Sparse Adaptive Filtering and Applications

Mohammad Naser Sabet Jahromi

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Electrical and Electronic Engineering

Eastern Mediterranean University
August 2017
Gazimagusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Mustafa Tümer
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Doctor of Philosophy in Electrical and Electronic Engineering.

Prof. Dr. Hasan Demirel
Chair, Department of Electrical and
Electronic Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality as a thesis of the degree of Doctor of Philosophy in Electrical and Electronic Engineering.

Asst. Prof. Dr. M. Shukri Salman
Cosupervisor

Prof. Dr. Aykut Hocanın
Supervisor

Examining Committee

1. Prof. Dr. Emin Anarım
2. Prof. Dr. A. Enis Çetin
3. Prof. Dr. Hasan Demirel
4. Prof. Dr. Aykut Hocanın
5. Asst. Prof. Dr. M. Shukri Salman

ABSTRACT

In recent years, sparse signal estimation has become an important paradigm in the field of signal processing due to its vast amount of applications. Among the wide range of applications, system identification and echo cancelation are likely two of the most challenging signal estimation problems for many practical channels with sparse nature. For such channels, due to low convergence speed and sensitivity to highly correlated inputs, conventional adaptive filtering algorithms such as least-mean-square (LMS) algorithm and its variants, recursive least-squares (RLS)algorithm and Kalman filters are incapable of exploiting the channel sparsity efficiently. To overcome the difficulties associated with sparse system identification and echo cancelation, l_0-norm constraint LMS (l_0-LMS) modifies the conventional LMS algorithm to capture and utilize the sparsity of the channel . This modification results in a zero-point attraction to all filter-taps. The l_0-norm addition, however, causes the optimization problem to be non-convex and hence not tractable.

In this thesis, we propose three different types of novel sparse adaptive filtering algorithms to achieve faster convergence rate while decreasing the mean-square deviation (MSD). Furthermore, all the novel approaches are transformed into convex optimization problem by imposing either l_1-norm or logarithmic penalty on the filter-tap during the adaptation process. The first algorithm is referred as weighted zero-attracting leaky LMS (WZA-LLMS) algorithm where the original cost function of the leaky-LMS algorithm is modified by an addition of a log-sum penalty that produces an adjustment term in the update equation. The adjustment causes the proposed algorithm to attract the zeros of sparse channel and improves the performance. For system identification and echo

cancelation setting, the proposed algorithm not only yields lower MSD for highly sparse channels but converges at the same rate as the standard zero-attracting-LMS (ZA-LMS) algorithm. In the case of fully non-sparse channels, the WZA-LLMS algorithm performs better than both the LLMS and ZA-LMS algorithms in the same settings. These filters can also be efficiently implemented for potential application such as in finite-precision hardware.

Due to an extra logarithmic cost function, however, the WZA-LLMS algorithm is computationally complex. To reduce the complexity while achieving lower MSD, a zero attractor-variable step-size LMS (ZA-VSSLMS) algorithm is introduced. This algorithm imposes an l_1-norm penalty to the original quadratic cost function of the VSSLMS algorithm which captures the system sparsity during adaptation process. For highly sparse channel, this process accelerates the final convergence and improves the error performance. The convergence analysis for ZA-VSSLMS algorithm is studied when the white process presents at the input of the system. The stability condition of the algorithm is presented. Next, the steady-state mean square deviation (MSD) analysis of the algorithm is carried out. A steady-state MSD expression for the ZA-VSSLMS algorithm is derived mathematically in terms of the system parameters for general white noise process.A crucial upper bound of the zero-attractor controller (ρ) which yields minimum MSD is theoretically shown. The effect of both zero-attractor controller (ρ) and the forgetting factor (α) in ZA-VSSLMS are investigated. Furthermore, the behavior of the ZA-VSSLMS algorithm is studied in the presence of noise with different probability density functions.

Finally, to further improve the ZA-VSSLMS filter when the sparsity of the channel

decreases, with a slight cost in the number of computations, the WZA-VSSLMS algorithm is introduced by adding the same log-sum penalty as in WZA-LLMS algorithm into original cost function of VSSLMS algorithm.

The performance of the ZA-VSSLMS, WZA-VSSLMS and WZA-LLMS algorithms are examined with respect to the standard ZA-LMS, VSSLMS, leaky-LMS, set-member- ship normalized LMS (SM-NLMS) and LMS algorithms in system identification, echo cancelation and image deconvolution problems. Simulation results show that the theoretical and simulation results of the ZA-VSSLMS algorithm not only outperforms the aforementioned algorithms but further are in good agreement with a wide range of parameters, different channels, input signal and noise types.

Keywords: Adaptive Filters, Sparse Signal, Compressive Sensing, LMS Algorithm, Zero Attractor.

ÖZ

Son yıllarda, ayrık sinyal kestirimi, geniş uygulama olanakları sunduğu için, sinyal işlemede önemli bir araştırma alanı olarak ortaya çıkmıştır. Uygulama alanları arasında, ayrık yapıya sahip gerçek kanallar için sistem tanılama ve yankı giderme en önemlileridir. Bu tür kanallar için, LMS, RLS ve Kalman benzeri geleneksel uyarlanır filtre algoritmaları, ayrık yapıya sahip özellikleri kullanamamakta ve yavaş yakınsama ve ilintili gürültüde düşük başarım sağlama gibi sorunlara yol açmaktadırlar. Ayrık yapıyı kullanabilmek için $l0$-norm kısıtı eklenerek LMS algoritması güncellenmektedir. Bu değişiklik, filtre katsayılarının sıfıra doğru yaklaşmalarını sağlamaktadır. Bununla beraber, l_0-norm kısıtının eklenmesi, eniyileştirme problemini dışbükey olmaktan çıkarmakta ve çözümünü zorlaştırmaktadır.

Bu tezde, daha hızlı yakınsama ve ortalama karesel sapmayı (MSD) azaltan üç özgün ayrık uyarlanır filtre önerilmiştir. Bunlara ek olarak, dışbükey olmayan eniyileştirme problemleri, filtrede l_1-norm kısıtı kullanılarak uyarlama adımları sırasında dışbükey hale dönüştürülmüştür. Önerilen birinci algoritma, ağırlıklı sıfıra yaklaşan kaçaklı LMS algoritması, WZA-LLMS, olarak adlandırılmış ve logaritmik toplama dayalı bir ek kısım eklenerek maliyet işlevi güncellenmiştir. Kanalın yapısında bulunan sıfır katsayılarına daha hızlı yaklaşılarak başarım artırılmaktadır. Sistem tanılama ve yankı giderme uygulamalarında, ayrık kanallar için daha düşük MSD elde edilmekte, yakınsama hızı ise standard sıfıra yaklaşan LMS algoritmasına (ZA-LMS) ile benzer olmaktadır. Ayrık olmayan kanallar için de, WZA-LLMS LLMS ve ZA-LMS algoritmalarından daha yüksek başarım göstermektedir. Önerilen filtreler gerçek donanımlarda etkin algoritmaların uygulamasında da kullanılabilmektedir.

Eklenen logaritmik maliyet işlevi nedeniyle WZA-LLMS algoritması, yüksek işlem karmaşıklığına sahiptir. İşlem karmaşıklığını azaltmak için değişken adım büyüklüğüne sahip ZA-VSSLMS algoritması tasarlanmıştır. Bilinen VSSLMS algoritmasına $l1$-norm kısıtı eklenerek ayrık kanal yapısının özellikleri kullanılmıştır. Yüksek ayrık özellikleri olan kanallarda, ZA-VSSLMS algoritması yüksek başarım göstermektedir. Beyaz Gauss gürültüsü altında, algoritma kuramsal olarak analiz edilerek MSD sonucui türetilmiştir. Kalıcıdurum başarımına, sıfırayaklaştırıcı ρ ve α adım uzunluğu parametrelerinin etkileri incelenmiş ve ρ için üst sınır belirlenmiştir. Ayrıca, farklı olasılık dağılımına sahip gürültü dağılımlarının başarıma etkisi araştırılmıştır.

Son olarak, ZA-VSSLMS algoritmasını daha da iyileştirmek için, maliyet işlevine bir logaritmik toplama terimi eklenerek WZA-VSSLMS elde edilmiştir.

Önerilen ZA-VSSLMS, WZA-VSSLMS, WZA-LLMS algoritmalarının, bilinen standard algoritmalar ile başarımları, sistem tanılama, yankı giderme ve imge ters-evrişim problemlerinde kıyaslanmıştır. Benzetim ve kuramsal sonuçlar, önerilen ZA-VSSLMS algoritmasının, farklı gürültü dağılımlarında daha yüksek başarım sağladığını göstermektedir. Ayrıca, kuramsal ve benzetim değerleri farklı parametre değerleri için örtüşmektedir.

Anahtar Kelimeler: Uyarlanır filtre, Ayrık sinyal, Sıkıştırmalı algılama, LMS algoritması, Sıfıra-yaklaşan.

Dedicated to my lovely fiancê, S. Hosseini , who has been standing by me miles away in the last three years.

ACKNOWLEDGMENT

First and foremost, I would like to offer my sincere gratitude to Prof. Aykut Hocanın who has not been only my supervisor but also a true life coach, good friend and inspiring personality to me since beginning of my journey in PhD study. Undoubtedly, without his unflinching supports and encouragements during difficult times, reaching to this stage was not possible. Thank you and thank you and thank you again.

I would like also to appreciate my co-supervisor, Asst. Prof. Dr. Mohammad Salman for all of his continuous supports, brilliant ideas and very crucial role in completion of this research. He has been rather a kind elder brother to me. Forever, I will remain indebted to him.

I also cannot pass these lines without paying enough thanks to my fiancê for being extraordinary kind, supportive and most of all patience in last three years of distance communication. I know, miles away, she suffered a lot throughout the time I had fully dedicated myself to the work rather than her. Thank you very much indeed and let me remind yourself that this moment is yours. My particular thanks go to our chairman Prof. Hasan Demirel for all his supports and guidance. My special gratitude to Prof. Osman Kükrer for his scientific help, enriching ideas and fruitful discussions.

My extra special thanks to Dr. Rouzbeh vaziri and his wife Mrs. Solmaz for sharing their house, time and good memories with me during the last year of my PhD study. I am grateful to them.

I should also be thankful to my friends for their moral supports and helpful tips: Dr. Reza Abrishambaf, Dr. Gholamreza Anbarjafari, Dr. Ehsan Kiani and all the individuals that contributed to this work.

Last but not least, I wish to extend my deepest appreciation to my parents and sisters for their understanding and presence. They mean a lot to me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

α	Steady-State forgetting factor
γ	Positive constant
δ	Misalignment vector
ζ	Positive constant
λ	Regularization parameter
μ	Rate of convergence
ρ	Zero-attractor controller
υ	Leakage parameter
J	Cost function
k	Time Index
σ^2	Variance
σ^4	Fourth Moment
$\mathbf{I}$	Identity matrix
N	Filter length
$\mathbf{p}$	Cross-correlation
$\mathbf{R}$	Autocorrelation matrix
T	Transposition operator
Tr	Trace operator
$\mathbf{S}$	Covariance matrix
$sgn(.)$	Point-wise sign function
$Vec(.)$	Vec operator
$\mathbf{w}$	Filter weights vector

x	Filter input vector
1D	one dimensional data
2D	two dimensional data
AEC	Acoustic echo canceller
AWGN	Additive white Gaussian noise
CS	Compressive sensing
FIR	Finite impulse response
GM	Gauss-Markov model
NZ	Non-zero set
LMS	Least-mean square
MSD	mean-square-deviation
MSE	mean squared error
pdf	probability density function
PSF	Point spread function
RLS	Recursive-least-squares
SM	Set-Membership
SNR	Signal-to-noise ratio
LLMS	Leaky Least-mean square
VSSLMS	Variable step size Least-mean square
ZA	Zero-attractor

Chapter 1

INTRODUCTION

1.1 Introduction

For several decades, adaptive filters have been studied by many researchers due to their wide range of applications in electronic devices such as digital cameras and smart phones [1, 2, 3]. The advantage of self-modification of an adaptive filter for real-time input and its iterative solution has resulted in its application in an extensive range of problems including system identification, channel estimation, echo cancelation and many others. To address such problems, various adaptive algorithms have been developed to reach optimal solution based on a certain optimization criterion. Among these algorithms, least mean square (LMS) algorithm and its variants [1]-[8] are some of the most popular methods that have been widely used for this purpose. However, these methods suffer from low convergence rate, high power consumption and sensitivity to highly correlated input for applications where the systems are sparse in nature [9, 10].

In the recent years, sparse signal recovery has become a popular approach in the field of signal estimation due to its vast range of applications. This inspires the design of more effective adaptive filters for the sparse signal recovery problem that leads to improved performance in both convergence behavior and error cost function.

1.2 Understanding Sparse Signal

A sparse signal is one that contains only a few relatively large amplitudes while the rest are zero or very small. Fig. 1.1 shows a typical sparse signal that resulted from an acoustic echo path in a communication system [11]. As it can be seen, the impulse response of the acoustic echo path has a few large active coefficients for all delay durations. Consequently, the impulse response of an acoustic echo is composed of mainly negligible coefficients (zero or near-zero). As a result of this property, the impulse response produces a sparse signal.

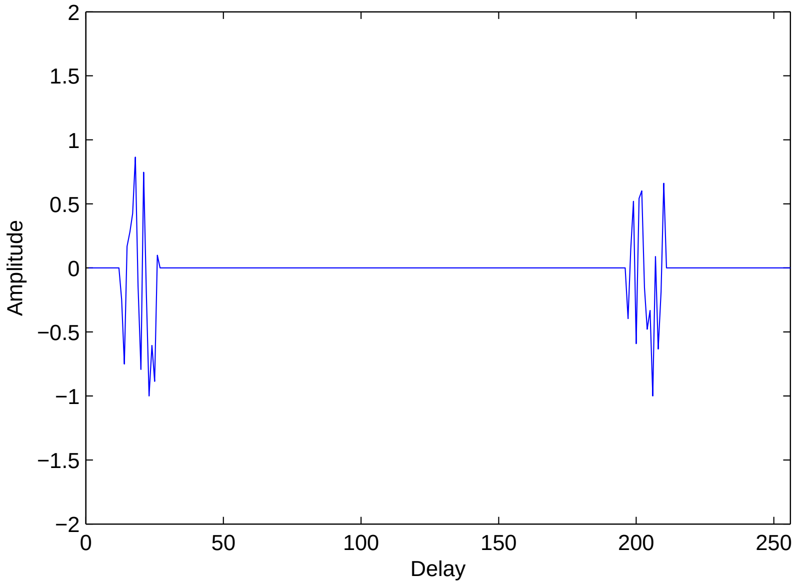


Figure 1.1: An acoustic echo path impulse response; a typical sparse signal, [11].

Such a sparse structure in systems can be found in many real world applications such as digital TV transmission [12], acoustic echo channel [13] and system identification [14] . Recently, Compressive Sensing (CS) [15] has emerged to be an exciting area of research in solving many inverse problems. Sparse signal processing of compressive sensing has an excellent capability of reducing the sampling rate which results in lower implementation cost [15].

1.3 Compressive Sensing

Compressive Sensing (CS) is a novel framework in signal processing that allows estimating a sparse signal via sampling at a much lower rate than Nyquist rate [15]. The main idea behind the CS theory is to solely acquire large amplitudes of the signal with most of its coefficients have values close to zero using a proper minimization technique for reconstruction [16, 17]. In general, estimating these sparse signals involves l_0-norm minimization of sparse vector **w** in the cost function of the form:

$$\min_{\mathbf{w}} ||\mathbf{w}||_0. \tag{1.1}$$

where $||.||_0$ denotes l_0-norm that returns the number of nonzero entries in sparse vector **w**. This gives the minimization process the ability of attracting all zero coefficients [18]. However, the cost function in (1.1) is a non-convex optimization problem and often is very hard to tackle. Many good alternative approximations of l_0-norm such as l_1-norm and a log-sum penalty function [10] have been proposed to overcome such drawback, as these are mathematically more tractable in the minimization process.

1.4 Thesis Contributions

In this work, in order to reduce the effects of the main problems such as low convergence rate and high power consumption of sparse signal estimation that exist in conventional adaptive filtering schemes, three novel versions of LMS-based sparse adaptive algorithms are proposed:

1. A weighted zero-attracting leaky LMS (WZA-LLMS) algorithm [19] that achieves enhanced performance in terms of both convergence rate and MSD by incorporating a logarithmic penalty term into the main cost function of leaky-LMS algorithm.

2. A zero attractor-variable step-size LMS (ZA-VSSLMS) algorithm [20] is introduced which reduces the computational complexity compared to the WZA-LLMS algorithm.

3. A weighted zero attractor-variable step-size LMS (WZA-VSSLMS) algorithm [20] is proposed to further improve the ZA-VSSLMS filter performance when the sparsity of the channel decreases.

For the proposed ZA-VSSLMS algorithm which captures the system sparsity by imposing the l_1-norm penalty into the quadratic cost function of the VSSLMS algorithm during adaptation process, the following studies are conducted:

1. The convergence analysis for ZA-VSSLMS algorithm [21] is studied when the input to the system is white.

2. A steady-state MSD expression for the ZA-VSSLMS algorithm is derived in terms of the system parameters for general white noise process [22].

3. The crucial upper bound of zero-attractor controller (ρ) which yields minimum MSD is theoretically presented [22].

4. The behavior of the ZA-VSSLMS algorithm is studied in the presence of four noise types: Gaussian, uniform, Laplacian, Impulsive. [22].

5. The effects of both the zero-attractor controller (ρ) and the forgetting factor (α) in ZA-VSSLMS algorithm are investigated [22].

The performance of the ZA-VSSLMS, WZA-VSSLMS and WZA-LLMS algorithms are evaluated with respect to the standard ZA-LMS, VSSLMS, leaky-LMS, set-membership normalized LMS (SM-NLMS) [23] and LMS algorithms in system identification, channel estimation, echo cancelation and image deconvolution problems. Simulation results show that the theoretical and simulation results of the ZA-VSSLMS algorithm are in good agreement within a wide range of parameters, different channel, input signal and noise types and outperform the aforementioned algorithms.

1.5 Thesis Outline

The structure of the thesis are arranged in the following order. After a brief introduction on sparsity phenomena in signal processing and its application to adaptive filtering problem in Chapter 1, In Chapter 2 a general background on the adaptive filtering and few LMS-type algorithms, that will be used in all proposed techniques, is given.

In Chapter 3 the proposed algorithms are described. In particular, convergence analysis for ZA-VSSLMS algorithm is studied when the white process presents at the input of the system. The stability condition of the algorithm is presented. Furthermore, the steady-state mean square deviation (MSD) analysis of the algorithm is carried out. A steady-state MSD expression for the ZA-VSSLMS algorithm is derived in terms of the system parameters for general white noise process. The crucial upper bound of zero-attractor controller (ρ) which yields minimum MSD is theoretically shown. The effect of both zero-attractor controller (ρ) and the forgetting factor (α) in ZA-VSSLMS are investigated. Furthermore, the behavior of the ZA-VSSLMS algorithm is studied in the presence of noise with different probability distributions.

The superiority of the proposed methods are presented through the simulation results in Chapter 4 for a wide range of parameters, different channel, input signal and noise types. We conclude the thesis in Chapter 5 by giving a summary of the results, discussions and future work.

Chapter 2

REVIEW OF ADAPTIVE FILTERING

2.1 Background

Adaptive filtering is a popular technique that has been used in a wide spectrum of signal processing and communication problems. In many real world scenarios, it is dealt with unknown time-varying processes which result in undesired distortion of signals [25]. In order to eliminate such unknown distortions, adaptive systems are known to be efficient tools for this purpose. By an adaptive system, we mean the self-designing finite impulse response (FIR) filter which relies on a recursive algorithm to converge to optimum Wiener solution in some statistical sense [1]. In each successive iteration, the output of adaptive filter attempts to minimize the error signal with respect to its desired response in order to update the filter coefficients. In the coming sections, a brief description of three common adaptive filtering algorithms, that will be used throughout this thesis, will be reviewed.

2.2 The Least-Mean-Square (LMS) Algorithm

The least-mean-square (LMS) adaptive algorithm, introduced by Widrow and Hoff [3], is the most widely used method that appears in many application areas, such as adaptive noise cancelation [3], channel equalization [26] and system identification [27], etc. The main reason for the popularity of LMS algorithm is its robustness, low computational complexity and easy hardware implementation [1]. The LMS algorithm has the following important characteristics:

1. The optimum filter [1] solution can be efficiently estimated without computing matrix inversion. Furthermore, the autocorrelation and cross correlation matrices

are not required [28].

2. A step-size, μ, is readily selected in order to control the convergence speed and stability of the algorithm [29].

3. The algorithm is robust and stable in solving many practical adaptive signal processing problems [3, 30, 31].

Table 2.1 summarize the LMS algorithm.

Table 2.1: The LMS algorithm (LMS)

Least-Mean-Square Algorithm	
Data	Input Signal : $\mathbf{x}(k)$ Desired Response: $d(k)$
Initialization	Set Filter-Tap: $\mathbf{w}(0) = 0$ Select Step-Size: μ
Computation	For $k = 0, 1, 2, ..., n$ Error Vector: $e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k)$ Update Filter-Tap: $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu e(k)\mathbf{x}(k)$

2.3 The Leaky Least Mean Square(LLMS)Algorithm

The leaky LMS algorithm (LLMS) has been proposed to mainly combat the numerical instability of the filter in the digital implementation of LMS algorithm [1, 3]. A leakage prevents overflow in finite-precision by providing a trade-off between minimizing the MSE and the energy in the filter's coefficients. This is achieved by adding a regularization term (l_2-norm of filter-taps) into the cost function of the LMS algorithm [1]. Table 2.2 summarize the LLMS algorithm..

Table 2.2: LLMS Algorithm

Leaky Least-Mean-Square Algorithm algorithm Algorithm (LLMS)	
Data	Input Signal : $\mathbf{x}(k)$ Desired Response: $d(k)$
Initialization	Set Filter-Tap: $\mathbf{w}(0) = 0$ Select Step-Size: μ Select Leakage Factor: υ ; Small Positve Number $<< 1$
Computation	For $k = 0, 1, 2, ..., n$ Error Vector: $e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k)$ Update Filter-Tap: $\mathbf{w}(k+1) = (1 - \mu\upsilon)\mathbf{w}(k) + \mu e(k)\mathbf{x}(k)$

2.4 The Variable Step-Size LMS (VSSLMS) algorithm

The variable step size LMS (VSSLMS) algorithm was proposed to establish a balance needed for faster convergence speed and lower MSE of the LMS algorithm with a fixed step-size μ [6]. By allowing each filter coefficient a separate time-varying step-size, the adaptation process is capable to accelerate the convergence speed by selecting a large step-size at the beginning. As the VSSLMS algorithm reaches the steady-state solution, the step-size μ decreases in order to reduce MSE. Table 2.3 summarize the VSSLMS algorithm..

Table 2.3: The VSSLMS algorithm

Variable Step-Size Least-Mean-Square Algorithm (VSSLMS)	
Data	Input Signal : $\mathbf{x}(k)$ Desired Response: $d(k)$
Initialization	Set Filter-Tap: $\mathbf{w}(0) = 0$ Select Variable Step-Size: $\mu(0) = \mu_{max}$
Computation	For $k = 0, 1, 2, ..., n$ Error Vector: $e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k)$ Update Filter-Tap: $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k)e(k)\mathbf{x}(k)$ Step-Size Update : $\mu(k+1) = \begin{cases} \mu_{max}, & \text{if } \alpha\mu(\mathrm{k}) + \gamma \mathrm{e}^2(\mathrm{k}) > \mu_{\max} \\ \mu_{min}, & \text{if } \alpha\mu(\mathrm{k}) + \gamma \mathrm{e}^2(\mathrm{k}) < \mu_{\min} \\ \alpha\mu(k) + \gamma e^2(k), & \text{otherwise} \end{cases}$ $\alpha\ , \gamma \in [0, 1]$

2.5 Sparse Adaptive Algorithms

Conventional adaptive algorithms, such as LMS and its variants [1, 3, 6], Kalman filters [7] and RLS [1], suffer from being sensitive to highly correlated inputs, low convergence and high-power consumption. Furthermore, the adaptive methods mentioned above are not capable taking advantage of a priori information available about the system structure such as sparsity. Using such a priori information can be very crucial in obtaining good performance by the adaptive filtering algorithm.

Consequently, research studies have been conducted to address these difficulties experienced by adaptive schemes in the context of recently emerging field of sparse signal processing [17, 18, 32, 33, 34].

In this work, new sparsity-aware adaptive filtering algorithms are proposed within the sparse signal processing framework for improved performance.

Chapter 3

PROPOSED SPARSE ADAPTIVE FILTERING ALGORITHMS

3.1 Introduction

As stated before, the LMS-type adaptive filtering algorithms suffer from low convergence rate and high power consumption when a long impulse response of an unknown system contains many small (near-zero) amplitudes and a few large ones. In other words, the system impulse response has a sparse structure. In many communication and signal processing systems, identifying the sparse impulse response is a challenging problem [10, 35]. Under such scenarios, the conventional LMS-type algorithms as well as the such as RLS and Kalman filtering techniques, are incapable of addressing system sparsity and hence fail to result in promising performance [17]. This inspires the development of efficient adaptive filtering algorithms that utilize the sparse nature of the system to be estimated.

Motivated by the LASSO [36] and the recent developments in the field of compressive sensing [15], the sparsity is addressed by combining l_0-norm penalty function into the original cost function of the LMS algorithm. Adding the l_0-norm to the cost function of the LMS-type algorithm causes the optimization problem to be non-convex. In order to avoid this drawback, the l_1-norm is used as an approximation to the l_0 norm [10]. This gives the adaptation process the ability of attracting zero (or nearly zero) filter coefficients, and is named zero-attracting LMS (ZA-LMS) algorithm[10]. A reweighed zero-attracting LMS (RZA-LMS) algorithm is additionally developed in [10] that, by using modified zeros-attractor term, employs a selective zero-forcing mechanism on the filter taps with small magnitude rather than uniformly forcing induce zero on all the filter taps. This results in improved performance when the sparsity of the system decreases. In this chapter, we design three new sparse LMS-type algorithms with zero attractors based on ZA-LMS and RZA-LMS algorithms that aim to achieve faster convergence rate while decreasing the mean-square deviation (MSD) compared to existing ones. The first algorithm is referred as weighted zero-attracting leaky LMS (WZA-LLMS)algorithm where a logarithmic penalty term is added into the leaky-LMS algorithm cost function in order to adjust the update equation. The adjustment causes the proposed algorithm to attract the zeros of sparse channel and improves the performance compared to LLMS and ZA-LMS algorithms. Furthermore, a zero attractor-variable step-size LMS (ZA-VSSLMS) algorithm is introduced with a lower computational complexity than the WZA-LLMS algorithm . The ZA-VSSLMS algorithm imposes an l_1-norm penalty to the original quadratic cost function of the VSSLMS algorithm which captures the system sparsity during adaptation process. For highly sparse channel, this process accelerates the convergence speed and improves the error performance. For this particular algorithm, the convergence analysis of the algorithm is derived when the white process presents at the input of the system and stability condition of the algorithm is presented. In addition, the steady-state MSD analysis of the algorithm is carried out in detail. A steady-state MSD expression for the ZA-VSSLMS algorithm is derived in terms of the system parameters for general white noise process. Most importantly, the crucial upper bound of zero-attractor controller (ρ) which yields minimum MSD is theoretically shown. Finally, to further improve the performance of the ZA-VSSLMS algorithm when the sparsity of the channel decreases, with a slight cost in the number of computations, the weighted zero-attracting-variable step-size LMS (WZA-VSSLMS) algorithm is introduced by adding the same log-sum penalty as in the WZ-LLMS algorithm into original cost function of the VSSLMS algorithm.

The rest of this chapter is organized as follows. Section 3.2 briefly reviews the ZA-LMS and RZA-LMS algorithms. In Section 3.3, the proposed WZA-LLMS algorithm is presented. In Section 3.4, the proposed ZA-VSSLMS will be provided together with detailed analysis in both convergence (Section 3.5) and the steady-state condition (Section 3.6) of the algorithm. Finally, in Section 3.7, the WZA-VSSLMS is derived which provides further improvement over the ZA-VSSLMS algorithm.

3.2 Review of the Zero Attracting Algorithms

3.2.1 The LMS Algorithm

Consider the input-output relation of a linear time-invariant (LTI) system described by

$$y(k) = \mathbf{h}^T \mathbf{x}(k) + v(k), \tag{3.1}$$

where $\mathbf{h}$ is the actual system response of length N, T is the transposition operator, $\mathbf{x}(k)$ is a white input signal, $y(k)$ is the output and $v(k)$ is an additive noise process which is independent from $x(k)$.

In the standard LMS algorithm, the cost function $J(k)$ is defined as

$$J(k) = \frac{1}{2} e^2(k), \tag{3.2}$$

where $e(k)$ is the error signal computed as,

$$e(k) = y(k) - \mathbf{w}^T(k)\mathbf{x}(k), \tag{3.3}$$

with $\mathbf{w}(k)$ is the coefficient weight vector of the adaptive algorithm of the length N.

By the steepest descent method, the filter coefficient vector is updated according to

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) - \frac{\mu}{2}\boldsymbol{\nabla} J(k) \\ &= \mathbf{w}(k) + \mu e(k)\mathbf{x}(k) \end{aligned} \tag{3.4}$$

where μ is the adaptation step-size which controls the convergence and steady-state behavior of the LMS algorithm.

3.2.2 ZA-LMS and RZA-LMS Algorithms

In order to estimate the sparse system, a new class of convex adaptive filtering algorithm is proposed [10] by adding the l_1-norm penalty to the cost function given in 3.2 as follows

$$J_1(k) = \frac{1}{2}e^2(k) + \lambda\|\mathbf{w}(k)\|_1 \tag{3.5}$$

where λ is a positive constant. By the gradient method, the tap-weights update equation takes the form,

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) - \frac{\mu}{2}\boldsymbol{\nabla} J_1(k) \\ &= \mathbf{w}(k) + \mu(k)e(k)\mathbf{x}(k) - \rho f(\mathbf{w}(k)) \end{aligned} \tag{3.6}$$

where $\rho = \lambda\mu$, and $f(\mathbf{w}(k))$ is the sign function $(f(\mathbf{w}(k)) = sgn(\mathbf{w}(k)))$.

Comparing (3.4) and (3.6), in (3.6) there is an extra term $(-\rho f(\mathbf{w}(k)))$. This term forces the tap coefficients to become zero. In other words, if the large number of the coefficients of **w** is zero, the zero-attractor will accelerate the convergence behavior. ρ controls the strength of the zero-attractor. Due to this, the algorithm is called zero-attracting LMS (ZA-LMS). The summary of the ZA-LMS algorithm is shown in Table 3.1.

Table 3.1: ZA-LMS Algorithm

Zero-Attracting Least- Mean Square Algorithm (ZA-LMS)	
Data	Input Signal : $\mathbf{x}(k)$ Desired Response: $d(k)$
Initialization	Set Filter-Tap: $\mathbf{w}(0) = 0$ Select Step-Size: μ Select Zero-Attractor Strength: $\rho = \lambda\mu$; $\rho \in [0, 1]$; λ is Small Positive Constant.
Computation	For $k = 0, 1, 2, ..., n$ Error Vector: $e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k)$ Update Filter-Tap: $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k)e(k)\mathbf{x}(k) - \rho f(\mathbf{w}(k))$

The shrinkage in the ZA-LMS algorithm does not differentiate the zero taps from the non-zero taps. Hence, its performance get worse in the case of less sparse systems. This is due to the fact that the zero-attractor term, $-\rho f(\mathbf{w}(k))$, in the ZA-LMS algorithm uniformly updates all filter taps. So weighting the zero-attractor term in (3.6), will enhance its performance if the system is less sparse [10]. This is achieved by replacing the l_1- norm penalty with a log-sum penalty, which resembles l_0-norm more than the l_1 norm, into the cost function in (3.2) as

$$J_2(k) = \frac{1}{2}e^2(k) + \lambda' \sum_{i=1}^{N} log\left(1 + \frac{|w_i|}{\zeta'}\right) \tag{3.7}$$

where λ' and ζ' are positive constants. Then, the same as before, by applying the gradient method we get

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k)e(k)\mathbf{x}(k) - \rho\frac{sgn\left[\mathbf{w}(k)\right]}{1+\zeta|\mathbf{w}(k)|} \tag{3.8}$$

where $\rho = \frac{\mu\lambda'}{\zeta'}$, $\zeta = \frac{1}{\zeta'}$ and $|\mathbf{w}(k)| = \sqrt{\sum_{i=1}^{N}(w_i^2)}$.

The weighted zero-attracting effect appears only on the taps that have magnitude comparable to $\frac{1}{\zeta}$ and there is a little shrinkage exerted on the taps whose magnitude is much greater than $\frac{1}{\zeta}$. As a result of that, the bias of the weighted zero-attracting LMS (RZA-LMS) algorithm can be reduced. The summary of the ZA-LMS algorithm is shown in Table 3.2.

3.3 Proposed Weighted Zero-Attracting Leaky-LMS Algorithm

By adding the cost function associated with LLMS algorithm [5]) together with a log-sum penalty function as in the RZA-LMS algorithm, we propose the weighted zero-attracting leaky-LMS algorithm which forms the new cost function as

$$J_3(k) = \frac{1}{2}e^2(k) + \upsilon\mathbf{w}^T(k)\mathbf{w}(k) + \gamma' \sum_{i=1}^{L} log\left(1 + \frac{|w_i|}{\zeta'}\right) \tag{3.9}$$

with υ is a small positive constant known as the leakage factor.

Applying the gradient decent method, the WZA-LLMS algorithm is updated recursively

Table 3.2: RZA-LMS Algorithm

Reweighted Zero-Attracting Least- Mean Square Algorithm (RZA-LMS)	
Data	Input Signal : $\mathbf{x}(k)$ Desired Response: $d(k)$
Initialization	Set Filter-Tap: $\mathbf{w}(0) = 0$ Select Step-Size: μ Select Zero-Attractor Strength: $\rho = \frac{\mu\lambda'}{\zeta'}$ $\lambda\mu$; $\rho \in [0, 1]$; λ' and ζ' are Small Positive constants
Computation	For $k = 0, 1, 2, ..., n$ Error: $e(k) = d(k) - \mathbf{w}^T(k)\mathbf{x}(k)$ Update Filter-Tap: $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k)e(k)\mathbf{x}(k) - \rho\frac{sgn[\mathbf{w}(k)]}{1+\zeta\lvert\mathbf{w}(k)\rvert}$

as,

$$\mathbf{w}(k+1) = (1 - \mu\upsilon)\mathbf{w}(k) + \mu e(k)\mathbf{x}(k) - \rho\frac{sgn\left[\mathbf{w}(k)\right]}{1 + \zeta\lvert\mathbf{w}(k)\rvert} \tag{3.10}$$

with a segment zero attractor vector $-\rho\frac{sgn[\mathbf{w}(n)]}{1+\zeta\lvert\mathbf{w}(n)\rvert}$. Simulation results in chapter 4 show that the WZA-LLMS algorithm outperforms the LLMS and ZA-LMS algorithms in terms of convergence speed and MSD.

3.4 Proposed Zero-attracting Variable Step-size LMS Algorithm

In this section, we propose a new approach called the zero-attracting variable step-size LMS (ZA-VSSLMS) algorithm that has the order of computational complexity as that of the LMS algorithm ($O(N)$) but with better performance compared with those of the well-known LMS, VSSLMS and ZA-LMS algorithms. Unlike the ZA-LMS and RZA-LMS algorithms, the ZA-VSSLMS algorithm employs a varying zero-attractor controller, $\rho(k)$, that results in improved MSD. It combines the ℓ_1-norm penalty function with the original cost function of the VSSLMS to utilize the sparsity of the system.

Using the same cost function as in (3.5) and the error signal given by (3.3), by the gradient method, the ZA-VSSLMS update equation becomes

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k)e(k)\mathbf{x}(k) - \rho(k)sgn(\mathbf{w}(k)). \tag{3.11}$$

Here, $\rho(k) = \lambda\mu(k)$, $sgn(.)$ is the pointwise sign function and $\mu(k)$ is a variable step-size [6] which is given by

$$\mu(k) = \begin{cases} \mu_{max} & \text{if } \mu'(k+1) > \mu_{max} \\ \mu_{min} & \text{if } \mu'(k+1) < \mu_{min} \\ \mu'(k+1) & \text{otherwise} \end{cases} \tag{3.12}$$

where μ_{max} and μ_{min} are the upper and lower bound of $\mu(k)$, respectively.

$\mu'(k+1)$ can be estimated in the following form

$$\mu'(k+1) = \alpha\mu'(k) + \gamma e^2(k), \tag{3.13}$$

with $0 < \alpha < 1$ and $\gamma > 0$.

3.4.1 Discussion

The recursive update formula in the VSSLMS algorithm, can be viewed as

$$\begin{pmatrix} \text{present filter} \\ \text{weights} \end{pmatrix} = \begin{pmatrix} \text{past filter} \\ \text{weights} \end{pmatrix} + \begin{pmatrix} \text{gradient} \\ \text{correction} \end{pmatrix} \tag{3.14}$$

where the filter taps are updated in the direction of the negative gradient. Also, the update equations of the ZA-VSSLMS algorithm can be written as

$$\begin{aligned} \begin{pmatrix} \text{present filter} \\ \text{weights} \end{pmatrix} = \begin{pmatrix} \text{past filter} \\ \text{weights} \end{pmatrix} &+ \begin{pmatrix} \text{gradient} \\ \text{correction} \end{pmatrix} \\ &+ \begin{pmatrix} \text{zero} \\ \text{attraction} \end{pmatrix} \end{aligned} \tag{3.15}$$

where the zero attraction term in (3.15) (equivalent to $-\rho(k)f(\mathbf{w}(k))$ in (3.11), imposes an attraction to zero on small filter-taps. Explicitly, for positive value of filter-taps the zero attractor term will subtract from the update equation and if the filter-taps is negative, it will add up to the update equation respectively.

In (3.11), $\rho(k)$ results in a compromise between the adaptation quality and its speed. A large value of $\rho(k)$ leads to a faster convergence as the ability of zero-forcing increases. At the same time, with the value of $\rho(k)$ increased, the steady-state misalignment also increases. At steady state, due to the sparsity nature, majority of filter weights are close to zero. Therefore, those near-zero coefficients $w_i(k)$ will move randomly in the small neighborhood of zero, caused by both attraction and the gradient noise terms. Hence, a large $\rho(k)$ results in a large steady-state misalignment.

Furthermore, in (3.11), $-\rho(k)sgn(\mathbf{w}(k))$ is bounded between $-\rho(k)$ and $\rho(k)$ or $-\lambda\mu_{max}$

and $-\lambda\mu_{min}$. Hence the convergence criterion of the ZA-VSSLMS algorithm is expected to be similar as that of the VSSLMS algorithm. We shall proof this mathematically in the next section.

3.5 Convergence Analysis of the ZA-VSSLMS Algorithm

In this section, we analyze the convergence of the ZA-VSSLMS algorithm when the input to the system is white. We start by defining the filter misalignment vector, mean and the misalignment vector covariance matrix respectively. Once all these parameters are well-defined, we use them to estimate the updated value of covariance matrix and hence deriving the stability condition based on the trace of this matrix. Here, we present the stability condition in terms of the input variance σ_x^2 and filter length N. The filter's misalignment vector can be defined as

$$\boldsymbol{\delta}(k) = \mathbf{h} - \mathbf{w}(k). \tag{3.16}$$

The mean and covariance matrices of $\boldsymbol{\delta}(k)$, respectively, as

$$\boldsymbol{\epsilon}(k) = E\{\boldsymbol{\delta}(k)\}, \tag{3.17}$$

$$\mathbf{S}(k) = E\{\mathbf{c}(k)\mathbf{c}^T(k)\}, \tag{3.18}$$

where a zero mean vector $\mathbf{c}(k)$ is computed as

$$\mathbf{c}(k) = \boldsymbol{\delta}(k) - E\{\boldsymbol{\delta}(k)\}. \tag{3.19}$$

We also define the instantaneous mean-square-deviation (MSD) in the following form

$$J(k) = E\{||\boldsymbol{\delta}(k)||_2^2\} = \sum_{i=0}^{N-1} \Lambda_i(k), \tag{3.20}$$

where $\Lambda_i(k)$ represents the *i-th* tap MSD given by

$$\Lambda_i(k) = E\{||\boldsymbol{\delta}_i(k)||_2^2\} = S_{ii}(k) + \epsilon_i^2(k), \qquad i = 0, ..., N-1. \tag{3.21}$$

In equation (3.21), $S_{ii}(k)$ represents the diagonal elements of the auto-covariance matrix $\mathbf{S}(k)$.

By combining (3.1), (3.3), (3.5), (3.16) and using the independence assumption we get

$$\boldsymbol{\delta}(k+1) = [\mathbf{I} - \mu(k)\mathbf{x}(k)\mathbf{x}^T(k)]\boldsymbol{\delta}(k) + \mu(k)\mathbf{x}(k)v(k) - \rho(k)sgn[\mathbf{w}(k)], \tag{3.22}$$

Taking the expectation of (3.22), we get

$$\begin{aligned} E\{\boldsymbol{\delta}(k+1)\} &= E\{[\mathbf{I} - \mu(k)\mathbf{x}(k)\mathbf{x}^T(k)]\boldsymbol{\delta}(k) + \mu(k)\mathbf{x}(k)v(k) \\ &\quad - \rho(k)sgn[\mathbf{w}(k)]\}. \end{aligned} \tag{3.23}$$

Due to the independence between x and v, the expectation of the second term becomes zero. Thus the expectation of (3.23) simplifies to

$$\boldsymbol{\epsilon}(k+1) = (1 - E\{\mu(k)\}\sigma_x^2)\boldsymbol{\epsilon}(k) - E\{\rho(k)sgn[\mathbf{w}(k)]\}. \quad (3.24)$$

Subtracting $E\{\boldsymbol{\delta}(k+1)\}$ from both sides of equation (3.22) and substituting in (3.24)

$$\begin{aligned}\mathbf{c}(k+1) &= (\mathbf{I} - \mu(k)\mathbf{x}(k)\mathbf{x}^T(k))\boldsymbol{\delta}(k) + \mu(k)\mathbf{x}(k)v(k) \\ &\quad - \rho(k)sgn[\mathbf{w}(k)] - \epsilon(k+1) \\ &= \mathbf{A}(k)\boldsymbol{\delta}(k) + \mu(k)\mathbf{x}(k)v(k) \\ &\quad - (1 - E\{\mu(k)\}\sigma_x^2)\boldsymbol{\epsilon}(k) + \mathbf{p}(k). \quad (3.25)\end{aligned}$$

Here

$$\mathbf{A}(k) = \mathbf{I} - \mu(k)\mathbf{x}(k)\mathbf{x}^T(k), \quad (3.26)$$

$$\mathbf{p}(k) = E\{\rho(k)\}E\{sgn[\mathbf{w}(k)]\} - \rho(k)sgn[\mathbf{w}(k)]. \quad (3.27)$$

Next, the term $\mu(k)\mathbf{x}(k)\mathbf{x}^T(k)\boldsymbol{\epsilon}(k)$ is added to both side of (3.25) and by rearranging the terms

we get

$$\mathbf{c}(k+1) = \mathbf{A}(k)\mathbf{c}(k) + \mu(k)\mathbf{x}(k)v(k) + \mathbf{B}(k)\boldsymbol{\epsilon}(k) + \mathbf{p}(k). \quad (3.28)$$

In the above equation,

$$\mathbf{B}(k) = E\{\mu(k)\mathbf{x}(k)\mathbf{x}^T(k)\} - \mu(k)\mathbf{x}(k)\mathbf{x}^T(k). \quad (3.29)$$

Next, we estimate $\mathbf{S}(k+1)$ based upon the independence among $x(k)$, $v(k)$, and $\boldsymbol{\delta}(k)$ as follows:

$$\begin{aligned}
\mathbf{S}(k+1) &= E\{\mathbf{c}(k+1)\mathbf{c}^T(k+1)\} \\
&= E\{\mathbf{A}(k)\mathbf{c}(k)\mathbf{c}^T(k)\mathbf{A}^T(k)\} + 2E\{\mathbf{A}(k)\mathbf{c}(k)\mathbf{p}^T(k)\} \\
&\quad + E\{\mathbf{B}(k)\boldsymbol{\epsilon}(k)\boldsymbol{\epsilon}^T(k)\mathbf{B}(k)\} \\
&= (1 - 2E\{\mu(k)\}\sigma_x^2 + 2E\{\mu^2(k)\}\sigma_x^4)\mathbf{S}(k) + E\{\mu^2(k)\}\sigma_x^4 tr[\mathbf{S}(k)]\mathbf{I} \\
&\quad + 2E\{\rho(k)\}(1 - 2E\{\mu(k)\}\sigma_x^2)E\{\mathbf{w}(k)\mathbf{p}^T(k)\} + E\{\mathbf{p}(k)\mathbf{p}^T(k)\} \\
&\quad + E\{\mu^2(k)\}\sigma_x^4(\boldsymbol{\epsilon}(k)\boldsymbol{\epsilon}^T(k) + tr[\epsilon(k)\epsilon^T(k)]\mathbf{I}) + E\{\mu^2(k)\}\sigma_x^2\sigma_v^2\mathbf{I},
\end{aligned} \tag{3.30}$$

The estimate in (3.30) is obtained by calculating input's fourth moment [37] as well as the symmetricity of the covariance matrix $\mathbf{S}(k)$.

Applying the trace operator to the both sides of the (3.30) yields,

$$\begin{aligned}
tr[\mathbf{S}(k+1)] &= (1 - 2E\{\mu(k)\}\sigma_x^2 + (N+2)E\{\mu^2(k)\}\sigma_x^4)tr[\mathbf{S}(k)] \\
&\quad + tr(E\{\mathbf{p}(k)\mathbf{p}^T(k)\}) + \ NE\{\mu^2(k)\}\sigma_x^2\sigma_v^2 \\
&\quad + 2E\{\rho(k)\}(1 - E\{\mu(k)\sigma_x^2\})E\{\mathbf{w}(k)\mathbf{p}^T(k)\} \\
&\quad + (N+1)E\{\mu^2(k)\}\sigma_x^4\epsilon^T(k)\epsilon(k).
\end{aligned} \tag{3.31}$$

In (3.31), $\mathbf{p}(k)$, $E\{\mathbf{w}(k)\}$, $E\{\mathbf{w}(k)\mathbf{p}^T(k)\}$ and $\boldsymbol{\epsilon}(k)$ are all bounded.

Thus, the adaptive filter is stable if the following holds

$$|1 - 2E\{\mu(k)\}\sigma_x^2 + (N+2)E\{\mu^2(k)\}\sigma_x^4| < 1. \tag{3.32}$$

This implies that as $k \longrightarrow \infty$, $E\{\mu^2(k)\} = E\{\mu(k)\}^2 = \mu^2(k)$. Hence the above equation simplifies to

$$0 < \mu(\infty) < \frac{2}{(N+2)\sigma_x^2}. \tag{3.33}$$

This result shows that if μ satisfies (3.33) the convergence of the ZA-VSSLMS is guaranteed. It is worthy to note that the stability condition in (3.33) is similar to that the standard LMS algorithm. The critical bound on μ for which the LMS algorithm converges in mean-square sense is given as [23]:

$$0 < \mu < \frac{1}{2\lambda_{max} + \sum_{j=0}^{N} \lambda_j}. \tag{3.34}$$

where λ_{max} corresponds to the largest eigenvalue of covariance matrix **S**. For the white input signal the stability condition in (3.34) reduces to (3.33) as above. Furthermore, for $\mu << 1$, the stability condition in (3.33) is identical to the NLMS algorithm at steady state [24]. Once the convergence criterion is known, we shall continue to examine the ZA-VSSLMS algorithm by studying the behavior of adaptive filter coefficients at steady state which is highly useful in actual design of the filter.

3.6 Mean-Square Deviation Analysis of ZA-VSSLMS Algorithm

In this section, we present the steady-state MSD analysis of the ZA-VSSLMS algorithm. A steady-state MSD expression for the ZA-VSSLMS algorithm is proved in terms of the system parameters. More importantly, an upper-bound of the zero-attractor controller (ρ)

which provides minimum MSD is derived. While the stability condition on the step size (μ) of ZA-VSSLMS is provided in presence of an additive Gaussian white noise in Section 3.4, the MSD analysis assumes any white noise distribution. It is later shown in Chapter 4, the theoretical and simulation results are in good agreement for a wide range of parameters, different channel, input signal and noise types.

Now we derive the MSD expression for the proposed algorithm. Our analysis will be based on the assumptions that:

1. The input signal is independent and identically distributed with zero mean and variance σ_x^2 [1].

2. The input-tap vector $\mathbf{x}(k)$ is independent from $\mathbf{w}(l)$ for $l \leq k$ [1].

3. The observation noise has zero mean and variance σ_v^2 and is independent from $\mathbf{x}(k)$.

The variable step-size $\mu(k)$ in (3.13) can be estimated recursively as

$$\mu(k+1) = \gamma \sum_{i=0}^{k-1} \alpha^i e^2(k-i) \tag{3.35}$$

where μ_1 is assumed to be initially at rest ($\mu_1 = 0$). Combining (3.1) and (3.3) yields,

$$e(k) = \mathbf{x}^T(k)\boldsymbol{\delta}(k) + v(k). \tag{3.36}$$

Inserting (3.16) and (3.35) into (3.11) gives the ZA-VSSLMS update equation as,

$$\begin{aligned} \boldsymbol{\delta}(k+1) &= \boldsymbol{\delta}(k) - \gamma\mathbf{x}(k)\mathbf{x}^T(k)\boldsymbol{\delta}(k)g(i,k) - \gamma\mathbf{x}(k)v(k)g(i,k) \\ &+ \lambda\gamma g(i,k)sgn[\mathbf{w}(k)] \end{aligned} \tag{3.37}$$

where $g(i,k) = \sum_{i=0}^{k-2} \alpha^i e^2(k-i-1)$.

Multiplying both sides of (3.37) by $\boldsymbol{\delta}^T(k+1)$ we obtain,

$$\begin{aligned}
\boldsymbol{\delta}(k+1)\boldsymbol{\delta}^T(k+1) = \boldsymbol{\delta}(k)\boldsymbol{\delta}^T(k) - \bigg(&\gamma\boldsymbol{\delta}(k)\boldsymbol{\delta}^T(k)\mathbf{x}(k)\mathbf{x}^T(k) \\
&+ \gamma\boldsymbol{\delta}(k)\mathbf{x}^T(k)v(k) - \lambda\gamma\boldsymbol{\delta}(k)sgn[\mathbf{w}^T(k)]\bigg)g(i,k) \\
&- \bigg(\gamma\mathbf{x}(k)\mathbf{x}^T(k)\boldsymbol{\delta}(k)\boldsymbol{\delta}^T(k) + \gamma\mathbf{x}(k)\boldsymbol{\delta}^T(k)v(k) \\
&- \lambda\gamma sgn[\mathbf{w}(k)]\boldsymbol{\delta}^T(k)\bigg)g(i,k) + \bigg(\gamma^2\mathbf{x}(k)\mathbf{x}^T(k)\boldsymbol{\delta}(k)\boldsymbol{\delta}^T(k) \\
&\mathbf{x}(k)\mathbf{x}^T(k) + \gamma^2\mathbf{x}(k)\mathbf{x}^T(k)\boldsymbol{\delta}(k)\mathbf{x}^T(k)v(k) - \lambda\gamma^2 \\
&\mathbf{x}(k)\mathbf{x}^T(k)\boldsymbol{\delta}(k)sgn[\mathbf{w}^T(k)]\bigg)g(i,k)g(i,k) \\
&+ \bigg(\gamma^2\mathbf{x}(k)\boldsymbol{\delta}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)v(k) + \gamma^2\mathbf{x}(k)\mathbf{x}^T(k)v^2(k) \\
&- \lambda\gamma^2\mathbf{x}(k)sgn[\mathbf{w}^T(k)]v(k)\bigg)g(i,k)g(i,k) \\
&- \bigg(\lambda\gamma^2 sgn[\mathbf{w}(k)]\boldsymbol{\delta}^T(k)\mathbf{x}(k)\mathbf{x}^T(k) + \lambda\gamma^2 sgn[\mathbf{w}(k)] \\
&\mathbf{x}^T(k)v(k) - \lambda^2\gamma^2 sgn[\mathbf{w}(k)]sgn[\mathbf{w}^T(k)]\bigg)g(i,k)g(i,k) \quad (3.38)
\end{aligned}$$

In order to obtain the expression for the MSD, we need to compute the expectation of (3.38) as $k \to \infty$. However, computing the expectation of (3.38) in the original matrix form is very difficult. To tackle this difficulty, we take advantage of both matrix stacking operator [38] and Kronecker product property at the same time. The matrix stacking operator maps the columns of an arbitrary matrix into a single column vector. This process is called vectorization and denoted by $vec(.)$. Then by Kronecker product property [39], for given arbitrary matrices A, B and C of compatible sizes, $vec(ABC) = (C^T \bigotimes A)vec(B)$ [40].

Using the above, the expectation of (3.38) in the vector form as $k \to \infty$ is:

$$\begin{aligned} vec\left(\mathbf{\Delta}_\infty\right) &= vec(\mathbf{\Delta}_\infty) + \gamma^2 \xi E\left[\mathbf{x}_\infty \mathbf{x}_\infty^T \otimes \mathbf{x}_\infty \mathbf{x}_\infty^T\right] vec\left(\mathbf{\Delta}_\infty\right) \\ &\quad - \frac{\gamma E\left(e_\infty^2\right)}{(1-\alpha)} \left(E\left[\mathbf{I} \otimes \mathbf{x}_\infty \mathbf{x}_\infty^T\right] + E\left[\mathbf{x}_\infty \mathbf{x}_\infty^T \otimes \mathbf{I}\right]\right) vec\left(\mathbf{\Delta}_\infty\right) \\ &\quad + \left[\gamma^2 \sigma_v^2 - \frac{2\gamma^2\lambda^2}{\sigma_x^2}\right] \xi vec(\mathbf{R}) + \frac{\lambda^2\gamma^2}{N} \xi \sum_{j=0}^{N-1} sgn[|w_j|] vec(\mathbf{I}) \end{aligned} \quad (3.39)$$

where $vec\left(\mathbf{\Delta}_\infty\right) = E\left[vec\left(\boldsymbol{\delta}_\infty \boldsymbol{\delta}_\infty^T\right)\right]$ is a vector of size $N^2 \times 1$, $sgn[|w_j|] = 1$ if $w_j \neq 1$, $\xi = \left[\frac{2\alpha\left(E\left(e_\infty^2\right)\right)^2}{(1-\alpha)^2(1+\alpha)} + \frac{E\left(e_\infty^4\right)}{(1-\alpha^2)}\right]$ and $\mathbf{R} = E[\mathbf{x}_\infty \mathbf{x}_\infty^T]$.

The detailed derivation of equation (3.39) can be found in the Appendix.

Solving equation (3.39) at steady state we can get

$$\begin{aligned} vec(\mathbf{\Delta}_\infty) = &\left\{ \left(E\left[\mathbf{I} \otimes \mathbf{x}_\infty \mathbf{x}_\infty^T\right] + E\left[\mathbf{x}_\infty \mathbf{x}_\infty^T \otimes \mathbf{I}\right]\right) E\left(e_\infty^2\right) \right. \\ &\quad - \frac{\gamma}{(1-\alpha)} \left[\frac{2\alpha\left(E\left(e_\infty^2\right)\right)^2 + (1-\alpha)E\left(e_\infty^4\right)}{1+\alpha}\right] \\ &\left. \times E\left[\mathbf{x}_\infty \mathbf{x}_\infty^T \otimes \mathbf{x}_\infty \mathbf{x}_\infty^T)\right]\right\}^{-1} \left\{ (1-\alpha)\xi \left(\left[\gamma\sigma_v^2 - \frac{2\gamma\lambda^2}{\sigma_x^2}\right] vec(\mathbf{R}) \right.\right. \\ &\quad \left.\left. + \frac{\lambda^2\gamma}{N} \sum_{j=0}^{N-1} sgn[|w_j|] vec(\mathbf{I}) \right)\right\} \end{aligned} \quad (3.40)$$

For $\gamma \ll (1-\alpha)$, the term,

$$\frac{\gamma}{(1-\alpha)} \left[\frac{2\alpha\left(E\left(e_\infty^2\right)\right)^2 + (1-\alpha)E\left(e_\infty^4\right)}{1+\alpha}\right] E\left[\mathbf{x}_\infty \mathbf{x}_\infty^T \otimes \mathbf{x}_\infty \mathbf{x}_\infty^T)\right],$$

in (3.40) can be neglected. Additionally, as $k \to \infty$ from (3.36) $e(\infty) \approx v(\infty)$. Hence, we can assume that $E\left(e_\infty^2\right) = \sigma_v^2$ and $E\left(e_\infty^4\right) = E\left(v_k^4\right)$. Therefore, equation (3.40) reduces to

$$vec(\boldsymbol{\Delta}_\infty) = \frac{1}{1-\alpha}\left[\frac{2\alpha(\sigma_v^2)^2 + (1-\alpha)E(v_k^4)}{(1+\alpha)\sigma_v^2}\right]\left(E\left[\mathbf{I}\otimes\mathbf{x}_\infty\mathbf{x}_\infty^T\right] + E\left[\mathbf{x}_\infty\mathbf{x}_\infty^T\otimes\mathbf{I}\right]\right)^{-1} \times \left[\left(\gamma\sigma_v^2 - \frac{2\gamma\lambda^2}{\sigma_x^2}\right)vec(\mathbf{R}) + \frac{\lambda^2\gamma}{N}\sum_{j=0}^{N-1} sgn[|w_j|]vec(\mathbf{I})\right] \quad (3.41)$$

From (3.41), we evaluate the mean square deviation (MSD) of the proposed algorithm by computing the trace of the covariance matrix of the misalignment vector (MSD $= Tr(E\left[(\boldsymbol{\delta}_\infty\boldsymbol{\delta}_\infty^T)\right])$. To do so, we first apply $vec^{-1}(.)$ operation to both sides of (3.41) and then using the property $Tr(AB) = (vec(A^T))^T vec(B)$ [41].

For white input($R = \sigma_x^2 I$), the MSD is

$$MSD = N\left[\frac{2\alpha(\sigma_v^2)^2 + (1-\alpha)E(v_k^4)}{2(1-\alpha^2)\sigma_v^2}\right]\left[\gamma\sigma_v^2 - \frac{2\gamma\lambda^2}{\sigma_x^2} + \frac{\gamma\lambda^2}{N\sigma_x^2}\sum_{j=0}^{N-1} sgn[|w_j|]\right] \quad (3.42)$$

From (3.42) we note the following:

1. For relatively small λ, the sparse VSSLMS algorithm will converge to the steady-state if the conventional VSSLMS algorithm converges.

2. The term $\left(-\frac{2\gamma\lambda^2}{\sigma_x^2} + \frac{\gamma\lambda^2}{N\sigma_x^2}\sum_{j=0}^{N-1} sgn[|w_j|]\right) \leq 0$ and, hence, the steady-state MSD of the sparse VSSLMS algorithm will always be less than or equal to that of the conventional VSSLMS algorithm.

3. The term $\left(\gamma\sigma_v^2 - \frac{2\gamma\lambda^2}{\sigma_x^2} + \frac{\gamma\lambda^2}{N\sigma_x^2}\sum_{j=0}^{N-1} sgn[|w_j|]\right) \geq 0$ and hence an upper-bound of $\rho(k)$ can be found to be:

$$0 < \rho(k) \leq \mu_{max} \sqrt{\frac{N\sigma_v^2\sigma_x^2}{2N - \sum\limits_{j=0}^{N-1} sgn[|w_j|]}}. \tag{3.43}$$

This upper bound always guarantees better MSD than that of the VSSLMS algorithm. It is also worthy to note that in a situation where the system is completely non-sparse, the MSD expression in (3.42) reduces to

$$MSD = N\left[\frac{2\alpha(\sigma_v^2)^2 + (1-\alpha)E(v_k^4)}{2(1-\alpha^2)\sigma_v^2}\right]\left[\gamma\sigma_v^2 - \frac{2\gamma\lambda^2}{\sigma_x^2}\right] \tag{3.44}$$

The MSD in (3.44) is still lesser than that provided by equation (21) in [6].

3.7 Proposed Weighted ZA-VSSLMS Algorithm

To further improve the ZA-VSSLMS filter when the sparsity of the channel decreases, with a slight cost in the number of computations, the WZA-VSSLMS is introduced by adding the same log-sum penalty as in WZ-LLMS algorithm into original cost function of VSSLMS algorithm.

Using the same cost function given in equation(3.8), the update equation for the WZA-VSSLMS algorithm becomes

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu(k)e(k)\mathbf{x}(k) - \rho(k)\frac{sgn\left[\mathbf{w}(k)\right]}{1+\zeta|\mathbf{w}(k)|} \tag{3.45}$$

where $\rho(k) = \lambda\mu(k)$ as before. All the properties of the WZA-VSSLMS algorithm is similar to the RZA-LMS algorithm except the adaptation uses the VSSLMS algorithm which results in improved performance.

Chapter 4

SIMULATION RESULTS

4.1 Introduction

In this chapter, we aim to evaluate the performance of the proposed algorithms derived in Chapter 3, through various numerical simulations.

In the following sections, *MATLAB* Software is employed for the simulation of the standard LMS, LLMS, VSSLMS, ZA-VSSLMS, WZA-LLMS, WZA-VSSLMS and set-membership normalized LMS (SM-NLMS) algorithms.

Simulations are performed in order to assess the effectiveness of the proposed algorithms with a wide range of parameters in additive white Gaussian noise (AWGN), general white noise with different probability distributions and correlated input signal for the following settings:

(a) system identification

(b) echo cancelation

(c) image deconvolution

Before we highlight the details of each conducted experiments, we shall briefly discuss on the general nature of the above mentioned settings where the proposed algorithms are used respectively.

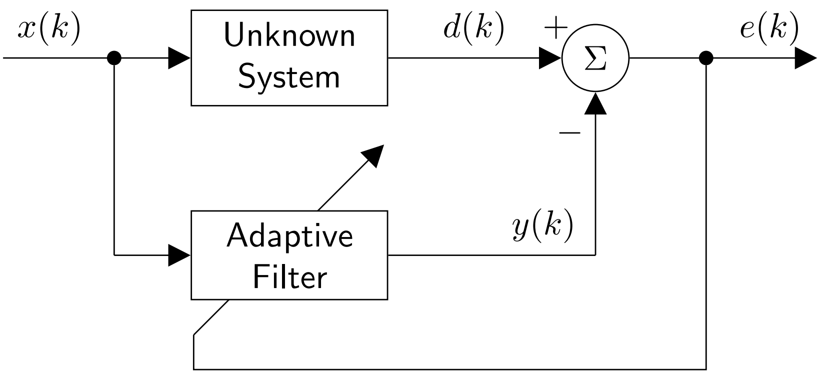


Figure 4.1: Block diagram of the system identification.

4.2 System Identification

System identification attempts to estimate and determine the input-output relationship of an unknown dynamic model (system) based only on data available at input-out of the unknown model. The block diagram of the system identification problem is shown in Fig.4.1.

The identification problem requires a set of model structure, a validation criterion and an aim [42]. This technique have been applied in many problems where the analyze, predict, interaction and control strategy designs for an unknown model is highly crucial [43].

In the context of classical adaptive filtering, however, the conventional algorithms are not suited for identifying an unknown sparse system. Therefore, in the following section, we show that by applying the proposed sparse algorithms, the identification problem can be further improved in terms of performance criterion (e.g. convergence speed, MSD).

4.3 Experimental Results

In this section, we compare the theoretical and the experimental results of the proposed algorithms in the system identification problem [1, 6] shown in Fig. 4.1. In all the experiments, if not explicitly mentioned otherwise, the input signal is designed to be white with zero-mean and the observed noise ($v(k)$) is assumed to be a white random sequence [44] with zero-mean and variance (σ_v^2) adjusted to provide the desired signal-to-noise ratio (SNR) in each experiment. Unless stated otherwise, all simulation results are averaged over 100 independent runs. In addition, throughout this chapter, the zero-attracting controller parameter (ρ) is strictly set to be less than the theoretical upper-bound of ρ_∞ given in (3.43). The performance of all experiments are evaluated using the MSD criterion defined as follows:

$$MSD(k)_{dB} = 10\log_{10} E||\mathbf{h} - \mathbf{w}(k)||^2. \tag{4.1}$$

4.3.1 Unknown Sparse Systems Identification with Different Lengths and Sparsity Levels

In order to investigate the tracking ability of the proposed algorithms (ZA-VSSLMS, WZA-VSSLMS and WZ-LLMS), here, we carry out number of experiments similar to the setup discussed in [10] but with different filter length and sparsity level.

In the first experiment, the proposed WZA-VSSLMS and ZA-VSSLMS algorithms are compared with that standard VSSLMS algorithm. The simulation involves identifying an unknown system with a filter length $N = 10$, that has only one filter-tap equal to one in the first 500 iterations while the others remain zero(to obtain a sparsity degree of $\frac{1}{10}$). After 500 iterations, all the 5 random taps are set to 1 and the rest kept to be zero, i.e., a sparsity of $\frac{5}{10}$. Finally, after 1000 successive iterations all the taps are set to values -1 and 1 randomly, producing a completely

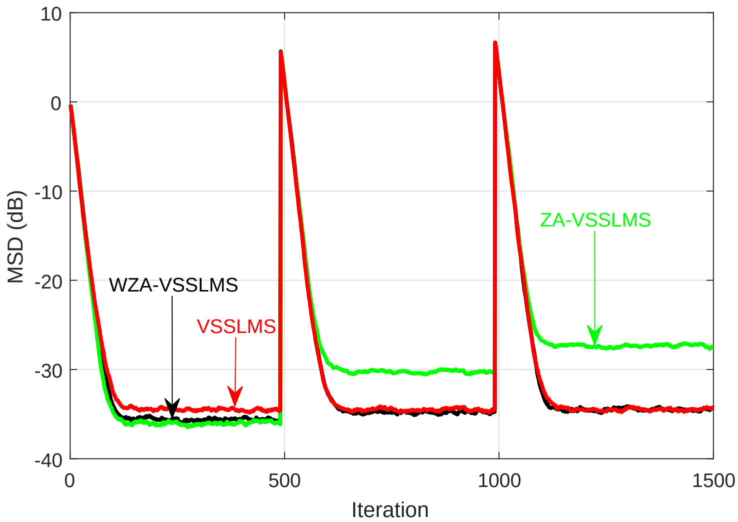


Figure 4.2: Comparison of MSDs between the WZA-VSSLMS, ZA-VSSLMS algorithms in AWGN. The filter length is $N = 10$.

non-sparse system. In order to have a 30dB SNR, the simulation assume both the input and noise signals are the white Gaussian random sequences with variances 1 and 10^{-3}, respectively. The remaining parameters For the VSSLMS are as follows: $\mu_{min} = 0.05$, $\mu_{max} = 0.07$, $\gamma = 0.048$ and $\alpha = 0.97$. For the ZA-VSSLMS: $\mu_{min} = 0.05$, $\mu_{max} = 0.07$, $\gamma = 0.048$, $\alpha = 0.97$ and $\rho = 1 \times 10^{-3}$. For the WZA-VSSLMS: $\mu_{min} = 0.05$, $\mu_{max} = 0.07$, $\gamma = 0.048$, $\alpha = 0.97$ and $\rho = 1 \times 10^{-3}$. Fig. 4.2 shows the average MSD curve for all algorithms in dB.

As it can be seen from the MSD results, before the 500^{th} iteration, with a significantly sparse system, the ZA-VSSLMS and the WZA-VSSLMS algorithms the converge speed is similar to that of the VSSLMS algorithm but with lower steady-state MSD (1dB better). After the 500^{th} iteration, as the number of non-zero taps increases, we see that the performance of the ZA-VSSLMS algorithm deteriorates since the shrinkage in the ZA-VSSLMS algorithm does not diffrentiate the zero taps from non-zero ones. However, the WZA-VSSLMS algorithm converges at the same rate to same MSD as that of the VSSLMS algorithm even if the system is non-sparse.

In the second experiment , the effectiveness of the proposed WZA-LLMS algorithm is tested with respect to the standard algorithms such as leaky-LMS and ZA-LMS. With the similar procedures as in the first experiment, the filter length is set to be $N = 16$ in this experiment. The SNR value is 30 dB as before and the remaining parameters for the leaky-LMS are set to be: $\mu = 0.035$, $\gamma = 0.001$. For the WZA-LLMS: $\mu = 0.035$, $\gamma = 0.001$, $\zeta = 10$ and $\rho = 5 \times 10^{-4}$. For ZA-LMS: $\mu = 0.035$ and $\rho = 5 \times 10^{-4}$.

Fig.4.3 demonstrates the average mean-square-devation curve of both algorithms all in dB. It can be observed that in the case of system with high degree of sparsity (before the 500^{th} iteration), the proposed WZA-LLMS algorithm has similar convergence speed as those in the other algorithms except it yields 1 dB lower MSD than the leaky-LMS .In the next 500^{th} iterations, by increasing the sparsity level to $\%50$ (with 8 non-zero taps), we observe that the WZA-LLMS algorithms converges with 30 iterations faster relative to the leaky-LMS at the same MSD rate. In addition, with reference to the ZA-LMS algorithm, it achieves 9.5dB better MSD result at moderately faster convergence speed. In the next 500 iterations where all the filter-taps are set to one(fully non-sparse system scenario), the performance of the WZA-LLMS is almost identical to the leaky-LMS algorithm but much better than the ZA-LMS algorithm.

We carry on with the performance of the WZA-LLMS algorithm for correlated (non-white) input signal in the third experiment. The input signal $x(k)$ is generated by using correlated first-order Gauassian-markov model ($x(k) = 0.8x(k-1) + v_o(k)$) [44] where $v_o(k)$ is a white Gaussian process. In addition, the abrupt changing time mechanism of the simulation where the sparsity

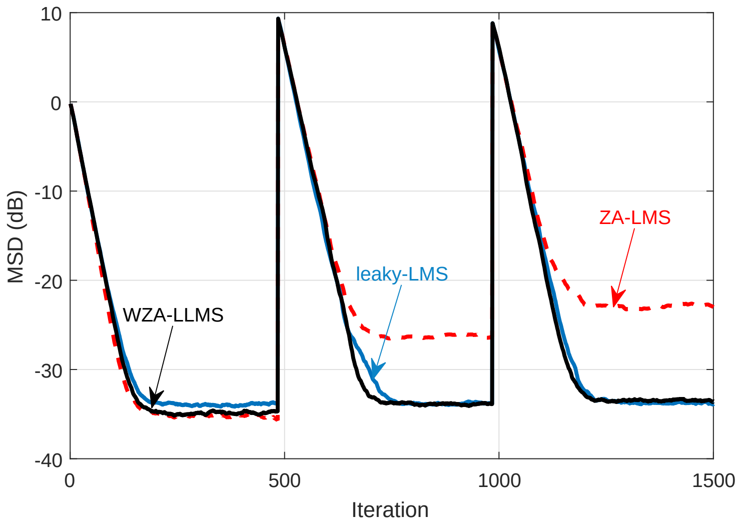


Figure 4.3: Comparison of MSDs in dB between the WZA-LLMS, ZA-LMS and LLMS algorithms in AWGN. The filter length is $N = 16$.

level of the channel alters in the previous experiments are adjusted to be at 5000^{th} and the 10000^{th} iteration respectively. The observed noise and SNR level are the same as before. Simulations are done with the following parameters: For the leaky-LMS: $\mu = 0.015$, $\gamma = 0.0001$. For the WZA-LLMS: $\mu = 0.015$, $\gamma = 0.0001$, $\zeta = 10$ and $\rho = 2 \times 10^{-4}$. For ZA-LMS: $\mu = 0.015$ and $\rho = 3 \times 10^{-5}$. Fig. 4.4 shows the average MSD curve for all algorithms in dB.

Referring to the MSD results obtained by the simulation, at highly sparse region, the WZA-LLMS has 400 iterations faster converges speed than the leaky-LMS but as same as the ZA-LMS algorithm. In addition, the WZA-LLMS algorithm results in 3.5 dB lower MSD relative to LLMS algorithm. By increasing the number of non-zero taps to eight, we observe that the WZA-LLMS algorithms converges faster than both algorithms with 0.5dB better MSD. in the case of complete non-sparse system, the WZA-LLMS algorithm still has better performance both algorithms.

To conclude this section, we repeat the same procedure for the fourth experiment

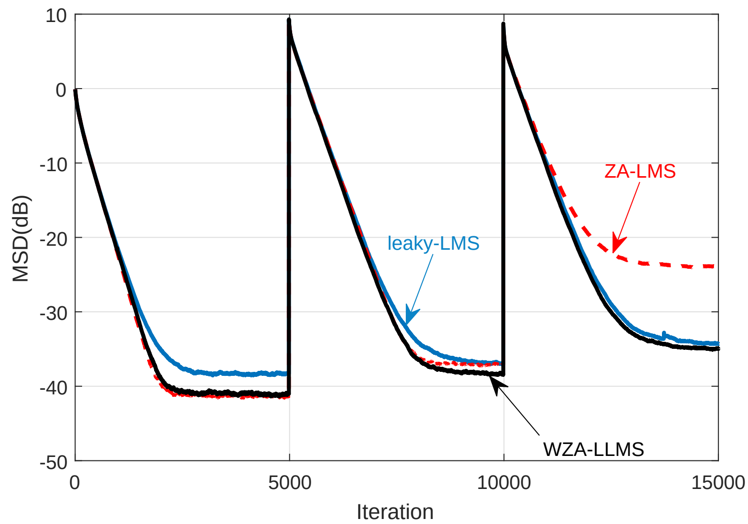


Figure 4.4: Comparison of MSDs between the WZA-LLMS, ZA-LMS and LLMS algorithms for correlated input signal and filter length $N = 16$.

with filter length set to be $N = 50$ and implemented with 200 independent runs explicitly. In this setting, the filter coefficients are designed such that the abrupt change occurs at certain periods of time. Initially, a random tap of the unknown system is set to 1 and the rest are zeros. After each 1500 iterations, 4 and 14 coefficients are set to ones and others to zeros, respectively. The performance of the ZA-VSSLMS algorithm is compared to those of the standard VSSLMS, LMS and ZA-LMS algorithms.

Once again, the simulation result confirms that for significantly sparse systems all proposed algorithm results in lower MSD compare to those conventional LMS-type algorithm as well as the sparse ZA-LMS algorithm. As the system sparsity decreases to completely non-sparse scenario the performance difference between the proposed sparse algorithm with respect to conventional adaptive filtering algorithms becomes smaller. However, it still performs comparably better than those conventional ones.

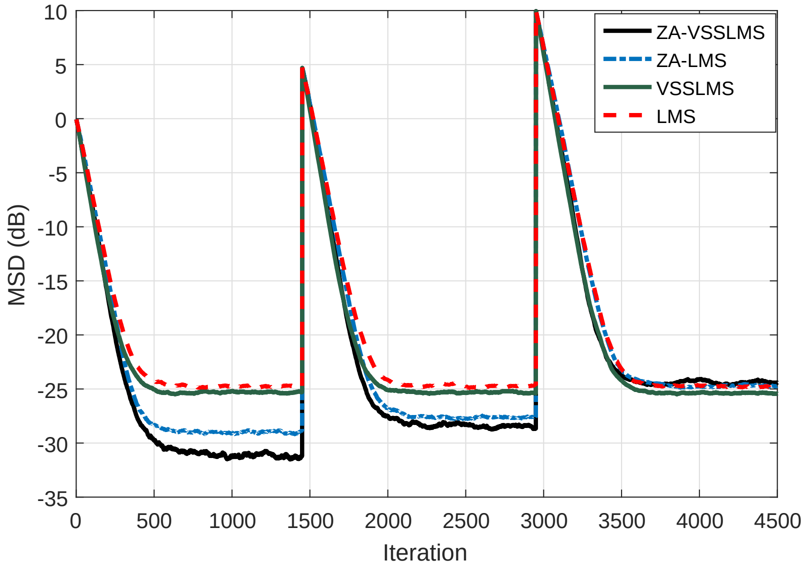


Figure 4.5: Comparison of MSDs of the VSSLMS, ZA-VSSLMS, LMS and ZA-LMS algorithms in AWGN. The filter length is $N = 50$.

4.3.2 The Effect of Zero-Attractor controller parameter On the ZA-VSSLMS Algorithm in System Identification

In this section, we investigate the effect of the varying parameter ρ on the performance of the ZA-VSSLMS algorithm in the system identification setting . The parameter ρ in the ZA-VSSLMS algorithm that were defined in the equation (3.11) controls the strength of the zero attractor in order to increases the convergence speed and decreases the MSD, by forcing the small filter taps to move toward the origin. Its value is changing within an interval $0 < \rho < 1$.

In order to observe the effectiveness of the MSD analysis in Section (3.6), One can select all different value of ρ's to be less than the theoretical upper-bound of ρ_∞ given in Eq. 3.43. By this, one can find the experimentally optimal value of ρ that results in optimal performance in the ZA-VSSLMS algorithm through simulation and employ it in the real design problem.

To show this, we carry out two similar experiments with slightly different

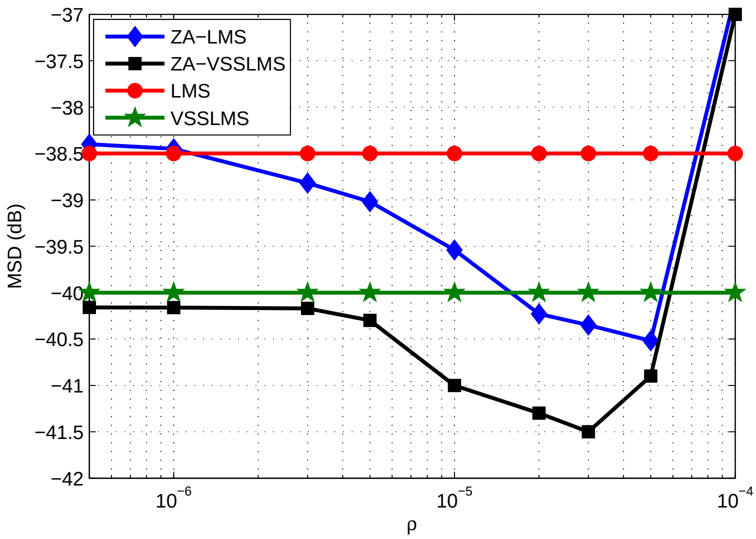


Figure 4.6: MSD vs ρ of the ZA-VSSLMS, ZA-LMS, VSSLMS and LMS algorithms in AWGN. The filter length is $N = 10$.

parameters. In the both experiments, as explained before, all the ρ's are chosen based on the upper-bound given in Eq. 3.43. In the first setting of system identification problem, the channel was assumed to be of length $N = 20$ with two nonzero coefficients distributed randomly (90% sparsity). The observation noise was assumed to be additive white Gaussian noise (AWGN) providing an SNR of 30 dB. The remaining parameters of the ZA-VSSLMS algorithm are: $\mu_{min} = 0.01$, $\mu_{max} = 0.1$, $\gamma = 0.0001$ and $\alpha = 0.97$.

As we see in Fig. 4.6, with the proposed bound on the parameter ρ, the performance of the ZA-VSSLMS algorithm keeps improving compared to those of ZA-LMS and VSSLMS algorithms by changing the ρ value until it reaches its minimum MSD at $\rho = 3 \times 10^{-5}$ ($\rho_{exp-opt} < \rho_{upper}$). At this optimal point, the ZA-VSSLMS algorithm shows 1 dB and 1.5 dB improvements over the ZA-LMS and VSSLMS algorithms, respectively. With $\rho > \rho_{exp-opt}$, the ZA-VSSLMS algorithm starts to lose its ability to attract the zero taps and hence yields a higher MSD compared to standard LMS and VSSLMS algorithms when $\rho > 6 \times 10^{-5}$.

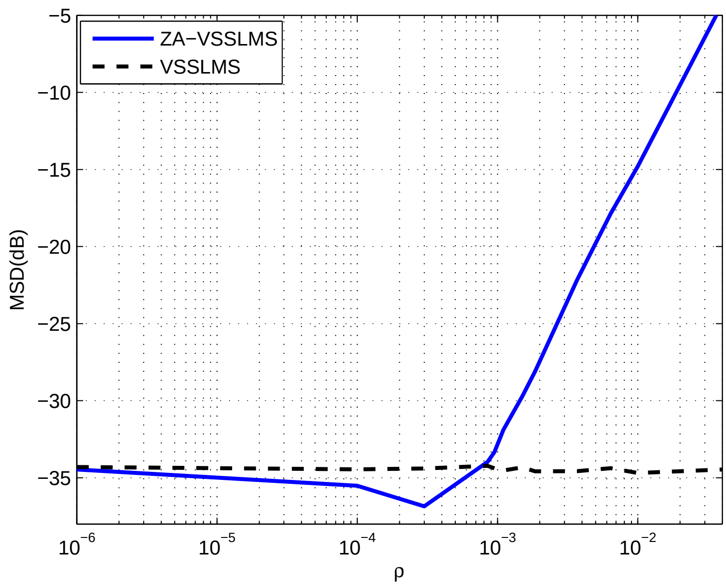


Figure 4.7: MSD vs ρ of the ZA-VSSLMS and VSSLMS algorithms in AWGN. The filter length is $N = 256$.

Similarly, we repeat the same experiment only for the ZA-VSSLMS and VSSLMS algorithms for a filter length of $N = 256$. Once again from Fig. 4.7, we confirm that the performance of the ZA-VSSLMS algorithm follows similar behavior as in the previous simulation until it reaches the minimum MSD at $\rho = 4 \times 10^{-4}$. At this point, the ZA-VSSLMS algorithm shows 4 dB improvement over the standard VSSLMS algorithm.

4.3.3 The Effect of forgetting factor On the ZA-VSSLMS Algorithm in System Identification

The forgetting factor parameter α given explicitly in (3.35) reveals the behavior of the step-size in the ZA-VSSLMS algorithm and plays a vital role on the steady-state performance of the algorithm. We derived that the MSD expression given in (3.42) is directly proportional to the the step-size parameter α of the ZA-VSSLMS algorithm.

In this section, we study the effect of parameter α on the MSD performance of ZA-VSSLMS for different values (0.2, 0.7 and 0.9) in the system identification problem.

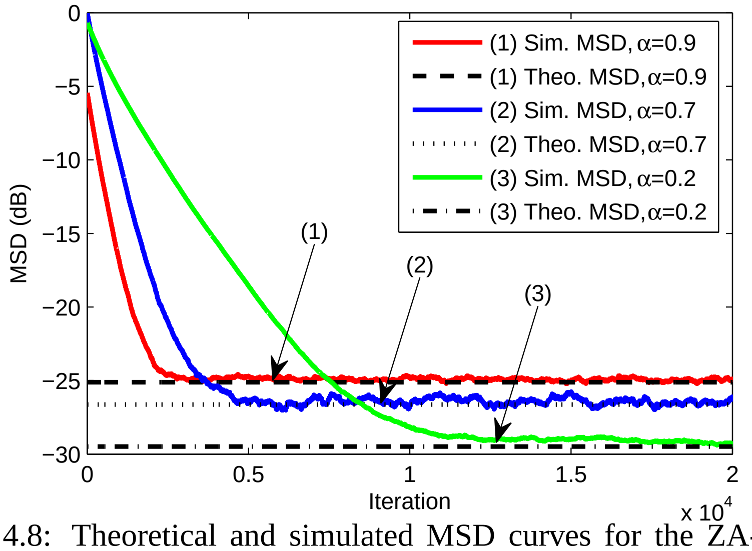


Figure 4.8: Theoretical and simulated MSD curves for the ZA-VSSLMS algorithm in AWGN with different α's.

In the following simulation, the channel is assumed to be of length $N = 4$ with 1 nonzero coefficient distributed randomly (75% sparsity). The observation noise was assumed to be AWGN with SNR set at 0 dB. The remaining parameters of the ZA-VSSLMS algorithm are: $\mu_{min} = 0.001$, $\mu_{max} = 0.1$, $\gamma = 0.0001$ and $\rho = 3 \times 10^{-5}$ (the experimentally optimum value in Fig. 4.6).

Fig. 4.8 shows the average MSDs for 3 values of α. For $\alpha = 0.2$, The ZA-VSSLMS has 3 dB and 4.5 dB lower MSD with respect to $\alpha = 0.7$ and 0.9 respectively. However, at the same α, the ZA-VSSLMS algorithm converges to the steady-state at a slower rate. Simulation results prove the validity of the theoretical studies that carried out in Chaper 3. Moreover, it shows a trade-off [45, 46] between lower MSD and faster convergence rate of the algorithm based on the value of parameter α.

4.3.4 The Effect of General White Noise parameters on the ZA-VSSLMS Algorithm in System Identification

Similar to step-size parameter α, the overall MSD in (3.42) is affected by the noise variance (σ^2) and the fourth moment (σ_v^4) respectively. Here we investigate the behavior of the ZA-VSSLMS algorithm is in the presence of white noise with different probability distributions.

In this experiment, the channel was assumed to be the same as in previous experiment in Section (4.3.3). The observation noises were assumed to be Uniform, Gaussian, Laplacian and Impulsive respectively. The SNR is set at 0 dB for all cases. The remaining parameters of the ZA-VSSLMS algorithm are: $\mu_{min} = 0.001$, $\mu_{max} = 0.1$, $\gamma = 0.0001$, $\alpha = 0.97$ and $\rho = 3 \times 10^{-5}$. All Simulation results in this experiment are averaged over 200 independent runs.

As it can be seen from Fig. 4.9, the ZA-VSSLMS performs better in a uniformly-distributed noise environment as compared to the Gaussian, Laplacian and Impulsive distributed ones. The performance is the lowest in Impulsive-distributed noise environment. The effect of the noise distribution is due to the different values of the fourth moment, $E(v_k^4)$, MSD expression given in (3.42). The theoretical values for this parameter with respect to its standard deviation (σ_v) are $1.8(\sigma_v^4)$, $3(\sigma_v^4)$, $6(\sigma_v^4)$ and $14(\sigma_v^4)$ respectively [47]. Once again, Simulation results indicate excellent agreement with the theoretical results.

4.3.5 The Effect of Non-White Input Signal on the ZA-VSSLMS Algorithm in System Identification

In this section, we consider the effect of a non-white input signal on the steady state performance of ZA-VSSLMS, the VSSLMS and , in particular, an alternative variable-step size adaptive algorithm known as the set-membership normalized LMS (SM-NLMS), [23], algorithms in system identification setting.

In this experiment, the input is generated by using correlated first-order

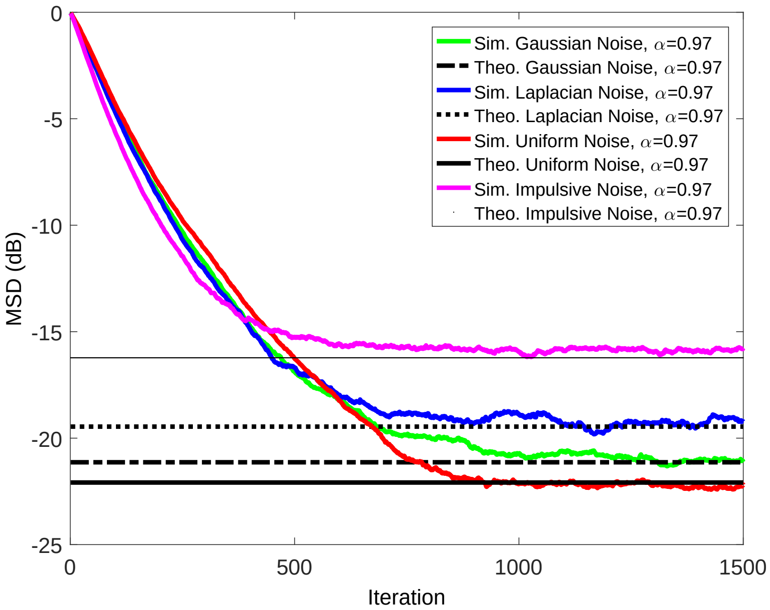


Figure 4.9: Theoretical and simulated MSD curves for the ZA-VSSLMS algorithm with different noise types.

Guassian-Markov model ($x(k) = a_1 x(k-1) + u(k)$)[44] with $a_1 = 0.8$. The channel was assumed to be the same as in previous section. The SNR is set at 0 dB and simulation results are the averaged deviation of 200 independent trials. The remaining parameters of the ZA-VSSLMS algorithm are: $\mu_{min} = 0.01$, $\mu_{max} = 0.1$, $\gamma = 0.0001$, $\alpha = 0.97$ and $\rho = 3 \times 10^{-5}$. For the VSSLMS algorithm: $\mu_{min} = 0.01$, $\mu_{max} = 0.1$, $\gamma = 0.0001$ and $\alpha = 0.97$. For the SM-NLMS algorithm: $\bar{\gamma} = \sqrt{5\sigma_n^2}$, where σ_n^2 is the noise variance, and $\gamma = 10^{-6}$. Fig. 4.10 shows that before 400^{th} iteration, the ZA-VSSLMS shows maximum 4 dB lower MSD and faster convergence rate compared to VSSLMS algorithm. After 400^{th} iteration, the performance difference between ZA-VSSLMS and VSSLMS algorithm becomes smaller. At steady state, however, the ZA-VSSLMS still performs comparably better than the VSSLMS. The SM-NLMS algorithm converges to 11 dB higher MSD than the other algorithm.

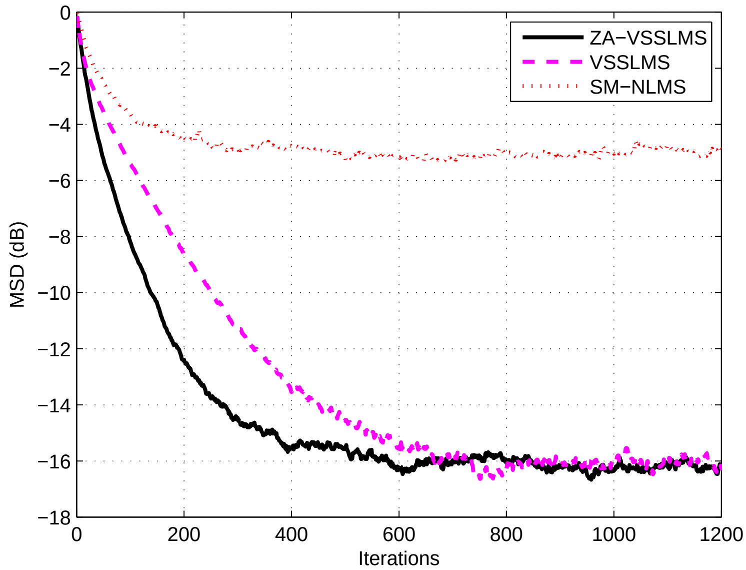


Figure 4.10: The MSD performance curve of the ZA-VSSLMS, VSSLMS and SM-NLMS algorithms for non-white input.

4.4 Echo Cancelation

Acoustic Echo Cancellation (AEC) is perhaps one of the most challenging problem in the domain of telecommunication systems [11, 48, 49]. over Several years, various adaptive filtering algorithms have been proposed to address this problem to deliver excellent performance [2, 23]. By emerging new technologies in the telecommunication industry such as internet phones and hand-free telephone, however, the demand for enhancing the speech/listening quality is still an active area of research in recent years. The main task of an echo canceler is to identify a replica of echo at the output of adaptive filter [50]. Fig. 4.11 shows the block diagram of a typical echo canceler.

From Figure, $\mathbf{x}(k)$ represents a speech signal from the far-end side that is transmitted in an acoustic room via a loudspeaker. A near-end speech signal,$\mathbf{v}(k)$, is then recorded the speech signal by microphone in the room and transmits back to the far-end side.

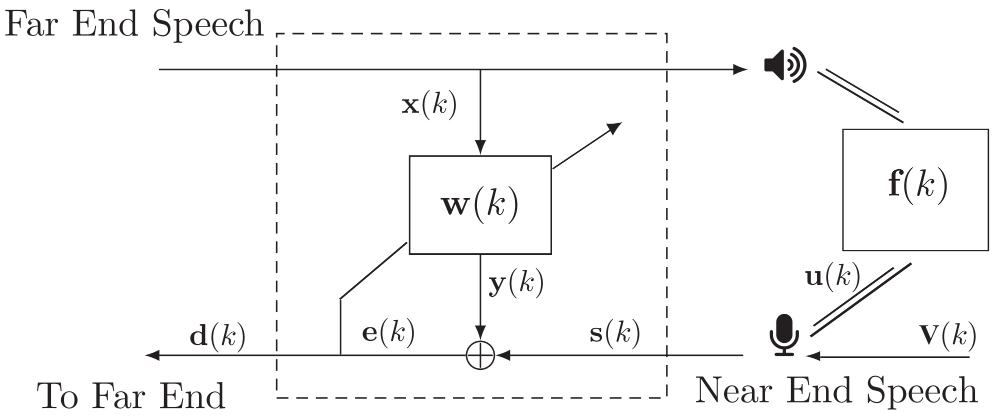


Figure 4.11: Schematic of an acoustic echo canceler.

This results in an acoustic echo path between the loudspeaker and the microphone. In other word, the recorded signal, $\mathbf{s}(k) = \mathbf{u}(k) + \mathbf{v}(k)$, is composed of an echo component $\mathbf{u}(k)$ and the near-end speech signal component $\mathbf{v}(k)$. In such a scenario, one can model the transfer function of the echo path using an FIR filter $\mathbf{f}(k)$, that turns the echo component as a filtered version of the loudspeaker signal, ($\mathbf{u}(k) = \mathbf{x}(k) * \mathbf{f}(k)$).

The basic task of an acoustic echo canceler is to identify the unknown room impulse response , $\mathbf{f}(k)$, to effectively eliminate the echo signal from the microphone signal. Consequently, the desired speech signal (with no echo) which is transmitted to the far-end side has the form $\mathbf{d}(k) = \mathbf{s}(k) - \mathbf{x}(k) * \mathbf{w}(k)$ [51]. It should be noted that $\mathbf{w}(k)$ corresponds to an estimate of $\mathbf{f}(k)$).

The problems of high power consumption and slow convergence speed makes the traditional adaptive algorithms impractical to use for an acoustic echo cancelation of the sparse system [14, 35, 52]. In many practical situations, the impulse response of a classic echo path is long and sparse [53]. In other words, the region where the impulse response coefficients have large magnitude, is relatively small [53, 54]. By taking advantage of this property, we employ the proposed algorithms through simulation in order to confirm the effectiveness of each method compared to that of conventional ones.

4.4.1 Experimental Results

In this section, we consider the convergence behavior of the proposed the WZA-LLMS, WZA-VSSLMS and the ZA-VSSLMS algorithms and compare to those of the LLMS, VSSLMS, ZA-VSSLMS and the SM-NLMS algorithm. All the experiments is set up for an echo cancelation problem by estimating the room impulse response such as shown in Fig. 1.1.

In the first experiment, We apply the proposed WZA-VSSLMS algorithm to an acoustic echo canceler consist of a 256-tap system with 28 non-zero coefficient distributed in a random fashion. The driving signal is white noise with the desired SNR level of 30 dB. The simulations are performed over 200 independent run. Simulations are done with the following parameters: For the leaky-LMS: $\mu = 0.005$, $\gamma = 0.002$.

For the WZA-LLMS: $\mu = 0.005$, $\gamma = 0.002$, $\zeta = 10$ and $\rho = 4 \times 10^{-4}$. For ZA-LMS: $\mu = 0.005$ and $\rho = 3 \times 10^{-5}$. Fig. 4.12 shows the average MSD curve for all algorithms in dB. As illustrated, the WZA-LLMS algorithm has the same convergence speed relative to the ZA-LMS algorithm but with 2dB lower MSD. With reference to the leaky-LMS, the WZA-LLMS algorithm has as much as 1000 iterations faster convergence speed with 3dB lower MSD.

In the similar fashion as the fist experiment of this section, the convergence performance of the ZA-VSSLMS algorithm is now tested and compared to those of

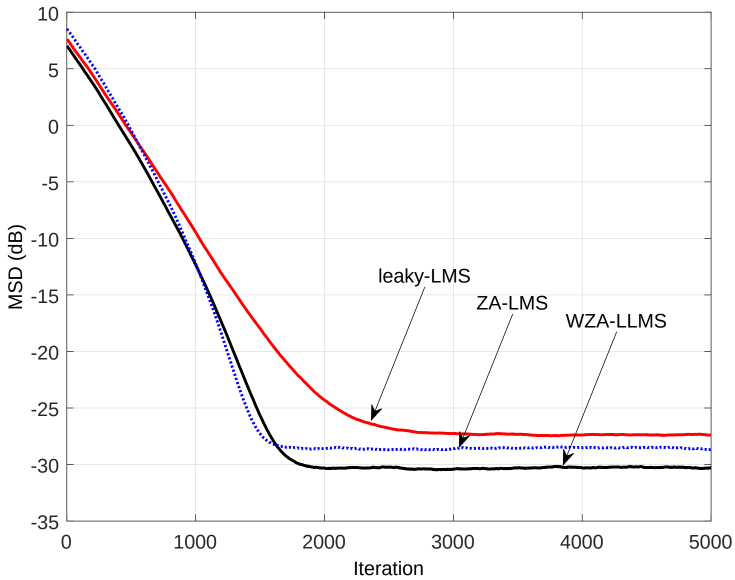


Figure 4.12: Convergence behaviors of the proposed WZA-LLMS algorithm in an acoustic echo cancelation setup. Filter length is $N = 256$.

the standard VSSLMS, LMS and ZA-LMS algorithms. The filter length is same as before with 10% nonzero coefficients distributed randomly. In this, the following parameters were used : For the VSSLMS: $\mu_{min} = 0.008$, $\mu_{max} = 0.012$, $\gamma = 0.0048$ and $\alpha = 0.97$. For the ZA-VSSLMS: $\mu_{min} = 0.008$, $\mu_{max} = 0.012$, $\rho = 0.0001$. For the ZA-LMS : $\mu = 0.005$. The simulations are performed for 200 independent runs. Fig. 4.13 demonstrates the averaged MSD curve for all algorithms in dB. As shown, the ZA-VSSLMS algorithm converges to 2 dB and 9 dB lower MSDs compared to the VSSLMS, ZA-LMS algorithms respectively. In addition, the proposed algorithm converges faster when room impulse response has sparse structure implying the sensitivity of the ZA-VSSLMS to such a scenario.

Finally to observe that the analytical results in Section(3.6) are consistent with our simulated result, we study the convergence behavior of the proposed ZA-VSSLMS algorithm in echo canceler problem and compare its result to those of VSSLMS and SN-NLMS algorithm. In this experiment, the room impulse response is assumed to

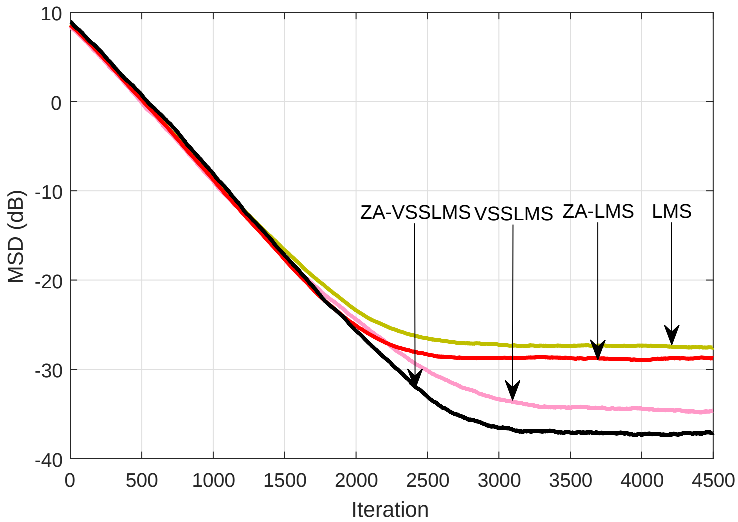


Figure 4.13: Convergence rate of the proposed ZA-VSSLMS algorithm in acoustic echo cancelation setup driven by a white input signal. Filter length is $N = 256$

be sparse with a total of 128 coefficients ($N = 128$); randomly 6 taps are set to 1 while the others are kept zero. The algorithms were simulated with the following parameters: $\mu_{min} = 0.001$, $\mu_{max} = 0.004$, $\gamma = 0.0001$, $\alpha = 0.97$ and $\rho = 5 \times 10^{-5}$. For the VSSLMS algorithm: $\mu_{min} = 0.001$, $\mu_{max} = 0.004$, $\gamma = 0.0001$ and $\alpha = 0.97$. For the SM-NLMS algorithm: $\bar{\gamma} = \sqrt{5\sigma_n^2}$ and $\gamma = 10^{-6}$.

As illustrated in Fig. 4.14, the ZA-VSSLMS algorithm converges to 1.5 dB lower MSD compared to the VSSLMS algorithm. Moreover, The ZA-VSSLMS algorithm converges much faster and to a 1.5 dB lower MSD than the SM-NLMS algorithm. This shows the advantage of the variable step-size update of the VSSLMS algorithm over that of the SM-NLMS algorithm.

4.5 Image Deconvolution

Blind image deconvolution [55], [56]-[58] addresses the problem of reconstructing the true image from a corrupted observation image without having the knowledge of either the original 2-D source or the degradation function.

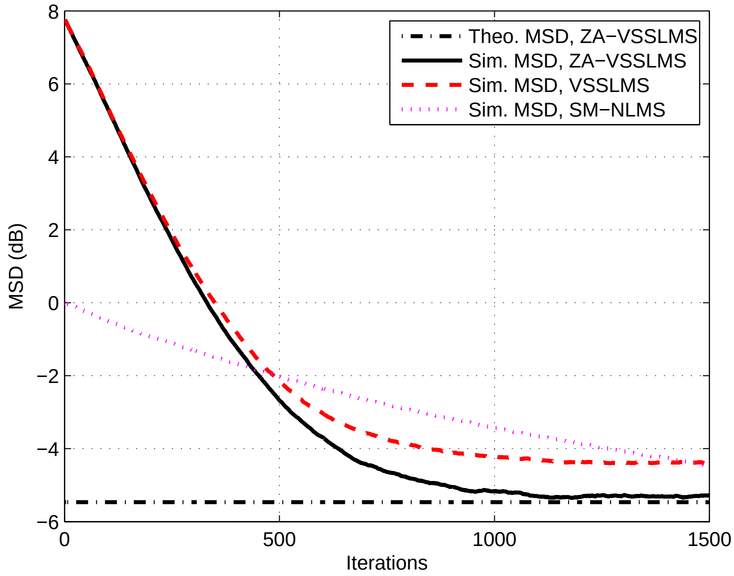


Figure 4.14: Convergence rate of ZA-VSSLMS, VSSLMS and SM-NLMS algorithms, for echo canceler driven by white input. The SNR is set at 0 dB.

No image is a perfect representation of the real world. All images have noise in them caused by the detection process in the camera. All images are also blurred to some extent, whether by focus problems, fundamental limitations or errors in the optics, motion blur, or the effects of air currents in the atmosphere [59]. All of these blurring effects can be modeled by a single Point-Spread Function (PSF) which is also known as a linear shift-invariant blur(LSI). An observed image can be modeled by the 2-D convolution of the true image with the PSF . Mathematically, this can be written as [55]:

$$s(x, y) = p(x, y) * m(x, y), \tag{4.2}$$

where $*$ represents the convolution operation in 2D, $s(x, y)$ is the corrupted image, $p(x, y)$ is the true image and $m(x, y)$ is the Point Spread Function (PSF) given in (4.7). In most systems, the PSF is unknown, or we may have partial information about it. Common techniques for reconstructing images include

Maximum-Likelihood (ML) approaches and the classical method of Least-Squares (LS), when the statistical properties of the noise are at hand. In both cases, we are lead to ill-posed problems and must ensure that appropriate regularizing measures are taken [60]. When the input image has a sparse structures, one may be interested in applying, to a certain extent, the proposed sparse adaptive algorithms in order to gain a possible improvements in reconstruction performance. Before we do that, it is important to generalize the proposed 1-D adaptive filters model into 2-D counterpart using the same approach [60, 61]. In the next section, we show how 1-D ZA-VSSLMS algorithm transforms into 2-D version and the sparsity is addressed to improve the performance of the filter in terms of both convergence rate and MSD.

The extended version has relatively low computational complexity (the computational complexity of the proposed algorithm is $O(N^2)$ which is of the same order as that of the VSSLMS algorithm). In addition, The filter has the ability of updating its coefficients by scanning along both the horizontal and vertical directions on a 2-D spatial coordinate providing better exploitation of information as well as the casuality condotion in the cases that it matters [62].

4.5.1 The two-dimensional zero-attracting variable step-size LMS

The update equation (3.11) can be readily extended to 2-D form as follows:

$$\begin{aligned} \mathbf{w}_{k+1}(p_1, p_2) &= \mathbf{w}_k(p_1, p_2) + \mu_k e_k(p_1, p_2)\mathbf{x}_k(l_1, l_2) \\ &\quad - \rho_k sgn(\mathbf{w}_k(p_1, p_2)) \end{aligned} \tag{4.3}$$

where $\mathbf{w}_{k+1}$ is the 2-D filter's coefficient vector with dimensions of $N \times N$, $\mathbf{x}_k(l_1, l_2)$ is the filter input vector and $p_1, p_2, l_1, l_2 = 0, 1, \ldots, N-1$, respectively.

More precisely, The filter input $\mathbf{x}_k(l_1, l_2)$ and coefficient vectors $\mathbf{w}_{k+1}$ can be written as the column vectors in the following form:

$$\mathbf{x}_k(l_1, l_2) = \begin{pmatrix} x(l_1, l_2) \\ \vdots \\ x(l_1, l_2 - N + 1) \\ \vdots \\ x(l_1 - N + 1, l_2) \\ \vdots \\ x(l_1 - N + 1, l_2 - N + 1) \end{pmatrix} \tag{4.4}$$

$$\mathbf{w}_k(p_1, p_2) = \begin{pmatrix} w(0, 0) \\ \vdots \\ w(0, N - 1) \\ \vdots \\ w(N - 1, 0) \\ \vdots \\ w(N - 1, N - 1) \end{pmatrix} \tag{4.5}$$

The filter output can be evaluated by a 2-D convolution as follows:

$$y(l_1, l_2) = \sum_{p_1=0}^{N-1} \sum_{p_2=0}^{N-1} w(p_1, p_2) x(l_1 - p_1, l_2 - p_2). \tag{4.6}$$

In the design of the 2-D filter, one should determine how to scan the input data so that it can be reused. There are many ways that data can be reused [58]. In this thesis we use the same method discussed in [62] to exploit the data efficiently.

4.6 Experimental Results

In the first part of the experiments, the performance of the proposed 2-D filter is compared to that of the 2-D VSSLMS algorithm in the image deconvolution setting. The test image we used for comparison is a sparse image 'Checker Board' in the shape of Fig. 4.15(a). Following parameters are used in this simulation: For the 2-D ZA-VSSLMS: $\mu_{min} = 0.005$, $\mu_{max} = 0.009$, $\gamma = 0.00048$, $\alpha = 0.97$ and $\rho = 4 \times 10^{-5}$. For the VSSLMS: $\mu_{min} = 0.005$, $\mu_{max} = 0.009$, $\gamma = 0.00048$ and $\alpha = 0.97$. The filter sizes in both cases are 3×3.

Fig. 4.15(a) indicated the test image 'Checker Board', Fig. 4.15(b) shows the blurred image which is obtained by using Eq. 4.2 where $p(x, y)$ is the original image and $s(x, y)$ is the 3×3 Gaussian PSF described as follows:

$$m(x,y) = \begin{pmatrix} -0.035 & -0.65 & -0.35 \\ 0.45 & 0.09 & 0.45 \\ 0.13 & -0.65 & 0.13 \end{pmatrix} \tag{4.7}$$

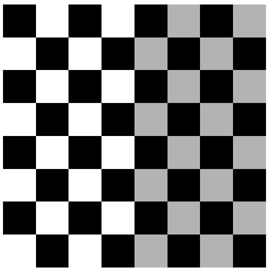
Fig. 4.15(c) illustrates the image recovered by 2-D VSSLMS and Fig. 4.15(d) shows the image recovered by the proposed 2-D ZA-VSSLMS algorithm. As it can be seen from Fig. 4.14(d), the proposed sparse algorithm results in a visually improved restored image. Also, the proposed 2-D ZA-VSSLSM algorithm yields approximately 1.0 dB improvement over the 2-D VSSLMS algorithm in terms of PSNR.

Even though the proposed algorithm performs better than the 2-D VSSLMS, its performance will be much better if we further assume that the Gaussian PSF has relatively many near-zero entries. To show this, with the same setting and parameters above, both algorithms were simulated using the PSF given in (4.8).

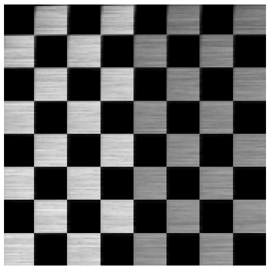
Fig. 4.16(b) demonstrates the degraded image. Fig. 4.16(c) illustrates the image recovered by 2-D VSSLMS and Fig. 4.16(d) indicates the image recovered by the proposed 2-D ZA-VSSLMS algorithm. As it can be seen, the proposed algorithm results in a much better performance (3.5 dB) than the 2-D VSSLMS algorithm when the input is sparse and PSF has meany near-zero coefficients.

$$h(x,y) = \begin{pmatrix} 0.035 & -0.0065 & -0.035 \\ 0.045 & 0.09 & 0.45 \\ 0.013 & -0.065 & 0.13 \end{pmatrix} \tag{4.8}$$

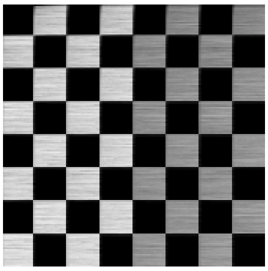
This shows, by assuming a sparse structure for the input image, small PSF coefficients and a proper input data reuse, one can design an efficient 2-D filter that considerably improve the performance of the adaptive filter. For an image deconvolution setup the simulation results show higher PSNR compared to 2-D VSSLMS algorithm.



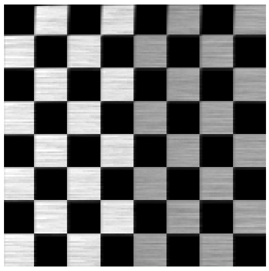
(a)



(b)

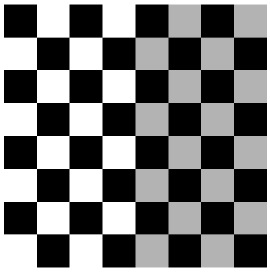


(c)

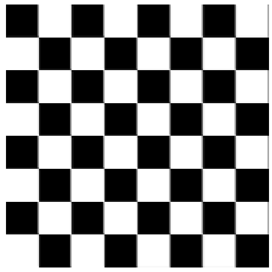


(d)

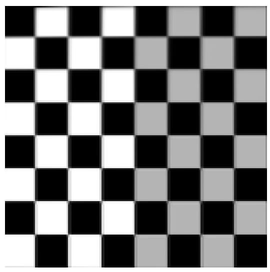
Figure 4.15: a) Test image, b) Degraded image, d) Recovered image by 2-D VSSLMS (PSNR=27.10 dB) and d) Recovered image by 2-D ZA-VSSLMS (PSNR=28.1 dB)



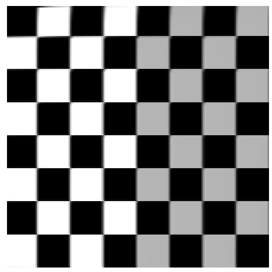
(a)



(b)



(c)



(d)

Figure 4.16: a) Test image, b) degraded image, d) Recovered image by 2-D VSSLMS (PSNR=30.5 dB) and d) Recovered image by 2-D ZA-VSSLMS (PSNR=34 dB)

Chapter 5

CONCLUSION AND FUTURE WORK

5.1 Conclusions

Sparsity is a feature present in many practical signals and systems. Efficient signal processing techniques which can exploit the sparseness property are very attractive from an implementation perspective. In this thesis, a set of novel sparsity-aware adaptive filtering algorithms are proposed. The proposed algorithms modifies the cost function of the conventional LMS, LLMS, VSSLMS algorithms in order to utilize the system sparsity. These modifications resulted in attracting zero or near-zero coefficients of adaptive filter-taps that significantly improve the performance of the adaptive filter in terms of both the convergence rate and MSD.

The ZA-LLMS and ZA-VSSLMS algorithms have been proposed in a similar optimization strategy by incorporating the l_1-norm into the cost function of the LLMS and VSSLMS algorithms respectively. The performance improves significantly when the impulse response of the channel is highly sparse. For a less sparse channel, however, the performance of the proposed l_1-norm based adaptive filters may deteriorate as the attractor term uniformly forces the filter-taps to become zero. To overcome this difficulty, by employing the logarithmic approximation to the l_0-norm, the WZA-LLMS and the WZA-VSSLMS algorithms are proposed.

For the ZA-VSSLMS algorithm, the convergence analysis has been presented to determine the stability condition. Moreover, the steady-state analysis has been carried out to derive the MSD expression for a general white input which is crucial in the real filter implementation. Furthermore, the upper bound on the zero-attractor controller (ρ) which yields minimum MSD has been theoretically derived. We have shown that by choosing the zero attracting controller parameter (ρ) smaller than that of the theoretical upper-bound the superiority of the ZA-VSSLMS algorithm is guaranteed. A 2-D extension of the ZA-VSSLMS has been presented to produce excellent results in imaging applications where sparsity is assumed.

The behavior of the ZA-VSSLMS algorithm has been investigated in the presence of four noise types. In order to achieve improved performance, by selecting optimal parameters, the effects of both the zero-attractor controller (ρ) and the step-size (α) on the MSD performance of ZA-VSSLMS algorithm have been studied.

The performance of the ZA-VSSLMS, WZA-VSSLMS and WZA-LLMS algorithms are compared with the standards ZA-LMS, VSSLMS, leaky-LMS, SM-NLMS and LMS algorithms in system identification, channel estimation, echo cancelation and image deconvolution problems. Results show that the theoretical and simulation results of the ZA-VSSLMS algorithm are in good agreement within a wide range of parameters, different channel, input signal and also noise types and outperform the standard algorithms.

5.2 Future Work

The tracking analysis of the proposed algorithms in both stationary and non-stationary environments could be a potential study for future work. Another area for investigation is to come up with more interesting applications and scenarios where sparse adaptive filtering can deliver promising performance. Finally, the development of an efficient hardware implementation of the proposed algorithms can be included in the future study.

REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*, 4^{th} ed., Prentice Hall, Upper Saddle River, NJ, 2002.

[2] B. F. Boroujeny, *Adaptive Filters: Theory and Applications*, John Wiley, Baffins Lane, Chichester, 1998.

[3] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice Hall, Eaglewood Cliffs, N.J., 1985

[4] B. Widrow and E. Walach, *Adaptive Inverse Control, Reissue Edition: A Signal Processing Approach*, Wiley-IEEE Press, 2007.

[5] K. Mayyas and T. Aboulnasr, "Leaky LMS algorithm: MSE analysis for Gaussian data," *IEEE Transactions on Signal Processing*, vol. 45, no. 4, pp. 927-934, 1997.

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

[7] A. H. Sayed, *Adaptive Filters*, John Wiley & Sons, NJ, 2008.

[8] M. G. Bellanger, *Adaptive Digital Filters*, 2^{nd} ed., Marcel Dekker, New York, 2001.

[9] K. Shi and P. Shi, “Convergence analysis of sparse LMS algorithms with ℓ_1-norm penalty based on white input signal,” *Signal Processing*, vol. 90, pp. 3289-3293, 2010.

[10] Y. Chen, Y. Gu and A. O. Hero, “Sparse LMS for system identification,” *IEEE International Conference on Acoustic, Speech, and Signal Processing*(ICASSP2009), pp. 3125-3128, May 2009.

[11] S.L. Gay and J. Benesty, *Acoustic Signal Processing for Telecommunication*, Kluwer Academic Publishers, Norwell, 2000.

[12] W. F. Schreiber, “Advanced television systems for terrestrial broadcasting: some problems and some proposed solutions,” *Proceedings of the IEEE*, vol. 83, pp. 958-981, June 1995.

[13] D. L. Duttweiler, “Proportionate normalized least-mean-squares adaptation in echo cancellers,” *IEEE Transactions on Speech Audio Processing*, vol. 8, pp. 508-518, September 2000.

[14] D. M. Etter, “Identification of sparse impulse response systems using an adaptive delay filter,” *Proceedings of ICASSP*, pp. 1169-1172, April 1985.

[15] D. Donoho, “compressive sensing,” *IEEE Transactions On Information Theory*, vol. 52, pp. 1289-1306, April 2006.

[16] W. U. Bajwa, J. Haupt, G. Raz and R. Nowak, “Compressed channel sensing,” *Proceedings of the 42nd annual conference on Information Sciences and Systems*, pp. 5-10, March 2008.

[17] Y. Gu, J. Jin, and S. Mei, “A stochastic gradient approach on compressive sensing signal reconstruction based on adaptive filtering framework,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 409-420, 2010.

[18] Y. Gu, J. Jin, and S. Mei,“ℓ_0 norm constraint LMS algorithm for sparse system identification,” *IEEE Signal Processing Letters*, vol. 16, pp. 774-777, September 2009.

[19] M. S Salman, M. N. S. Jahromi, O. Kukrer and A. Hocanin, “A Weighted Zero-Attracting Leaky-LMS Algorithm,” *SoftCOM 2012 International Conference on Software, Telecommunications and Computer Networks*, Croatia, September 2012.

[20] M. S. Salman, M. N. Jahromi, A. Hocanin and O. Kukrer, “A zero-attracting variable step-size LMS algorithm for sparse system identification,” *IX International Symposium on Telecommunications*, pp. 25-27, October 2012.

[21] M. N. S. Jahromi, M. S. Salman, A. Hocanin and O. Kukrer, “Convergence analysis of the zero-attracting variable step-size LMS algorithm for sparse system identification,” *Signal Image and Video Processing, Springer*, vol. 9, pp. 1353-1356, 2015.

[22] M. N. S. Jahromi, M. S. Salman, A. Hocanin and O. Kukrer, “Mean-square deviation analysis of the zero-attracting variable step-size LMS algorithm,” *Signal Image and Video Processing, Springer*, vol. 11, pp. 533540, 2017.

[23] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, 4th ed. NewYork, NY, USA: Springer, 2013.

[24] N. J. Bershad, “Analysis of the normalized LMS Algorithm with Gaussian Inputs,” *IEEE Transactions on Acoustic Speech and Signal Processing*, vol. ASSP-34, No. 4, pp. 793-806, August 1986.

[25] B. Allen and M. Ghavami, *Adaptive Array Systems: Fundamentals and Applications*, John Wiley & Sons Ltd, West Sussex, England, 2005.

[26] C. V. Sinn and J. Gotze, “Comparative study of techniques to compute FIR filter weights in adaptive channel equalization,” *IEEE International Conference on Acoustic, Speech and Signal Processing*, vol. 6, pp. 217-220, April 2003.

[27] X. Guan, X. Chen; and G. Wu, “QX-LMS adaptive FIR filters for system identification,” 2^{nd} *International Congress on Image and Signal Processing (CISP2009)*, 2009, pp. 1-5.

[28] B. Widrow and M. E. Hoff, “Adaptive switching circuits,” *IRE WESCON Convention Record*, vol. 4, 1960, pp. 96-104.

[29] G. Tummarello, F. Nardini and F. Piazza, “Stepsize control in NLMS acoustic echo cancellation using a neural network approach,” *International Symposium on Circuits and Systems*, vol. 5, pp. 705-708, May 2003.

[30] M. M. Chansarkar and U. B. Desai, “A Robust recursive least squares algorithm,” *IEEE Transactions on Signal Processings*, vol. 45, no. 7, pp. 1726-1735, July 1997.

[31] M.Z.A. Bhotto and A. Antoniou, “Robust recursive least-squares adaptive-filtering algorithm for impulsive-noise environments,” *IEEE Signal Processing Letters*, vol. 18, no. 3, pp. 185-188, January 2011.

[32] G. Gui, W. Peng, F. Adachi, “Improved Adaptive Sparse Channel Estimation Based on the Least Mean Square Algorithm,” *IEEE Wireless Communications and Networking Conference*, Shanghai,China, pp. 3105-3109, April 2013.

[33] FY. Wu, F. Tong, “Non-Uniform Norm Constraint LMS Algorithm for Sparse System Identification,” *IEEE Communications Letters*, vol. 2, pp. 385-388, 2014.

[34] G. Su, J. Jin, Y. Gu, J. Wang, “Performance analysis of l_0 norm constraint least mean square algorithm,” *IEEE Transactions on Signal Processing*, 2012, vol. 60, pp. 2223 - 2235, 2012.

[35] S. Kawamura and M. Hatori, “A tap selection algorithm for adaptive filters,”

Proceedings of ICASSP, vol. 11, pp. 2979-2982, April 1986.

[36] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society Series B*, 58, pp. 267-288, 1996.

[37] J. F. Kenney and E. S. Keeping, "Kurtosis," *In Mathematics of Statistics*, Pt. 1, 3rd ed. Princeton, NJ: Van Nostrand, pp. 102-103, 1962.

[38] H. V. Henderson and S. R. Searle, "The Vec-Permutation Matrix, The Vec Operator, and Kronecker Products, A Review," *Linear and Mulitilinear Algebra*, vol. 9, no. 4 pp. 271-288, 1981.

[39] G. Alexander, *Kronecker Products and Matrix Calculus with application*, New York, NY, USA: Halsted, 1981.

[40] W. H. Steeb, *Kronecker product of matrices and applications*, Mannheim : B. I. Wissenschaftsverlag, 1990.

[41] D. A. Turkington, *Generalized Vectorization, Cross-Products, and Matrix Calculus*, Cambridge University Press, 2012.

[42] L. Ljung , T. Glad, *Modeling of dynamic systems*, Prentice Hall, Englewood Cliffs, 1994.

[43] P. Eykhoff, *System identification*, Wiley, London, 1974.

[44] A. Papoulis, *Probability, Random variables, and Stochastic Processes*, 3rd ed. NewYork, NY, USA: McGraw-Hill, 1991.

[45] C. Y. Chi, C. C. Feng, C. H. Chen and C. Y. Chen, *Blind Equalization and System Identification: Batch Processing Algorithms, Performance and Applications*, Springer, Germany, 2006.

[46] V. Panuska, "An adaptive recursive-least-squares identification algorithm," 8^{th} *IEEE Symposium on Adaptive Processes and Decision and Control*, vol. 8, part 1, pp. 65-69, November 1969.

[47] G. Cowan, *Statistical Data Analysis*, Oxford Science Publications, Clarendon Press, 1998.

[48] M. S. Ahmad, O. Kukrer and A. Hocanin,"Recursive inverse adaptive filtering algorithm in acoustic echo cancellation," 6^{th} *International Symposium on Electrical & Electronic Engineering and Computer Systems* (EEECS 2010), Lefke, North Cyprus, November 2010.

[49] J. Benesty, T. Gnsler, D. R. Morgan, M. M. Sondhi and S. L. Gay, *Advances in Network and Acoustic Echo Cancellation*, Springer-Verlag , Berlin, Germany, 2001.

[50] J. Benesty, C. Paleologu, S. Ciochina and B.H. Juang, *Sparse Adaptive Filters for Echo Cancellation*, Morgan and Claypool Publishers, 2013.

[51] M. Rages and K. C. Ho, "Limits on echo return loss enhancement on a voice coded speech signal," *The* 45^{th} *Midwest Symposium on Circuits and Systems* (MWSCAS-2002), vol. 2, pp. 152-155, August 2002.

[52] Y. Li, Y. Gu, K.Tang, " Parallel NLMS filters with stochastic active taps and step-size for sparse system identification," *Proceedings of ICASSP*, pp. 1169-1172, May 2006.

[53] C. Yang, *Adaptive Filter Design for Sparse Signal Estimation* , Ph.D. Dissertations, University of Minnesota, 2011.

[54] Y. Chen, *Regularized Estimation of High-dimensional Covariance Matrices*, Ph.D. Dissertations, University of Michigan, 2011.

[55] D. Kundur and D. Hatzinakos, "A Novel blind deconvolution scheme for image restoration using recursive filtering," *IEEE Transactions on Signal Processing*, vol. 26, no. 2, pp. 375-390, February 1998.

[56] C. Vural and W. A. Sethares, "Blind deconvolution of noisy blurred images via dispersion minimization," 14^{th} *International Conference on Digital Signal Processing*, vol. 2, pp. 783-786, 2002.

[57] C. Vural and W. A. Sethares, "Recursive blind image deconvolution via dispersion minimization," 14^{th} *International Conference on Digital Signal Processing*, vol. 2, pp. 787-790, 2002.

[58] A. Giannoula, “Classification-based adaptive filtering for multiframe blind image restoration,” *IEEE Transactions on Image Processing*, vol. 20, no. 2, pp. 382-390, August 2011.

[59] M. N. S. Jahromi, A. Hocanin, O. Kukrer and M.S. Salman, “Two dimensional zero-attracting variable step-size LMS algorithm for sparse system identification,” *Proceedings of IEEE SIU 2013 (21st National Signal Processing and Applications Conference)*, Girne, North Cyprus, April 2013.

[60] M. M. Hadhoud and D. W. Thomas, “The two-dimensional adaptive LMS (TDLMS) algorithm,” *IEEE Transactions on Circuits and Systems*, vol. 34, pp. 1633-1642, July 1992.

[61] R. A. Soni and W. K. Jenkins, “Projection algorithms for two dimensional adaptive filtering applications,” *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers*, vol. 1, 1997.

[62] M. Ahmad, O. Kukrer and A. Hocanin, “A 2-D Recursive Inverse Adaptive Algorithm,” *Signal Image and Video Processing, Springer*, vol. 7, pp. 221226, March 2011.

[63] J. I. Nagumo and A. Noda, “A Learning method for system identification,” *IEEE Transactions on Automation and Control*, vol. AC-12, pp. 282-287, 1967.

[64] F. PirahanSiahi, S. N. H. S. Abdullah and S. Sahran, “Adaptive image segmentation based on peak signal-to-noise ratio for a license plate recognition

system," *International Conference on Computer Applications and Industrial Electronics* (ICCAIE 2010), pp. 468-472, 2010.

[65] T. Aboulnasr and K. Mayyas, "A robust step-size LMS-type algorithm: analysis and simulations," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 631-639, 1997.

APPENDICES

Appendix A: Proof of the MSD result in equation 3.38

Here we show how we obtain mathematically the result stated in equation (3.39) from (3.38). By using the Kronecker product property [39], for a given arbitrary matrices A, B and C of compatible sizes, $vec(ABC) = (C^T \bigotimes A)vec(B)$. Then, the expression given in (3.39) can be transformed into the following $vec(.)$ form and taking the expectation of both sides yields,

$$
\begin{aligned}
E\Big[vec\Big(\boldsymbol{\delta}(k+1)\boldsymbol{\delta}^T(k+1)\Big)\Big] &= E\Big[vec\left(\boldsymbol{\delta}(k)\boldsymbol{\delta}^T(k)\right)\Big] \\
&- \gamma E\Big[\underbrace{\Big(\mathbf{x}(k)\mathbf{x}^T(k) \otimes \mathbf{I})vec\left(\boldsymbol{\delta}(k)\boldsymbol{\delta}^T(k)\right)\Big)(g(i,k))}_{①}\Big] \\
&- \gamma E\Big[\underbrace{\Big((\mathbf{I} \otimes \mathbf{x}(k)\mathbf{x}^T(k))vec\left(\boldsymbol{\delta}(k)\boldsymbol{\delta}^T(k)\right)\Big)(g(i,k))}_{②}\Big] \\
&+ \gamma^2 E\Big[\underbrace{\Big((\mathbf{x}(k)\mathbf{x}^T(k) \otimes \mathbf{x}(k)\mathbf{x}^T(k))vec\left(\boldsymbol{\delta}(k)\boldsymbol{\delta}^T(k)\right)\Big)(g(i,k))}_{③}\Big] \\
&+ \gamma^2 E\Big[\underbrace{\Big(vec(\mathbf{x}(k)\mathbf{x}^T(k))v^2(k)\Big)(g(i,k))}_{④}\Big] \\
&- \lambda\gamma E\Big[\underbrace{\Big(\boldsymbol{\delta}(k)sgn[\mathbf{w}^T(k)]\Big)g(i,k)}_{⑤}\Big] \\
&- \lambda\gamma^2 E\Big[\underbrace{\Big(\mathbf{x}(k)\mathbf{x}^T(k)\delta(k)sgn[\mathbf{w}^T(k)]\Big)g(i,k)g(i,k)}_{⑥}\Big]
\end{aligned}
$$

$$+ \lambda\gamma E\Big[\underbrace{\Big(sgn[\mathbf{w}(k)]\boldsymbol{\delta}^T(k)\Big)g(i,k)}_{⑦}\Big]$$

$$- \lambda\gamma^2 E\Big[\underbrace{\Big(sgn[\mathbf{w}(k)]\boldsymbol{\delta}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)\Big)g(i,k)g(i,k)}_{⑧}\Big]$$

$$+ \lambda^2\gamma^2 E\Big[\underbrace{\Big(sgn[\mathbf{w}(k)]sgn[\mathbf{w}^T(k)]\Big)g(i,k)g(i,k)}_{⑨}\Big]$$

$$+ E\Bigg\{v(k)\Bigg[-\gamma\boldsymbol{\delta}(k)\mathbf{x}^T(k)g(i,k) + \gamma^2\mathbf{x}(k)\mathbf{x}^T(k)$$

$$\boldsymbol{\delta}(k)\boldsymbol{\delta}^T(k)g(i,k)g(i,k) - \gamma\mathbf{x}(k)\boldsymbol{\delta}^T(k)g(i,k)$$

$$+ \gamma^2\mathbf{x}(k)\boldsymbol{\delta}^T(k)\mathbf{x}(k)\mathbf{x}^T(k)g(i,k)g(i,k)$$

$$- \lambda\gamma^2\mathbf{x}(k)sgn[\mathbf{w}^T(k)]g(i,k)g(i,k) - \lambda\gamma^2 sgn[\mathbf{w}(k)]$$

$$\mathbf{x}^T(k)g(i,k)g(i,k)\Bigg]\Bigg\} \tag{A.1}$$

In steady-state we can assume the following:

1- $E\left\{e_{k-i-1}^2 e_{k-j-1}^2\right\} = E\left\{e_{k-i-1}^2\right\} E\left\{e_{k-j-1}^2\right\}$ for $i \neq j$.[65]

2- The expressions $(g(i,k))$, $(g(i,k))\,(g(i,k))$ and $\big(\boldsymbol{\delta}(k)\boldsymbol{\delta}^T(k)\big)$ are assumed to be independent.

Hence, employing the above assumptions we obtain (refer to next Section for the proof):

$$E\left\{(g(i,k))\,(g(i,k))\right\} = \frac{2\alpha\left(E\left(e_\infty^2\right)\right)^2}{(1-\alpha)^2(1+\alpha)} + \frac{E\left(e_\infty^4\right)}{(1-\alpha^2)} \tag{A.2}$$

and

$$E\left\{g(i,k)\right\} = \frac{E\left(e_{\infty}^{2}\right)}{1-\alpha}$$

From (A.1) it is easy to show that the expectation of the last term is zero ($E\{.\} = 0$). Now let us evaluate the expectations of terms ⑤, ⑥ and ⑨, individually.

Starting by term ⑤ and using the independence assumption,

$$E\left[\lambda\gamma\boldsymbol{\delta}(k)sgn[\mathbf{w}^{T}(k)]\left(g(i,k)\right)\right]$$
$$= \lambda\gamma E\left[(\mathbf{h}-\mathbf{w}(k))sgn[\mathbf{w}^{T}(k)]\right] \times E\left[g(i,k)\right] \quad \text{(A.3)}$$

To find expectation of (A.3) at steady state, we need to compute $E[\mathbf{w}_{\infty}]$. To do so, by applying the above assumption (2), we calculate the expectation of equation (3.37) in the article as $\lim_{k\to\infty}$ we obtain,

$$\begin{aligned} E[\boldsymbol{\delta}_{\infty}] &= \left(1 - \frac{\gamma\sigma_x^2 E\left(e_{\infty}^{2}\right)}{1-\alpha}\right) E[\boldsymbol{\delta}_{\infty}] + \frac{\lambda\gamma E\left(e_{\infty}^{2}\right)}{1-\alpha} E[sgn(\mathbf{w}_{\infty})] \\ &= \frac{\lambda}{\sigma_x^2} E[sgn(\mathbf{w}_{\infty})]. \end{aligned} \quad \text{(A.4)}$$

Substituting the result of (A.4) in (3.16) as $k \to \infty$ we get,

$$E[\mathbf{w}_{\infty}] = \mathbf{h} - \frac{\lambda}{\sigma_x^2} E[sgn(\mathbf{w}_{\infty})], \quad \text{(A.5)}$$

or in scalar form, (A.5) can be written as

$$E[w_{j,\infty}] = h_j - \frac{\lambda}{\sigma_x^2} E[sgn(w_{j,\infty})], \qquad j \in NZ \tag{A.6}$$

where $j \in NZ$ is the index that corresponds to a non-zero (NZ) coefficient. By [9], for sufficiently small $\frac{\lambda}{\sigma_x^2}$, the sign of (A.6) is,

$$E\left[sgn\left(w_{j,\infty}\right)\right] = sgn[h_j] \tag{A.7}$$

And hence, using (A.6) and (A.7) gives

$$E\left[w_{j,\infty}\right] E\left[sgn\left(w_{j,\infty}\right)\right] = \begin{cases} |h_j| & j \in \text{NZ} \\ \frac{-\lambda}{\sigma_x^2} E^2\left[sgn(w_{j,\infty})\right] & j \in \text{Z} \end{cases} \tag{A.8}$$

Additionally, from (A.7), it is straight forward to show that for $j \in$ NZ we have,

$$E\left[h_j sgn\left(w_{j,\infty}\right)\right] = |h_j|. \tag{A.9}$$

Substituting (A.6), (A.8), (A.9) in (A.3) and letting $E^2\left[sgn\left(w_{j,\infty}\right)\right] \approx 0$ if $j \in$ Z, the expectation in equation(A.3) at steady state reduces to,

$$\lambda\gamma E\left[(\mathbf{h} - \mathbf{w}(k))sgn[\mathbf{w}^T(k)]\right] E\left[g(i,k)\right] = \begin{cases} \frac{\gamma\lambda^2}{(1-\alpha)\sigma_x^2} E\left(e_\infty^2\right) & j \in \text{NZ} \\ 0 & j \in \text{Z} \end{cases} \tag{A.10}$$

In similar way it is easy to verify that the expectations of terms (6) and (9) in equation (A.1) are as follows,

$$\lambda\gamma^2 E\left[\mathbf{x}(k)\mathbf{x}^T(k)\delta(k)sgn[\mathbf{w}^T(k)]g(i,k)g(i,k)\right]$$

$$= \begin{cases} \frac{\gamma^2\lambda^2}{\sigma_x^2}\left[\frac{2\alpha\left(E\left(e_\infty^2\right)\right)^2}{(1-\alpha)^2(1+\alpha)} + \frac{E\left(e_\infty^4\right)}{(1-\alpha^2)}\right]vec(\mathbf{R}) & j \in \text{NZ} \\ 0 & j \in \text{Z} \end{cases} \tag{A.11}$$

and

$$\lambda^2\gamma^2 E\left[sgn[\mathbf{w}(k)]sgn[\mathbf{w}^T(k)]g(i,k)g(i,k)\right]$$

$$= \begin{cases} \frac{\lambda^2\gamma^2}{N}\sum_{j=0}^{N-1} sgn[|w_j|]\left[\frac{2\alpha\left(E\left(e_\infty^2\right)\right)^2}{(1-\alpha)^2(1+\alpha)} + \frac{E\left(e_\infty^4\right)}{(1-\alpha^2)}\right]vec(\mathbf{I}) & j \in \text{NZ} \\ 0 & j \in \text{Z} \end{cases} \tag{A.12}$$

The expectations of term ⑦ and ⑧ in equation(A.1) are identical to term ⑤ and ⑥, respectively.

As $k \to \infty$, let $vec\left(\boldsymbol{\Delta}_\infty\right) = E\left[vec\left(\boldsymbol{\delta}_\infty\boldsymbol{\delta}_\infty^T\right)\right]$ be a vector of size $N^2 \times 1$, $\xi = \left[\frac{2\alpha\left(E\left(e_\infty^2\right)\right)^2}{(1-\alpha)^2(1+\alpha)} + \frac{E\left(e_\infty^4\right)}{(1-\alpha^2)}\right]$ and $\mathbf{R} = E[\mathbf{x}_\infty\mathbf{x}_\infty^T]$. Hence, substituting the results of (A.10), (A.11) and (A.12) into (A.1) and using assumption (2) we get

$$\begin{aligned} vec\left(\boldsymbol{\Delta}_\infty\right) &= vec(\boldsymbol{\Delta}_\infty) + \gamma^2\xi E\left[\mathbf{x}_\infty\mathbf{x}_\infty^T \otimes \mathbf{x}_\infty\mathbf{x}_\infty^T\right]vec\left(\boldsymbol{\Delta}_\infty\right) \\ &\quad - \frac{\gamma E\left(e_\infty^2\right)}{(1-\alpha)}\left(E\left[\mathbf{I}\otimes\mathbf{x}_\infty\mathbf{x}_\infty^T\right] + E\left[\mathbf{x}_\infty\mathbf{x}_\infty^T\otimes\mathbf{I}\right]\right)vec\left(\boldsymbol{\Delta}_\infty\right) \\ &\quad + \left[\gamma^2\sigma_v^2 - \frac{2\gamma^2\lambda^2}{\sigma_x^2}\right]\xi vec(\mathbf{R}) + \frac{\lambda^2\gamma^2}{N}\xi\sum_{j=0}^{N-1} sgn[|w_j|]vec(\mathbf{I}) \end{aligned}$$

which is identical to equation (3.39) in the Section (3.6).

Appendix B :Proof of equation A.2

Recalling that $g(i,k) = \sum_{i=0}^{k-2} \alpha^i e^2(k-i-1)$, we want to evaluate the expectation of the form,

$$E\{g(i,k)g(i,k)\} = \sum_{i=0}^{k-2}\sum_{j=0}^{k-2} \alpha^{i+j} E\Bigg\{ e^2(k-i-1)e^2(k-j-1) \Bigg\} \quad \text{(A.13)}$$

To calculate the above expectation we divide (A.13) into two cases as following:

Case I: when $i = j$, as $k \to \infty$

$$\sum_{i=0}^{k-2}\sum_{j=0}^{k-2} \alpha^{i+j} E\Bigg\{ e^2(k-i-1)e^2(k-j-1) \Bigg\} = \frac{1}{1-\alpha^2} E(e_\infty^4)$$

Case II: when $i \neq j$, as $k \to \infty$ and by assumption (2) in A.1., in Appendix I, we have,

$$\sum_{i=0}^{k-2}\sum_{j=0}^{k-2} \alpha^{i+j} E\Bigg\{ e^2(k-i-1)e^2(k-j-1) \Bigg\} = \sum_{\substack{i=0 \\ (i \neq j)}}^{\infty} \sum_{j=0}^{\infty} \alpha^i \alpha^j \left(E(e_\infty^2)\right)^2 \quad \text{(A.14)}$$

Combining the results of case (I) & (II), the expectation of (A.13) becomes,

$$E\{g(i,k)g(i,k)\} = \frac{2\alpha \left(E\left(e_\infty^2\right)\right)^2}{(1-\alpha)^2(1+\alpha)} + \frac{E\left(e_\infty^4\right)}{(1-\alpha^2)}$$

Which is identical to (A.2).