.

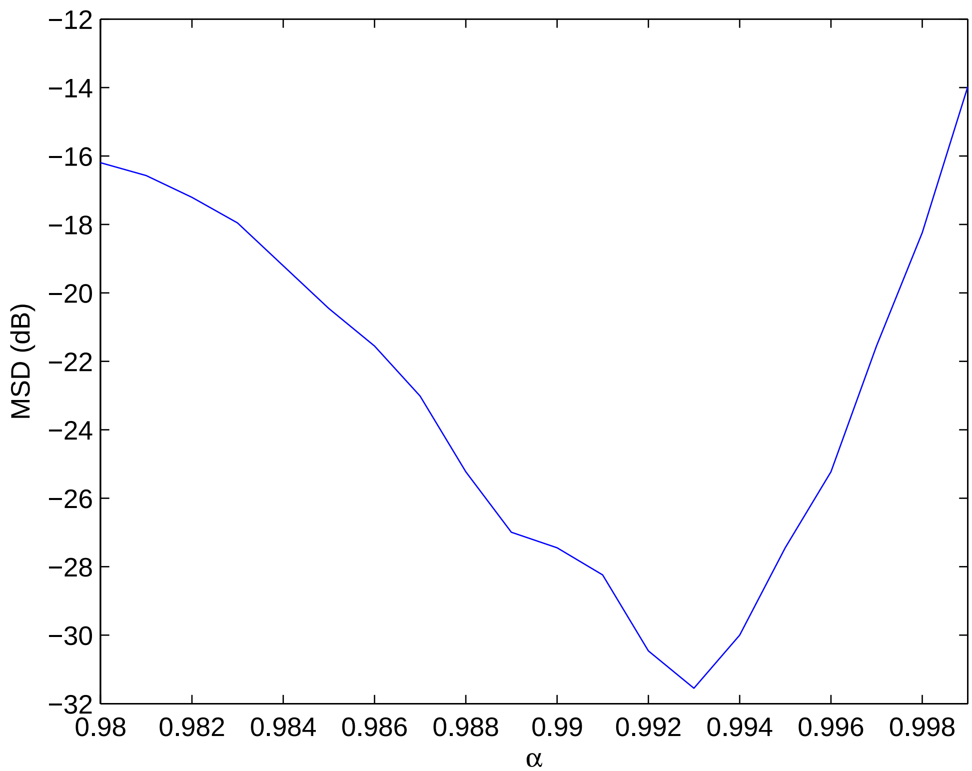


Figure 4.1. MSD vs. $\alpha$.

In this section, the performances of the proposed algorithm are compared to those of the VSSLMS, RVSSLMS, FC-VSSLMS and RZA-LMS algorithms in sparse and non-sparse system identification models. All the experiments are implemented with 200 independent runs.

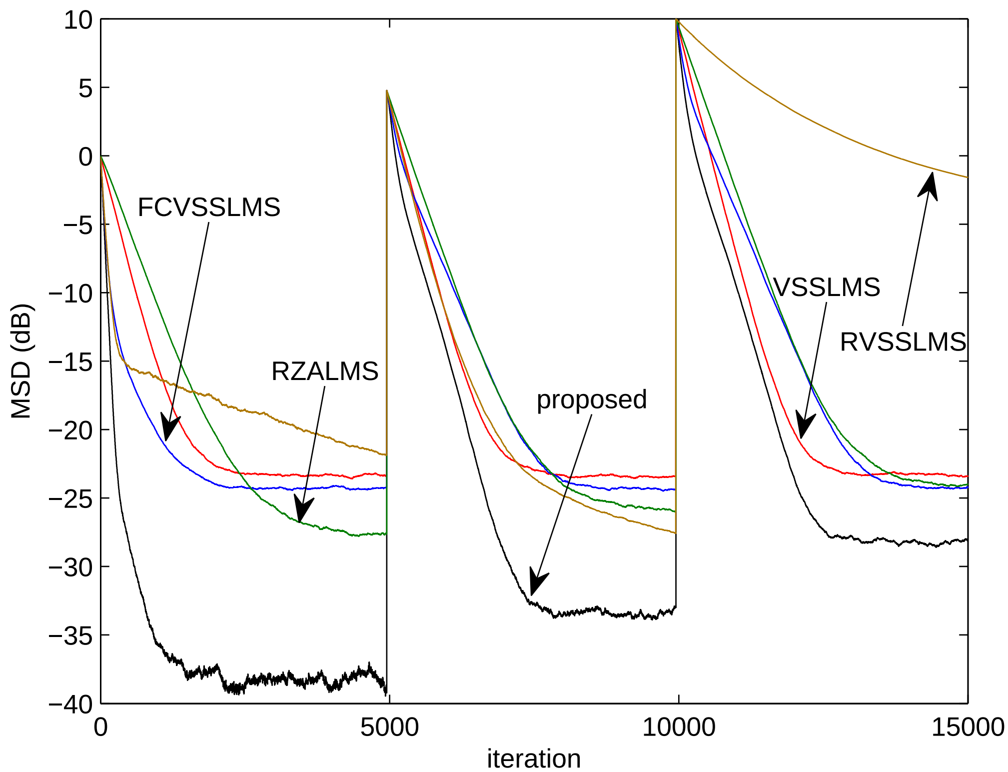


Figure 4.2. Tracking and steady state behaviors of a 50 tap adaptive filter driven by a white input signal.

In the first experiment, three different filters of 50 coefficients which have only one randomly placed coefficient with value ‘1’ for the first 5000 iterations, four randomly placed coefficients as ‘1’ for the second 5000 iterations and fourteen randomly placed coefficients as ‘1’ for the last 5000 iterations, were used. In order to obtain a 10 dB signal-to-noise ratio (SNR), the input signal and the observed noise are both assumed to be white Gaussian random sequences with variances 1.5 and 0.15, respectively. The used performance measure is the $MSD = E\{\|\mathbf{h} - \mathbf{w}(k)\|^2\}$, where $\mathbf{h}$ represents the impulse response of the unknown system.

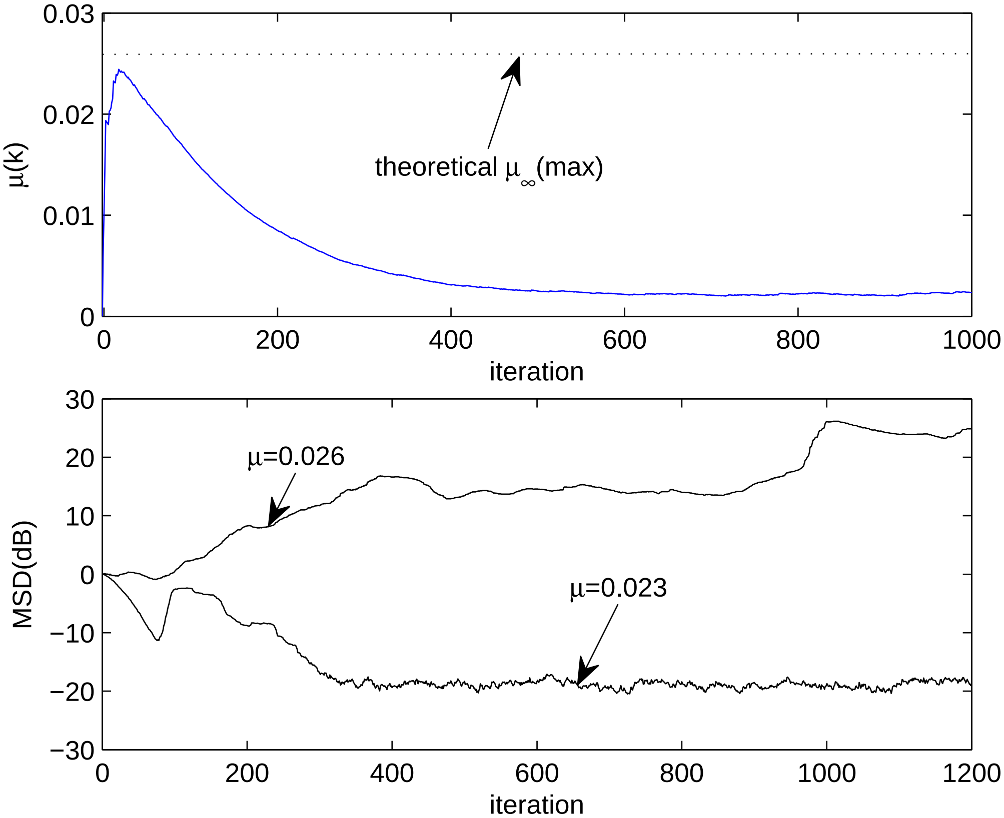

Figure 4.3. (a) Update behavior of $\mu(k)$, (b) performance of the proposed algorithm with $\mu(\infty) < \text{upperbound}$ (converges) and $\mu(\infty) > \text{upperbound}$ (diverges).

Since $\mu$ is automatically updated for the proposed algorithm, we will try to calculate a proper value of $\alpha$ by simulation. Basically, we slowly change the value of $\alpha$ until we find the optimum $\alpha$ value that provides the minimum MSD with a 50 tap unknown system. This value is shown in Fig. 4.1 and corresponds to 0.993. Hence, in the following experiments an $\alpha = 0.993$ will be used.

Simulations were done with the following parameters: For the VSSLMS: $\mu_{min} = 0.0012$, $\mu_{max} = 0.002$, $\gamma = 0.0005$ and $\alpha = 0.97$. For the RVSSLMS: $\alpha = 0.9997$, $\gamma = 0.00002$, $a = 0.9$ and $b = 1 - 10^{-5}$. For the FC-VSSLMS: $\alpha = 0.993$, $\beta = 0.99$, $\gamma = 0.002$ and $L = 200$. For the RZA-LMS: $\mu_{max} = 0.0015$, $\epsilon = 10$ and $\rho = 5.5 \times 10^{-4}$. For the proposed algorithm: $\mu(0) = 0$, $\alpha = 0.993$ (selected by the scheme given in Fig. 4.1), $\beta = 0.99$, $\gamma_s = 0.0029$, $\rho = 5.5 \times 10^{-4}$ (the same as that of the RZA-LMS), $\lambda = 8$ (should be a positive relatively large number to provide a good approximation to the $l_0$-norm and $L = 200$ (used in [55]). Fig. 4.2 gives the MSD measure for the five algorithms. It is seen from the figure that when the system has a high sparsity, the proposed algorithm has a very low MSD compared to the other algorithms. Even if the sparsity decreases, it still provides a significant high performance, compared to the other algorithms, in terms of both the convergence rate and MSD.

The step-size parameter used in the first experiment, should be less than 0.0256 according to the theoretical $\mu(\infty)$ in (4.22). Hence, in simulations, $\mu(k)$ should not exceed this value. Fig. 4.3(a) shows the update behavior for $\mu(k)$ for the first part of experiment 1. The figure shows that $\mu(k)$ never exceeds the theoretical $\mu(\infty)$. Fig. 4.3(b) shows the performance of the algorithm with two different values of $\mu(\infty)$. The figure shows that the algorithm converges for $\mu = 0.023$ (smaller than the upper bound ($\mu(\infty)$)), but it diverges for $\mu = 0.026$ (larger than the upper bound).

In the second experiment, the MSD of the algorithms at different sparsity levels is measured with the same SNR and other parameters as in the first experiment. Fig. 4.4 shows that the proposed algorithm has a higher performance up to 30% and 15% sparsity than the FC-VSSLMS and VSSLMS, respectively, and better than RZA-LMS and RVSSLMS for all levels of sparsity.

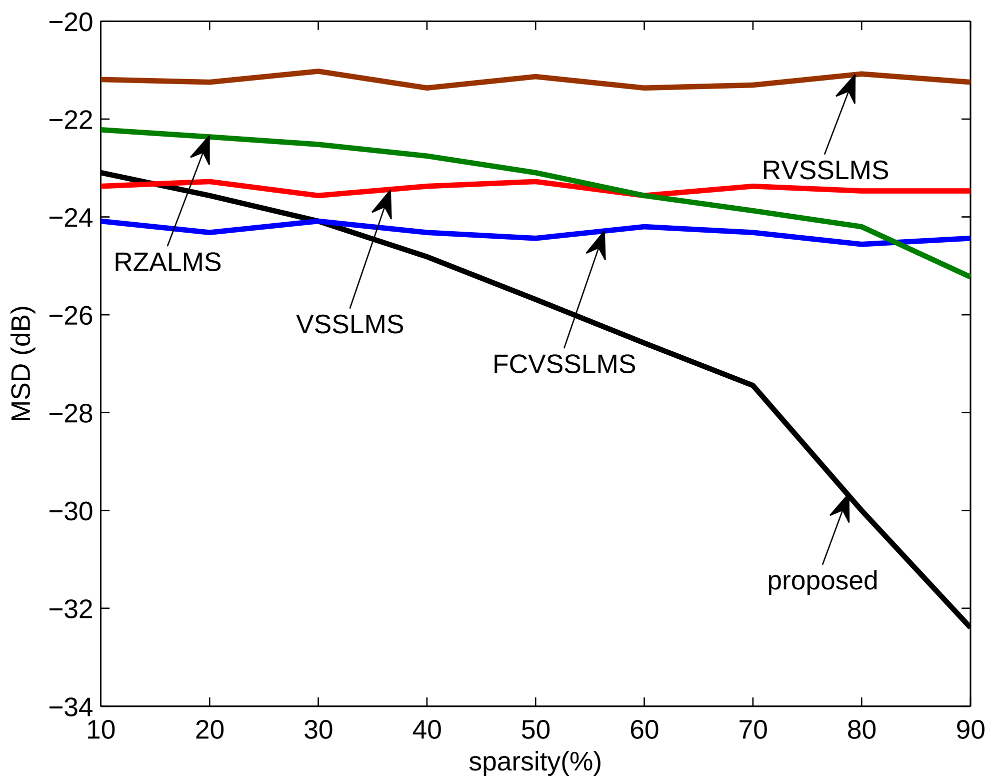

Figure 4.4. MSD vs. sparsity at 10 dB SNR.

In the third experiment, MSD of the algorithms are measured for different SNR and the results are compared with each other, keeping the sparsity at 20 $\%$ level and without changing other parameters. As seen in the Fig. 4.5, we obtain the best performance again using the proposed algorithm.

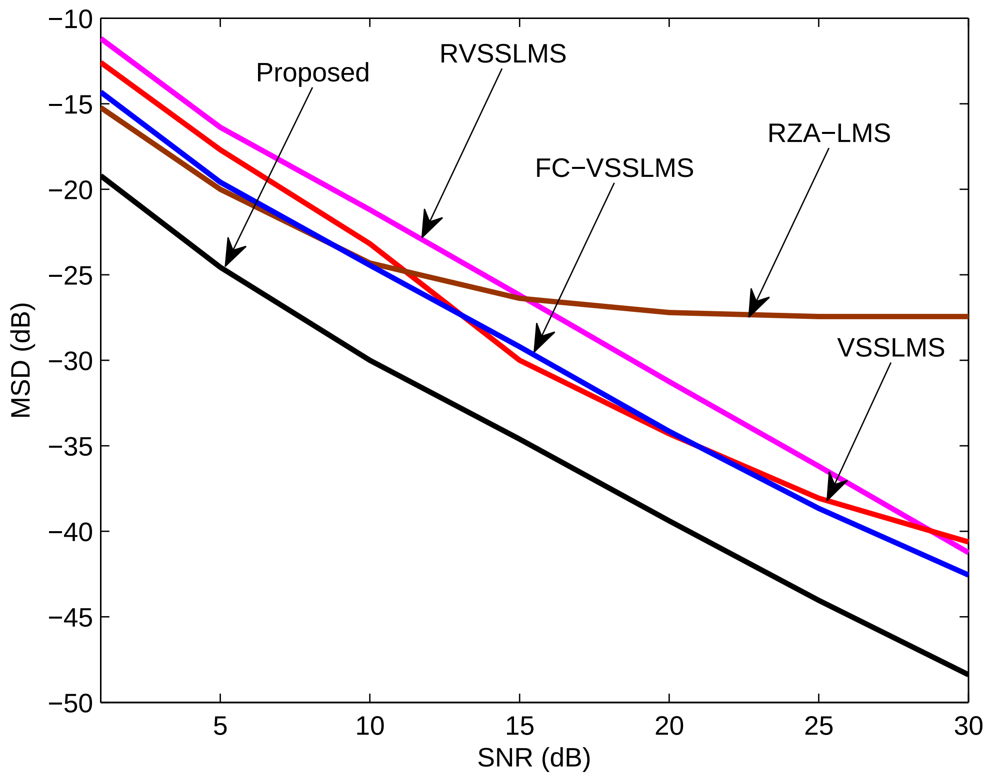


Figure 4.5. MSD vs. SNR at 20 $\%$ sparsity.

In the fourth experiment, the performance of the algorithms are compared with 128 taps filter with thirty non-zero random coefficients between [0, 1] (sparsity of 77 %) and 10 dB SNR. The observed noise is assumed to be correlated Gaussian random sequence. The correlated noise is created by passing a white Gaussian noise with zero mean and variance 0.1 through an AR(1) process with correlation coefficient 0.7. It is seen from Fig. 4.6 that the proposed algorithm converges faster than other algorithms and has a lower MSD than the others.

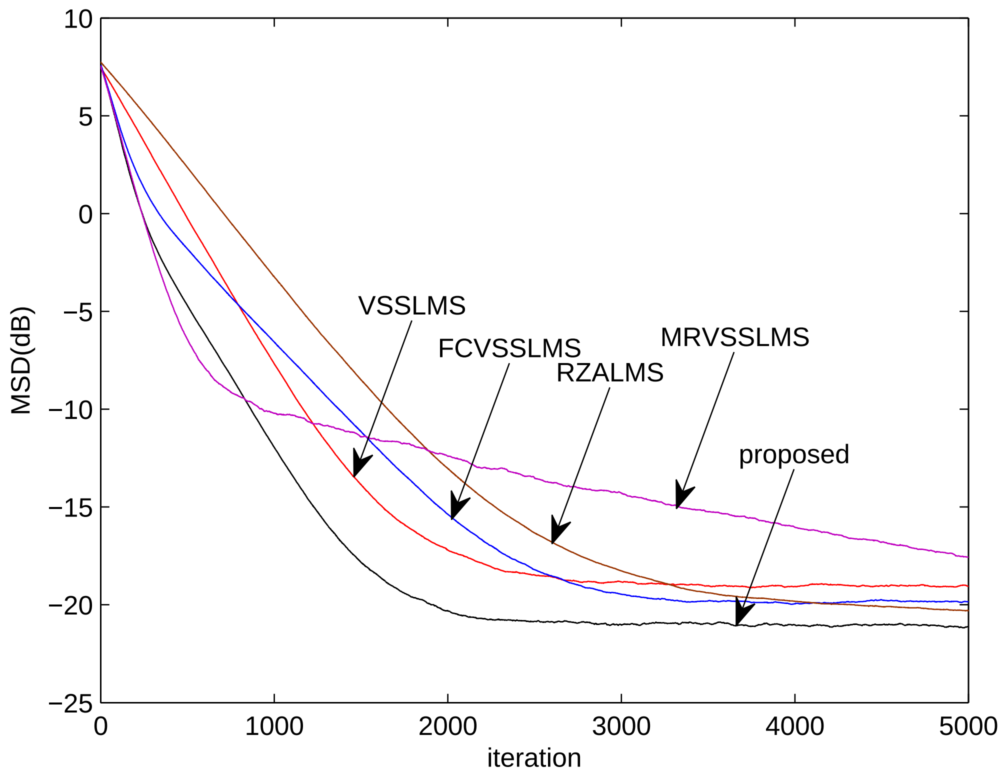

Figure 4.6. MSD comparisons for 128 tap filters with 77 % sparsity and 10 dB SNR.

# CHAPTER 5

## PROPOSED ALGORITHM IN TRANSFORM DOMAIN

### 5.1. Introduction

Recently, many proposals have shown that the performances of LMS-type algorithms can be improved further in system identification settings when the system is sparse (digital tv transmission channel [26], echo paths [73], etc. ). In [56], the authors have proposed sparse LMS algorithms that exploit the sparsity of the system. However, still these algorithms suffer from the constant step-size problem. In [63], we have proposed a sparse function controlled variable step-size LMS (SFC-VSSLMS) algorithm. The algorithm takes the advantages of sparsity and variable step-size, and provides prominent results, when the additive noise is white. However, similar to the other algorithms, the performance of the SFC-VSSLMS algorithm deteriorates when the input signal and/or the additive noise are/is correlated (the eigenvalue spread of the autocorrelation matrix of the input signal is relatively high [14]).

Many proposals appeared to deal with the problem of the high eigenvalue spread of the autocorrelation matrix [77, 80, 49]. For example in [80], the authors show that transforming the input signal into another domain reduces the eigenvalue spread of its autocorrelation matrix, which consequently enhances the performance of the adaptive filter. In order to exploit sparsity on top of the transformation, authors in [49] propose a transform domain reweighted zero attracting LMS (TD-RZALMS) algorithm. Still this algorithm suffers from the constant step-size.

Up to our knowledge, there is no algorithm that exploits the sparsity of the system, uses a variable step-size and transformation of the input signal to reduce the eigenvalue spread of the autocorrelation matrix at the same time.

In this chapter, we propose a transform domain sparse function controlled variable step-size algorithm that combines all of the aforementioned properties. The proposed algorithm imposes the transform domain on the SFC-VSSLMS algorithm in which an approximate $l_0$-norm penalty is added to the cost function of the FC-VSSLMS algorithm.

The chapter is organized as follows: In Section II, brief reviews of the transform domain-LMS (TD-LMS) and SFC-VSSLMS algorithms are provided and the proposed algorithm is derived. In Section III, the convergence analysis of the proposed algorithm is presented. In Section IV, simulation results that compare the performance of the proposed algorithm to other algorithms are provided and discussed.

## 5.2. The Proposed Algorithm

### 5.2.1. Review of the Transform Domain LMS Algorithm

Consider a linear system with input-tap vector $\mathbf{x}\left(k\right) = \left[x_0, ..., x_{N-1}\right]^T$ and output $d(k)$ related by

$$d\left(k\right) = \mathbf{h}^T\mathbf{x}\left(k\right) + n\left(k\right) \tag{5.1}$$

where $\mathbf{h} = \left[h_0, \ldots, h_{N-1}\right]^T$ is the unknown system coefficients with length $N$, $T$ is the transposition operator and $n(k)$ is the observation noise. For the TD-LMS algorithm, the input vector $\mathbf{x}(k)$ is processed by a unitary transform such as discrete Fourier transform (DFT) or discrete cosine transform (DCT). Once the filter order $\mathbf{N}$ is fixed, the transform is simply an $N \times N$ matrix $\mathbf{T}$, which is in general complex, with orthonormal rows. And the transformed vector is obtained as

$$\mathbf{X}(k) = \mathbf{T}\mathbf{x}(k), \tag{5.2}$$

where $\mathbf{T}$ is a unitary matrix that is $\mathbf{T}^T\mathbf{T} = \mathbf{T}\mathbf{T}^T$. The filter output is then

$$y(k) = \mathbf{W}^T(k)\mathbf{X}(k) \tag{5.3}$$

and the corresponding estimation error is

$$e\,(k) = d\,(k) - y(k) \tag{5.4}$$

where $\mathbf{W}(k)$ is the transform domain filter coefficients vector. We may note that although $\mathbf{X}(k)$ is in the transform domain, the filter output $y(k)$ and the estimation error $e(k)$ are both in time domain. The filter coefficients of TD-LMS are then updated by

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \mu\mathbf{D}^{-1}e(k)\mathbf{X}(k), \tag{5.5}$$

where $\mathbf{D}$ is an $N \times N$ diagonal matrix whose elements are the transform domain signal power components $E[|X_i|^2]$ [49] which can be calculated by a recursive equation as:

$$D_{ii}(k+1) = \beta D_{ii}(k) + (1-\beta)|X_{ii}(k)|^2. \tag{5.6}$$

In (5.5) $\mu(k) = \mu\mathbf{D}^{-1}$, and it is clear that the speed of convergence for TD-LMS algorithm depends on $\mathbf{D}^{-1}\mathbf{R_{XX}}$. In Appendix A, we show that, with a proper orthogonal transformation, the eigenvalue spread of the input autocorrelation matrix can be reduced.

### 5.2.2. Proposed Algorithm

Remembering the SFC-VSSLMS algorithm, in which the aim is to find the optimum vector $\mathbf{h}$ as,

$$\mathbf{h} = \arg\min_{\mathbf{w}} \left\{ \frac{1}{2} \left| e\left(k\right) \right|^{2} + \xi \|\mathbf{w}\left(k\right)\|_{0} \right\}, \tag{5.7}$$

where $e(k)$ is defined in (5.4), $\xi$ is a small positive constant and $\|.\|_0$ denotes the $l_0$-norm of the weights vector. Since (5.7) is an NP-hard problem, $\|\mathbf{w}\left(k\right)\|_0$ is approximated in [74] as

$$\|\mathbf{w}\left(k\right)\|_{0} \simeq \sum_{k=0}^{N-1} (1 - e^{-\lambda|\mathbf{w}(k)|}), \tag{5.8}$$

where $\lambda$ is a positive parameter. The update equation of the SFC-VSSLMS algorithm can be written as:

$$\mathbf{w}\left(k+1\right) = \mathbf{w}\left(k\right) + \mu\left(k\right) e\left(k\right) \mathbf{x}\left(k\right) - \rho(k) sgn\left[\mathbf{w}\left(k\right)\right] e^{-\lambda|\mathbf{w}(k)|}, \tag{5.9}$$

where $\rho(k) = \xi\lambda\mu(k)$ and $\mu\left(k\right)$ is the variable step-size parameter and given by [55] as,

$$\mu(k+1) = \alpha\mu(k) + \gamma_s f(k) \frac{|e(k)|^2}{\widehat{e}^2_{ms}(k)}, \tag{5.10}$$

where $0 < \alpha < 1$, $\gamma_s > 0$ is an updating parameter, $f(k) = 1/k$ if $k < L$ else $f(k) = 1/L$ and $\widehat{e}^2_{ms}(k)$ is the estimated MSE defined as,

$$\widehat{e}^2_{ms}(k) = \beta\widehat{e}^2_{ms}(k-1) + (1-\beta)|e(k)|^2. \tag{5.11}$$

where $\beta$ is the weighting factor $0 \ll \beta < 1$.

In this work, we propose a new cost function using inverse transfer domain coefficient vector $\mathbf{W}(k)$ obtained by transfer domain input vector $\mathbf{X}(k)$, hence

$$\mathbf{H} = \arg\min_{\mathbf{W}} \left\{ \frac{1}{2} \left| e\left(k\right) \right|^2 + \xi \| \mathbf{T}^T \mathbf{W}\left(k\right) \|_0 \right\}. \tag{5.12}$$

$\|.\|_0$ denotes the $l_0$-norm of the weights vector and can be approximated as given below:

$$\| \mathbf{W}\left(k\right) \|_0 \simeq \sum_{k=0}^{N-1} \left(1 - e^{-\lambda |\mathbf{W}(k)|}\right), \tag{5.13}$$

where $\lambda$ is a positive parameter. Deriving (5.12) with respect to $\mathbf{W}\left(k\right)$ and substituting in

$\mathbf{W}(k+1) = \mathbf{W}(k) - \mu\left(k\right) \dfrac{\partial J\left[\mathbf{W}\left(k\right)\right]}{\partial \mathbf{W}\left(k\right)}$ yields,

$$\mathbf{W}\left(k+1\right) = \mathbf{W}\left(k\right) + \mu\left(k\right) \mathbf{D}^{-1} e\left(k\right) \mathbf{X}\left(k\right) - \rho(k) \mathbf{D}^{-1} \mathbf{T}^T sgn[\mathbf{T}^T \mathbf{W}(k)] e^{-\lambda |\mathbf{T}^T \mathbf{W}(k)|}, \tag{5.14}$$

where $\mathbf{X}(k)$ is given in (5.2), $\mathbf{W}(k)$ is the transform domain vector of $\mathbf{w}(k)$ and $\rho(k) = \xi\lambda\mu(k)$ is the sparsity aware parameter. The update equation in (5.14) has an additional term $-\rho(k)\mathbf{D}^{-1}\mathbf{T}^T sgn[\mathbf{T}^T\mathbf{W}(k)]e^{-\lambda|\mathbf{T}^T\mathbf{W}(k)|}$ which always attracts the tap coefficients to zero. This is called a zero-attractor because its strength is controlled by $\rho$. In other words, it will speed-up convergence when most of the system coefficients are zeros, that is, the system is sparse.In this work we use the DCT due to its real valued components. The proposed algorithm is summarized in Table 5.1.

### 5.3. Computational Complexity of the Proposed Algorithm

Computational complexity of the proposed algorithm can be calculated with the additional complexity of the TD-LMS algorithm. We need $5N$ multiplications and $5N$ additions for DCT transform; $5N$ multiplications, $2N$ additions and one division for power normalization [51]. Overall with SFC-VSSLMS complexity, $15N + 9$ multiplications, $10N + 3$ additions, $N$ comparisons and one division. That is, the proposed algorithm has $O(N)$ complexity.

Table 5.1. Summary of the used Algorithms

| TD-RZALMS | SFC-VSSLMS | Proposed |
|---|---|---|
| Initialize: | Initialize: | Initialize: |
| $\mu,\ \rho$ | $ems(-1) = 0,\ \mu(-1) = 0,\ \rho$ | $ems(-1) = 0,\ \mu(-1) = 0,\ \rho$ |
| for $k = 1, ..., N$ | for $k = 1, ..., N$ | for $k = 1, ..., N$ |
| $e(k) = d(k) - \mathbf{w}(k)^T\mathbf{x}(k)$ | $ems(k) = \beta ems(k-1) + (1-\beta)\|e(k)\|^2$ | $ems(k) = \beta ems(k-1) + (1-\beta)\|e(k)\|^2$ |
| $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\mathbf{x}(k)$ | $\mu(k) = \alpha\mu(k-1) + \gamma f(k)\|e(k)\|^2/ems(k)$ | $\mu(k) = \alpha\mu(k-1) + \gamma f(k)\|e(k)\|^2/ems(k)$ |
| $-\rho sgn(\mathbf{w}(k))$ | where $e(k) = d(k) - \mathbf{w}(k)^T\mathbf{x}(k)$ | where $e(k) = d(k) - \mathbf{w}(k)^T\mathbf{x}(k)$ |
| | $f(k) = 1/k$ if $k < L$ else $f(k) = 1/L$ | $f(k) = 1/k$ if $k < L$ else $f(k) = 1/L$ |
| | $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k)e(k)\mathbf{x}(k)$ | $\mathbf{X}(k) = \mathbf{T}\mathbf{x}(k)$ and $\mathbf{W}(k) = \mathbf{T}\mathbf{w}(k)$ |
| | $-\rho(k)sgn(\mathbf{w}(k))e^{-\lambda\|\mathbf{w}(k)\|}$ | $\mathbf{W}(k+1) = \mathbf{W}(k) + \mu(k)\mathbf{D}^{-1}e(k)\mathbf{X}(k)$ |
| | | $-\rho(k)\mathbf{D}^{-1}\mathbf{T}^T sgn(\mathbf{T}^T\mathbf{W}(k))e^{-\lambda\|\mathbf{T}^T\mathbf{W}(k)\|}$ |

### 5.4. Convergence Analysis of the Proposed Algorithm

In this section we perform the convergence analysis of the proposed algorithm assuming that the input signal and noise are statistically independent. Denoting $\mathbf{H}$ to be the transformed optimal filter coefficients, as

$$\mathbf{H} = \mathbf{T}\mathbf{h}. \tag{5.15}$$

Substituting (5.15) and (5.2) into (5.1) yields

$$d(k) = \mathbf{H}^T\mathbf{X}(k) + n(k). \tag{5.16}$$

The transform domain misalignment vector of the LMS algorithm is defined as

$$\boldsymbol{\theta}(k) = \mathbf{W}(k) - \mathbf{H}, \tag{5.17}$$

combining (5.3), (5.4), (5.16) and (5.17) gives

$$e(k) = -\mathbf{X}^T(k)\boldsymbol{\theta}(k) + n(k), \tag{5.18}$$

substituting the results of (5.17) and (5.18) in (5.14) provide,

$$\begin{aligned}\boldsymbol{\theta}(k+1) =& [\mathbf{I}_N - \mu(k)\mathbf{D}^{-1}\mathbf{X}(k)\mathbf{X}^T(k)]\boldsymbol{\theta}(k) + \mu(k)\mathbf{D}^{-1}\mathbf{X}(k)n(k) \\ & - \rho(k)\mathbf{D}^{-1}\mathbf{T}^T sgn[\mathbf{T}^T\mathbf{W}(k)]e^{-\lambda|\mathbf{T}^T\mathbf{W}(k)|},\end{aligned} \tag{5.19}$$

taking the expectation of (5.19) with the independence assumption we obtain,

$$E[\boldsymbol{\theta}(k+1)] = [I_N - \mu(k)\mathbf{D}^{-1}\mathbf{R}_{XX}]E[\boldsymbol{\theta}(k)] - \rho(k)\mathbf{D}^{-1}\mathbf{T}^T E[sgn[\mathbf{T}^T\mathbf{W}(k)]e^{-\lambda|\mathbf{T}^T\mathbf{W}(k)|}]. \tag{5.20}$$

In (5.20), $\rho(k)\mathbf{D}^{-1}\mathbf{T}^T E[sgn[\mathbf{T}^T\mathbf{W}(k)]e^{-\lambda|\mathbf{T}^T\mathbf{W}(k)|}]$ is bounded and hence $E[\boldsymbol{\theta}(k)]$ converges if the maximum eigenvalue of $[\mathbf{I}_N - \mu(k)\mathbf{D}^{-1}\mathbf{R}_{XX}] \in (-1, 1)$ and this, in turn, guarantees the convergence of the algorithm in the mean sense.

### 5.5. Simulation Results

In this section, the performance of the proposed algorithm is compared to those of the SFC-VSSLMS and TD-RZALMS algorithms in sparse system identification models in the presence of white and correlated input signals. In all experiments, the filter length is assumed to be 16 taps and the signal-to-noise ratio (SNR) is tuned to be 30 dB. The performance measure used is the

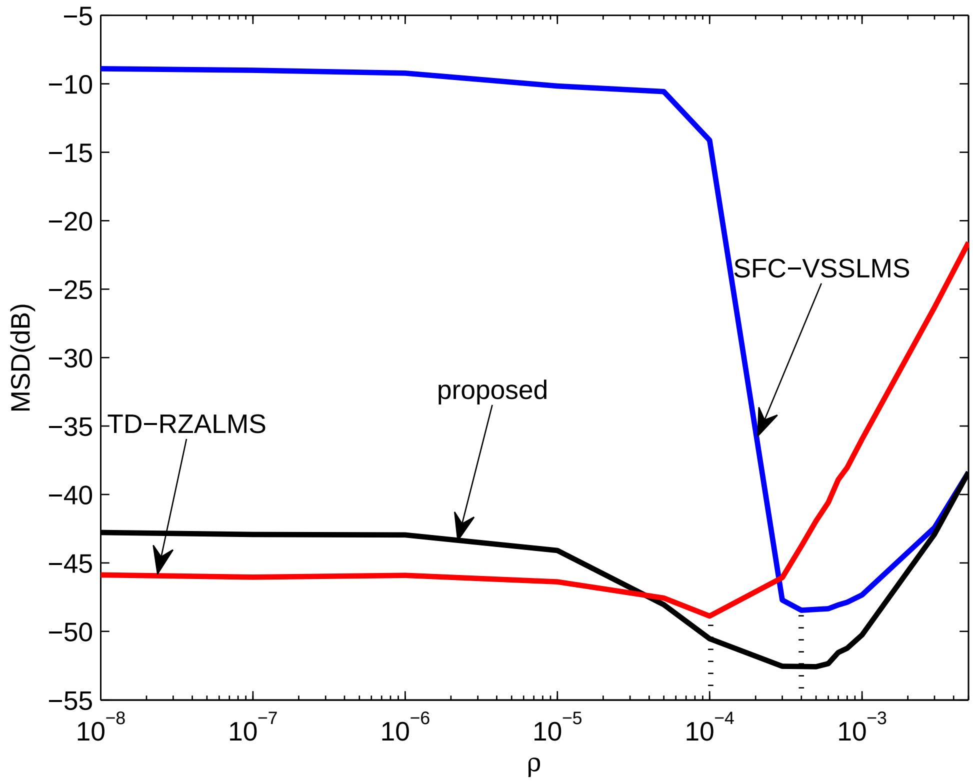


Figure 5.1. optimum $\rho$ for TD-RZALMS, SFC-VSSLMS and the proposed algorithms by extensive simulations on experiment 1.

$MSD = E\{\|\mathbf{H} - \mathbf{W}(k)\|^2\}$. All the experiments are implemented with 200 independent runs.

In the first experiment, a 16 taps time varying unknown system is used. In the first 5000 iterations, a random coefficient is set to '1' and the rest are zeros. In the second 5000 iterations, 4 random coefficients are set to '1' and the rest are zeros, and in the last 5000 iterations, 8 random coefficients are set to '1' and the rest are assumed to be zeros. The input signal and the observed noise are assumed to be white Gaussian random sequences with zero mean and variances to provide 30 dB SNR. Simulations are done with the following parameters: For SFC-VSSLMS and the proposed algorithms: $\alpha = 0.997$, $\beta = 0.75$, $\gamma = 0.004$, $L = 200$, $\lambda = 8$ and $\rho = 5 \times 10^{-4}$. For TD-RZALMS algorithm: $\rho = 10^{-4}$, $\epsilon = 10$ and $\mu = 0.005$. The most important parameter selection is the sparsity-aware parameter $\rho$. We select $\rho$ by assuming 1/16 sparsity of the unknown system and find $\rho$ that gives minimum MSD, that is optimum result for each algorithm, as shown in Fig. 5.1 and generalized to the other parts of the experiment. However, for TD-RZALMS, we found that $\rho$ needs to be regularized if the sparsity changes, so we have selected different $\rho$ than the found optimum one in order to guarantee convergence of the algorithm in the other sparsity regions. Fig. 5.2 shows that the proposed algorithm always outperforms the TD-RZALMS algorithm in both MSD and convergence rate, and behaves exactly the same as the SFC-VSSLMS algorithm.

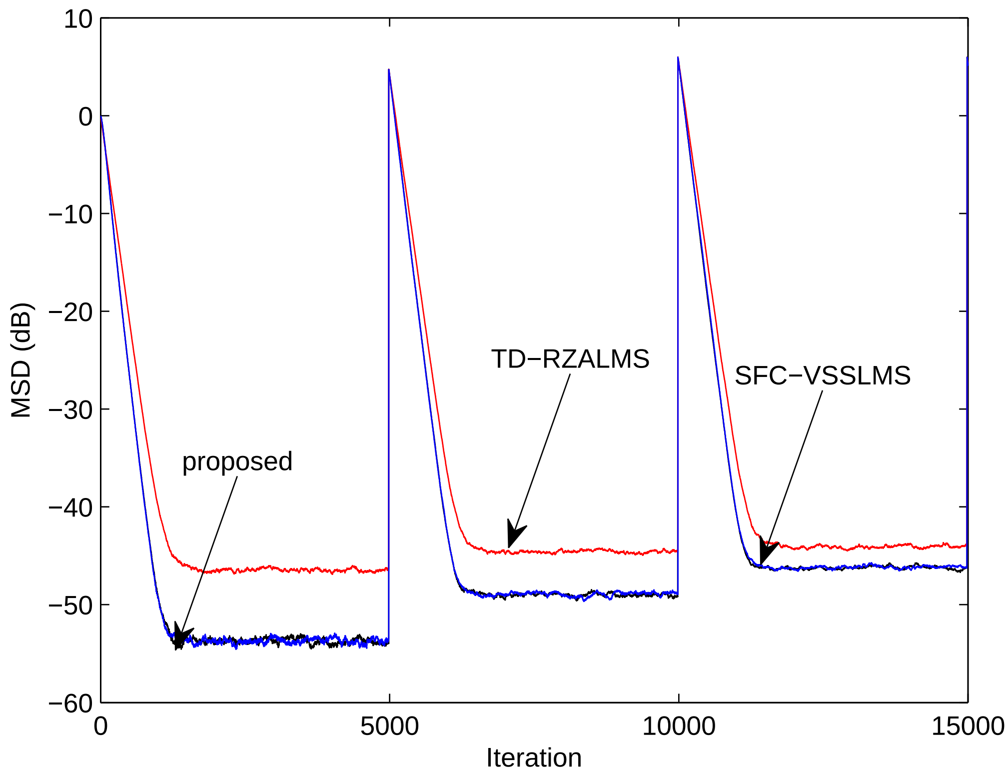


Figure 5.2. Tracking and steady state behaviors of the SFC-VSSLMS, TD-RZALMS and the proposed algorithms with 16 taps sparse adaptive filter driven by a white input signal.

Since the input signal in the first experiment is white, the advantage of the proposed algorithm is not prominent. Hence, in the second experiment, the input signal $\mathbf{x}(k)$ is assumed to be an AR(4) process generated as $x(k) = 1.79x(k-1) - 1.85x(k-2) + 1.27x(k-3) - 0.41x(k-4) + n_0(k)$ [81], where $n_0(k)$ is a zero-mean white Gaussian sequence with variance $\sigma^2_{n_0} = 0.15$. The eigenvalue spread of the autocorrelation matrix of the input signal $\mathbf{x}(k)$ is measured to be $\frac{\lambda_{max}}{\lambda_{min}} = 944$. The observed noise is assumed to be an additive white Gaussian noise (AWGN) with zero mean and variance that, again, provides 30 dB SNR. The unknown system is assumed to be a 16 taps time varying unknown system a random coefficient is set to '1' in the first 15000 iteration and the rest are zeros (here we increase the iteration number in order to be able to notice the convergence in the steady-state). In the second 15000 iterations, 4 random coefficients are set to '1' and the rest are zeros, and in the last 15000 iterations, 8 random coefficients are set to '1' and the rest are assumed to be zeros. Simulations are done with the following parameters: For the proposed algorithm the same parameters in experiment 1 are used. For SFC-VSSLMS algorithm: $\alpha = 0.999$, and the rest are the same as those in experiment 1. For TD-RZALMS algorithm: $\rho = 10^{-5}$, $\epsilon = 10$ and $\mu = 0.003$. Note that when the input signal is changed, the other algorithms required parameter tuning where it is not the case of the proposed algorithm. In Fig. 5.3, the virtue of the transform domain appears clearly in the proposed algorithm. It is seen that the proposed algorithm provides faster convergence and lower MSD than the other algorithms in all regions. However, it should be noted that, in region 3; where the sparsity is relatively low, the performance of the TD-RZALMS and SFC-VSSLMS algorithms severely deteriorate where it is not the case for the proposed algorithm.

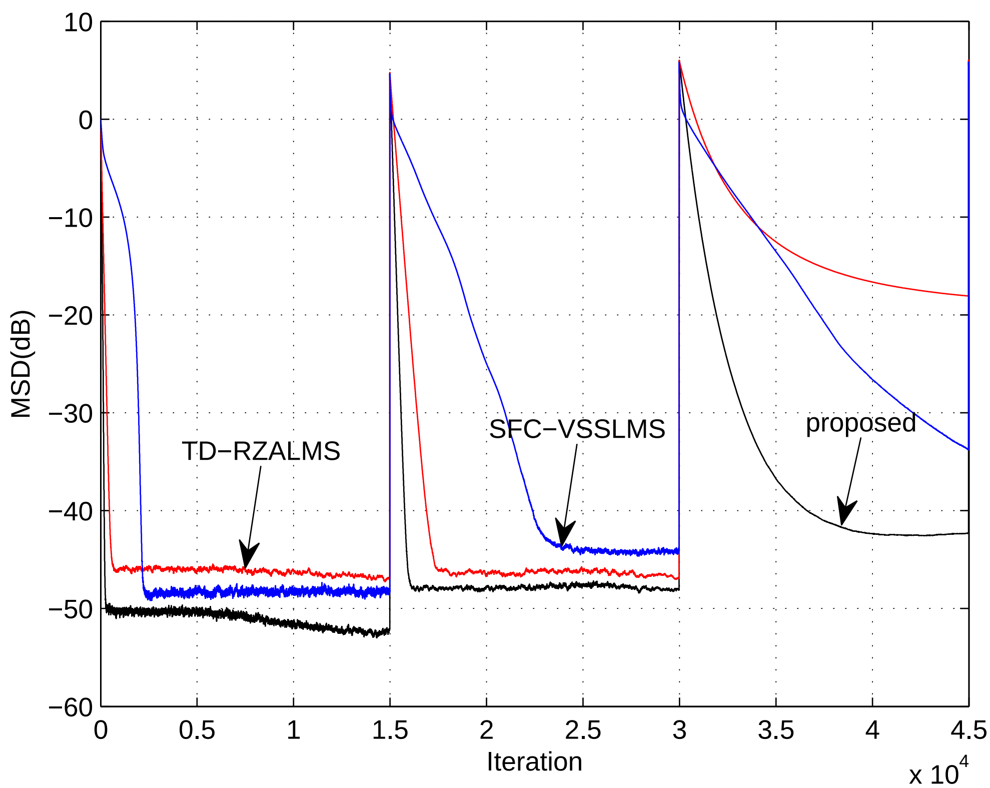


Figure 5.3. Tracking and steady state behaviors of the SFC-VSSLMS , TD-RZALMS and proposed algorithms with 16 taps adaptive filter driven by a colored input signal.

In the third experiment, in order to observe the performance of the algorithms for a higher filter tap and with a correlated Gaussian noise, their performances are compared for a 150 taps filter with thirty randomly distributed coefficients with value “1" (80 % sparsity) and the SNR 30 dB. The algorithms are simulated with the following parameters. For the TD-RZALMS: $\rho = 10^{-5}$, $\epsilon = 10$

and $\mu = 0.003$. For SFC-VSSLMS and the proposed algorithms: $\alpha = 0.99$, $\beta = 0.75$, $\gamma = 0.003$, $L = 200$, $\lambda = 8$ and $\rho = 5 \times 10^{-4}$. Note that $\rho$ is selected in the same way explained in experiment I (please see Fig. 5.4). Figure 5.5 shows that the convergence of SFC-VSSLMS is very slow (here the advantage of the transformation is very clear). Whereas, the proposed algorithm converges faster than the TD-RZALMS algorithm by almost 1000 iteration and 6 dB lower MSD.

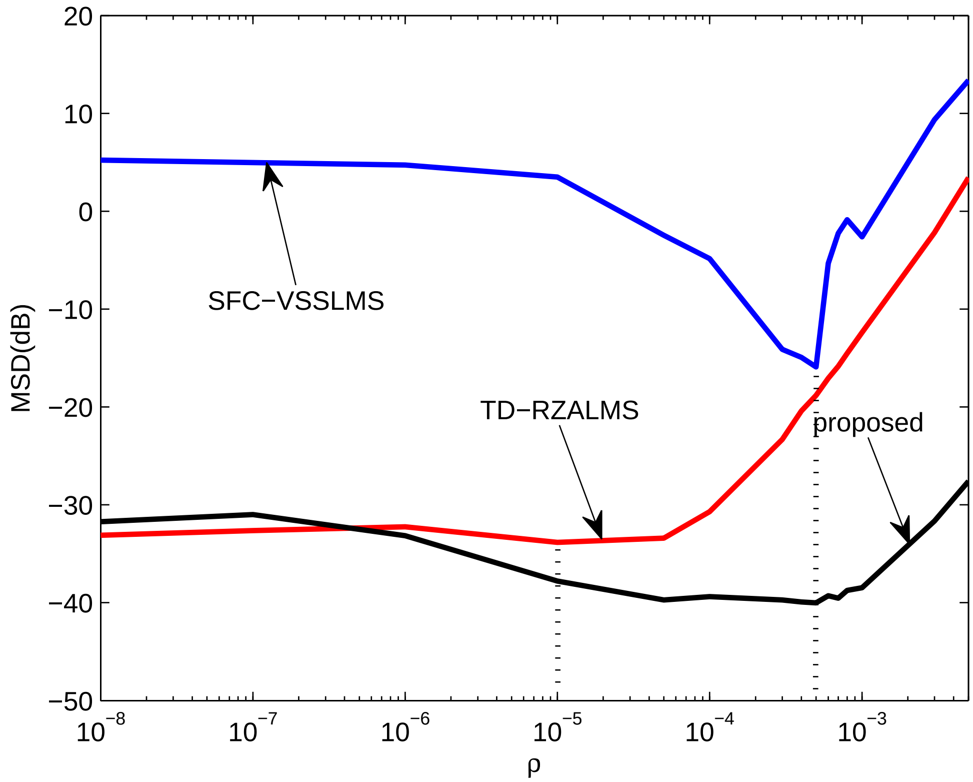

Figure 5.4. Optimum $\rho$ for all algorithms by extensive simulations on experiment II.

From the previous experiments, we see the virtues of combining variable step-size (faster convergence) and transform domain (lower MSD) very clearly.

To see the performance of the proposed algorithm for highly correlated real-time signals, we pass ECG and EMG through the unknown sparse system and try to find the filter coefficients. So we apply our proposed filter to a normalized highly correlated Electrocardiography (ECG) signals (which are electrical signals taken from the heartbeats of a patient) of a healthy person (please see Fig.5.6). The filter is of length 16 taps with four random coefficients are set to be '1' and the rest are set to zeros ($\%75$ sparse system). Simulations are done with the following parameters: For TD-RZALMS algorithm: $\rho = 10^{-4}$, $\epsilon = 10$ and $\mu = 0.0004$. For the proposed algorithm: $\alpha = 0.999$,

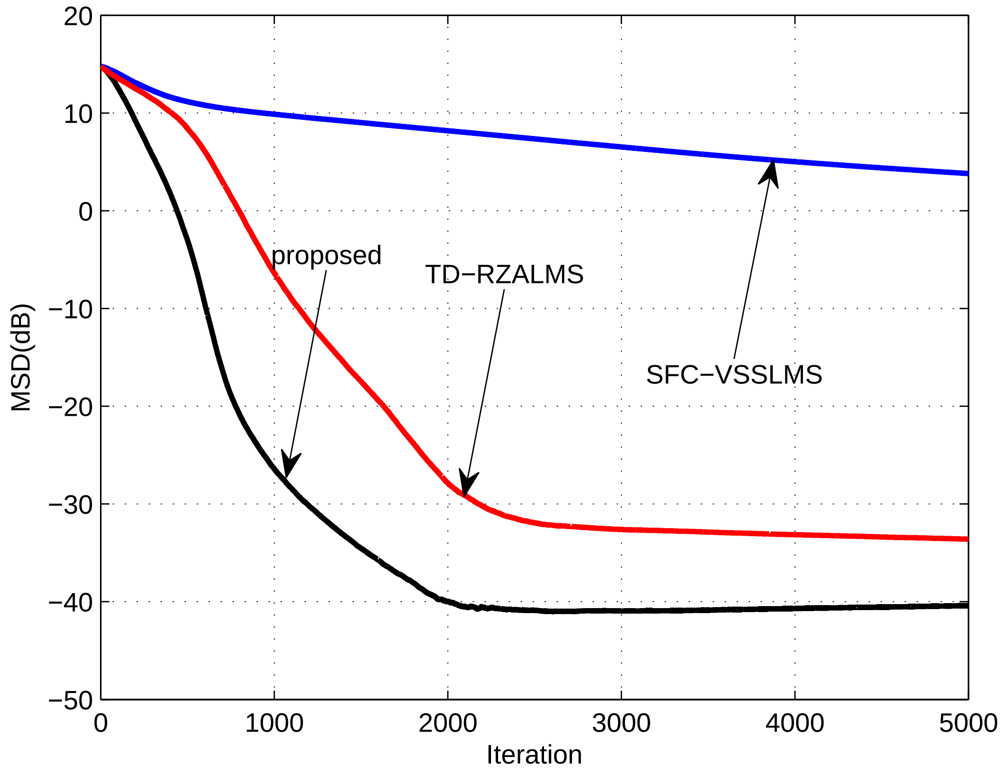


Figure 5.5. MSD learning curves of the SFC-VSSLMS, TD-RZALMS and proposed algorithms for a 150 taps filter with 30 random coefficients are set to be “1".

$\beta = 0.7$, $\gamma = 0.0001$, $L = 200$, $\lambda = 8$ and $\rho = 10^{-5}$. Fig.5.7 shows a MSD performance comparison between the proposed and the TD-RZALMS algorithms (here the SFC-VSSLMS algorithm is not included because it converges very slowly). Even though the TD-RZALMS converges slightly faster than the proposed algorithm but the proposed algorithm has lower MSD (4 dB better than the TD-RZALMS algorithm).

To see the robustness of the proposed algorithm due to changing the input signal, experiment IV is repeated with a normalized highly correlated Electromyography (EMG) signals (which are generated by muscle fibers prior to the production of muscle force) of a healthy person (please see Fig.5.8). Fig. 5.9 shows that the proposed algorithm converges much faster than the TD-RZALMS algorithm (7000 iterations faster) with 5 dB lower MSD. Also, we should note that the TD-RZALMS algorithm is highly affected by changing the input signal where it is not the case for the proposed algorithm.

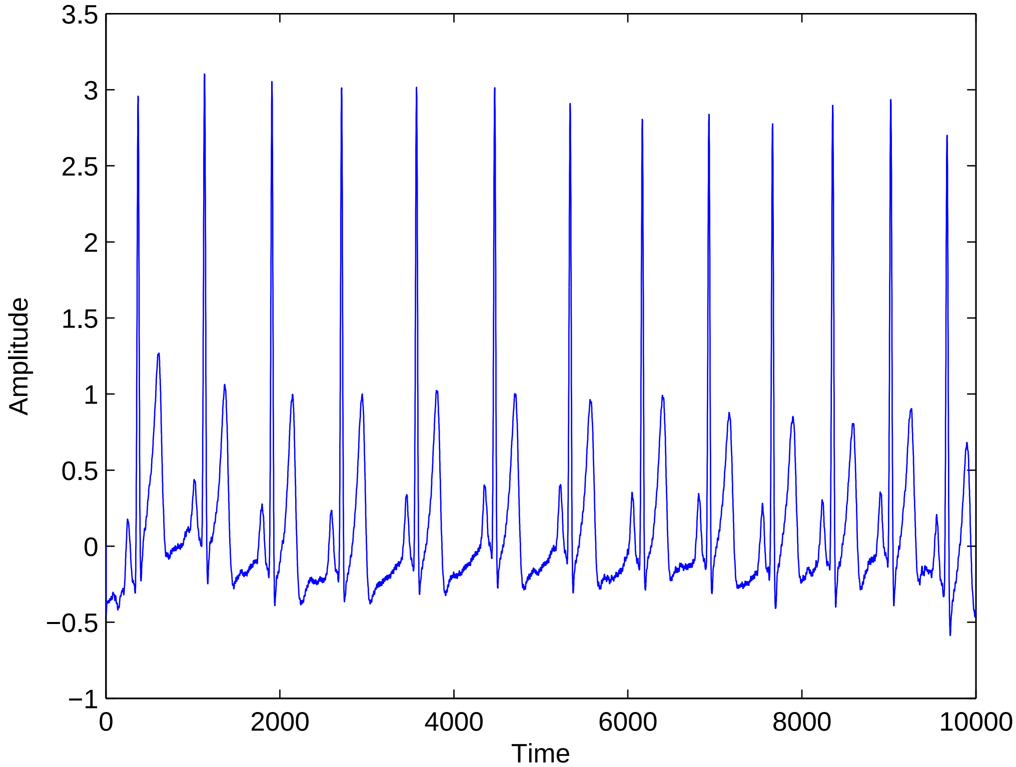


Figure 5.6. ECG signal used in simulations (cited from www.physionet.org/physiobank/database).

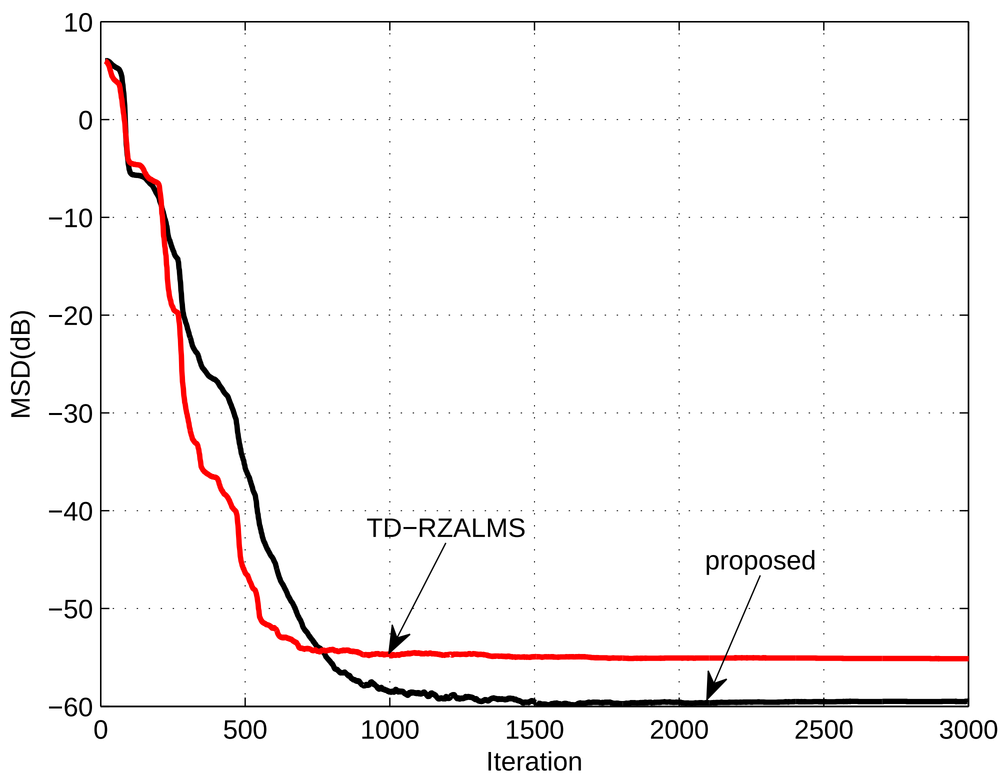


Figure 5.7. MSD curves of the of the proposed and TD-RZALMS algorithms for an ECG input signal.

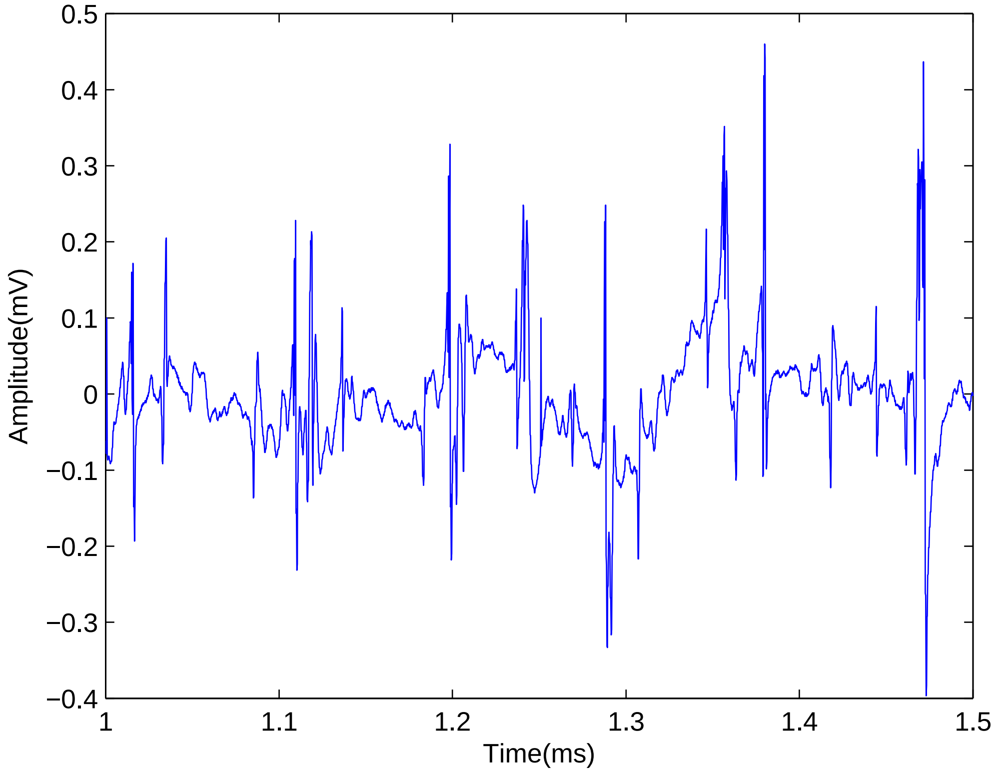


Figure 5.8. EMG signal used in simulations (cited from www.physionet.org/physiobank/database).

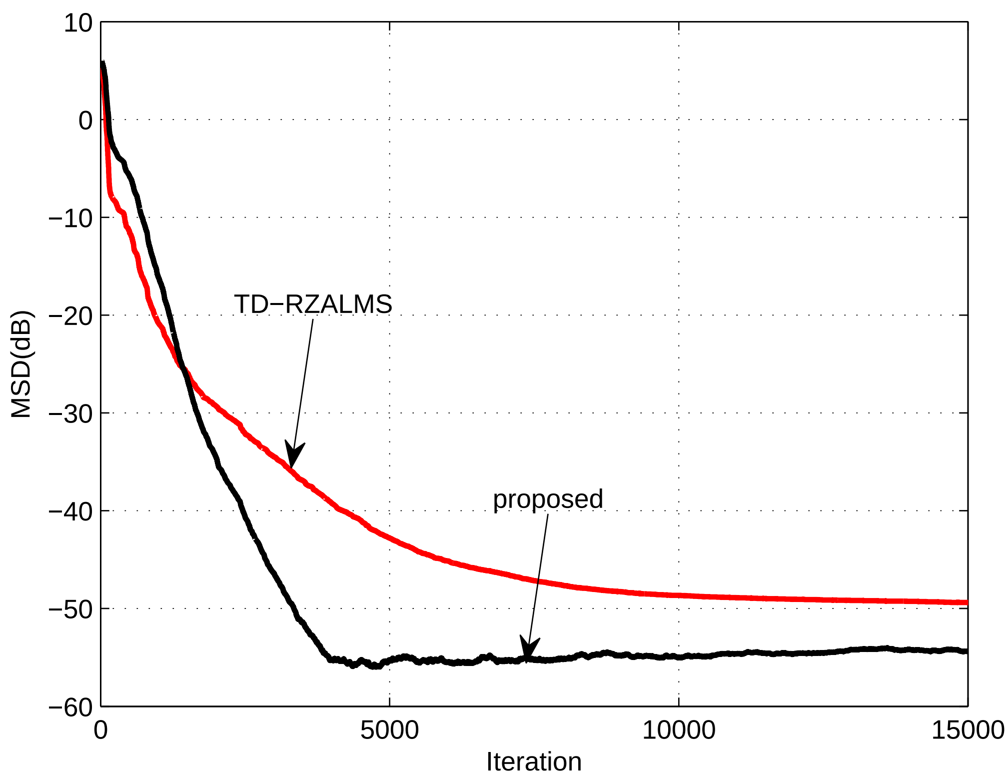


Figure 5.9. MSD curves of the of the proposed and TD-RZALMS algorithms for an EMG input signal.

# CHAPTER 6

## BLOCK IMPLEMENTATION OF PROPOSED ALGORITHM

### 6.1. Introduction

As mentioned before, the LMS-type algorithms have been used successfully in system identification using adaptive filters [18, 19] (see Fig. 6.1). The echo cancellation problem can be mentioned as a basic SI problem with a specific property that can be used to improve the adaptation process. This is the sparsity of both network and acoustic echo paths which can be described as the small percentage of the impulse response components have a significant magnitude while the rest are zero or small [22]. The advantage of sparsity for such systems have been addressed in many works recently [82, 83, 84].

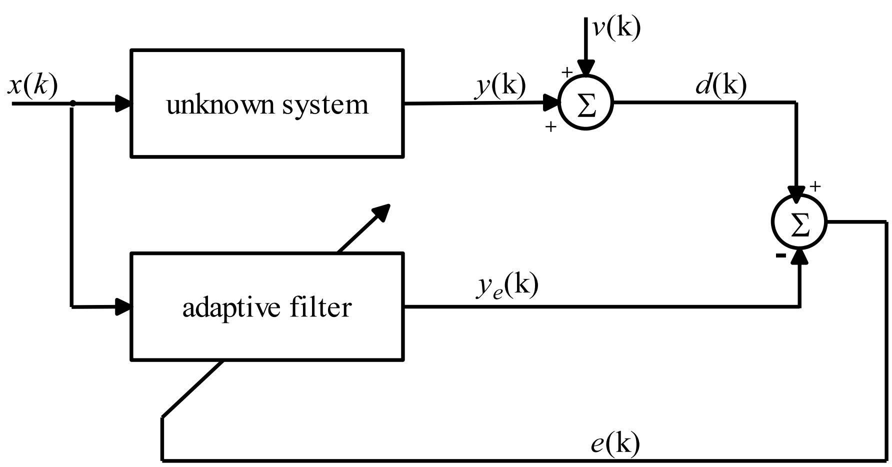


Figure 6.1. Block diagram of the system identification process.

Combining the advantages of sparsity and variable step-size, a sparse function controlled variable step-size LMS (SFC-VSSLMS) algorithm is proposed in [63]. In that algorithm, very remarkable results are obtained for sparse system identification which is generally used for acoustic echo cancellation. Actually, the conventional LMS algorithm has a high computational complexity in some adaptive filtering applications with long impulse response. For instance, acoustic echo cancellation, channel equalization and active noise control require a sufficiently high-order FIR filter. For such adaptive filtering applications, a block-LMS (BLMS) algorithm has been proposed in [85, 86] to speed up the calculation time of LMS algorithm. In that algorithm, the adaptive filter is realized by blockwise processing of the data in order to gain computational advantage. In the conventional LMS algorithm, filter parameters are updated for each data sample. Unlike the LMS algorithm, the BLMS algorithm adjusts the weights one per block of data in parallel processors.

In this chapter, we propose a new algorithm that combines the advantages of sparsity, variable step-size and block-LMS algorithm. The proposed algorithm imposes the block implementation of SFC-VSSLMS in which an approximate $l_0$-norm penalty in the cost function of the FC-VSSLMS algorithm is used. In the Section 6.2, a brief review of the BLMS and SFC-VSSLMS are provided and the proposed algorithm is derived. After that, the convergence analysis of the algorithm has been performed and finally, simulation results that compare the performance of the proposed algorithm to the BLMS, the zero-attracting BLMS (ZA-BLMS) and the reweighted ZA-BLMS (RZA-BLMS) algorithms are provided and discussed.

### 6.2. The Proposed Algorithm

Consider a linear system identification setting with input-tap vector $\mathbf{x}(n) = [x(n), x(n-1), ..., x(n-N+1)]^T$ and output of the unknown system $d(n)$ related by

$$d(n) = \mathbf{h}^T\mathbf{x}(n) + \upsilon(n), \tag{6.1}$$

where $\mathbf{h} = [h_0, \ldots, h_{N-1}]^T$ is the unknown system coefficients with length $\mathbf{N}$ and $\upsilon(n)$ is the observation noise. The update equation of the conventional LMS algorithm is

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}(n). \tag{6.2}$$

In BLMS algorithm (see Fig. 6.2), the filter tap weights are updated once after a collection of every block of data samples. Using $k$ to denote the block index, the vector formulation of the BLMS algorithm is derived as follows:
Define the matrix

$$\mathbf{X}(k) = [\mathbf{x}(kL), \mathbf{x}(kL+1), ..., \mathbf{x}(kL+L-1)]^T,$$

and the column vectors

$$\mathbf{d}(k) = [d(kL), d(kL+1), ..., d(kL+L-1)]^T,$$

$$\mathbf{e}(k) = [e(kL), e(kL+1), ..., e(kL+L-1)]^T,$$

where L is the block length and

$$\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{X}(k)\mathbf{w}(k).$$

Then, the update equation of the BLMS algorithm can be derived as

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_B \sum_{i=0}^{L-1} \mathbf{x}(kL+i)e(kL+i), \quad (6.3)$$

where $\mu_B$ is the block step-size and $i = 0, ..., L-1$. In another form as in [15]:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_B \mathbf{X}^T(k)\mathbf{e}(k). \quad (6.4)$$

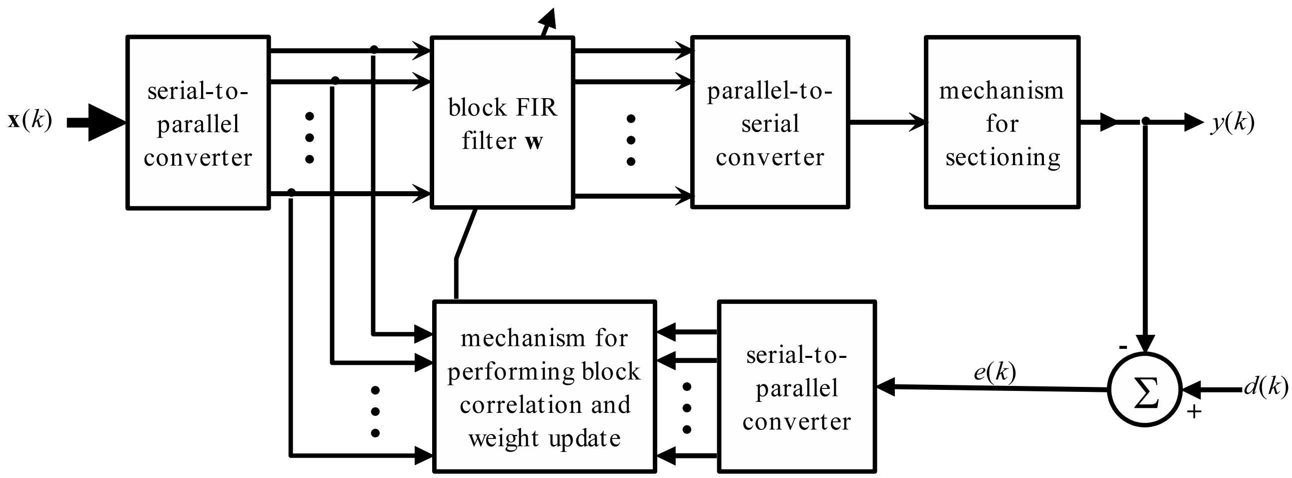


Figure 6.2. Block diagram of a general BLMS algorithm.

The block length is naturally chosen as the same that of the filter length in most applications. Because, when $L$ is greater than $N$, the gradient estimation uses more information than the filter, resulting in redundant operations. For $L$ less than $N$, the filter length is larger than the input block being processed, which is a waste of filter weights. So we have used the same length for block-length and filter-length in our computations.

In this work, we propose a block version of the SFC-VSSLMS algorithm proposed in [63]. The update equation of the SFC-VSSLMS algorithm is:

$$\mathbf{w}\left(n+1\right)=\mathbf{w}\left(n\right)+\mu\left(n\right)e\left(n\right)\mathbf{x}\left(n\right)-\rho(n)sgn\left[\mathbf{w}\left(n\right)\right]e^{-\lambda|\mathbf{w}(n)|}. \tag{6.5}$$

Using the block implementation of the LMS algorithm, the update equation of the proposed algorithm can be written as

$$\mathbf{w}(k+1)=\mathbf{w}(k)+\mu_B(k)\mathbf{X}^T(k)\mathbf{e}(k)-\rho_B(k)sgn\left[\mathbf{w}(k)\right]e^{-\lambda|\mathbf{w}(k)|}, \tag{6.6}$$

where $\rho_B(k)$ is the block sparsity aware parameter and $\mu_B\left(k\right)$ is the block variable step-size parameter and given by [55]

$$\mu_B(k+1) = \alpha_B \mu_B(k) + \gamma_B f(k) \frac{\overline{\mathbf{e}(k)}^2}{\widehat{e}_{ms}^2(k)}, \tag{6.7}$$

where $0 < \alpha_B <, \gamma_B > 0$ are some positive constants and $\overline{\mathbf{e}(k)}$ is mean value of error vector. $\widehat{e}_{ms}^2(k)$ is the estimated mean-square-error (MSE) and defined as

$$\widehat{e}_{ms}^2(k) = \beta_B \widehat{e}_{ms}^2(k-1) + (1-\beta_B)\overline{\mathbf{e}(k)}^2, \tag{6.8}$$

where $\beta_B$ is a weighting factor given as $0 \ll \beta_B < 1$ and $f(k)$ is a control function given in [55]. The proposed algorithm is summarized in Table 6.1.

## 6.3. Computational Complexity of the Proposed Algorithm

Although the proposed algorithm is much more faster than the time domain SFC-VSSLMS algorithm, their computational complexities are the same. We just have the advantage of parallel calculation during the iteration of the update equation.

Table 6.1. Summary of the Proposed Algorithm.

| | |
|---|---|
| define $N, L, \mu_B, \rho_B, \alpha_B, \gamma_{sB} and \beta_B$ | |
| initialize $\mathbf{w}(0)$, | |
| for $k = 1, 2, ...$ | |
| Estimation Error | $\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{X}(k)\mathbf{w}(k)$ |
| Tap-Weight Vector Adaptation | $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu_B(k)\mathbf{X}^T(k)\mathbf{e}(k) - \rho_B(k) sgn[\mathbf{w}(k)] e^{-\lambda\|\mathbf{w}(k)\|}$ |
| where | |
| Step-Size Adaptation | $\mu_B(k+1) = \alpha_B \mu_B(k) + \gamma_B f(k) \frac{\overline{\mathbf{e}(k)}^2}{\widehat{e}_{ms}^2(k)}$ |
| Estimated MSE | $\widehat{e}_{ms}^2(k) = \beta_B \widehat{e}_{ms}^2(k-1) + (1-\beta_B)\overline{\mathbf{e}(k)}^2$ |

## 6.4. Convergence Analysis of the Proposed Algorithm

In this section we perform the convergence analysis of the proposed algorithm assuming that the input signal and the noise are statistically independent. We use the misalignment vector which is defined in the followings in our analysis:

$$\boldsymbol{\theta}\left(k\right) = \mathbf{w}\left(k\right) - \mathbf{h} \tag{6.9}$$

Subtracting the system coefficient vector $\mathbf{h}$ from both side of the update equation (6.3),

$$\mathbf{w}(k+1) - \mathbf{h} = \mathbf{w}(k) - \mathbf{h} + \mu_B \sum_{i=0}^{L-1} \mathbf{x}(kL+i)(d(kL+i) - \mathbf{x}(kL+i)^T \mathbf{w}(k)), \tag{6.10}$$

we obtain the misalignment vector of the BLMS algorithm as

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) + \mu_B \sum_{i=0}^{L-1} \mathbf{x}(kL+i)(d(kL+i) - \mathbf{x}(kL+i)^T \mathbf{w}(k)). \tag{6.11}$$

Substituting (6.1) in (6.11) and rearranging it, we get

$$\boldsymbol{\theta}(k+1) = [\mathbf{I} - \mu_B \sum_{i=0}^{L-1} \mathbf{x}(kL+i)\mathbf{x}(kL+i)^T]\boldsymbol{\theta}(k) + \mu_B \sum_{i=0}^{L-1} \mathbf{x}(kL+i)\upsilon(kL+i)). \tag{6.12}$$

Taking the expectation of (6.12) with the independence assumption, we obtain

$$E[\boldsymbol{\theta}(k+1)] = [\mathbf{I} - \mu_B E[\mathbf{x}(kL+i)\mathbf{x}(kL+i)^T] E[\boldsymbol{\theta}(k)] = (\mathbf{I} - \mu_B \mathbf{R}) E[\boldsymbol{\theta}(k)]. \tag{6.13}$$

Applying the same method to (6.6) we get,

$$E[\boldsymbol{\theta}(k+1)] = (\mathbf{I} - \mu_B \mathbf{R})E[\boldsymbol{\theta}(k)] - \rho_B(k)sgn\left[\mathbf{w}(k)\right]e^{-\lambda|\mathbf{w}(k)|}. \tag{6.14}$$

In (6.14), $\rho_B(k)sgn\left[\mathbf{w}(k)\right]e^{-\lambda|\mathbf{w}(k)|}$ is bounded. Therefore $E[\boldsymbol{\theta}(k)]$ converges to zero if $(\mathbf{I} - \mu_B\mathbf{R}) \in (-1, 1)$. This is the same stability condition as that of the conventional LMS algorithm.

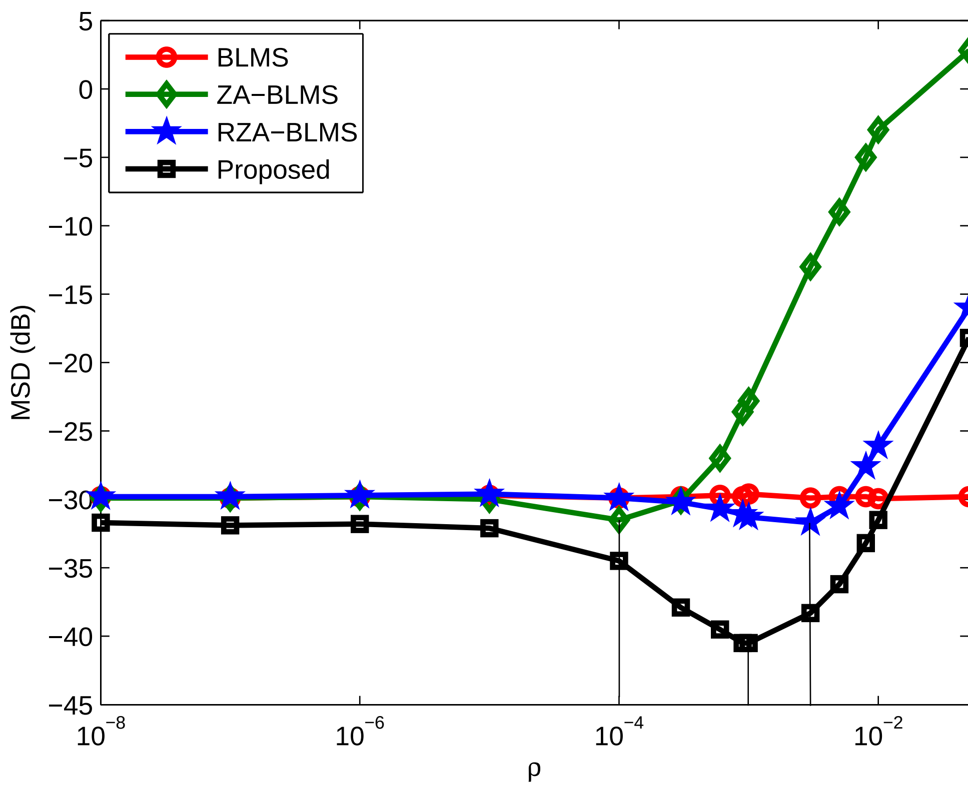


Figure 6.3. $\rho$ vs. MSD of the BLMS, ZA-BLMS, RZA-BLMS and the proposed algorithm.

### 6.5. Simulation Results

In the first part of this section, the performance of the proposed algorithm is compared to those of the BLMS, ZA-BLMS and the RZA-BLMS algorithms in sparse system identification settings in the presence of white Gaussian signal in a stationary environment. All the experiments are implemented with 100 independent runs.

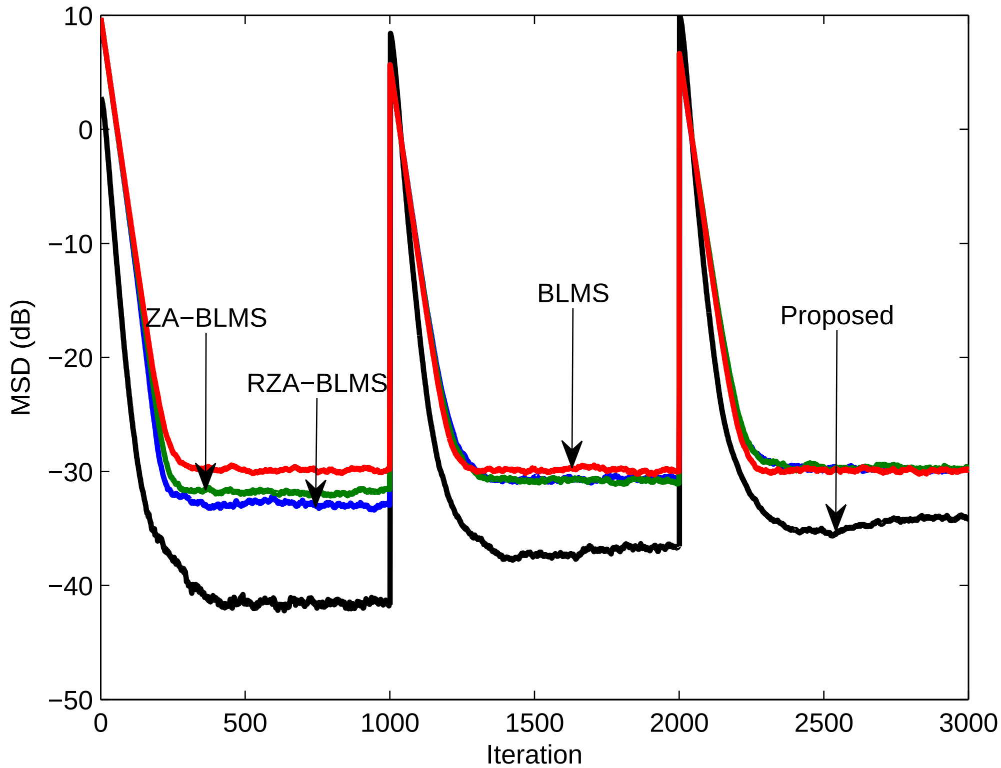


Figure 6.4. Steady state behavior of the BLMS, ZA-BLMS, RZA-BLMS and the proposed algorithm in 95%, 75% and 50% sparse systems, respectively.

In the first experiment, the proposed algorithm has been compared to the BLMS, ZA-BLMS and the RZA-BLMS algorithms with three different filters of 20 coefficients which have one coefficient as '1' (95% sparsity) for the first 20000 iterations, five random coefficients as '1' (75% sparsity) for the second 20000 iterations and 10 random coefficients as '1' (50% sparsity) for the third 20000 iterations. The observed noise and the input signal are assumed to be a white Gaussian random sequence which have appropriate variances so that the SNR is 10 dB. The performance measure used is $MSD = E\{\|\mathbf{h}-\mathbf{w}(k)\|^2\}$. Simulations were done with the parameters for BLMS: $\mu = 0.001$; ZA-BLMS: $\mu = 0.001$, $eps = 10$ and $\rho = 10^{-4}$; RZA-LMS: $\mu = 0.001$, $\epsilon = 10$ and $\rho = 3.10^{-3}$ and for the proposed algorithms as: $\alpha = 0.99$, $\beta = 0.99$, $\gamma = 0.003$, $L = 400$, $\lambda = 8$ and $\rho = 10^{-3}$. The optimum $\rho$ for each algorithm has been calculated by extensive simulations and shown in Fig. 6.3. Fig. 6.4 gives the MSD vs. iteration number for the four algorithms. It is seen from the figure that although the proposed algorithm has a slightly slow convergence but has much lower MSD than that of the others.

In the second experiment, sparsity degree vs. MSD has been analyzed and simulated in Fig. 6.5. Starting from 2 (90% ), the number of non-zero filter coefficients have been increased by units of 2 till 20 (0 % ). It seems that BLMS does not depend on sparsity besides the MSD of the proposed algorithm decreases with decreasing sparsity. But for all degree of sparsity, the proposed algorithm has a lower MSD than that of the other algorithms.

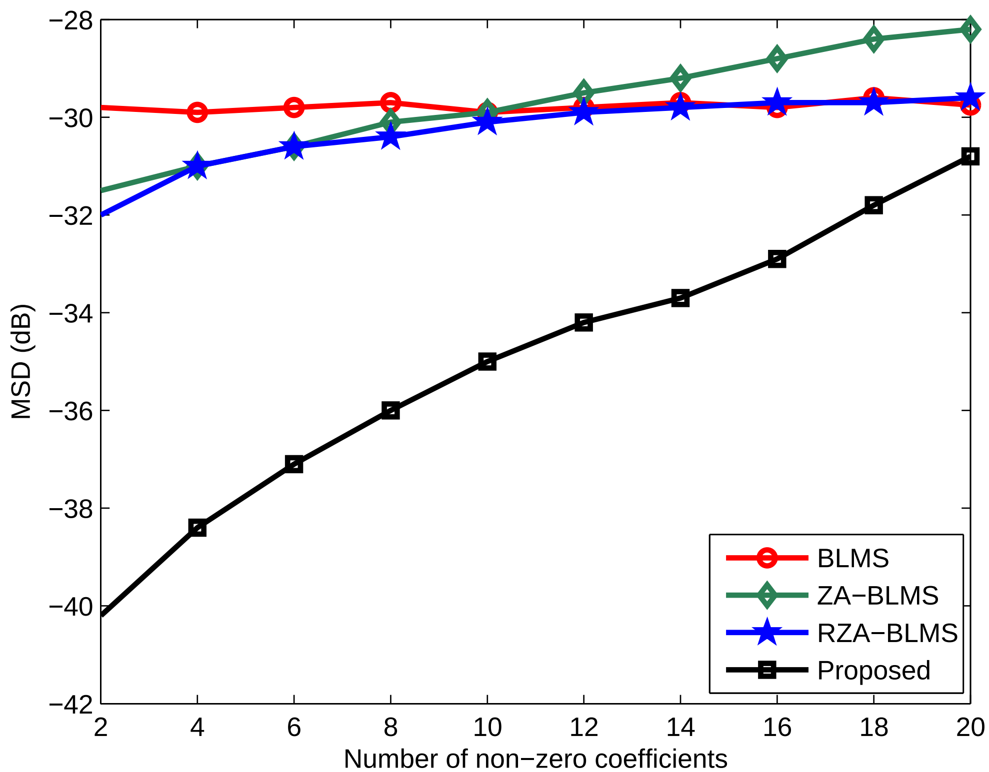

Figure 6.5. Sparsity vs. MSD for BLMS and the proposed algorithm.

In the third experiment, the effects of the filter length and, hence, the block length on the MSD of the algorithms were simulated and shown in Fig. 6.6. An increase in filter length from 20 to 200 causes an increase in the MSD by 11 dB for BLMS algorithm, while the MSD of the proposed algorithm increases by 6 dB (less affected by filter length than the BLMS algorithm).

In the fourth experiment, we investigated how SNR affects the performance of the algorithms in terms of MSD. the SNR value is changed from 0 dB to 30 dB. Fig. 6.7 shows that although the MSD of BLMS algorithm decreases linearly, the proposed algorithm always has a lower MSD

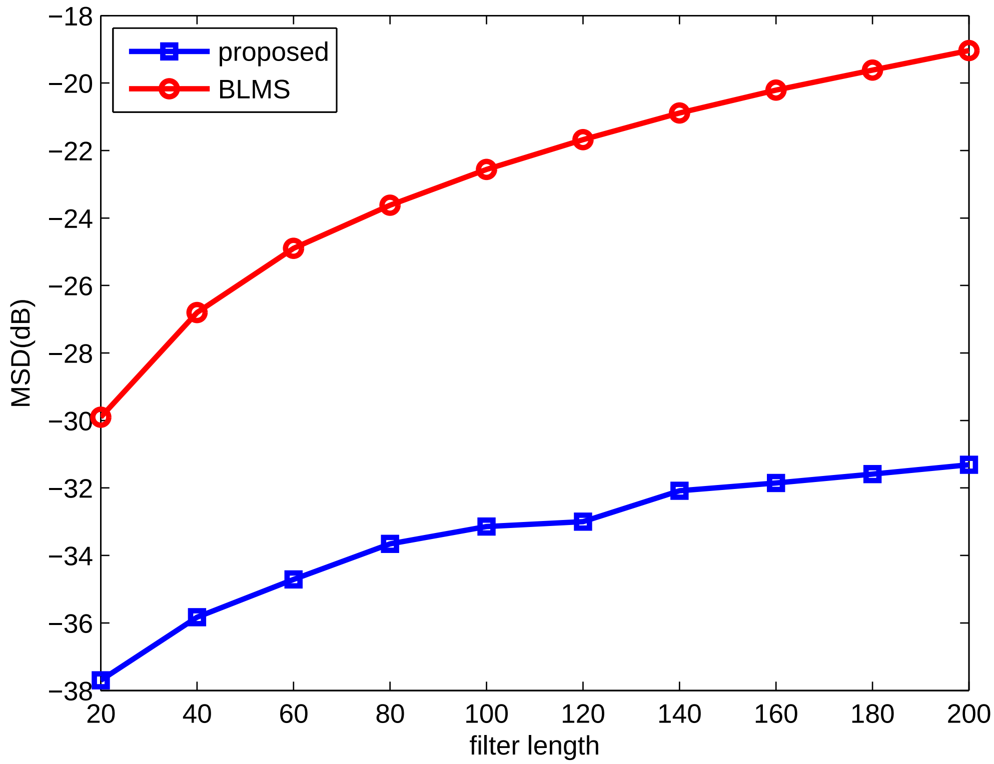

Figure 6.6. Filter length vs. MSD for BLMS and the proposed algorithm.

performance than that of the BLMS algorithm.

In the second part, we investigate the performance of the proposed [87] algorithm in non-stationary environment under different parameters and noise types. The performance of the proposed algorithm is compared to those of the BLMS and RZA-BLMS algorithms in non-stationary sparse system identification settings in the presence of AWGN and additive uniformly distributed noise (AUDN) sequences. All the experiments are implemented with 200 independent runs. The system is assumed to be slowly changing in time to represent a time-varying unknown system defined in [23] as:

$$\mathbf{h}(k) = \varepsilon\mathbf{h}(k-1) + \sqrt{1-\varepsilon^2}\mathbf{s}(k) \tag{6.15}$$

where $\varepsilon = 0.99999, \mathbf{h}(k) = [h_0(k), h_1(k), ..., h_{N-1}(k)]$ and $\mathbf{s}(k) = [s_0(k), s_1(k), ..., s_{N-1}(k)]^T$ is a random sequence with elements drawn from a normal distribution with zero mean and unit

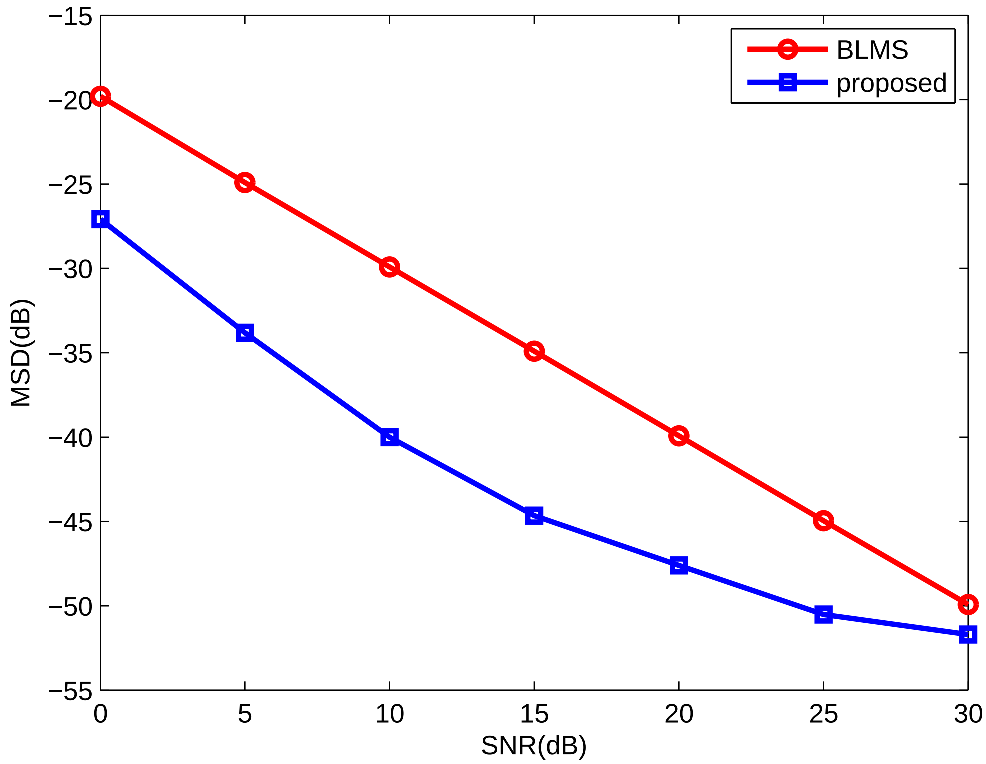

Figure 6.7. SNR vs. MSD for BLMS and the proposed algorithm.

variance.

In the first experiment, the performance of the proposed algorithm is compared to those of the BLMS and RZA-BLMS algorithms with a filter length of 20 coefficients. The unknown system is assumed to be initially sparse with 2 coefficients set to '1' and 18 coefficients set to '0' (90 $\%$ sparsity) and then is fitted to (6.15). The observed noise ( $v(k)$ in Fig. 6.1) and the input signal are assumed to be a white Gaussian random sequences with 10 dB signal-to-noise ratio (SNR). The performance measures used here are the MSD and the convergence rate. Simulations are done with the following parameters. For the BLMS: $\mu = 0.001$. For the RZA-LMS: $\mu = 0.001$, $\epsilon = 10$ and $\rho = 10^{-3}$. For the proposed algorithm: $\alpha = 0.99$, $\beta = 0.99$, $\gamma = 0.0001$, $L = 400$, $\lambda = 8$ and $\rho = 5.10^{-5}$. The optimum $\rho$ for each algorithm is calculated by extensive simulations and shown in Fig. 6.8 (no $\rho$ parameter in the BLMS algorithm, hence the graph of the MSD is almost constant). Fig. 6.9 provides the MSD vs. iteration number of the three algorithms. It can be seen from the figure that although the proposed algorithm has the same convergence rate as the algorithms but it has much lower MSD than the others. In addition to that, the same experiment is repeated with the

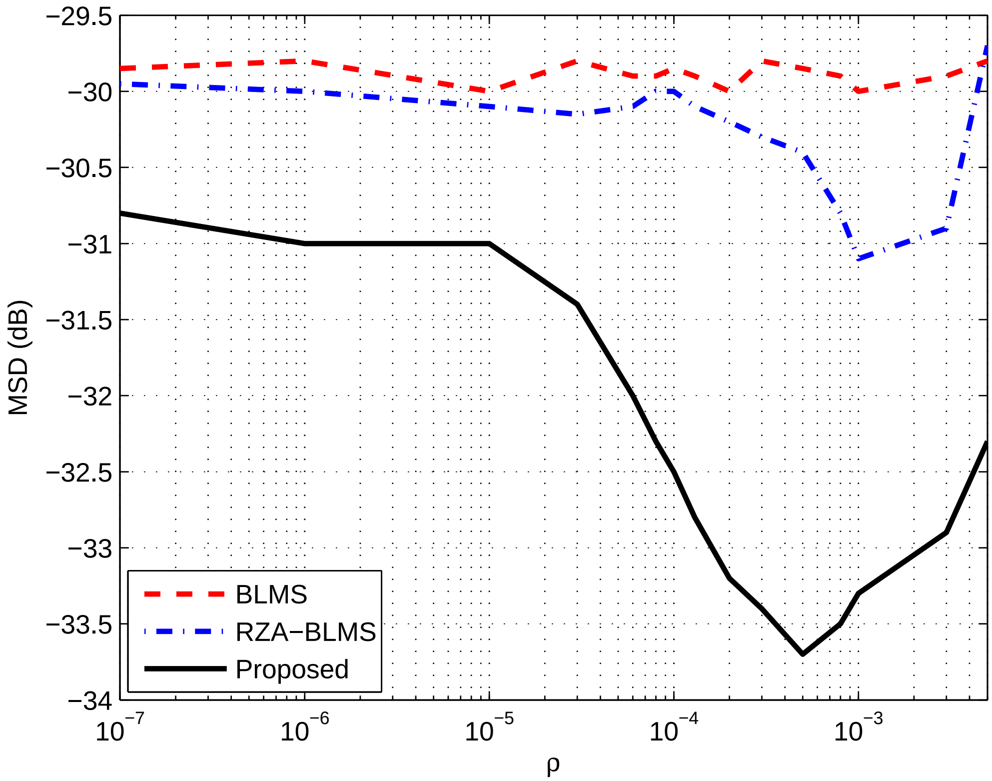

Figure 6.8. $\rho$ vs. MSD of the BLMS, RZA-BLMS and the proposed algorithm.

same parameters but with different levels of sparsity. Table 6.2 shows that the proposed algorithm always outperforms the other algorithms.

In the second experiment, the proposed algorithm is compared to the other algorithms with the same settings and parameters as in the previous experiment but with AUDN. Fig. 6.10 gives the MSD vs. iteration number for the three algorithms. Although the MSD performance of all algorithms is worse than that of the first experiment (due to the nature of the additive noise), still

Table 6.2. Convergence rate and MSD comparisons of the algorithms for different sparsity levels.

| | 95 % sparsity | | 75 % sparsity | | 50 % sparsity | |
|---|---|---|---|---|---|---|
| | Conv.rate(itr.) | MSD(dB) | Conv.rate(itr.) | MSD(dB) | Conv.rate(itr.) | MSD(dB) |
| BLMS | 250 | -29.8 | 280 | -29.8 | 280 | -29.8 |
| RZA-BLMS | 250 | -31.3 | 300 | -30.5 | 320 | -30.1 |
| Proposed | 250 | -33.8 | 280 | -32.9 | 290 | -32.2 |

the proposed algorithm outperforms the other algorithms.

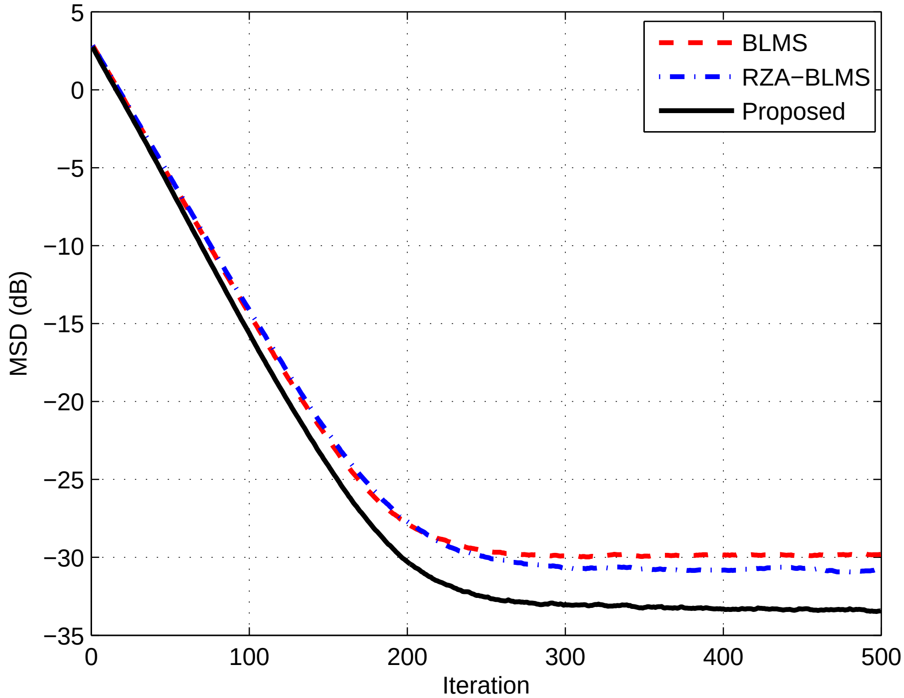


Figure 6.9. Steady state behavior of the BLMS, RZA-BLMS and the proposed algorithm with AWGN for 90 % sparsity.

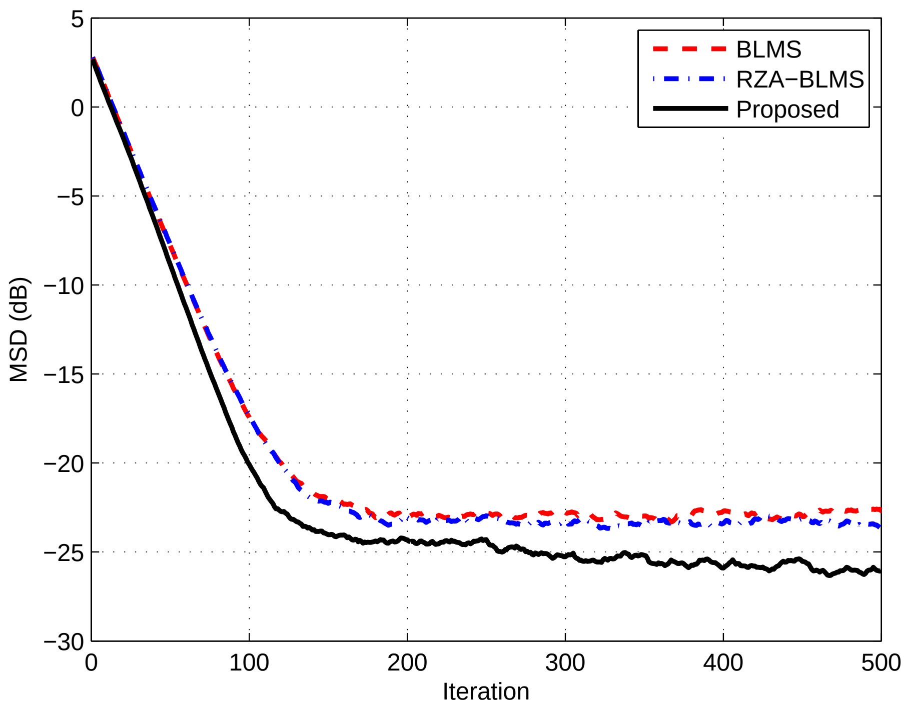


Figure 6.10. Steady state behavior of the BLMS, RZA-BLMS and the proposed algorithm with AUDN for for 90 % sparsity.

# CHAPTER 7

## CONCLUSIONS AND RECOMMENDATIONS

In this thesis, different solutions to acoustic echo problems in telecommunications through adaptive algorithms were proposed. Since acoustic echo path behaves as a sparse system and AEC is a system identification problem so we proposed novel algorithms for sparse system identification. The proposed algorithms are based on the FC-VSSLMS algorithm. The performances of the proposed algorithms are compared to those of the LMS, VSSLMS, FC-VSSLMS, ZA-LMS and RZA-LMS algorithms in terms of convergence speed and MSD's.

Initially, we proposed the SFC-VSSLMS algorithm by adding an approximate $l_0$-norm penalty to the cost function of the FC-VSSLMS algorithm. The analysis of the algorithm is presented and its stability criterion is derived. Then, to overcome the correlated signal problems, a transform domain version of the SFC-VSSLMS algorithm is proposed and its convergence analysis is derived. In order to decrease the time of computation for long filter taps, a block implementation of the proposed algorithm is derived with analysis. For all three versions of the proposed algorithm, the computational complexities are performed and compared to the other algorithms.

All the experiments are performed in MATLAB. Simulations show that the proposed algorithms always outperform the aforementioned algorithms in different environments and different settings.

As a future work, the advantages of the sparsity, variable step-size and $l_0$-norm penalty can be applied to different adaptive algorithms to improve their performances. Another investigation is the use of the proposed algorithms in other areas of adaptive filtering techniques such as interference cancellation and channel equalization. Additionally, since a sparse system identification has a better performance than that of a dispersive system, using a transformation that provides a sparse system from a dispersive one, will improve the performance.

Actually in this work, the proposed algorithms were tested using MATLAB program. Both input and noise signals were artificially generated in the same manner as in other related works. So, it is recommended that, the proposed algorithms be tested in real-time applications with real echo and noise signals as a future investigation.

# APPENDIX A

Without loss of generality, assume that the power of the input signal is unity, i.e, $E(x_n^2) = 1$. From matrix theory [88], for any square matrix A with size $N \times N$ , a maximum eigenvalue $(\lambda_{max})$ and a minimum eigenvalue $(\lambda_{min})$,

$$\lambda_{max} \leq Tr(A) \tag{A1}$$

and

$$\lambda_{min} \geq Det(A), \tag{A2}$$

where *Tr* and *Det* are trace and determinant operators, respectively. Therefore the ratio of

$$\psi(A) = \frac{Tr(A)}{Det(A)} \geq \frac{\lambda_{max}}{\lambda_{min}}. \tag{A3}$$

Defining

$$R_{XX} = E[\mathbf{X}\mathbf{X}^T] = \mathbf{T}R_{xx}\mathbf{T}^T, \tag{A4}$$

where $R_{XX}$ and $R_{xx}$ are the autocorrelation matrices of the transformed and non-transformed input signals, respectively.

$$Tr(D^{-1}R_{XX}) = Tr(R_{xx}) = N \tag{A5}$$

and

$$Det(D^{-1}R_{XX}) = Det(D_{-1})Det(R_{xx}). \tag{A6}$$

Therefore, dividing (A5) by (A6)

$$\begin{aligned}\psi(D^{-1}R_{XX}) &= \frac{N}{Det(D^{-1})Det(R_{xx})} \\ &= Det(D)\psi(R_{xx})\end{aligned} \tag{A7}$$

Since the $Det(D)$ is always assured to be less than or equal to unity , hence

$$\psi(D^{-1}R_{XX}) \leq \psi(R_{xx}). \tag{A8}$$

In other words, (A8) shows that, with a proper orthogonal transform, eigenvalue spread can be reduced.

# REFERENCES


[1] Antoniou A.: Digital Signal Processing: Signals, Systems and Filters, 2006, McGraw-Hill, Canada.

[2] Manolakis D. and Ingle V. K.: Applied Digital Signal Processing: Theory and Practice, 2011, Cambridge, NY .

[3] Bellanger M. G.: Adaptive Digital Filters, 2001, Second Ed., Marcel Dekker, NY.

[4] Vashegi S. V.: Advanced Digital Signal Processing and Noise Reduction, 2008, Fourth Ed., Wiley, Singapore.

[5] Ingle V. K. and Proakis J. G.: Digital Signal Processing using Matlab, 2012, Third Ed., Cengage Learning, USA.

[6] Ozbay Y., Karlik B. and Kavasoglu A. R.: A Windows-Based Digital Filter Design, Mathematical and Computational Applications, 2003, 8, 3, 287-294.

[7] White S.: Digital Signal Processing: A Filtering Approach, 2000, Delmer, USA.

[8] Winder S.: Analog and Digital Filter Design, 2002, Second Ed., Elsevier, USA.

[9] Singh R. R.: Network Analysis and Synthesis, 2013, McGraw Hill Ed., India.

[10] Mandal M. and Asif A.: Continuous and Discrete Time Signals and Systems, 2007, Camridge, NY.

[11] Zaknich A.: Principles of Adaptive Filters and Self-learning Systems, 2005, Springer, Germany.

[12] Diniz P. S. R.: Adaptive Filtering Algorithms and Practical Implementation, 2013, Fourth Ed., Springer, NY.

[13] Lee K. A., Gan W. S. and Kuo S.M.: Subband Adaptive Filtering: Theory and Implementation, 2009, Wiley, GB.

[14] Vijay K. M. and Douglas B. W.: Subband Adaptive Filtering: Theory and Implementation, 1998, CRC Press, USA.

[15] Behrouz F. B.: Adaptive Filters: Theory and Applications, 2013, Second Ed., Wiley, USA.

[16] Salman M. S., Kukrer O. and Hocanin, A.: Adaptive Filtering Fundamentals and Applications, 2011, LAP Lambert, Germany.

[17] Widrow B. and Stearns S. D.: Adaptive Signal Processing, 1985, Prentice Hall, USA.

[18] Vega L. R. and Rey H.: A Rapid Introduction to Adaptive Filtering, 2013, Springer, NY.

[19] Morales L. G.: Adaptive Filtering Applications, 2011, InTech, Croatia.

[20] Ozbay Y. and Kavsaoglu A. R. : An optimum algorithm for adaptive filtering on acoustic echo cancellation using TMS320C6713 DSP, Journal of Digital Signal Processing, Elsevier, 2010, 20, 133-148.

[21] Morales L. G.: Adaptive Filtering, 2011, InTech, Croatia.

[22] Paleologu C., Benesty J. and Ciochina S.: Sparse Adaptive Filters for Echo Cancellation, Synthesis Lectures on Speech and Audio Processing, 2010, Morgan- Claypool.

[23] Loganathan P.: Sparseness-Controlled Adaptive Algorithms for Supervised and Unsupervised System Identification, Ph.D. Thesis, 2011, Imperial College, London.

[24] Starck J. L., Murtagh F. and Fadili J. M.: Sparse Image and Signal Processing: Wavelets, Curvelets, Morphological Diversity, 2010, Cambridge, USA.

[25] Elad M.: Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing, 2010, Springer, NY.

[26] Schreiber W.F.: Advanced television systems for terrestrial broadcasting: some problems and some proposed solutions, Proceedings of the IEEE, 1995, 83, 958-981.

[27] Gu Y., Jin J. and Mei S.: $l_0$-Norm Constraint LMS for Sparse System Identification, IEEE Signal Processing Letters, 1985, 9, 774-777.

[28] Stergiopoulos S.: Advanced Signal Processing Handbook : Theory and implementation for radar, sonar, and medical imaging real-time systems, 2001, CRC Press, USA.

[29] Mathews V. J. and Zhenhua X.: Stochastic Gradient Adaptive Filters with Gradient Adaptive Step-Sizes, International Conference on Acoustics, Speech, and Signal Processing (ICASSP1990), April 1990, 1385-1388.

[30] Mulgrew B. and Cowan C. F. N.: Adaptive Filters and Equalisers, 1988, Kluwer Academic Publishers, USA.

[31] Ogunfunmi T.: Adaptive Nonlinear System Identification: The Volterra and Wiener Model Approaches, 2007, Springer, USA.

[32] Benvenuto N. and Cherubini G.: Algorithms for Communications Systems and their Applications, 2002, Wiley, England.

[33] Panuska V.: An adaptive recursive-least-squares identification algorithm, IEEE Symposium on Adaptive Processes Decision and Control, 1969, 65.

[34] Cioffi J. M.: A covariance RLS lattice for adaptive filtering, IEEE International Conference on Acoustics,Speech, and Signal Processing, 1986, 11, 2115-2118.

[35] Liu Z., Liu Y. and Li C.: Distributed Sparse Recursive Least-Squares Over Networks, IEEE Transactions on Signal Processing, 2014, 62, 6, 1386-1395.

[36] Xiao L., Wu M., Yang J. and Tian J.: Robust RLS via the nonconvex sparsity prompting penalties of outlier components, IEEE China Summit and International Conference on Signal and Information Processing, 2015, 997-1001.

[37] Mathews V. J. and Douglas S. C.: Adaptive Filters, 2001, Prentice Hall, USA.

[38] Van Saders J. G. and Bar-Ness Y.: Study of nonlinear variants of the least mean square (LMS) algorithm, IEEE Conference Record. Bridging the Gap. Interoperability, Survivability, Security on Military Communications, 1989, 1, 274-278.

[39] Haykin S.: Adaptive Filter Theory, 1996, Prentice Hall, USA.

[40] Poularikas A. D. and Ramadan Z.M.: Adaptive Filtering Primer with Matlab, 2006, CRC Press, USA.

[41] Ang W-P. and Boroujeny B. F.: A New Class of Gradient Adaptive Step-Size LMS Algorithms, IEEE Transaction on Signal Processing, 2001, 49, 4, 805-810.

[42] Salman M. S., Hameed A. and Karlik B.: Convex Combination of Recursive Inverse Algorithms, Turkish Journal of Electrical Engineering and Computer Sciences (DOI: 10.3906/elk-1306-232)

[43] Jamel T. M. and Mohamed H. L.: Performance Improvements of Adaptive Noise Canceller Using New Adajusted Step Size LMS Algorithm, 3rd IEEE International Conference on Signal Processing Systems (ICSPS 2011), 2011, Yantai, China.

[44] Harris R., Chabries D. M. and Bishop F.: A variable step (VS) adaptive filter algorithm, IEEE Transactions on Acoustics, Speech and Signal Processing, 1986, 34, 2, 309-316.

[45] Aboulnasr T. and Mayyas K.: A robust variable step-size LMS-type algorithm: analysis and simulations, IEEE Transaction on Signal Processing, Computers and Applications, 1997, 45, 3, 631-639.

[46] Chen Y., Tian J. and Liu Y.: Variable step size LMS algorithm based on modified Sigmoid function, IEEE International Conference on Audio, Language and Image Processing (ICALIP), 2014, 627-630.

[47] Sayed A. H.: Fundamentals of Adaptive Filtering, 2003, Wiley, USA.

[48] Sayed A. H.: Adaptive Filters, 2008, Wiley, USA.

[49] Shi K. and Ma X.: Transform domain LMS algorithms for sparse system identification, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2010, 3714-3717.

[50] Alkassim M. A.: Sparse Adaptive Filters for Echo Cancellation, M.S. Thesis, 2014, Mevlana (Rumi) University, Konya, Turkey.

[51] Dogancay K.: Partial-Update Adaptive Filters and Adaptive Signal Processing: Design, Analysis and Implementation, 2008, Elsevier, Hungary.

[52] Poularikas A. D.: Adaptive Filtering: Fundamentals of Least Mean Squares with Matlab, 2015, CRC Press, USA.

[53] Kwong R. H. and Johnston E.W.: A Variable Step-Size LMS Algorithm, IEEE Transaction on Signal Processing, 1992, 40, 7, 1633-1641.

[54] Wu X., Gao L. and Tan Z.: An improved variable step size LMS algorithm, IEEE International Conference on Measurement, Information and Control (ICMIC), 2013, 1, 533-536.

[55] Li M., Li L. and Tai H-M.: Variable Step Size LMS Algorithm Based on Function Control, Circuits, Systems and Signal processing, 2013, 32, 6, 3121-3130.

[56] Chen Y., Gu Y. and Hero A. O.: Sparse LMS for System Identification, IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP2009), Taipei, Taiwan, April 2009, 3125-3128.

[57] Petraglia Y. F. and Delores M. E.: Improved PNLMS algorithm employing wavelet transform and sparse filters, IEEE 16th European Signal Processing Conference, 2008, 1-5.

[58] Cheng Y. F. and Delores M. E.: Analysis of an Adaptive Technique for Modeling Sparse Systems, IEEE Transactions on Acoustics, Speech and Signal Processing, 1989, 37, 2, 254-264.

[59] Martin R.K. and Johnson C.R.: NSLMS: a proportional weight algorithm for sparse adaptive filters, IEEE Asilomar Conference on Signals, Systems and Computers, 2001, 2, 1530-1534.

[60] Deng H. and Doroslovacki M.: New Sparse Adaptive Algorithms Using Partial Update, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2004, 2, ii-845-8.

[61] Bin Saeed M. O. and Sheikh A. U. H.: A New LMS Strategy for Sparse Estimation in Adaptive Networks, IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2012, 1728-1733.

[62] Salman M. S., Jahromi N., Hocanin A. and Kukrer O.: A Zero-Attracting Variable Step-Size LMS Algorithm for Sparse System Identification, IX International Symposium on Telecommunications (BIHTEL2012), Sarajevo, Bosnia and Herzegovina, October 2012, 1-4.

[63] Turan C. and Salman M. S.: A Sparse Function Controlled Variable Step-size LMS Algorithm for System Identification, IEEE 22nd Signal Processing and Communications Applications Conference (SIU2014), Trabzon, Turkey, April 2014, pp 329-332.

[64] Tibshirani R.: Regression Shrinkage and Selection via the LASSO, Journal of Royal Statistical Society, Series B, 1996, 58, 267-288.

[65] Candes E.: Compressive sampling, International Congress of Mathematics, 2006, 3, 1433–1452.

[66] Candes E. J., Wakin M. and Boyd. S.: Enhancing Sparsity By Reweighted $l_1$-Minimization, Journal of Fourier Analysis and Applications, October 2008, 14, 877-905.

[67] Baraniuk R.: Compressive sensing, IEEE Signal Processing Magazine, March 2007, 25, 21–30.

[68] Rey H. and Vega L. R.: A rapid Introduction to Adaptive Filtering, Springer, NY, 2013.

[69] Jamel T. M. and Al-Magazachi K. K.: Simple variable step size LMS algorithm for adaptive identification of IIR filtering system, IEEE International Conference on Communications, Computers and Applications, 2012, 23-28.

[70] Hwang J. K. and Li Y. P.: Variable step-size LMS algorithm with a gradient-based weighted average, IEEE Signal Processing Letters, 2009, 16, 12, 1043-1046.

[71] Zhao S., Jones D. L., Khoo S. and Man Z.: New Variable Step-Sizes Minimizing Mean-Square Deviation for the LMS-Type Algorithms, Springer, Circuits, Systems and Signal processing, 33: 2251-2265, 2014.

[72] Zhao S., Man Z., Khoo S. and Wu R. H.: Variable Step-Size LMS Algorithm with a quotient form, Elsevier, Signal processing, 2009, 89, 1, 67-76.

[73] Duttweiler D. L.: Proportionate normalized least-mean-sqaures adaptation in echo cancelers, IEEE Transaction on Speech Audio Processing, 2000, 8, 5, 508-518.

[74] Sing-Long C. A., Tejos C. A. and Irarrazaval P.: Evaluation of continuous approximation functions for the $l_0$-norm for compressed sensing, Proc. Intl. Soc. Mag. Reson. Med., 2009, 17.

[75] Kenney J. F. and Keeping E. S.: Kurtosis, In Mathematics of Statistics, Princeton, NJ, 1962.

[76] Bismor D.: Extension of LMS stability condition over a wide set of signals, International Journal of Adaptive Control and Signal Processing, 2014; DOI: 10.1002/acs.2500.

[77] Salman M. S.: Sparse leaky-LMS algorithm for system identification and its convergence analysis, International Journal of Adaptive Control and Signal Processing, 2014, 28, 1065-1072.

[78] Shi K. and Shi P.: Convergence Analysis of Sparse LMS Algorithms with $l_1$-norm Penalty based on White Input Signal, Journal of Signal Processing, Elsevier, 2010, 90, 12, 3289–3293.

[79] Jin J., Gu Y. and Mei S.: A Stochastic Gradient Approach on Compressive Sensing Signal Reconstruction Based on Adaptive Filtering Framework, IEEE Journal of Selected Topics in Signal Processing, 2010, 4, 2, 409-420.

[80] Narayan S., Peterson A.M. and Narasimha M.J.: Transform domain LMS algorithm, IEEE Transactions on Acoustics, Speech and Signal Processing, 1983, 31, 3, 609-615.

[81] Bilcu R. C., Kuosmnen P. and Egiazarian K.: A transform domain LMS adaptive filter with variable step-size, IEEE Signal Processing Letters, 2002, 9, 2, 51-53.

[82] Li Y. and Hamamura M.: Zero-attracting variable-step-size least mean square algorithms for adaptive sparse channel estimation, International Journal of Adaptive Control and Signal Processing, 2015, DOI: 10.1002/acs.2536.

[83] Das B. K. and Chakraborty M.: Sparse Adaptive Filtering by an Adaptive Convex Combination of the LMS and the ZA-LMS Algorithms, IEEE Transactions on Circuits and Systems, 2014, 61, 5, 1499-1507.

[84] Feng Y., Wu J., Zeng R. and Luo L.: Gradient compared $lp$-LMS algorithms for sparse system identification, IEEE Control and Decision Conference (CCDC), 2015, 6280-6284 .

[85] Clark G. A., Mitra S. K. and Parker S. R.: Block Implementation of Adaptive Digital Filters, IEEE Transactions on Acoustics, Speech and Signal Processing, 1981, 29, 3, 744-752.

[86] Benesty J. and Duhamel P.: A fast exact least mean square adaptive algorithm, IEEE Transactions on Signal Processing, 1992, 40, 12, 2904-2920.

[87] Turan C., Salman M. S. and Eleyan A.: A block LMS-type algorithm with a function controlled variable step-size for sparse system identification, IEEE 57th International Symposium ELMAR, 2015, Zadar, Croatia, 1-4.

[88] Strang G.: Linear Algebra and its Applications, New York, Academic Press, 1976.