DERIVATIVE BASED PROPORTIONATE TYPE ADAPTIVE FILTERING OVER SPARSE ECHO CHANNELS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MURAT BABEK SALMAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2018

Approval of the thesis:

DERIVATIVE BASED PROPORTIONATE TYPE ADAPTIVE FILTERING OVER SPARSE ECHO CHANNELS

submitted by **MURAT BABEK SALMAN** in partial fulfillment of the requirements for the degree of **Master of Science in Electrical and Electronics Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Tolga Çiloğlu
Head of Department, **Electrical and Electronics Engineering** ______________

Prof. Dr. Tolga Çiloğlu
Supervisor, **Electrical and Electronics Engineering Dept., METU** ______________

Examining Committee Members:

Assoc. Prof. Dr. Umut Orguner
Electrical and Electronics Engineering Dept., METU ______________

Prof. Dr. Tolga Çiloğlu
Electrical and Electronics Engineering Dept., METU ______________

Assist. Prof. Dr. Gökhan Muzaffer Güvensen
Electrical and Electronics Engineering Dept., METU ______________

Assist. Prof. Dr. Sevinç Figen Öktem
Electrical and Electronics Engineering Dept., METU ______________

Assoc. Prof. Dr. Cenk Toker
Electrical and Electronics Engineering Dept., Hacettepe Uni. ______________

Date: 05.09.2018

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

Name, Last name : Murat Babek Salman

Signature :

ABSTRACT

DERIVATIVE BASED PROPORTIONATE TYPE ADAPTIVE FILTERING OVER SPARSE ECHO CHANNELS

Salman, Murat Babek
MSc., Department of Electrical and Electronics Engineering
Supervisor : Prof. Dr. Tolga Çiloğlu

September 2018, 104 pages

Adaptive filters are the intelligent mechanisms, which are extensively used in the various real world applications such as system identification and equalization problems. Therefore, many well established adaptive filtering algorithms were developed in the literature. In this thesis, adaptive filters will be considered as a tool to identify an unknown impulse response. Impulse responses that are considered in current identification problems have special characteristics. These impulse responses are long and they are mainly composed of zero coefficients, only a few number of non-zero coefficients present in the impulse response. Due to this fact, well known adaptive filtering algorithms such as Normalized Least Mean Squares and Affine Projection Algorithm yield slow convergence. In order to deal with this performance problem, proportionate type algorithms were developed. Main idea behind the proportionate algorithms is to apply coefficient specific step-size by exploiting the sparse characteristics of the impulse response. In this thesis, previously proposed proportionate algorithms are investigated and their advantages and disadvantages are discussed. Furthermore, a novel approach using the dynamic behavior of filter coefficients is presented. In this approach, time derivatives of the filter coefficients are used in the adaptation process. Moreover, mathematical and geometrical analysis on the convergence of the proportionate algorithms are provided. The proposed algorithm is also extended to the situations in which non-Gaussian impulsive noise is present. In the presence of the non-Gaussian impulsive noise standard algorithms show poor performance in terms of robustness. Therefore, developed approach is combined with the algorithms that are robust against non-Gaussian impulsive noise. Superiority of the proposed approach is observed via the computer simulations.

Keywords: Sparse channels, System identification, Proportionate algorithms

ÖZ

SEYREK EKO KANALLARI İÇİN TÜREV TABANLI ORANTILI SÜZGEÇLEME

Salman, Murat Babek
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Tolga Çiloğlu

Eylül 2018, 104 sayfa

Uyarlanır süzgeçler, sistem tanılama ve denkleştirme gibi uygulamalarda sıklıkla kullanılan akıllı mekanizmalardır. Bu nedenle, literatürde çok sayıda iyi bilinen uyarlanır süzgeç algoritmaları geliştirilmiştir. Bu tezde, uyarlanır süzgeçler bilinmeyen bir sistemi tanılamak için kullanılmaktadır. Güncel tanılama problemlerine konu olan dürtü yanıtlarının önemli özellikleri vardır. Bu dürtü yanıtları uzundurlar ve çok sayıda sıfır büyüklüklü katsayıdan oluşmaktadır, sadece az sayıda sıfırdan farklı büyüklüğe sahip katsayı bulunmaktadır. Bu özellikleri, çok bilinen "Normalized Least Mean Squares" ve "Affine Projection Algorithm" gibi yöntemlerin yavaş yakınsamasına neden olmaktadır. Bu sorunu gidermek için, oratılı uyarlanır süzgeçler geliştirilmiştir. Uyarlanır süzgeç yönteminin arkasındaki temel mantık, dürtü yanıtının seyrek yapısından faydalanarak her katsayı için özel seçilmiş bir adım boyu uygulamaktır. Bu tezde, daha önce önerilen orantılı algoritmalar incelenmiş ve her birinin avantaj ve dezavantajları tartışılmıştır. Ayrıca, filtre katsayılarının dinamik davranışı düşünülerek yeni bir yaklaşım sunulmuştur. Bu tezde, filtre katsayılarının türevleri adaptasyon sürecine dahil edilmiştir. Ayrıca, önerilen algoritmanın matematiksel ve geometrik analizleri sunulmuştur. Önerilen yaklaşım aynı zamanda Gaussian olmayan gürültü içeren durumlar için de kullanılmıştır. Gaussian olmayan gürültü durumunda standart algoritmalar zayıf performans göstermektedirler. Bu nedenle geliştirilen yöntem Gaussian olmayan gürültüye karşı dayanıklı algoritmalarla birleştirilmiştir. Önerilen yöntemin üstünlüğü bilgisayar benzetimleriyle de gözlemlenmiştir.

Anahtar Kelimeler: Seyrek kanallar, Sistem tanılama, Orantılı algoritmalar

To My Family

TABLE OF CONTENTS

ABSTRACT v
ÖZ vii
TABLE OF CONTENTS x
LIST OF TABLES xiii
LIST OF FIGURES xiv
LIST OF ABBREVIATIONS xvi
CHAPTERS
1. INTRODUCTION 1
  1.1. Motivation 1
  1.2. Contributions 6
  1.3. Outline of the Thesis 7
2. PROBLEM STATEMENT AND ADAPTIVE FILTERING FRAMEWORK 9
  2.1. Introduction 9
  2.2. Problem Statement 9
  2.3. Steepest Descent Method 11
  2.4. Least Mean Squares (LMS) Algorithm 13
  2.5. Normalized Least Mean Squares (NLMS) Algorithm 15
  2.6. Affine Projection Algorithm (APA) 17
  2.7. NLMS and APA from Newton's Method Perspective 18
  2.8. Geometrical Perspective of the NLMS and APA 19
  2.9. Evaluation Criteria for Adaptive Filtering 21
3. ADAPTIVE FILTERS FOR SPARSE CHANNEL IDENTIFICATION 23
  3.1. Introduction 23
  3.2. Sparse Impulse Response Concept and Measure of Sparseness 23
  3.3. Proportionate Type Algorithms 24
  3.4. Proportionate NLMS (PNLMS) and Proportionate APA (PAPA) 25
  3.5. Improved PNLMS (IPNLMS) and Improved PAPA (IPAPA) 28
  3.6. Gradient Controlled IPNLMS and IPAPA 29
  3.7. Difference Based IPNLMS (D-IPAPA) and IPAPA (D-IPAPA) 32

3.8. Simulation Results of Proportionate Type Algorithms 33
3.9. Computational Complexities of the Proportionate Type Algorithms 36
3.10. A Mathematical Derivation of Proportionate Type Algorithms 37
4. SIGN ALGORITHMS:ROBUSTNESS AGAINST IMPULSIVE INTERFERENCE 43
4.1. Introduction 43
4.2. Motivation for Sign Algorithms 43
4.3. Normalized Sign Algorithm (NSA) 44
4.4. Affine Projection Sign Algorithm (APSA) 45
4.5. Proportionate Type Sign Algorithms 46
4.6. Performance Comparisons for Sign Algorithms 48
5. DERIVATIVE BASED PROPORTIONATE ALGORITHMS AND GEOMETRICAL INTERPRETATION 53
5.1. Introduction 53
5.2. Motivation for the Proposed Algorithm 53
5.3. Proposed Derivative Based Proportionate approach 58
5.4. Geometrical Interpretation of the Proportionate Type Algorithms 63
6. SIMULATION RESULTS 67
6.1. Introduction 67
6.2. Configuration of the Simulation Environment 67
6.3. Performance Comparisons of DB-IPNLMS and DB-IPAPA 68
6.3.1. Effect of Sparseness of the Channel 69
6.3.2. Colored Input Signal Case 71
6.3.3. Tests for Network Echo Channels from ITU-T Standards 72
6.3.4. Tracking Performance 75
6.3.5. Effect of Step-size Parameter 76
6.3.6. Effect of α Parameter 78
6.3.7. Effect of the Projection Order 80
6.3.8. Effect of SNR 81
6.3.9. Selection of Lm value 82
6.3.10. Misalignment Performances of Proportionate Type Sign Algorithms ... 84
6.3.11. Selection of Step-size Parameter for DB-IPAPSA 84

6.3.12. Comparisons of Time Evolutions of Proportionate Factors of DB-IPAPA, D-IPAPA and IPAPA 87

6.3.13. A Benchmark for the Proportionate Type Algorithms 91

7. CONCLUSION AND FUTURE WORK 93

REFERENCES 95

APPENDICES 99

Appendix A 99

Appendix B 103

LIST OF TABLES

TABLES

Table 3.1 Number of operations required for adaptive filtering algorithms 37

LIST OF FIGURES

Figure 1.1 Adaptive filtering schemes for (a) Identification (b) Inverse Modelling (c) Prediction (d) Interference Cancelling 2
Figure 2.1 Adaptive Filtering Scheme 10
Figure 2.2 Error surface for adaptive filter with length 2 12
Figure 2.3 Instantaneous Error Surface 14
Figure 2.4 Trajectory of the filter coefficients for Steepest Descent and Newton's algorithms 19
Figure 2.5 Geometrical representation of the update of NLMS 20
Figure 2.6 Geometrical representation of the update of APA 21
Figure 3.1 (a) Sparse and (b) Dispersive Impulse Responses 24
Figure 3.2 Evolutions of the filter coefficients for NLMS and PNLMS algorithms 27
Figure 3.3 Proportionate Factor Comparisons for IPAPA and GC-IPAPA 31
Figure 3.4 Performance comparisons of NLMS, PNLMS and IPNLMS algorithms for the sparse channel with $\xi = 0.9297$ 34
Figure 3.5 Performance comparisons of NLMS, PNLMS and IPNLMS algorithms for the dispersive channel with $\xi = 4947$ 34
Figure 3.6 Performance comparison of IPNLMS, D-IPNLMS and GC-IPNLMS 35
Figure 3.7 Performance comparison of IPAPA, D-IPAPA and GC-IPAPA 36
Figure 4.1 Performance comparisons of Sign Algorithms with conventional algorithms for SIR = 30 dB 49
Figure 4.2 Performance comparisons of Sign Algorithms with conventional algorithms for SIR = -10 dB 49
Figure 4.3 Performance comparisons of Proportionate type Sign Algorithms and APSA for SIR = -10 dB 50
Figure 4.4 Convergence behavior of the filter coefficients for APA and APSA 51
Figure 5.1 Coefficient evolutions for both major and minor coefficients for IPAPA 54
Figure 5.2 Coefficient evolutions for the same coefficient for AR(1) and Speech Input Signals for the proposed algorithm 60
Figure 5.3 Geometrical illustration of update process of NLMS and DB-IPNLMS 65
Figure 6.1 Performance comparison of proportionate type algorithms for $\xi = 0.7961$ 70
Figure 6.2 Performance comparison of proportionate type algorithms for $\xi = 0.8896$ 70
Figure 6.3 Performance comparison of proportionate type algorithms for $\xi = 0.5719$ 71
Figure 6.4 Performance comparison of proportionate type algorithms for $M = 2$ 72
Figure 6.5 Performance comparison of proportionate type algorithms for EPM-1 73
Figure 6.6 Performance comparison of proportionate type algorithms for EPM-2 73
Figure 6.7 Performance comparison of proportionate type algorithms for EPM-1 for Speech Input 74
Figure 6.8 Performance comparison of proportionate type algorithms for EPM-2 for Speech Input 75
Figure 6.9 Tracking performance comparison of proportionate type algorithms 76
Figure 6.10 Misalignment curves of DB-IPAPA for different step-sizes $\mu = 0.1$ to $\mu = 0.5$ 77

Figure 6.11 Misalignment curves of DB-IPAPA for different step-sizes $\mu = 1.8$ to $\mu = 2.2$ 77
Figure 6.12 Misalignment curves of IPAPA for different step-sizes $\mu = 0.1$ to $\mu = 0.5$..... 78
Figure 6.13 Misalignment curves of DB-IPAPA for different α values 79
Figure 6.14 Misalignment curves of IPAPA for different α values 79
Figure 6.15 Misalignment curves of the proposed algorithm for different projection order 80
Figure 6.16 Misalignment curves of the proposed algorithm for different SNR values 81
Figure 6.17 Performance comparison of the proportionate type algorithms for SNR=15 dB case........ 82
Figure 6.18 Performance of DB-IPAPA for different Lmvalues for AR(1) input signal...... 83
Figure 6.19 Performance of DB-IPAPA for different Lmvalues for speech input signal..... 83
Figure 6.20 Performance comparisons of proportionate type sign algorithms for SIR=-30 dB 84
Figure 6.21 Misalignment curves for DB-IPAPSA for different step-sizes........ 85
Figure 6.22 Performance comparisons of proportionate type sign algorithms when double talk occurs 86
Figure 6.23 Performance comparisons of sign and non-sign algorithms when double talk occurs 86
Figure 6.24 Proportionate factors of DB-IPAPA and IPAPA for a major coefficient.......... 88
Figure 6.25 Proportionate factors of DB-IPAPA and IPAPA for a minor coefficient.......... 89
Figure 6.26 Proportionate factors of DB-IPAPA and D-IPAPA for a major coefficient...... 90
Figure 6.27 Proportionate factors of DB-IPAPA and D-IPAPA for a minor coefficient...... 91
Figure 6.28 Misalignment curves of the proposed algorithm and the benchmark 92

LIST OF ABBREVIATIONS

AEC	Acoustic Echo Cancellation
APA	Affine Projection Algorithm
APSA	Affine Projection Sign Algorithm
BG	Bernoulli Gaussian
DB-IPAPA	Derivative Based Improved Proportionate Affine Projection Algorithm
DB-IPAPSA	Derivative Based Improved Proportionate Affine Projection Sign Algorithm
D-IPAPA	Difference Based Improved Proportionate Normalized Affine Projection Algorithm
D-IPNLMS	Difference Based Improved Proportionate Normalized Least Mean Squares
DTD	Double Talk Detector
FIR	Finite Impulse Response
GC-IPAPA	Gradient Controlled Improved Proportionate Affine Projection Algorithm
GC-IPAPSA	Gradient Controlled Improved Proportionate Affine Projection Sign Algorithm
GC-IPNLMS	Gradient Controlled Improved Proportionate Normalized Least Mean Squares
G-PAPA	Generalized Proportionate Affine Projection Algorithm
IAF-PAPA	Proportionate Affine Projection Algorithm with Individual Activation Factors

IPAPA	Improved Proportionate Affine Projection Algorithm
IPAPSA	Improved Proportionate Affine Projection Sign Algorithm
IPNLMS	Improved Proportionate Normalized Least Mean Squares
LMS	Least Mean Squares
NLMS	Normalized Least Mean Squares
NSA	Normalized Sign Algorithm
msl	Normalized steady state misalignment
PAPA	Proportionate Affine Projection Algorithm
PAPSA	Proportionate Affine Projection Sign Algorithm
PNLMS	Proportionate Normalized Least Mean Squares Algorithm
WGN	White Gaussian Noise
ZA-PNLMS	Zero Attracting Proportionate Normalized Least Mean Squares

CHAPTER 1

INTRODUCTION

1.1. Motivation

Filters are excessively employed in variety of engineering fields such as digital signal processing, biomedical engineering and control engineering [1-3]. These applications may involve filtering process for several purposes such as noise removal, signal enhancement, information extraction and prediction [1], [3], [4]. Depending on the application, filtering operations can be carried out either in the digital or analog domain. However, digital filters are generally chosen over analog filters due to the enhancements in the hardware technologies, which make the implementation of the digital filters much simpler [5]. Moreover, filters can be classified as linear and non-linear filters. Much of the attention in the literature, was paid to linear filtering since linear filtering has been extensively studied throughout the years [2]. Therefore, filters are often implemented as finite impulse response (FIR) filter, which eventually reduces the filtering problem to the determination of the FIR filter coefficients. Prior information about the signal statistics makes it possible to find the optimal values of the filter coefficients. However, statistical information is not always available; therefore, alternative methods should be developed to estimate the filter parameters.

Adaptive filtering is an approach to iteratively estimate the filter coefficients without need for prior information [6]. An adaptive filter can be defined as a time varying filter whose parameters are adjusted according to an adaptation rule so that the desired signal is obtained at the output. Adaptive filtering is an important tool of statistical signal processing since it provides implementation friendly solutions for many real-world applications such as telecommunications, echo cancellation, noise reduction, radar and sonar. [6]. Due to this fact, many complex adaptive filtering algorithms have been proposed for different applications which employ different adaptive filtering schemes. These schemes are shown in Figure 1.1.

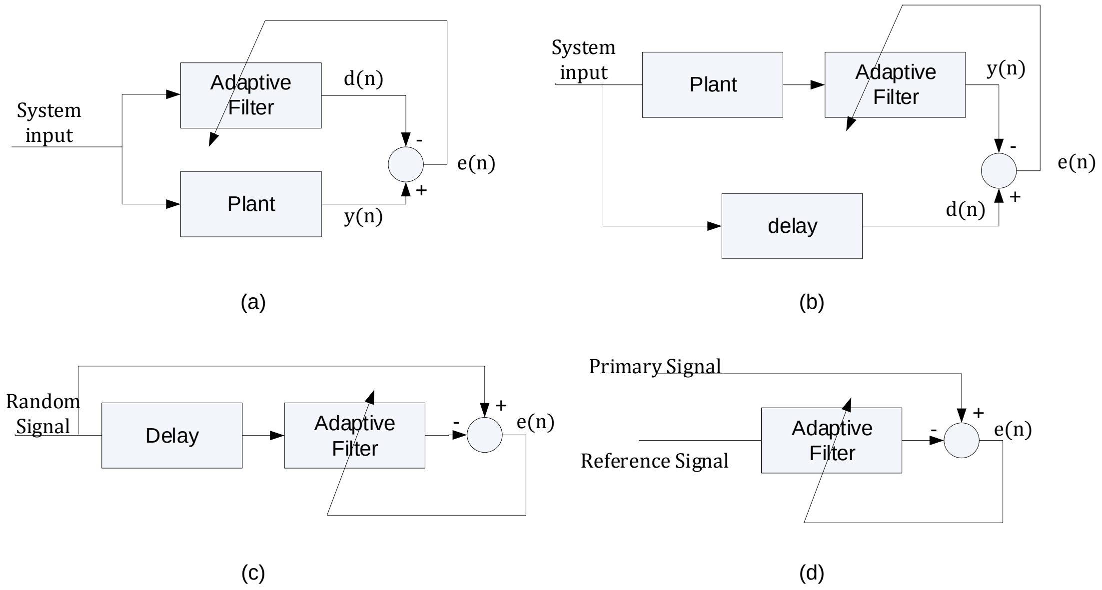

Figure 1.1 Adaptive filtering schemes for (a) Identification (b) Inverse Modelling (c) Prediction (d) Interference Cancelling

Main focus of this thesis is on the system identification scheme. System identification is a commonly encountered problem that employs adaptive filtering in order to model an unknown system by replicating the impulse response of the system [4]. In system identification scheme, both the adaptive filter and the unknown system receive the same input signal as shown in Figure 1.1. Coefficients of the adaptive filter is modified based on the difference between the outputs of the adaptive filter and the unknown system. For instance, in acoustic echo cancellation (AEC) [7] applications adaptive filters are used to replicate the impulse response of the echo path so that the undesired echo signal can be eliminated by subtracting the replicated echo signal generated at the output of the adaptive filter.

Many of the impulse responses that are considered in the recent studies have sparse characteristic. In these channels, energy is distributed among a small number of coefficients in the impulse response. In other words, most of the coefficients in the impulse response have zero or very small magnitudes, which are called minor coefficients. Relatively small number of coefficients have significantly larger magnitudes compared to the minor coefficients and these are defined as major coefficients. In the literature, such channels are classified as sparse channels. Network echo channels and acoustic channels are the examples of sparse channels. Another distinctive characteristic of these channels is the length of their impulse responses. Sparse channels are known to have longer impulse responses compared to that of dispersive channels. For instance, satellite-linked communication systems include sparse echo channels, which have long delays between the active echo regions in the impulse response [8]. These long delays in the echo path yield minor coefficients in the impulse response and active regions yield major coefficients. Due to the excessive length of the impulse response of the sparse channels, conventional adaptive filtering algorithms such as Normalized Least Mean Squares Algorithm (NLMS) [9] and Affine Projection Algorithm (APA) [10] suffer from slow convergence since convergence time is proportional to the length of the impulse response [9]. Therefore, algorithms that exploit the sparsity of the impulse response have been proposed in literature.

Proportionate type algorithms can be considered as one of the most important methods regarding the sparse channel identification. These algorithms achieve the same estimation accuracy by using smaller number of data compared to conventional adaptive filtering algorithms. Firstly, Proportionate NLMS (PNLMS) [11] and Proportionate APA (PAPA) [12] were introduced to provide fast convergence for sparse channels but these algorithms suffer from performance degradation as the sparsity of the impulse response decreases. Later, Improved PNLMS (IPNLMS) [13] and Improved PAPA (IPAPA) [12] were proposed which solve performance degradation problem for dispersive channels. These proportionate type algorithms improve the convergence rate by assigning individual step-sizes called proportionate factors, to each coefficient as roughly proportional to its magnitude. Later, many different proportionate type algorithms stemming from the same idea have been proposed such as μ-law IPNLMS [14], Individual Activation Factor IPNLMS (IAF-

IPNLMS) [15] and Zero Attracting LMS (ZA-PNLMS) [16]. In the classical proportionate type algorithms, adaptation is done by adjusting the coefficient specific step-sizes proportional to the "current" magnitude of the respective filter coefficients. Initial development of the proportionate algorithms was based on this intuitive idea that does not rely on any mathematical derivation. After a decade, [17] provided a basis pursuit perspective for the proportionate approach. Therefore, a theoretical basis is obtained for proportionate type algorithms which is based on the l_1 norm optimization criterion.

Classical proportionate type algorithms only provide fast convergence at the initial stage; hence, as adaptive filter gets closer to steady state, convergence speed decreases. In order to overcome this issue, new proportionate type algorithms, which take the dynamic behavior of the filter coefficients into account, were introduced. Generalized PAPA (G-PAPA) [18] is the first algorithm that uses the time derivatives of the filter coefficients while assigning the individual step-sizes. Later, Gradient Controlled IPAPA (GC-IPAPA) [19] was proposed which uses the gradient of the error with respect to estimated filter coefficients in order to control the individual step-sizes. Gradient vector corresponds to update term in the update equation; hence, GC-IPAPA assigns larger step-sizes to coefficients which are subject to greater changes in magnitude. If the estimated filter coefficients are initialized as zero, then only major coefficients are subject to significant changes. Hence, they will receive relatively larger step-sizes compared to that of minor coefficients until they reach steady-state values. Consequently; GC-IPAPA assigns proportionate factors proportional to difference between current and optimal filter coefficient magnitudes. Coefficient Difference Based IPAPA (D-IPAPA) algorithm was proposed in [20] which is also based on the distance between current filter coefficients and the optimal filter coefficients. This method approaches the adaptation in a block by block manner assuming that filter coefficient values do not change significantly during a block period, P samples. At the beginning of each block, estimated filter coefficient values are stored as the initial coefficient vector for the adaptive filter. Then proportionate factors are calculated by using the difference between current filter coefficients and stored initial vector. As the estimated filter coefficients get close to their optimal values, difference values become small for major coefficients; hence, small coefficients can receive reasonable step-size to converge faster.

These classical algorithms are developed based on the Gaussian noise assumptions; however, non-Gaussian interference is another important problem, which is frequently encountered in echo-cancellation applications [21]. In such cases, output of the unknown system is subject to impulsive noise, which may cause divergence of the adaptive filter. Double talk situation [22] in acoustic echo cancellation applications is an example of such non-Gaussian interference. In echo cancellation problems, far-end speech is the input of the unknown channel and output signal at the near-end side is used in the adaptation algorithm. In the presence of double talk, the output signal is corrupted by the near-end speech which makes coefficients of the adaptive filter significantly deviate from their optimal values.

Many methods have been proposed in order to compensate the effects of double talk interference. One of the methods is based on the detection of the double talk. In this method, adaptation is modified to avoid divergence during the presence of the interference [23]. However, detection of the double talk (DTD) is another major challenge; therefore, this study focuses on another approach, which provides algorithms which are inherently robust against impulsive interference. It can be shown that divergence of the affine projection type algorithms stem from the fact that they are result of l_2 norm optimization. Therefore, optimization problems based on lower order norms are used in order to have robustness against impulsive noise. [24] proposes an l_p-norm optimization scheme with $0 \leq p \leq 1$ to obtain robust algorithms. If p is selected as 1, well-known Normalized Sign Algorithm (NSA) is obtained as the solution. NSA provides robustness against impulsive interference by using the sign of the error signal rather than magnitude of the error signal, which is significantly affected by the impulsive interference. However, NSA suffers from significant degradation of convergence speed. In [21], Affine Projection Sign Algorithm was proposed to achieve fast convergence while preserving robustness feature. Later, proportionate approach is applied to sign algorithms to achieve further improvement in convergence rate by exploiting the sparse nature of the echo channels that are exposed to impulsive interference. In [25], Improved Proportionate APSA (IPAPSA) and Proportionate APSA (PAPSA) were proposed by combining the APSA with IPAPA and PAPA. In addition, Gradient Controlled IPAPSA (GC-IPAPSA) was proposed in which time averaged gradient approach is applied to the proportionate type sign algorithms in order to improve the convergence speed [35].

In addition to adaptive filtering, compressed sensing has also attracted much attention in the context of the sparse channel identification [36]. In particular, compressed sensing techniques are commonly employed in the estimation of sparse multipath channels in communication applications [36-38]. However, in the literature, compressed sensing techniques are not considered in the echo cancellation applications. Therefore, in this thesis, only adaptive filtering framework is considered for sparse system identification.

1.2. Contributions

Aim of this thesis is to develop fast converging adaptive filtering algorithms that achieves superior performance compared to previously proposed algorithms. In addition, the proposed algorithm is extended to its sign algorithm counterpart to have robustness against the impulsive interference. Proposed Derivative Based Improved Proportionate Affine Projection Algorithm (DB-IPAPA) and Derivative Based Improved Proportionate Affine Projection Sign Algorithm (DB-IPAPSA) are based on the observations of the dynamic behavior of the filter coefficients. In the proposed algorithms, update energy is distributed among the filter coefficients proportional to the rate of change of their magnitudes. As a result, coefficient specific step-sizes are adjusted such that convergence speed does not degrade during the adaptation process. In the proposed algorithm, time derivatives of the filter coefficients are used to calculate the proportionate factors. Consequently, proposed algorithm significantly outperforms previously proposed algorithms in terms of convergence speed and steady-state estimation error. Lastly, it should be stated that a scientific paper related to the studies presented in the thesis is being prepared.

1.3. Outline of the Thesis

Organization of the thesis is given as follows. Chapter 2 provides problem definition and mathematical background for adaptive filtering, starting from the steepest descent method. Following the steepest descent approach, algorithms employing gradient descent such as LMS, NLMS and APA are presented. Some interpretations regarding NLMS and APA are given to provide insight about these algorithms. Lastly, performance assessment criteria are also defined in this chapter. In Chapter 3, sparseness of the channel is defined and measure of sparseness is given. Then, some important proportionate type algorithms are introduced without any mathematical derivation since these algorithms were developed intuitively. Performances of the proportionate type algorithms are compared with that of non-proportionate types. Later, mathematical derivation of proportionate type algorithms is provided from a basis pursuit perspective. In Chapter 4, sign algorithms are introduced. Then proportionate type algorithms are extended to sign algorithms. In Chapter 5, Motivation for the proposed method for calculation of the proportionate factors is presented. Then proposed derivative based approach for calculation of the proportionate factors is introduced and geometrical interpretation of proposed algorithm is given. In Chapter 6, simulations results are given for the proposed algorithm and performance of the proposed algorithm is compared with that of previously proposed algorithm. In Chapter 7, conclusions and future works are discussed.

CHAPTER 2

PROBLEM STATEMENT AND ADAPTIVE FILTERING FRAMEWORK

2.1. Introduction

In this chapter, system identification problem is stated and optimal solution for the problem is derived. Then, Steepest Descent method, which can be considered as the fundamental approach in adaptive filtering theory, is introduced. Three, widely used Gradient Descent algorithms, namely LMS, NLMS and APA, are examined in detail. Advantages and disadvantages of each algorithm are discussed. Derivations of NLMS and APA based on Newton's method is given to elaborate on the convergence behavior of these algorithms. In addition, geometrical interpretations of NLMS and APA are given to provide intuition about the adaptation process which will be considered later while investigating the proportionate type algorithms. Lastly, criteria, which are considered while evaluating performance of the adaptive filters, are defined.

2.2. Problem Statement

This study focuses on identification of an unknown system by means of adaptive filtering such that impulse response of the unknown system is replicated. In system identification applications, adaptive filter and unknown system receive the same input signal. Output of the unknown system is treated as the reference signal and it is desired to obtain the same signal at the output of the adaptive filter. Adaptive filtering configuration used in this thesis is illustrated in Figure 2.1 and it should be noted that throughout this thesis, unless otherwise stated, all signals have real values.

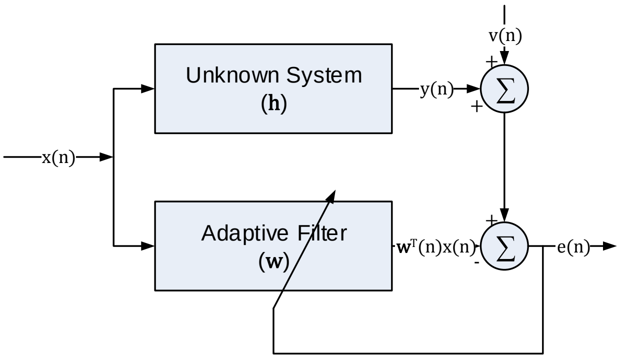

Figure 2.1 Adaptive Filtering Scheme

As Figure 2.1 depicts, a known signal at time n, $x(n)$, passes through an unknown channel, $\boldsymbol{h} = [h_0\ h_1 \ldots h_{L-1}]^T$with length L so that desired output signal, $y(n)$, is obtained as

$$y(n) = \boldsymbol{x}^T(n)\boldsymbol{h} + v(n), \tag{2.1}$$

where $\boldsymbol{x}(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^T$ is the input signal vector formed by the last L input signals, $v(n)$ is the background noise and superscript T denotes the transpose operation. $v(n)$ may have different statistical properties depending on the application. In most of the applications, $v(n)$ is modeled as white Gaussian noise (WGN); however, in some applications it is more suitable to model $v(n)$ as the combination of white Gaussian noise and Bernoulli Gaussian process (BG). Estimation error, $e(n)$, is obtained as the difference between desired and estimated output signals,

$$e(n) = y(n) - \boldsymbol{x}^T(n)\boldsymbol{w}(n) = y(n) - \boldsymbol{w}^T(n)\boldsymbol{x}(n), \tag{2.2}$$

where $\boldsymbol{w}(n) = [w_0(n), w_1(n), \ldots, w_{L-1}(n)]^T$ represents the estimated filter coefficients. In order to obtain a solution for $\boldsymbol{w}(n)$, a suitable cost function should be defined as a function of $\boldsymbol{w}(n)$. One of the most commonly utilized cost function in estimation theory is the Mean-Squared-Error [26] cost function which is given as,

$$J = E\{|e(n)|^2\} = E\{(y(n) - \boldsymbol{x}^T(n)\boldsymbol{w}(n))^T(y(n) - \boldsymbol{x}^T(n)\boldsymbol{w}(n))\}. \tag{2.3}$$

Wiener Filtering [27] provides the optimal solution for $\boldsymbol{w}$ so that J attains its minimum value. Since cost function defined in (2.3) is a convex function, stationary point of J corresponds to its global minimum. Consequently, solution of the following convex optimization problem gives the optimal filter coefficients,

$$\min_{\boldsymbol{w}} J = \min_{\boldsymbol{w}} E\{(y(n) - \boldsymbol{x}^T(n)\boldsymbol{w})^T(y(n) - \boldsymbol{x}^T(n)\boldsymbol{w})\}, \tag{2.4}$$

$$\begin{aligned}\min_{\boldsymbol{w}}[E\{y^2(n)\} - E\{y(n)\boldsymbol{w}^T\boldsymbol{x}(n)\} - E\{\boldsymbol{x}^T(n)\boldsymbol{w}(n)y^T(n)\}\\ + E\{\boldsymbol{w}^T\boldsymbol{x}(n)\boldsymbol{x}^T(n)\boldsymbol{w}\}].\end{aligned} \tag{2.5}$$

Equation (2.5) can be simplified as

$$\min_{\boldsymbol{w}}[E\{y^2(n)\} - 2E\{y(n)\boldsymbol{x}^T(n)\}\boldsymbol{w} + \boldsymbol{w}^T E\{\boldsymbol{x}(n)\boldsymbol{x}^T(n)\}\boldsymbol{w}]. \tag{2.6}$$

By taking the derivative of the cost function and equating it to zero, one can simply get the optimal solution,

$$\frac{\partial}{\partial \boldsymbol{w}}(E\{y^2(n)\} - 2\boldsymbol{w}^T E\{\boldsymbol{x}(n)y(n)\} + \boldsymbol{w}^T E\{\boldsymbol{x}(n)\boldsymbol{x}^T(n)\}\boldsymbol{w}) = 0, \tag{2.7}$$

$$-2E\{\boldsymbol{x}(n)y(n)\} + 2E\{\boldsymbol{x}(n)\boldsymbol{x}^T(n)\}\boldsymbol{w} = 0, \tag{2.8}$$

$$E\{\boldsymbol{x}(n)\boldsymbol{x}^T(n)\}\boldsymbol{w} = E\{\boldsymbol{x}(n)y(n)\} \tag{2.9}$$

$$\boldsymbol{R}_x\boldsymbol{w} = \boldsymbol{r}_{xy}, \tag{2.10}$$

where $\boldsymbol{R}_x = E\{\boldsymbol{x}(n)\boldsymbol{x}^T(n)\}$ is the $L \times L$ autocorrelation matrix of input signal and $\boldsymbol{r}_{xy} = E\{\boldsymbol{x}(n)y(n)\}$ is $L \times 1$ the cross-correlation vector of input and output signals. Solution of linear system of equations in (2.10) gives optimal filter coefficients. However, obtaining such statistics is a very difficult process and brings excessive computational burden to the system.

2.3. Steepest Descent Method

Steepest Descent is an iterative gradient based optimization method which is used to find a minimum of a certain function [4]. In adaptive filtering framework, Steepest Descent is employed to estimate the filter coefficients iteratively, which minimize the cost function provided that the cost function is differentiable at every point [4]. Let the cost function be defined by the Mean-Squared-Error criterion, $J = E\{|e(n)|^2\}$, which is a quadratic function of adaptive filter coefficients that can be represented by an error surface. Objective of the Steepest Descent is to reach the minimum point of the error surface step-by-step, starting from an arbitrary point on the error surface. At each step, it is desired to move to a point on the error surface, which is closer to the global minimum. Consider a case where optimal filter has two coefficients such that the error surface can be visualized in 3-D as shown in Figure 2.2. The goal is to reach the bottom of the error surface. In order to achieve this goal, it is clear that coefficients of the adaptive filter should move in the opposite direction of the gradient vector since gradient vector shows the direction along which cost function increases most.

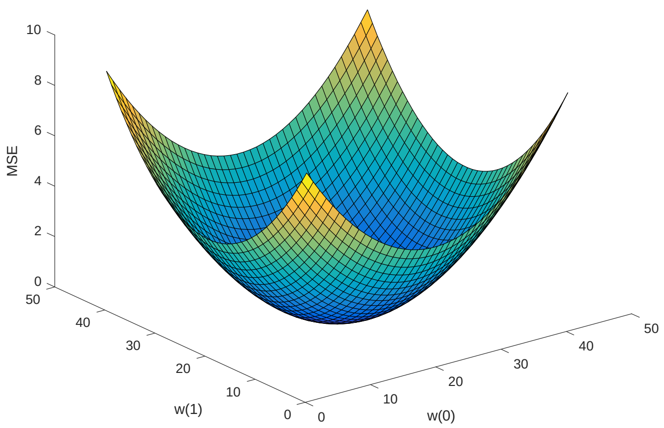

Figure 2.2 Error surface for adaptive filter with length 2

Mathematically, Steepest Descent method for adaptive filtering framework can be expressed as follows,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) - \mu \nabla_{\boldsymbol{w}} J, \qquad (2.11)$$

where μ is the step-size parameter which adjusts the adaptation speed of the algorithm, $\nabla_{\boldsymbol{w}}$ denotes the gradient operation with respect to $\boldsymbol{w}$. Cost function is already defined as ,$J = E\{|e(n)|^2\}$, then the gradient can be found as,

$$\nabla_{\boldsymbol{w}} J = -2\boldsymbol{r}_{xy} + 2\boldsymbol{R}_x \boldsymbol{w}(n). \tag{2.12}$$

By substituting $\nabla_{\boldsymbol{w}} J$ expression in (2.12) to (2.11), Steepest Descent Algorithm is obtained as,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu\left[\boldsymbol{r}_{xy} - \boldsymbol{R}_x \boldsymbol{w}(n)\right], \tag{2.13}$$

where the factor 2 in the $\nabla_{\boldsymbol{w}} J$ is included in μ. Similar to optimal filtering, Steepest Descent approach also requires prior knowledge on statistics of the related signals. Therefore, algorithms which use approximations of these correlation matrices, $\boldsymbol{r}_{xy}$ and $\boldsymbol{R}_x$, were developed.

Step-size is an important parameter that significantly affects the behavior of the adaptive filter. If step-size is selected too large, adaptive filter may diverge, on the other hand, if it is selected too small then its convergence speed decreases significantly. It was shown that choice of step-size depends on the magnitudes of the eigenvalues of the input correlation matrix. It is shown in Appendix A that, μ is bounded by $0 < \mu < 2/\lambda_{max}$ where, λ_{max} is the largest eigenvalue of $\boldsymbol{R}_x$ in order to ensure stability of the adaptive filter.

2.4. Least Mean Squares (LMS) Algorithm

In the literature, “Stochastic Gradient Descent” algorithms were proposed which employ approximations of the correlation matrices in the adaptation. LMS algorithm [28], is a Stochastic Gradient Descent algorithm which uses one of the simplest approximations for the correlation matrices. LMS algorithm employs instantaneous realizations of the signals as,

$$\boldsymbol{R}_x \approx \boldsymbol{x}(n)\boldsymbol{x}^T(n), \tag{2.14}$$

$$\boldsymbol{r}_{xy} \approx \boldsymbol{x}(n)y(n). \tag{2.15}$$

After inserting these approximations into (2.13), update equation becomes,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu[\boldsymbol{x}(n)y(n) - \boldsymbol{x}(n)\boldsymbol{x}^T(n)\boldsymbol{w}(n)], \tag{2.16}$$

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu\boldsymbol{x}(n)[y(n) - \boldsymbol{x}^T(n)w(n)], \tag{2.17}$$

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu\boldsymbol{x}(n)e(n). \tag{2.18}$$

As one can see from (2.18), update equation depends only on the instantaneous input vector and error signal. Therefore, implementation of LMS algorithm is extremely easy compared to the Steepest Descent Algorithm. Moreover, error surface changes at every step since approximated correlation matrices change at every step. A sample instantaneous error surface is shown in Figure 2.3. Change in the error surface creates random variations in the adaptive filter parameters since direction of the gradient changes randomly at each step. Hence, LMS algorithm suffers from gradient noise, which leads the adaptive filter to follow a noisy path during adaptation. Consequently, LMS algorithm has a slower convergence rate than the Steepest Descent Algorithm.

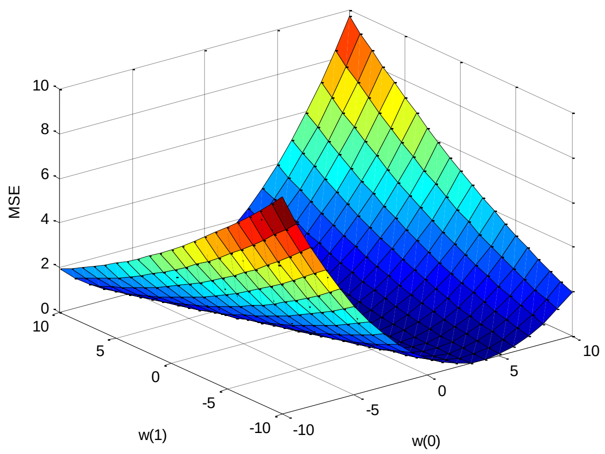

Figure 2.3 Instantaneous Error Surface

2.5. Normalized Least Mean Squares (NLMS) Algorithm

In LMS algorithm, magnitude of the input signal power has a significant effect on the adaption of the adaptive filter as can be observed in (2.18). Since input signal, $x(n)$, inherently includes the noise components, excessive increase in input signal power may yield gradient noise amplification problem [4]. Hence, new methods were needed in order to reduce the effects of the input signal power. For this purpose, Normalized LMS algorithm was proposed [9]. It is the solution of a constrained optimization problem which is defined as in (2.19) such that deviation of the filter coefficients from their current values is minimized to avoid fluctuations,

$$\begin{gathered}\min_{\boldsymbol{w}(n+1)} \|\boldsymbol{w}(n+1)-\boldsymbol{w}(n)\|_2^2 \\ \text{subject to } y(n)=\boldsymbol{x}^T(n)\boldsymbol{w}(n+1).\end{gathered} \tag{2.19}$$

This optimization problem can be solved by using the method of Lagrange Multipliers which gives the following cost function to be minimized,

$$J=\|\boldsymbol{w}(n+1)-\boldsymbol{w}(n)\|_2^2+\lambda[y(n)-\boldsymbol{x}^T(n)\boldsymbol{w}(n+1)], \tag{2.20}$$

where λ is called the “Lagrange multiplier”. In order to find the solution, cost function should be differentiated with respect to $\boldsymbol{w}(n+1)$ and be equated to zero,

$$\begin{aligned}&\frac{\partial J}{\partial \boldsymbol{w}(n+1)} \\ &=\frac{\partial\{[\boldsymbol{w}(n+1)-w(n)][\boldsymbol{w}(n+1)-w(n)]^T+\lambda[y(n)-\boldsymbol{x}^T(n)\boldsymbol{w}(n+1}{\partial \boldsymbol{w}(n+1)} \\ &=0,\end{aligned} \tag{2.21}$$

$$\frac{\partial J}{\partial \boldsymbol{w}(n+1)}=2[\boldsymbol{w}(n+1)-\boldsymbol{w}(n)]-\lambda\boldsymbol{x}(n)=0, \tag{2.22}$$

$$\boldsymbol{w}(n+1)=\boldsymbol{w}(n)+\frac{\lambda}{2}\boldsymbol{x}(n). \tag{2.23}$$

By substituting (2.23) into constraint in (2.19), λ can be obtained

$$y(n) - \boldsymbol{x}^T(n)\left[\boldsymbol{w}(n) + \frac{\lambda}{2}\boldsymbol{x}(n)\right] = 0, \tag{2.24}$$

$$y(n) - \boldsymbol{x}^T(n)\boldsymbol{w}(n) - \frac{\lambda}{2}\boldsymbol{x}^T(n)\boldsymbol{x}(n) = 0, \tag{2.25}$$

$$e(n) = \frac{\lambda}{2}\boldsymbol{x}^T(n)\boldsymbol{x}(n), \tag{2.26}$$

$$\lambda = \frac{2e(n)}{\boldsymbol{x}^T(n)\boldsymbol{x}(n)}, \tag{2.27}$$

where $\|\boldsymbol{x}(n)\|_2^2 = \boldsymbol{x}^T(n)\boldsymbol{x}(n)$ is the instantaneous input signal power. Then by substituting (2.27) into (2.23), update equation is obtained as

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + 2\frac{\boldsymbol{x}(n)e(n)}{\|\boldsymbol{x}(n)\|_2^2}. \tag{2.28}$$

In order to control the adaptation, a step-size parameter is inserted to (2.28) and overall update equation becomes,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu\frac{\boldsymbol{x}(n)e(n)}{\|\boldsymbol{x}(n)\|_2^2}. \tag{2.29}$$

It is clear from (2.29) that as the input signal power increases, normalization value, which is the denominator value of the update term, also increases such that effects of the input power are compensated. Hence, adaption becomes independent from the input signal power. However, there is a possibility of division by zero in (2.29).To avoid division by zero, a regularization term, ϵ, is also included,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu\frac{\boldsymbol{x}(n)e(n)}{\|\boldsymbol{x}(n)\|_2^2 + \epsilon}. \tag{2.30}$$

In order to guarantee the stability of the NLMS algorithm step-size is bounded by $0 < \mu < 2$ [4]. Although it provides robustness against excessive input signal power, NLMS algorithm suffers from performance degradation for colored input since it only uses the instantaneous realization of the input signals. When the correlated input signal is applied to the system instantaneous value of the input signal does not provide sufficient information. Therefore, instead of using instantaneous input signal vector, past input signal vectors should also be employed in the update process.

2.6. Affine Projection Algorithm (APA)

Instantaneous values of the signals were used in the approximations of the correlation matrices, $\boldsymbol{R}_x$ and $\boldsymbol{r}_{xy}$ in LMS and NLMS algorithms. However, these approximations do not satisfy performance requirements in case of colored input signals [10]. Consequently, in addition to the current values, past values of these signals are embedded into the constraint equation of the optimization problem in order to overcome performance degradation issue. Then, optimization problem becomes,

$$\begin{gathered}\min_{\boldsymbol{w}(n+1)} \|\boldsymbol{w}(n+1) - \boldsymbol{w}(n)\|_2^2 \\ \text{subject to } \boldsymbol{y}(n) = \boldsymbol{X}^T(n)\boldsymbol{w}(n+1),\end{gathered} \tag{2.31}$$

where $\boldsymbol{X}(n) = [\boldsymbol{x}(n), \boldsymbol{x}(n-1), \dots, \boldsymbol{x}(n-M+1)]$ is the input matrix formed by the last M input vectors, $\boldsymbol{y}(n) = [y(n), y(n-1), \dots, y(n-M+1)]^T$ is the output vector formed by last M output signals and M is the projection order of the algorithm. By using the method of Langrage Multipliers with multiple constraints, update equation of APA is obtained as,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu\boldsymbol{X}^T(n)(\boldsymbol{X}(n)\boldsymbol{X}^T(n) + \epsilon\boldsymbol{I})^{-1}\boldsymbol{e}(n), \tag{2.32}$$

where $\boldsymbol{I}$ is $M \times M$ identity matrix and $e(n) = \boldsymbol{y}(n) - \boldsymbol{X}^T(n)\boldsymbol{w}(n)$ is the error vector. Derivation of APA is given in Appendix B. One can observe the similarity between APA and NLMS algorithm. In fact, NLMS algorithm can be considered as a special case for APA when the projection order is $M = 1$.

APA provides higher convergence speed compared to NLMS algorithm. As the projection order increases, convergence speed also increases. However, improvement in the convergence rate decreases as the projection order increases. Main drawback of

APA is its computational complexity since it requires matrix multiplications and inversion during the adaptation. Moreover, computational complexity of APA depends on the projection order with $O(M^2)$; therefore, as the projection order increases computational complexity of the algorithm significantly increases. Therefore, choice of a reasonable projection order is an important issue.

2.7. NLMS and APA from Newton's Method Perspective

Another approach regarding the derivation of APA and NLMS is considering them as the simplified versions of the Newton's method [29]. In order to verify this point, update equation of the Newton's method is considered,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu \boldsymbol{R}_x^{-1}\left[\boldsymbol{r}_{xy} - \boldsymbol{R}_x \boldsymbol{w}(n+1)\right]. \tag{2.33}$$

If one inserts the approximations for the correlation matrices,

$$\boldsymbol{R}_x = \boldsymbol{x}(n)\boldsymbol{x}^T(n), \tag{2.34}$$

$$\boldsymbol{r}_{xy} = \boldsymbol{x}(n)y(n), \tag{2.35}$$

and after some matrix manipulations [29], (2.33) becomes,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) \tag{2.36}$$
$$+ \mu[\boldsymbol{x}^T(n)\boldsymbol{x}(n)]^{-1}[\boldsymbol{x}(n)y(n) - \boldsymbol{x}(\boldsymbol{n})\boldsymbol{x}^T(n)\boldsymbol{w}(n)],$$

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \frac{\mu \boldsymbol{x}(n)e(n)}{\|\boldsymbol{x}(n)\|_2^2}, \tag{2.37}$$

which is the update equation for the NLMS algorithm. In a similar way, this approach can be extended to APA [29].

Steepest descent method adjusts the filter coefficients in the opposite direction of the gradient. However, in Newton's method, filter coefficients move directly toward their optimum values since the gradient is rotated by multiplication with $\boldsymbol{R}_x^{-1}$.

Therefore, convergence speed of Newton's based algorithms is higher than that of Steepest Descent algorithms. In Figure 2.4. convergence of the filter coefficients for Steepest Descent and Newton's algorithms are shown. It can be observed that coefficients, which are updated according to Newton's algorithm, follow a straight path toward optimal point; however, coefficients follow a longer path when Steepest Descent algorithm is applied. This longer path also demonstrates the source of slow convergence of the Steepest Descent Algorithm.

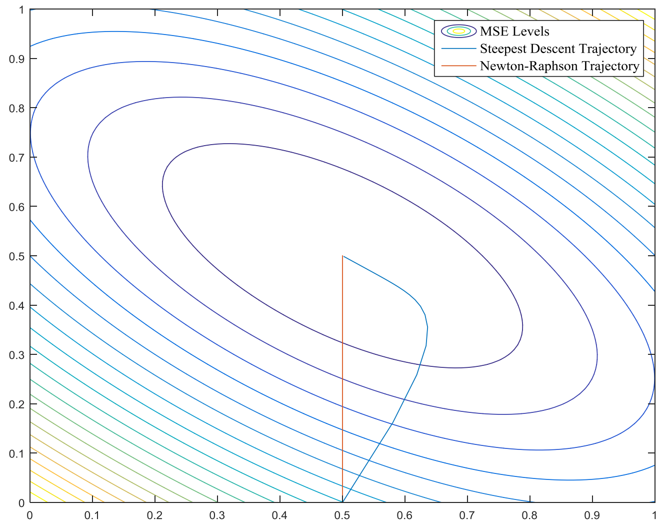

Figure 2.4 Trajectory of the filter coefficients for Steepest Descent and Newton's algorithms

2.8. Geometrical Perspective of the NLMS and APA

It is important to understand the geometrical reasoning behind the adaptation of NLMS and APA, therefore, in this section a geometrical insight for adaptation process is provided. From the update equation of the NLMS, it can be observed that adjustment of the coefficients is done in the positive or negative direction of the input vector depending on the sign of the error signal as in

$$\delta \boldsymbol{w}(n+1) = \frac{\boldsymbol{x}(n)e(n)}{\|\boldsymbol{x}(n)\|_2^2}, \tag{2.38}$$

where $\delta\boldsymbol{w}(n+1) = \boldsymbol{w}(n+1) - \boldsymbol{w}(n)$ is the adjustment vector and step-size is selected as $\mu = 1$ for simplicity. Adjustment of the filter coefficients can be visualized for $L = 3$ as in Figure 2.5. $\boldsymbol{w}(n+1)$ is obtained as the projection of $\boldsymbol{w}(n)$, $\boldsymbol{w}(n+1) = \boldsymbol{P}_{\Pi_n}\boldsymbol{w}(n)$, onto the hyperplane Π_n. Π_n is composed of the set of filter coefficients, $\widehat{\boldsymbol{w}}(n)$, that satisfies $y(n) = \boldsymbol{x}^T(n)\widehat{\boldsymbol{w}}(n)$ and $\boldsymbol{P}_{\Pi_n}$is the projection operator. Therefore, $\boldsymbol{w}(n+1)$ corresponds to the intersection point of Π_n and $\boldsymbol{x}(n)$. From Figure 2.5, it can also be concluded that performance of the algorithm can be improved by changing the direction of the update vector toward optimal point by manipulating the input vector. This idea is the main motivation of the development of the proportionate type algorithms.

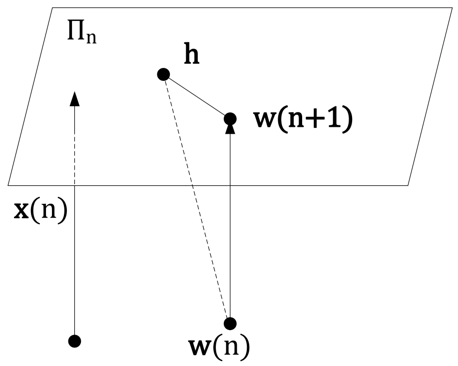

Figure 2.5 Geometrical representation of the update of NLMS

This approach can be directly extended to APA as shown in Figure 2.6. In this case, filter coefficients are not projected onto a single hyperplane but they are projected onto the intersection of many planes $\Pi_n \cap \Pi_{n-1} \cap \ldots \Pi_{n-M+1}$, to calculate $\boldsymbol{w}(n+1)$ where M is the projection order. In Figure 2.6, $\dot{\boldsymbol{w}}(n+1)$ represents the update of $\boldsymbol{w}(n)$ if NLMS update rule is applied. It is shown in [10] that, $\|\boldsymbol{h} - \boldsymbol{w}(n+1)\|_2 \leq \|\boldsymbol{h} - \dot{\boldsymbol{w}}(n+1)\|_2$ which means that better estimate is obtained by APA compared to NLMS estimate.

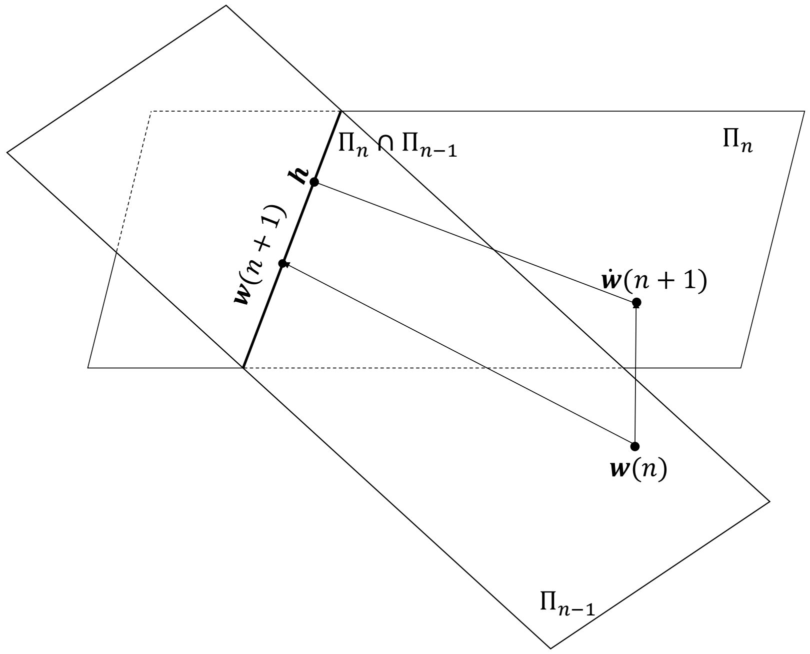

Figure 2.6 Geometrical representation of the update of APA

2.9. Evaluation Criteria for Adaptive Filtering

There are several criteria that are considered while evaluating the performances of adaptive filtering algorithms. In this thesis, two of them are accepted as the major criteria, which are normalized steady state misalignment and convergence rate. Normalized steady state misalignment, (msl), is a measure of the distance between estimated filter coefficients and the optimal filter coefficients and it is given as

$$msl(n) = 20 \log_{10} \frac{\|\boldsymbol{h} - \boldsymbol{w}(n)\|_2}{\|\boldsymbol{h}\|_2}. \tag{2.39}$$

Convergence rate is the measure of time, which is required to achieve steady-state. Consequently, major concern in this thesis study is to develop algorithms which achieve lower steady state misalignment with a higher convergence rate. There are other criteria which should be considered such as computational complexity and steady state mean square error, $\lim\limits_{n\to\infty} E\{|e(n)|^2\}$. Generally, computational complexity of an algorithm is measured by the number of arithmetic operations (additions, multiplications) required.

CHAPTER 3

ADAPTIVE FILTERS FOR SPARSE CHANNEL IDENTIFICATION

3.1. Introduction

In this chapter, firstly the concept of sparse impulse and a measure of sparseness are defined. Then, proportionate type adaptive filtering framework is established and fundamental proportionate type algorithms, PNLMS (PAPA) and IPNLMS (IPAPA) algorithms, are introduced. Later, other proportionate algorithms GC-IPNLMS (GC-IPAPA) and D-IPNLMS (D-IPAPA) which consider the dynamical behavior of the filter coefficients are presented. Then, performances of proportionate type and classical adaptive filtering algorithms are compared. Furthermore, these algorithms are also compared in terms of their computational complexities. Lastly, a mathematical basis for proportionate type algorithms is provided.

3.2. Sparse Impulse Response Concept and Measure of Sparseness

In this thesis, impulse responses of the systems are categorized into two classes based on the characteristics of the impulse responses. The first class is the sparse impulse responses in which magnitudes of many coefficients are zero or close to zero but a small number of coefficients have relatively larger magnitudes compared to these small coefficients. In the literature, small coefficients are referred to as "minor" coefficients and larger coefficients are referred to as "major" coefficients. The other class involves "dispersive" impulse responses for which many of the filter coefficients have significant magnitudes. In Figure 3.1, examples of sparse and dispersive responses are shown.

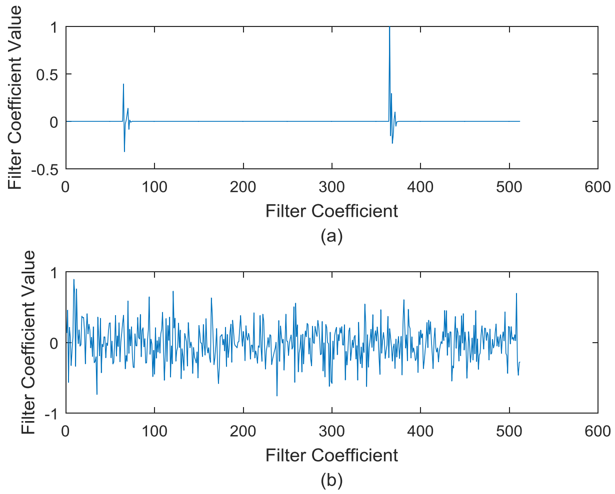

Figure 3.1 (a) Sparse and (b) Dispersive Impulse Responses

In order to place an impulse response in one of the classes, a mathematical measure for sparseness is needed. In [30], measure of sparseness is defined as

$$\xi \triangleq \frac{L}{L-\sqrt{L}}\left(1-\frac{\|h\|_1}{\sqrt{L}\|h\|_2}\right), \tag{3.1}$$

where $0 \le \xi \le 1$. As ξ increases, impulse response becomes sparser. In the extreme case of a single nonzero coefficient $\xi = 1$. On the contrary, if all of the coefficients have the same absolute value then $\xi = 0$. For instance, the sparseness of the channel shown in Figure 3.1 (a) is $\xi = 0.9297$ meaning that it is a highly sparse channel and that of in Figure 3.2 (b) is $\xi = 0.2268$ which corresponds to a highly dispersive channel. This study mainly focuses on impulse responses with ξ close to 1.

3.3. Proportionate Type Algorithms

Classical adaptive filtering algorithms assign the same step-size to all filter coefficients regardless of the characteristics of the unknown impulse response. However, in case of sparse channels, this leads to performance degradation. For instance, consider the case where all filter coefficients are initialized as zero such that minor coefficients are already close to their optimal values. However, at the beginning of the adaptation, these coefficients would significantly deviate from their optimal values due to the assignment of large step-sizes, which yields slow convergence of minor coefficients. In order to avoid this issue, smaller step-sizes should be assigned to minor coefficients. Therefore, proportionate type algorithms were proposed to assign coefficient specific step-sizes to filter coefficients so that faster convergence is achieved by taking the sparse nature of the unknown channel into account.

In order to assign individual step-sizes, a diagonal proportionate matrix is defined whose diagonal elements are the proportionate factors that contain the information about the structure of the impulse response. Let $\mathbf{\Omega}(n)$ be the $L \times L$ diagonal proportionate matrix composed of proportionate factors,

$$\mathbf{\Omega}(n) = diag([\omega_0(n), \omega_1(n), \ldots, \omega_{L-1}(n)]), \tag{3.2}$$

where $\omega_k(n)$ is the proportionate factor of the k^{th} coefficient. General form of the update equation of proportionate type NLMS algorithms is given as follows [11],

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu \frac{\mathbf{\Omega}(n)\boldsymbol{x}(n)e(n)}{\boldsymbol{x}^T(n)\mathbf{\Omega}(n)\boldsymbol{x}(n) + \delta}. \tag{3.3}$$

This approach can be extended to APA as [12],

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu\mathbf{\Omega}(n)\boldsymbol{X}(n)(\boldsymbol{X}^T(n)\mathbf{\Omega}(n)\boldsymbol{X}(n) + \delta\boldsymbol{I})^{-1}\boldsymbol{e}(n). \tag{3.4}$$

3.4. Proportionate NLMS (PNLMS) and Proportionate APA (PAPA)

PNLMS algorithm [11] is a milestone in the context of the sparse channel identification. It was proposed to achieve faster convergence compared to NLMS algorithm which has been commonly used in echo cancellation applications. PNLMS achieves faster convergence by assigning coefficient specific step-sizes to each coefficient independently. The elements of the proportionate matrix in PNLMS algorithm are obtained as follows,

$$e(n) = y(k) - \boldsymbol{w}^T(n)\boldsymbol{x}(n), \tag{3.5}$$

$$l_\infty(n) = \max\{|w_0(n)|, |w_1(n)|, \dots, |w_{L-1}(n)|\}, \tag{3.6}$$

$$l'_\infty(n) = \max\{\epsilon, l_\infty(n)\}, \tag{3.7}$$

$$\gamma_k(n) = \max\{\rho l'_\infty(n), |w_k(n)|\}, \tag{3.8}$$

$$\omega_k(n) = \frac{\gamma_k(n),}{\sum_{i=0}^{L-1} \gamma_i(n)}, \tag{3.9}$$

then the update equation becomes,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu \frac{\boldsymbol{\Omega}(n)\boldsymbol{x}(n)e(n)}{\boldsymbol{x}^T(n)\boldsymbol{\Omega}(n)\boldsymbol{x}(n) + \delta}. \tag{3.10}$$

In PNLMS, “update energy” is distributed over the filter coefficients by $\boldsymbol{\Omega}$ matrix and the proportionate factors, ω_k, represent the gain distributors. It is clear from (3.9) that proportionate factors, are roughly proportional to the instantaneous magnitudes of the filter coefficients. Therefore, minor coefficients receive relatively smaller step-sizes than those of major coefficients. Assignment of smaller step-sizes prevents deviation of the minor coefficients from their optimal values which results in faster convergence. In PNLMS algorithm, an individual step-size assigned to a minor coefficient is determined by the parameter ρ, which affects the overall distribution of the update energy since summation of the proportionate factors is equal to 1. Consequently, if ρ is selected to be too large then major coefficients cannot receive adequate update energy. Otherwise, if it is selected to be too small then any estimation error in minor coefficients may have significant effects on the convergence. Therefore, choice of ρ is an important issue regarding the performance of PNLMS algorithm. As a rule of thumb, it is selected to be $5/L$ [13].

It is shown in [11] that steady state misalignment of the PNLMS is the same as that of NLMS. This is an important result regarding the steady-state performance of PNLMS since it achieves faster convergence while having the same steady-state error.

In order to comprehend the logic behind the proportionate approach, it is better to observe the evolutions of the filter coefficients. For this purpose, in Figure 3.2 evolutions of adaptive filters for a 2 tap sparse impulse response, $\boldsymbol{h} = [1\ 0]^T$, are shown for NLMS and PNLMS algorithms. It can be observed that minor coefficient deviates from their optimal values if NLMS rule is applied even if its initial value is optimal. On the other hand, coefficients follow a more direct path in case of PNLMS since by assigning smaller step-sizes to minor coefficient deviation from the optimal value is prevented. Another comment about the convergence pattern of PNLMS may be its resemblance to Newton’s method since filter coefficients move toward their optimal value on a more direct path.

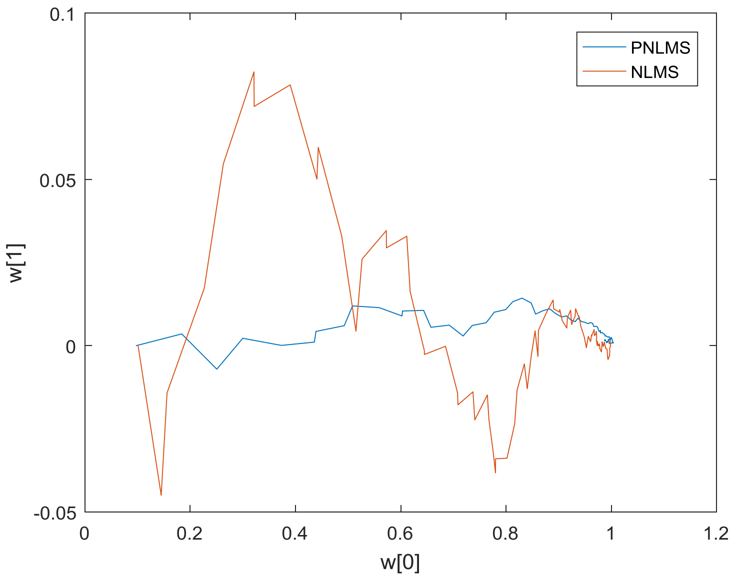

Figure 3.2 Evolutions of the filter coefficients for NLMS and PNLMS algorithms

This idea can be extended to its affine projection counterpart, which is called Proportionate APA (PAPA). Update equation of PAPA is given as

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu\boldsymbol{\Omega}(n)\boldsymbol{X}(n)(\boldsymbol{X}^T(n)\boldsymbol{\Omega}(n)\boldsymbol{X}(n) + \delta\boldsymbol{I})^{-1}\boldsymbol{e}(n). \quad (3.11)$$

PNLMS and PAPA provide fast convergence for only sparse impulse responses; on the other hand, for dispersive channels their performances significantly degrade. Even NLMS and APA outperform PNLMS and PAPA in case of dispersive channel identification. Therefore, methods, which provide better performance when the impulse response is dispersive, have been proposed.

3.5. Improved PNLMS (IPNLMS) and Improved PAPA (IPAPA)

Introduction of PNLMS algorithm formed a basis for new algorithms, which further improve the performance in case of sparse channels. IPNLMS is one of the widely employed approaches regarding the sparse channel identification. IPNLMS algorithm not only improves the convergence rate but also solves the performance degradation problem for dispersive channels. IPNLMS offers a smoother way to calculate proportionate factors so that negative effects of the estimation errors are eliminated. A proportionate factor of IPNLMS is calculated as

$$\omega_k(n) = \frac{1-\alpha}{2L} + (1+\alpha)\frac{|w_k(n)|}{2\sum_{i=0}^{L-1} w_i(n) + \epsilon} , \quad (3.12)$$

where $-1 < \alpha < 1$ is control parameter. For α values close to 1, IPNLMS algorithm behaves like PNLMS algorithm. If $\alpha = -1$ then IPNLMS becomes NLMS algorithm. The first term in (3.12) compensates the error made in the estimation of the current filter coefficients and the second term corresponds to proportionality of the IPNLMS which exploits the sparsity of the impulse response. By its hybrid approach, IPNLMS provides faster convergence for both sparse and dispersive impulse responses. Then, by inserting these proportionate factors into update equation IPNLMS can be obtained as

$$\boldsymbol{w}(n+1)=\boldsymbol{w}(n)+\mu\frac{\boldsymbol{\Omega}(n)\boldsymbol{x}(n)e(n)}{\boldsymbol{x}^{T}(n)\boldsymbol{\Omega}(n)\boldsymbol{x}(n)+\delta}\ . \tag{3.13}$$

Similarly, the hybrid approach of IPNLMS can be applied to IPAPA as [13],

$$\boldsymbol{w}(n+1)=\boldsymbol{w}(n)+\mu\boldsymbol{\Omega}(n)\boldsymbol{X}(n)(\boldsymbol{X}^{T}(n)\boldsymbol{\Omega}(n)\boldsymbol{X}(n)+\delta\boldsymbol{I})^{-1}\boldsymbol{e}(n). \tag{3.14}$$

3.6. Gradient Controlled IPNLMS and IPAPA

Both PNLMS (PAPA) and IPNLMS (IPAPA) algorithms only consider magnitudes of the instantaneous filter coefficients while assigning proportionate factors. These approaches provide fast initial convergence. However, overall convergence performance degrades when the estimated filter coefficients get close to the optimal point since these algorithms lead to random variations around the optimal point by assigning large step-sizes to major coefficients, which are close to optimal values. Therefore, it is necessary to modify the assignment of the coefficient specific step-sizes such that only coefficients, which are far away from their optimal values, receive larger update energy. Since optimal filter coefficients are not known, it is an impossible task to find the difference between current and optimal filter coefficients. Hence, [19] achieves this by assigning step-sizes proportional to the gradient vector which contains the information of the difference between optimal and the current filter coefficient. In order to justify this statement, firstly, consider the error vector,

$$\boldsymbol{e}(n)=\boldsymbol{X}^{T}(n)\boldsymbol{h}-\boldsymbol{X}^{T}(n)\boldsymbol{w}(n-1)+\boldsymbol{v}(n). \tag{3.15}$$

If noise is assumed to be sufficiently small, then error vector approximately becomes,

$$\boldsymbol{e}(n)\approx\boldsymbol{X}^{T}\widetilde{\boldsymbol{h}}(n-1), \tag{3.16}$$

where $\widetilde{\boldsymbol{h}}(n)=\boldsymbol{h}-\boldsymbol{w}(n)$ is the weight error vector. [19] states that the coefficient error vector can be utilized by considering the gradient vector,

$$\boldsymbol{\nabla}(n) = -\boldsymbol{X}(n)[\boldsymbol{X}^T(n)\boldsymbol{X}(n) + \delta\boldsymbol{I}]^{-1}\boldsymbol{e}(n), \tag{3.17}$$

since it inherently involves coefficient error vector. This can be observed by inserting (3.16) to (3.17),

$$\boldsymbol{\nabla}(n) = -\boldsymbol{X}(n)[\boldsymbol{X}^T(n)\boldsymbol{X}(n) + \delta\boldsymbol{I}]^{-1}\boldsymbol{X}^T\widetilde{\boldsymbol{h}}(n-1), \tag{3.18}$$

where $\boldsymbol{X}(n)[\boldsymbol{X}^T(n)\boldsymbol{X}(n) + \epsilon\boldsymbol{I}]^{-1}\boldsymbol{X}^T$ can be considered as the projection matrix onto the range space of $\boldsymbol{X}(n)$, $\mathbb{R}[\boldsymbol{X}(n)]$. Therefore, according to (3.18), $\boldsymbol{\nabla}(n)$ is the projection of $\widetilde{\boldsymbol{h}}(n-1)$ onto $\mathbb{R}[\boldsymbol{X}(n)]$. Hence, instead of directly using the weight error vector, its projection onto $\mathbb{R}[\boldsymbol{X}(n)]$, $\boldsymbol{\nabla}(n)$, is employed while calculating the proportionate factors. In order to avoid overshooting, smoothed version of the gradient vector is used,

$$\overline{\boldsymbol{\nabla}}(n) = \beta\overline{\boldsymbol{\nabla}}(n-1) - (1-\beta)\boldsymbol{X}(n)[\boldsymbol{X}^{\boldsymbol{T}}(n)\boldsymbol{X}(n) + \delta\boldsymbol{I}]^{-\boldsymbol{1}}\boldsymbol{e}(n), \tag{3.19}$$

where $0 < \beta < 1$. Consequently, proportionate factors are expressed as

$$\omega_l(n) = \frac{(1-\alpha)}{2L} + \frac{(1+\alpha)|\overline{\nabla}_l(n-1)|}{2\sum_{i=0}^{L-1}|\overline{\nabla}_i(n-1)| + \epsilon}\ , \tag{3.20}$$

where $\overline{\nabla}_l$ is the l^{th} element of the smoothed gradient vector. Similar to IPAPA, update equation of GC-IPAPA is given as

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu\boldsymbol{\Omega}(n)\boldsymbol{X}(n)(\boldsymbol{X}^T(n)\boldsymbol{\Omega}(n)\boldsymbol{X}(n) + \delta\boldsymbol{I})^{-1}\boldsymbol{e}(n). \tag{3.21}$$

In addition, GC-IPNLMS is the special case of GC-IPAPA where $M = 1$,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu\frac{\boldsymbol{\Omega}(n)\boldsymbol{x}(n)e(n)}{\boldsymbol{x}^T(n)\boldsymbol{\Omega}(n)\boldsymbol{x}(n) + \delta}\ . \tag{3.22}$$

In order to understand the improvement introduced by the GC-IPAPA, evolution of the proportionate factors should be investigated. Figure 3.3 shows the behavior of a specific proportionate factor for both IPAPA and GC-IPAPA. It can be seen that, proportionate factor of IPAPA increases up to a point and stays there for the rest of the adaptation. However, proportionate factor for GC-IPAPA increases until it converges to its optimal value. After reaching the optimal point, magnitude of the proportionate factor starts to decrease. This behavior prevents the fluctuations around the optimal point; therefore, GC-IPAPA provides smoother and faster convergence.

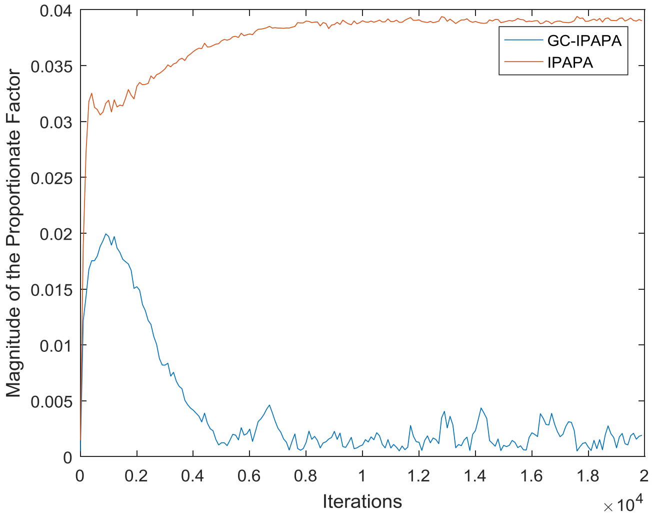

Figure 3.3 Proportionate Factor Comparisons for IPAPA and GC-IPAPA

Although GC-IPAPA provides faster convergence, it requires an extra matrix inversion while calculating the gradient vector. Therefore, the main drawback of the GC-IPAPA is its computational complexity. Especially, as the projection order increases computational complexity becomes a major problem.

3.7. Difference Based IPNLMS (D-IPAPA) and IPAPA (D-IPAPA)

D-IPAPA is another method proposed to combat the performance degradation problem after the initial stage of the adaptation [20]. D-IPAPA divides the adaptation into two stages, which are the initial stage and the second stage. At the initial stage, major coefficients quickly reach their optimal values. At the second stage, convergence of the minor coefficients begins; therefore, proportionate factors should be modified such that larger step-sizes are assigned to the minor coefficients. According to [20], PNLMS and IPNLMS have performance degradation problem since they do not have the capability to adjust their proportionate factors in the second stage. In order to modify the proportionate factors appropriately, D-IPAPA handles the adaptation block-by-block, assuming that convergence behavior of a certain filter coefficient does not change significantly within a block period. Therefore, during a block period, variation of a proportionate factor is assumed to be relatively slow. At each block period, a new set of proportionate factors are calculated by considering current values of the filter coefficients at the beginning of each block as the initial value during that block period. Consider the k^{th} block, it is assumed that adaptation has the initial values of $\boldsymbol{w}(kP)$, where P is the block period. Then proportionate factors are obtained as

$$\omega_l(kP+i) = \frac{(1-\alpha)}{2L} + \frac{(1+\alpha)|w_l(kP+i) - w_l(kP)|}{2\sum_{j=0}^{L-1}\left|w_j(kP+i) - w_j(kP)\right| + \epsilon} \; . \qquad (3.23)$$
$$i = 0,1,\dots,P-1$$

It is clear that if all filter coefficients are initialized as zero, at the initial stage D-IPNLMS and IPNLMS are identical. When the major coefficients converge, the difference, $w_l(kP+i) - w_l(kP)$**,** becomes negligible. Therefore, the minor coefficients, which are far away from their optimal value, may have the opportunity to receive necessary step-size so that their convergence is boosted.

3.8. Simulation Results of Proportionate Type Algorithms

In this section, misalignment performances of proportionate type algorithms are investigated. Length of the unknown impulse response is set to 512 for all cases and it is either a highly sparse one with $\xi = 0.9297$ or relatively dispersive one with $\xi = 0.4647$. Input of the unknown system is modeled as a white Gaussian signal and as a noise component an independent white Gaussian signal is added to the output of the system to have $30\ dB$ SNR. Step-size, μ, is set to 0.2, control variable, $\alpha,$ is set to 0 and decaying constant for GC-IPAPA is set to $\beta = 0.999$ and ρ is set to 0.01 for PNLMS. Results are obtained by ensemble averaging of 10 independent trials.

Firstly, improvements introduced by the proportionate approach is observed by comparing PNLMS and IPNLMS algorithms with NLMS algorithm for a sparse channel. Results are shown in Figure 3.4 and superiority of PNLMS and IPNLMS over NLMS is apparent. However, as can be seen in Figure 3.5 PNLMS algorithm cannot hold the superiority for a dispersive channel with $\xi = 0.5138$; on the other hand, IPNLMS still outperforms NLMS algorithm. Consequently, IPNLMS seems to be more appealing choice for the system identification applications. Therefore, in the remainder of this thesis IPNLMS is regarded as the reference method for the performance comparisons.

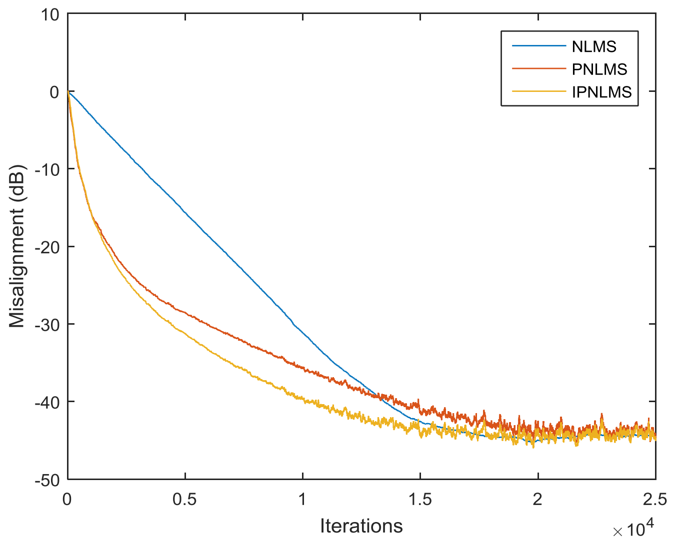

Figure 3.4 Performance comparisons of NLMS, PNLMS and IPNLMS algorithms for the sparse channel with $\xi = 0.9297$

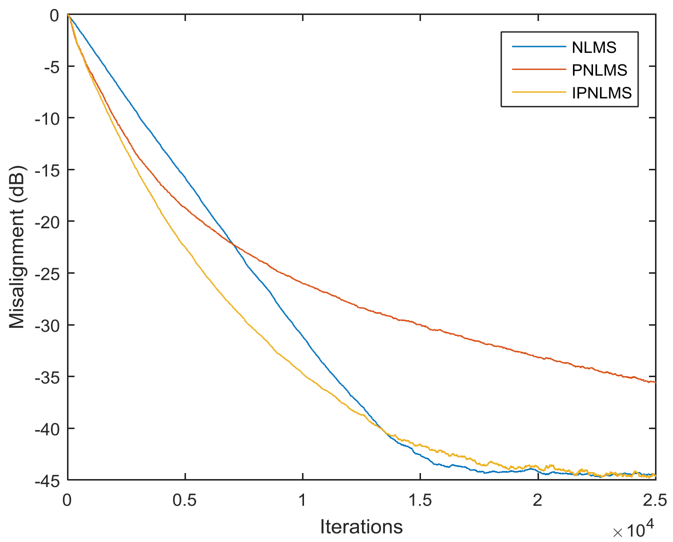

Figure 3.5 Performance comparisons of NLMS, PNLMS and IPNLMS algorithms for the dispersive channel with $\xi = 4947$

After defining the IPNLMS as the reference approach, it is compared with D-IPNLMS and GC-IPNLMS algorithms. In this case, a channel with a sparseness of $\xi = 0.8036$ is identified and all other parameters are kept same. Figure 3.6 shows the misalignment curves of these algorithms. It can be seen that both D-IPNLMS and GC-IPNLMS outperforms IPNLMS algorithm.

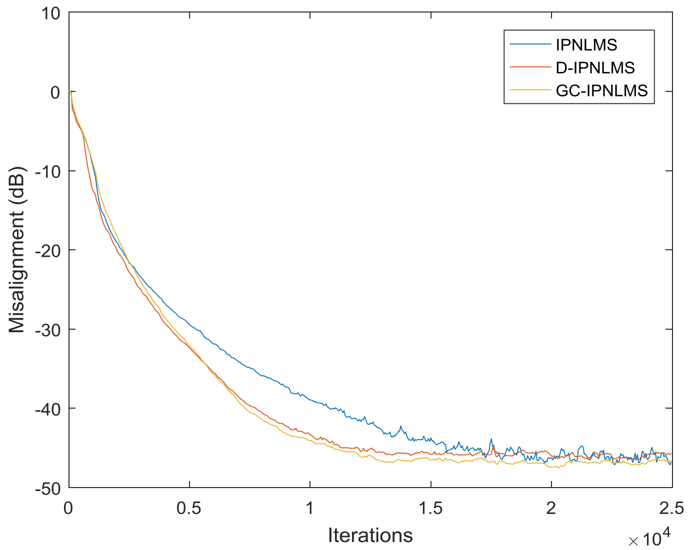

Figure 3.6 Performance comparison of IPNLMS, D-IPNLMS and GC-IPNLMS

Moreover, results for algorithms with higher order projection are obtained in order to compare IPAPA, D-IPAPA and GC-IPAPA for $M = 2$ and superior performance of D-IPAPA and GC-IPAPA can be observed from Figure 3.7.

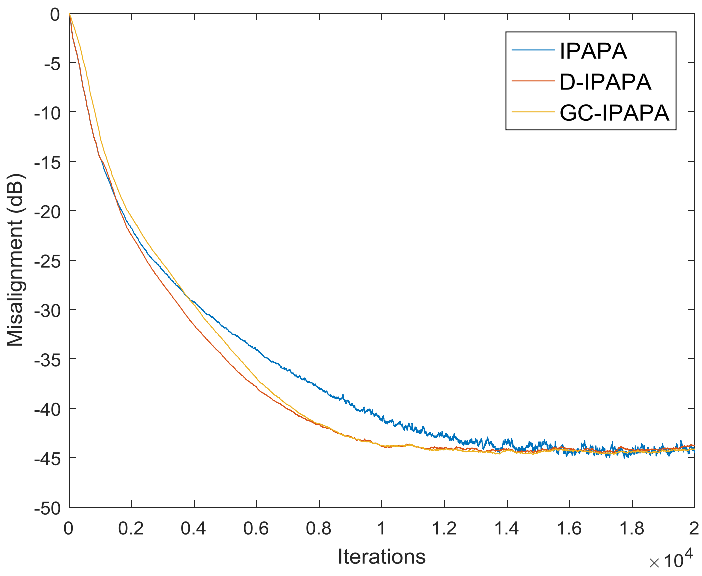

Figure 3.7 Performance comparison of IPAPA, D-IPAPA and GC-IPAPA

3.9. Computational Complexities of the Proportionate Type Algorithms

In this section computational complexities of the proportionate type algorithms are compared. For this purpose, the number of operations required by the algorithms, PAPA, IPAPA, D-IPAPA and GC-IPAPA are given in Table 3.1. In addition, the number of operations required for APA is also given in order to observe the extra complexity brought by the proportionate approach. Computational complexities of APA, PAPA, IPAPA and D-IPAPA are close to each other. However, GC-IPAPA requires almost twice as much operations as the other algorithms. Therefore, it can be concluded that proportionate type algorithms do not bring excessive computational burden except GC-IPAPA.

Table 3. 1 Number of operations required for adaptive filtering algorithms

Method	Summations	Multiplications	Comparisons	Memory
APA	$(M^2+M-1)L$	$(M^2+M+1)L+M^2$	0	0
PAPA	$(M^2+M-1)L$	$(M^2+M+2)L+M^2+2$	2L	L
IPAPA	$(M^2+M+1)L$	$(M^2+M+2)L+M^2$	0	L
D-IPAPA	$(M^2+M+2)L$	$(M^2+M+2)L+M^2$	0	L
GC-IPAPA	$(2M^2+2M+2)L$	$(2M^2+2M+2)L+2M^2$	0	2L

3.10. A Mathematical Derivation of Proportionate Type Algorithms

Initial development of proportionate type algorithms took place in an intuitive manner. Although proportionate approach is experimentally proved to be useful, it is necessary to provide a mathematical foundation. In [17], proportionate approach is derived from a basis pursuit perspective by employing l_1 norm optimization. First step of the derivation in [17] is to show that estimated coefficient vectors in NLMS and APA can be expressed as the sum of two orthogonal vectors. One of these vectors is the solution of an l_2 optimization problem, while the other one is considered as an initialization vector. Consequently, by changing the optimization problem to l_1 norm criterion, proportionate type algorithms can be obtained.

Consider the NLMS update equation without regularization constant, δ, and $\mu = 1$,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \frac{\boldsymbol{x}(n)e(n)}{\boldsymbol{x}^T(n)\boldsymbol{x}(n)}, \tag{3.24}$$

where $e(n) = y(n) - \boldsymbol{x}^T(n)\boldsymbol{w}(n) = y(n) - \hat{y}(n)$, $\hat{y}(n)$ is the estimate of $y(n)$. Then, (3.24) can be rewritten as

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \frac{\boldsymbol{x}(n)}{\boldsymbol{x}^T(n)\boldsymbol{x}(n)}[y(n) - \boldsymbol{x}^T(n)\boldsymbol{w}(n)], \tag{3.25}$$

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) - \frac{\boldsymbol{x}(n)\boldsymbol{x}^T(n)}{\boldsymbol{x}^T(n)\boldsymbol{x}(n)}\boldsymbol{w}(n) + \frac{\boldsymbol{x}(n)y(n)}{\boldsymbol{x}^T(n)\boldsymbol{x}(n)}, \tag{3.26}$$

$$\boldsymbol{w}(n+1) = \left[\boldsymbol{I} - \frac{\boldsymbol{x}(n)\boldsymbol{x}^T(n)}{\boldsymbol{x}^T(n)\boldsymbol{x}(n)}\right]\boldsymbol{w}(n) + \frac{\boldsymbol{x}(n)y(n)}{\boldsymbol{x}^T(n)\boldsymbol{x}(n)}, \tag{3.27}$$

$$\boldsymbol{w}(n+1) = \wp(n)\boldsymbol{w}(n) + \tilde{\boldsymbol{w}}(n), \tag{3.28}$$

where,

$$\wp(n) = \boldsymbol{I} - \frac{\boldsymbol{x}(n)\boldsymbol{x}^T(n)}{\boldsymbol{x}^T(n)\boldsymbol{x}(n)}, \tag{3.29}$$

and

$$\tilde{\boldsymbol{w}}(n) = \frac{\boldsymbol{x}(n)y(n)}{\boldsymbol{x}^T(n)\boldsymbol{x}(n)}. \tag{3.30}$$

In (3.28), $\tilde{\boldsymbol{w}}(n)$ is related to solution of the l_2 norm optimization and $\wp(n)\boldsymbol{w}(n)$ corresponds to the initialization component. It can be shown that constituent vectors of $\boldsymbol{w}(n+1)$ are orthogonal such that

$$[\wp(n)\boldsymbol{w}(n)]^T\tilde{\boldsymbol{w}}(n) = 0. \tag{3.31}$$

Hence, $\tilde{\boldsymbol{w}}(n)$ is in the null space of $\wp(n)$, $\tilde{\boldsymbol{w}}(n) \in \aleph(\wp(n))$,

$$\wp(n)\tilde{\boldsymbol{w}}(n) = \boldsymbol{0}. \tag{3.32}$$

Moreover, $\boldsymbol{w}(n)$ can be expressed as the sum of two vectors which are in the range and null spaces of $\wp(n)$,

$$\boldsymbol{w}(n) = \boldsymbol{w}_{\parallel}(n) + \boldsymbol{w}_{\perp}(n), \tag{3.33}$$

where

$$\wp(n)\boldsymbol{w}(n) = \boldsymbol{w}_{\parallel}(n), \tag{3.34}$$

$$\wp(n)\boldsymbol{w}_{\perp}(n) = \boldsymbol{0}. \tag{3.35}$$

Therefore, (3.28) becomes,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}_{\parallel}(n) + \widetilde{\boldsymbol{w}}(n). \tag{3.36}$$

From (3.31), it is clear that

$$\boldsymbol{w}_{\parallel}(n)^T \widetilde{\boldsymbol{w}}(n) = 0, \tag{3.37}$$

and the desired signal can be obtained as

$$\boldsymbol{x}^T(n)\boldsymbol{w}(n+1) = \boldsymbol{x}^T(n)\widetilde{\boldsymbol{w}}(n) = d(n)\,. \tag{3.38}$$

Furthermore, $\boldsymbol{w}_{\perp}(n)$ can be obtained in a straightforward manner as,

$$\boldsymbol{w}_{\perp}(n) = \boldsymbol{w}(n) - \wp(n)\boldsymbol{w}(n), \tag{3.39}$$

then by inserting (3.29) to (3.39), one can obtain,

$$\boldsymbol{w}_{\perp}(n) = \frac{\boldsymbol{x}(n)\hat{y}(n)}{\boldsymbol{x}^T(n)\boldsymbol{x}(n)}, \tag{3.40}$$

it is important to note that $\boldsymbol{w}_{\perp}(n)$ is also in the null space of $\wp(n)$. By using the definition of the error signal, it can be written as

$$e(n) = \boldsymbol{x}^T(n)\widetilde{\boldsymbol{w}}(n) - \boldsymbol{x}^T(n)\boldsymbol{w}_{\perp}(n) = \boldsymbol{x}^T(n)[\widetilde{\boldsymbol{w}}(n) - \boldsymbol{w}_{\perp}(n)], \tag{3.41}$$

which is independent of $\boldsymbol{w}_{\parallel}(n)^T$. Thus, [17] states that $\widetilde{\boldsymbol{w}}(n)$ is the solution of l_2 norm optimization problem which gives the overall update expression of NLMS,

$$\min_{\widetilde{\boldsymbol{w}}(n)} \|\widetilde{\boldsymbol{w}}(n)\|_2^2 \ subject\ to\ y(n) = \boldsymbol{x}^T(n)\widetilde{\boldsymbol{w}}(n). \tag{3.42}$$

So far, a basis for the derivation of the proportionate approach is constructed. By using l_1 norm criterion instead of l_2 norm criterion in (3.42), following optimization problem is obtained to get proportionate type algorithms,

$$\min_{\widetilde{\boldsymbol{w}}(n)} \|\widetilde{\boldsymbol{w}}(n)\|_1 \ subject\ to\ y(n) = \boldsymbol{x}^T(n)\widetilde{\boldsymbol{w}}(n), \tag{3.43}$$

and it can be solved by using Lagrange multipliers method,

$$J = \|\widetilde{\boldsymbol{w}}(n)\|_1 + \lambda[y(n) - \boldsymbol{x}^T(n)\widetilde{\boldsymbol{w}}(n)]. \tag{3.44}$$

By taking the derivative of J with respect to $\widetilde{\boldsymbol{w}}(n)$ and equating it to zero, one can get,

$$\frac{\partial J}{\partial \widetilde{\boldsymbol{w}}(n)} = sgn\big(\widetilde{\boldsymbol{w}}(n)\big) - \lambda \boldsymbol{x}(n) = 0. \tag{3.45}$$

where, $sgn()$ is the sign function, which gives signs of the elements in the vector. By multiplying (3.45) with matrix $\widetilde{\boldsymbol{\Omega}}(n) = diag(|\widetilde{w}_0(n)|, |\widetilde{w}_1(n)|, \ldots, |\widetilde{w}_{L-1}(n)|)$, following equality is obtained,

$$\widetilde{\boldsymbol{\Omega}}(n) sgn\big(\widetilde{\boldsymbol{w}}(n)\big) = \lambda \widetilde{\boldsymbol{\Omega}}(n)\boldsymbol{x}(n), \tag{3.46}$$

$$\widetilde{\boldsymbol{w}}(n) = \lambda \widetilde{\boldsymbol{\Omega}}(n)\boldsymbol{x}(n). \tag{3.47}$$

By inserting (3.47) in the constraint in (3.43), one can obtain λ as

$$d(n) = \lambda \boldsymbol{x}^T(n)\widetilde{\boldsymbol{\Omega}}(n)\boldsymbol{x}(n), \tag{3.48}$$

$$\lambda = \frac{d(n)}{\boldsymbol{x}^T(n)\widetilde{\boldsymbol{\Omega}}(n)\boldsymbol{x}(n)}. \tag{3.49}$$

Consequently, $\widetilde{\boldsymbol{w}}(n)$ becomes,

$$\widetilde{\boldsymbol{w}}(n) = \frac{\widetilde{\boldsymbol{\Omega}}(n)\boldsymbol{x}(n)d(n)}{\boldsymbol{x}^T(n)\widetilde{\boldsymbol{\Omega}}(n)\boldsymbol{x}(n)}, \tag{3.50}$$

since $\widetilde{\boldsymbol{\Omega}}(n)$ is not obtained at that moment, it is reasonable to replace it with $\boldsymbol{\Omega}(n)$, where $\boldsymbol{\Omega}(n) = diag(|w_0(n)|, |w_1(n)|, \dots, |w_{L-1}(n)|)$. Hence, (3.50) becomes,

$$\widetilde{\boldsymbol{w}}(n) = \frac{\boldsymbol{\Omega}(n)\boldsymbol{x}(n)d(n)}{\boldsymbol{x}^T(n)\boldsymbol{\Omega}(n)\boldsymbol{x}(n)}. \tag{3.51}$$

Next step is to form the projection matrix $\wp(n)$ as

$$\wp(n) = \boldsymbol{I} - \frac{\boldsymbol{\Omega}(n)\boldsymbol{x}(n)\boldsymbol{x}^T(n)}{\boldsymbol{x}^T(n)\boldsymbol{\Omega}(n)\boldsymbol{x}(n)}. \tag{3.52}$$

By combining these results update equation of PNLMS algorithm can be written as

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \frac{\boldsymbol{\Omega}(n)\boldsymbol{x}(n)e(n)}{\boldsymbol{x}^T(n)\boldsymbol{\Omega}(n)\boldsymbol{x}(n)}. \tag{3.53}$$

Other variants of the proportionate type algorithms can be obtained by using this framework. This framework also shows that proportionate and non-proportionate type algorithms are not irrelevant since they share similar ideas in their nature.

CHAPTER 4

SIGN ALGORITHMS: ROBUSTNESS AGAINST IMPULSIVE INTERFERENCE

4.1. Introduction

Development of many adaptive filtering algorithms including NLMS, APA etc., is based on the Gaussian noise assumption [24]. Therefore, performance degradation is inevitable if the noise in the environment is not Gaussian. Impulsive interference is a non-Gaussian noise, which extremely affects the performance of these algorithms since they are developed based on the mean-square-error cost function which uses square of the error signal. In order to overcome this problem, lower order normed cost functions are considered instead of the mean-squared-error. Stemming from this idea, sign algorithms are proposed which are based on the l_1 norm of the error signal since l_1 norm based optimization provides robustness against impulsive interference. In this chapter, firstly, some well-known sign algorithms, Normalized Sign Algorithm (NSA) [24] and Affine Projection Sign Algorithm (APSA) [21], are introduced. Later, proportionate approach is extended to APSA [25] for sparse channels which are afflicted by impulsive noise. Lastly, simulation results related to sign algorithms are presented.

4.2. Motivation for Sign Algorithms

In Section 2.2, it is stated that background noise $v(n)$ is generally modeled as WGN but for some cases it can be modeled as the combination of WGN and BG process to integrate impulsive interference into the system as

$$v(n) = t(n) + z(n), \tag{4.1}$$

where, $t(n)$ is a WGN and $z(n)$ is a BG process.

Conventional algorithms have been proposed under the assumption of $z(n) = 0$. Therefore, these algorithms fail when $z(n) \neq 0$ due to MSE criterion, $J = E\{|e(n)|^2\}$ since, in the presence of impulsive noise, magnitude of the error signal changes drastically which is directly used in the update equations (2.31) and (2.33). Therefore, resultant algorithms are significantly affected by the impulsive noise. As a result, sign algorithms were proposed to eliminate the effect of the impulsive noise.

4.3. Normalized Sign Algorithm (NSA)

NSA is one of the simplest forms of the sign algorithms, which can be interpreted as the sign extension of NLMS algorithm. NSA is obtained by employing a lower order normed cost function defined in (4.2) to provide robustness against impulsive noise,

$$J = E\{|e(n)|\}. \tag{4.2}$$

By using this cost function, [31] proposes the Sign Algorithm for cases in which non-Gaussian noise is present. Furthermore, stemming from Sign Algorithm, NSA has been proposed in [24],

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu \frac{\boldsymbol{x}(n)}{\|\boldsymbol{x}(n)\|_1 + \delta} sgn\big(e(n)\big) \tag{4.3}$$

According to (4.3), update equation is independent of the magnitude of the error signal, it only depends on the sign of the error signal. Hence, the effect of the impulsive noise is limited to $+1$ and -1.

NSA provides robustness against impulsive interference; however, it suffers from slow convergence especially for the colored input case. For instance, for Gaussian noise case, LMS algorithm significantly outperforms NSA. Therefore, Affine

Projection Sign Algorithm (APSA) is proposed as a solution of the slow convergence problem.

4.4. Affine Projection Sign Algorithm (APSA)

NSA uses only the instantaneous input and error signal for the adaptation. Therefore, multiple instances of these signal can be used in order to improve the performance of the algorithm. APSA is proposed such that affine projection idea is extended to sign algorithm framework. Consequently, APSA [21] is obtained as the solution of the following optimization problem,

$$\begin{aligned} &\min_{\boldsymbol{w}(n+1)} \|\boldsymbol{y}(n) - \boldsymbol{X}^T(n)\boldsymbol{w}(n+1)\|_1 \\ &subject\ to \quad \|\boldsymbol{w}(n+1) - \boldsymbol{w}(n)\|_2^2 < \psi^2 , \end{aligned} \tag{4.4}$$

where ψ is the disturbance constraint which keeps the variation in the filter coefficients sufficiently small such that filter coefficients do not diverge. ψ should be selected as small as possible. By using Lagrange multipliers, cost function is obtained as

$$\begin{aligned} J &= \|\boldsymbol{y}(n) - \boldsymbol{X}^T(n)\boldsymbol{w}(n+1)\|_1 + \lambda[\|\boldsymbol{w}(n+1) - \boldsymbol{w}(n)\|_2^2 - \psi^2] \\ &= \left\|\boldsymbol{e}_p(n)\right\|_1 + \lambda[\|\boldsymbol{w}(n+1) - \boldsymbol{w}(n)\|_2^2 - \psi^2], \end{aligned} \tag{4.5}$$

where $\boldsymbol{e}_p(n)$ is the a posteriori error vector. By taking the derivative of J with respect to $\boldsymbol{w}(n+1)$, one gets,

$$\frac{\partial J}{\partial \boldsymbol{w}(n+1)} = -\boldsymbol{X}(n)sgn\left(\boldsymbol{e}_p(n)\right) + 2\lambda[\boldsymbol{w}(n+1) - \boldsymbol{w}(n)], \tag{4.6}$$

and by equating it to zero, following update equation is obtained,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \frac{1}{2\lambda}\boldsymbol{X}(n)sgn\left(\boldsymbol{e}_p(n)\right). \tag{4.7}$$

In order to obtain λ, substitute (4.7) into the constraint in (4.4) to obtain

$$\frac{1}{2\lambda} = \frac{\psi}{\sqrt{sgn\left(\boldsymbol{e}_p^T(n)\right)\boldsymbol{X}(n)\boldsymbol{X}^T(n)sgn\left(\boldsymbol{e}_p(n)\right)}}, \tag{4.8}$$

and update equation becomes,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \frac{\psi\boldsymbol{X}(n)sgn\left(\boldsymbol{e}_p(n)\right)}{\sqrt{sgn\left(\boldsymbol{e}_p^T(n)\right)\boldsymbol{X}(n)\boldsymbol{X}^T(n)sgn\left(\boldsymbol{e}_p(n)\right)}}. \tag{4.9}$$

Note that $\boldsymbol{e}_p(n)$ is a function of $\boldsymbol{w}(n+1)$ which is not present at the moment. Therefore, $\boldsymbol{e}_p(n)$ is replaced with the priori error vector $\boldsymbol{e}(n)$. In order to satisfy the stability of the algorithm minimum disturbance parameter should be much smaller than 1 which can be considered as the step-size of the algorithm; hence, ψ is replaced with step-size parameter μ. Final form of the update equation becomes,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \frac{\mu\boldsymbol{x}_s(n)}{\sqrt{\boldsymbol{x}_s^T(n)\boldsymbol{x}_s(n) + \delta}}, \tag{4.10}$$

where $\boldsymbol{x}_s(n) = \boldsymbol{X}(n)sgn\big(\boldsymbol{e}(n)\big)$ and step-size should satisfy $0 < \mu \ll 1$ due to minimum disturbance constraint.

Most remarkable property of this algorithm is that there is no need of matrix inversion unlike other APA's. Consequently, computational complexity is significantly reduced.

4.5. Proportionate Type Sign Algorithms

APSA provides fast convergence and robust adaptation in case of existence of non-Gaussian impulsive interference. However, its performance can be improved further for sparse channels by extending the proportionate approach to sign algorithms. Proportionate type algorithms provide fast convergence for acoustic channels but they are not robust against impulsive interferences. However, acoustic channels, for example, are often exposed to double talks, which can be considered as the non-Gaussian interference. Therefore, integration of sign and proportionate type algorithms is crucial. Consequently, Proportionate APSA (PAPSA) and Improved PAPSA (IPAPSA) [25] were proposed to provide both fast convergence and robustness against impulsive interference. These algorithms are given below in (4.11)-(4.15) and (4.16)-(4.19), respectively. Resultant algorithms have slightly higher computational complexities compared to that of APSA.

$$\gamma_{min} = \rho \max(c, |w_0(n)|, |w_1(n)|, \dots, |w_{L-1}(n)|), \tag{4.11}$$

$$\omega_l(n) = \frac{\gamma_l}{\|\boldsymbol{\gamma}\|_1/L}, \tag{4.12}$$

$$\boldsymbol{\Omega}(n) = diag\big(\omega_0(n), \omega_1(n), \dots, \omega_{L-1}(n)\big), \tag{4.13}$$

$$\boldsymbol{x}_{gs}(n) = \boldsymbol{\Omega}(n)\boldsymbol{X}(n)sgn\big(\boldsymbol{e}(n)\big), \tag{4.14}$$

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \frac{\mu \boldsymbol{x}_{gs}(n)}{\sqrt{\boldsymbol{x}_{gs}^T(n)\boldsymbol{x}_{gs}(n) + \delta}}. \tag{4.15}$$

$$\omega_l(n) = \frac{1-\alpha}{2L} + (1+\alpha)\frac{|w_l(n)|}{2\sum_{i=0}^{L-1} w_i(n) + \epsilon}, \tag{4.16}$$

$$\boldsymbol{\Omega}(n) = diag\big(\omega_0(n), \omega_1(n), \dots, \omega_{L-1}(n)\big), \tag{4.17}$$

$$\boldsymbol{x}_{gs}(n) = \boldsymbol{\Omega}(n)\boldsymbol{X}(n)sgn\big(\boldsymbol{e}(n)\big), \tag{4.18}$$

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \frac{\mu \boldsymbol{x}_{gs}(n)}{\sqrt{\boldsymbol{x}_{gs}^T(n)\boldsymbol{x}_{gs}(n) + \delta}}. \tag{4.19}$$

4.6. Performance Comparisons for Sign Algorithms

In this section, firstly performances of the sign algorithms, namely NSA and APSA, will be compared to those of conventional algorithms, NLMS and APA. Later, proportionate type sign algorithms will be compared to NSA and APSA so that contribution of the proportionate approach in the improvement of the performance can be observed. The same configuration used in Section 3.8. is also used in this section. Input signal is generated as an $AR(1)$ process, passing a white noise through a first order IIR filter which has a pole at 0.8. Step-size is set to 0.003 for APSA in order to meet stability conditions and it is set to 0.2 for APA, 1 for NLMS and 0.1 for NSA. These steps-size values are those used in [21].

For simulations, impulsive interference is generated by BG process which is the combination of a Bernoulli process and a Gaussian process. Let $z(n)$ be the signal modeled by the BG process. It can be obtained by the multiplication of a Bernoulli signal, $b(n)$, with probability mass function $P(b) = 1 - P_b$ for $b = 0$ and $P(b) = P_b$ for $b = 1$ and a zero mean Gaussian signal, $\eta(n)$, with variance σ_η^2, $z(n) = \omega(n)\eta(n)$. Power of the BG signal is calculated by $P = P_b\sigma_\eta^2$. In this process, rate of impulses depends on P_b and their magnitudes depend on the σ_η^2. Therefore, it can be concluded that Bernoulli process is associated with the probability of occurrence of an impulse and Gaussian process is related to its magnitude.

Firstly, normalized misalignments of sign algorithms are compared to those of conventional algorithms for different Signal-to-Interference Ratio (SIR) levels. Adaptive filter has length $L = 512$. For this case, projection order for APA and APSA is selected to be $M = 2$ and P_b is set to 0.001. In Figure 4.1 results for a high SIR, $SIR = 30\ dB$, case is shown. It can be seen that APA exhibits the best performance compared to other algorithms and NSA has the slowest convergence speed. Effect of low SIR, $SIR = -10\ dB$ on the the algorithms is shown in Figure 4.2. Sign algorithms offer notable robustness against impulsive interference and it can be observed that conventional algorithms are affected by the interference such that they may almost diverge.

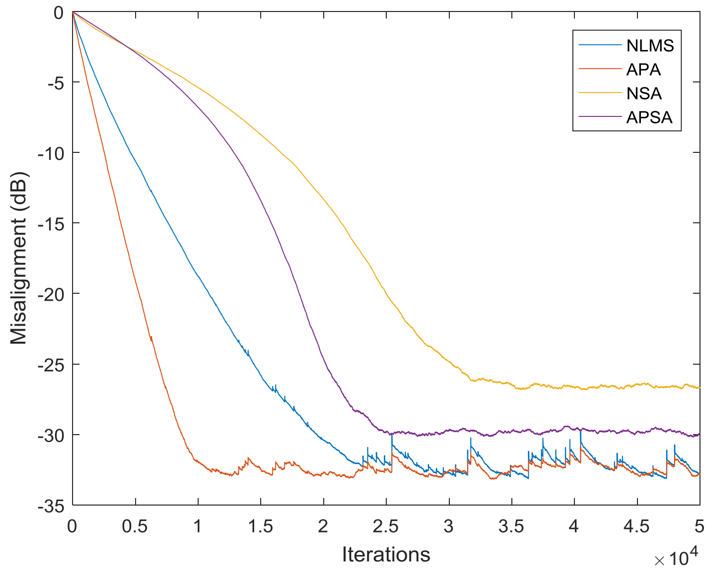

Figure 4.1 Performance comparisons of Sign Algorithms with conventional algorithms for SIR = 30 dB

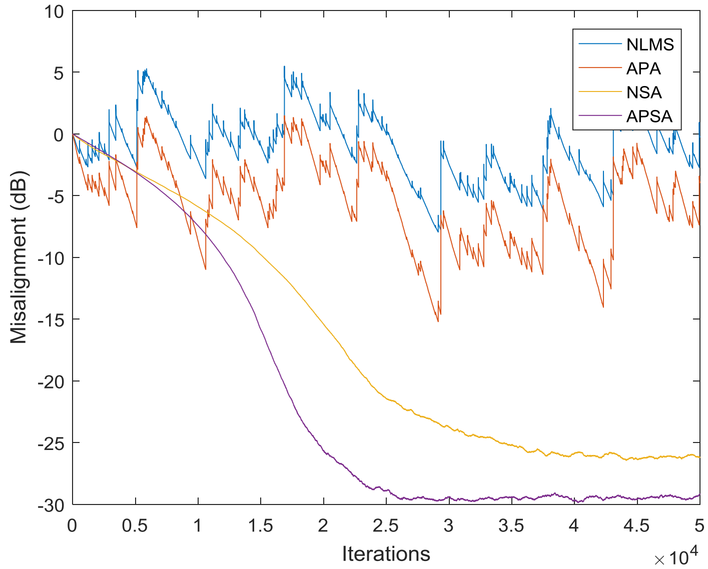

Figure 4.2 Performance comparisons of Sign Algorithms with conventional algorithms for SIR = -10 dB

Improvement introduced by the proportionate type sign algorithms are shown in Figure 4.3. Step-sizes for PAPA and IPAPA is set to 0.003. These steps-size values are those used in [25]. In this case, a channel with $\xi = 0.8723$ is identified and interference is added to system to have $SIR = -10\, dB$. Figure 4.3 shows that proportionate type sign algorithms significantly outperform APSA in terms of both convergence speed and steady-state misalignment.

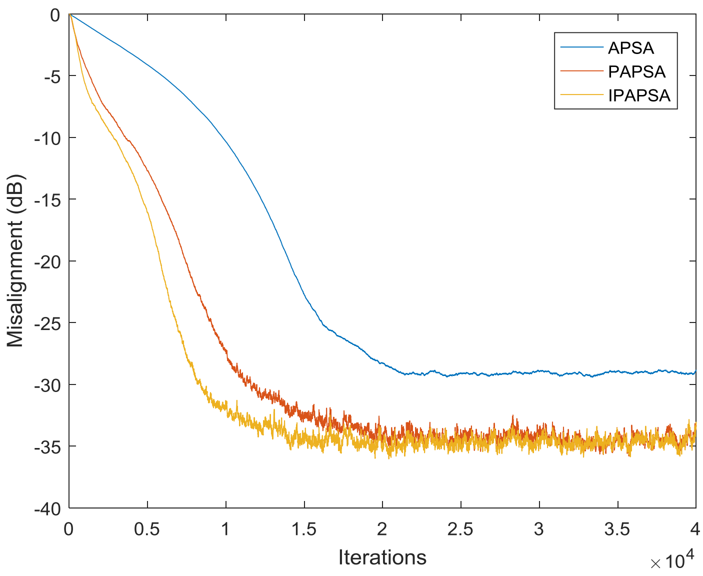

Figure 4.3 Performance comparisons of Proportionate type Sign Algorithms and APSA for SIR = -10 dB

Convergence behavior of the coefficients of APA and APSA are also compared. It can be seen from Figure 4.4 that when the desired signal is disturbed by an impulse, APA significantly changes coefficient values; however, APSA does not allow such significant change. Therefore, misalignment of APA drastically increases when an impulsive noise occurs; on the other hand, misalignment of APSA is not affected since APSA prevents deviation of the filter coefficients.

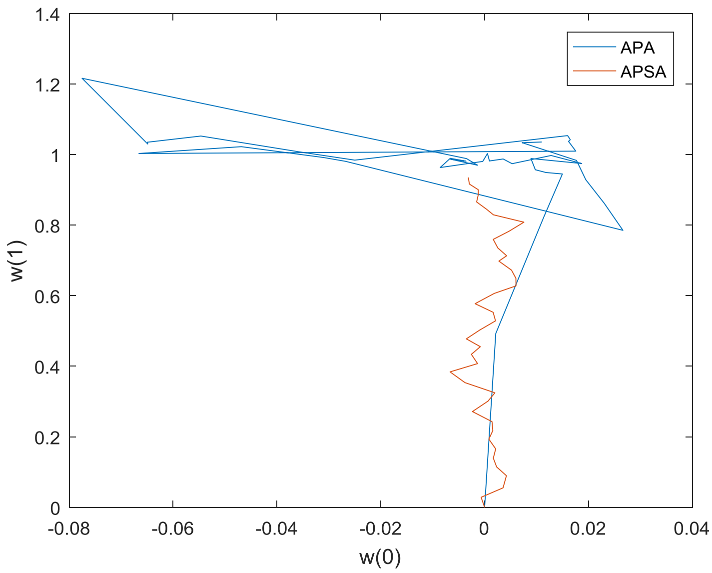

Figure 4.4 Convergence behavior of the filter coefficients for APA and APSA

CHAPTER 5

DERIVATIVE BASED PROPORTIONATE ALGORITHMS AND GEOMETRICAL INTERPRETATION

5.1. Introduction

In this chapter, a coefficient-derivative based method for sparse channel identification is introduced. In the proposed method, the proportionate factors are computed by using the time “derivatives” of filter coefficients. Firstly, the motivation, which leads to the development of the algorithm, is expressed by illustrating the convergence behavior of both minor and major coefficients. Then, the proposed method will be described and comments on the steady-state misalignment and computational complexity will be given. Lastly, a geometrical interpretation of the proportionate type algorithms will be presented to comprehend the reasoning behind the proportionate approach.

5.2. Motivation for the Proposed Algorithm

Proportionate type algorithms assign coefficient specific step-sizes to each coefficient in order to achieve faster convergence. Main purpose of these algorithms is to assign coefficient specific step-sizes in order to prevent the deviation of minor coefficients from their optimal values at the initial stage. In other words, proportionate type algorithms aim to reduce the gradient noise introduced due to the approximation of the correlation matrices. Conventional proportionate type algorithms make these assignments roughly proportional to magnitudes of the current filter coefficients. Later, approaches, which incorporate estimates of the differences between the current filter coefficients and the optimal filter coefficients, were proposed. The proposed algorithm is an example of the latter sort of algorithms since it involves proportionate factors proportional to time derivatives of the filter coefficients. Simulations on the evolution of filter coefficients show that filter coefficients tend to have higher derivative values if they are away from their optimal values. In Figure 5.1, evolutions of a major and a minor coefficient are plotted. It can be seen that the amount of variation of the minor coefficient is negligible compared to that of the major coefficient during the transient period. These variations can be quantified as the derivative of the corresponding coefficient. Therefore, derivative information possesses useful information regarding the structure of the impulse response, which can be used to determine the proportionate factors. Lastly, it can be seen that as the filter coefficients get close to their optimal value, major coefficient's rate of change also becomes small. Hence, smaller step-sizes are assigned to major coefficients at the steady-state which reduces random fluctuations around the optimal value.

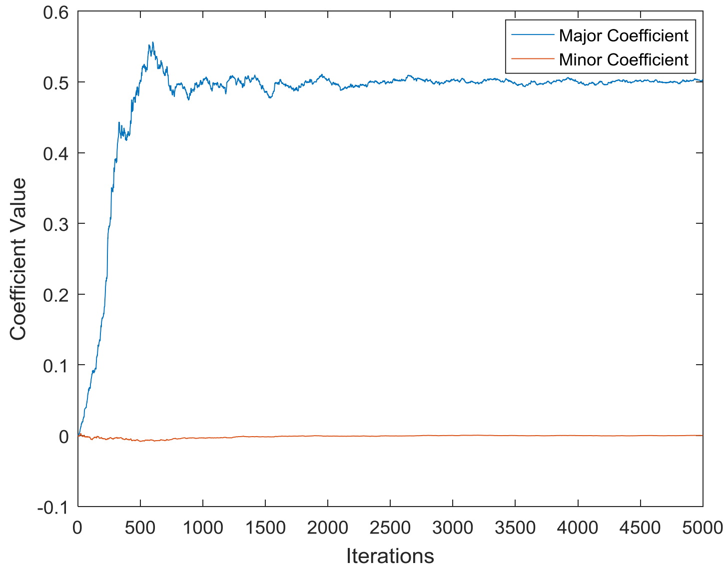

Figure 5.1 Coefficient evolutions for both major and minor coefficients for IPAPA

In order to provide a mathematical basis, firstly it can be shown that proportionate factors should be selected proportional to difference between current filter coefficients and the optimal filter coefficients. For this purpose, consider the update equation of a proportionate type NLMS algorithm,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu \frac{\boldsymbol{\Omega}(n)\boldsymbol{x}(n)}{\boldsymbol{x}^T(n)\boldsymbol{\Omega}(n)\boldsymbol{x}(n)} e(n). \tag{5.1}$$

Writing the weight error as $\widetilde{\boldsymbol{h}}(n) = \boldsymbol{h} - \boldsymbol{w}(n)$, (5.1) can be modified as

$$\widetilde{\boldsymbol{h}}(n+1) = \widetilde{\boldsymbol{h}}(n) - \mu \frac{\boldsymbol{\Omega}(n)\boldsymbol{x}(n)}{\sigma_{gx}^2} e(n), \tag{5.2}$$

where $\sigma_{gx}^2 = \boldsymbol{x}^T(n)\boldsymbol{\Omega}(n)\boldsymbol{x}(n)$. This update equation can be written for the i^{th} coefficient as,

$$\tilde{h}_i(n+1) = \tilde{h}_i(n) - \mu \frac{\omega_i(n)x_i(n)}{\sigma_{gx}^2} e(n). \tag{5.3}$$

To provide faster convergence, the following inequality should be satisfied to prevent deviation of the filter coefficients from their optimal values,

$$\left|\tilde{h}_i(n+1)\right| < \left|\tilde{h}_i(n)\right|, \tag{5.4}$$

$$\left|\tilde{h}_i(n) - \mu \frac{\omega_i(n)x_i(n)}{\sigma_{gx}^2} e(n)\right| < \left|\tilde{h}_i(n)\right|. \tag{5.5}$$

Inequality (5.5) is satisfied only if the following condition is met,

$$\left|\mu \frac{\omega_i(n)x_i(n)}{\sigma_{gx}^2} e(n)\right| < 2\left|\tilde{h}_i(n)\right|, \tag{5.6}$$

$$\mu\omega_i(n) \frac{|x_i(n)||e(n)|}{\sigma_{gx}^2} < 2\left|\tilde{h}_i(n)\right|, \tag{5.7}$$

$$\mu\omega_i(n) < \frac{2\left|\tilde{h}_i(n)\right|\sigma_{gx}^2}{|x_i(n)||e(n)|}. \tag{5.8}$$

From (5.8), it can be concluded that i^{th} proportionate factor, $\omega_i(n)$, is proportional to weight error of i^{th} coefficient, i.e. $\omega_i(n) \propto \tilde{h}_i(n)$. However, proportionate factors cannot be directly estimated since weight error vector is not available. Therefore, it is necessary to obtain an estimate of $\tilde{\boldsymbol{h}}(n)$. For this purpose, again consider the update equation of the NLMS algorithm over several iterations,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu \frac{\boldsymbol{x}(n)}{\boldsymbol{x}^T(n)\boldsymbol{x}(n)} e(n), \tag{5.9}$$

$$\boldsymbol{w}(n+2) = \boldsymbol{w}(n+1) + \mu \frac{\boldsymbol{x}(n+1)}{\boldsymbol{x}^T(n+1)\boldsymbol{x}(n+1)} e(n+1), \tag{5.10}$$

$$\vdots$$

$$\begin{aligned}\boldsymbol{w}(n+N) &= \boldsymbol{w}(n+N-1) \\ &\quad + \mu \frac{\boldsymbol{x}(n+N-1)}{\boldsymbol{x}^T(n+N-1)\boldsymbol{x}(n+N-1)} e(n+N-1).\end{aligned} \tag{5.11}$$

Summation of $\boldsymbol{w}(n+1)$ to $\boldsymbol{w}(n+N)$ gives,

$$\boldsymbol{w}(n+N) = \boldsymbol{w}(n) + \mu \sum_{j=0}^{N-1} \frac{\boldsymbol{x}(n+j)}{\boldsymbol{x}^T(n+j)\boldsymbol{x}(n+j)} e(n+j). \tag{5.12}$$

Since $e(n) = \boldsymbol{x}^T(n)\tilde{\boldsymbol{h}}(n) + v(n)$ 5.12 becomes,

$$\begin{aligned}&\boldsymbol{w}(n+N) - \boldsymbol{w}(n) \\ &\qquad = \mu \sum_{j=0}^{N-1} \frac{\boldsymbol{x}(n+j)\left[\boldsymbol{x}^T(n+j)\tilde{\boldsymbol{h}}(n+j) + v(n+j)\right]}{\boldsymbol{x}^T(n+j)\boldsymbol{x}(n+j)}.\end{aligned} \tag{5.13}$$

Inspiring from law of large numbers, summation in (5.13) can be replaced by expectation operator and it is assumed that the step-size μ is chosen such that the change in $\tilde{\boldsymbol{h}}(n)$ is sufficiently small. Then (5.13) becomes,

$$\boldsymbol{w}(n+N) - \boldsymbol{w}(n) \approx \mu N E\left\{\frac{\boldsymbol{x}\left[\boldsymbol{x}^T\overline{\boldsymbol{h}}(n+N) + v\right]}{\boldsymbol{x}^T\boldsymbol{x}}\right\}. \tag{5.14}$$

where $\overline{\boldsymbol{h}}(k+N)=\frac{\sum_{j=0}^{N-1}\widetilde{\boldsymbol{h}}(k+j)}{N}$ can be considered as the average coefficient error vector at time $k+N$ assuming that convergence behavior of filter coefficients does not change significantly within N samples. Since the background noise and the input signal are independent, the second term of the expectation becomes zero. Therefore, the following approximation is obtained,

$$\boldsymbol{w}(n+N)-\boldsymbol{w}(n)\approx\mu N E\left\{\frac{\boldsymbol{x}\boldsymbol{x}^T\overline{\boldsymbol{h}}(n+N)}{\boldsymbol{x}^T\boldsymbol{x}}\middle|\boldsymbol{w}(k)\right\}. \tag{5.15}$$

By using the assumptions made in [4], (5.15) turns into

$$\boldsymbol{w}(n+N)-\boldsymbol{w}(n)\approx\frac{\mu N E\{\boldsymbol{x}\boldsymbol{x}^T\}}{E\{\boldsymbol{x}^T\boldsymbol{x}\}}\widehat{\boldsymbol{h}}(n+N), \tag{5.16}$$

$$\boldsymbol{w}(n+N)-\boldsymbol{w}(n)\approx\frac{\mu N\boldsymbol{R}_x}{N\sigma_x^2}\widehat{\boldsymbol{h}}(n+N). \tag{5.17}$$

where $\widehat{\boldsymbol{h}}(n+\boldsymbol{N})=E\{\overline{\boldsymbol{h}}(n+N)\}$. If the signal is assumed to be white then $\boldsymbol{R}_x/\sigma_x^2=\boldsymbol{I}$. Therefore, N-step coefficient difference becomes,

$$\boldsymbol{w}(n+N)-\boldsymbol{w}(n)\approx\mu\widehat{\boldsymbol{h}}(n+N). \tag{5.18}$$

Consequently, it can be concluded that optimal proportionate factors can be calculated by using the difference of two instances of the filter coefficients. This difference in the discrete domain can be interpreted as an approximation of the time-averaged derivative in continuous domain. Therefore, time derivative of filter coefficients provides valuable information while calculating proportionate factors. Above analysis is carried out for white Gaussian input signals; however, it can be extended to APA for colored input.

5.3. Proposed Derivative Based Proportionate approach

Proposed method for the calculation of proportionate factors is composed of two parts. As the name of the algorithm implies, the first part of the algorithm is related to the "derivatives" of the filter coefficients. The term "derivative" is used generically since adaptive filters operate in discrete time. "Derivatives" of filter coefficients are defined as

$$\nabla_l(n) \triangleq |w_l(n) - \bar{\bar{w}}_l(n)|, \tag{5.19}$$

where $\bar{\bar{w}}_l(n)$ would be,

$$\bar{\bar{w}}_l(n) = w_l(n - T). \tag{5.20}$$

However, (5.20) requires excessive memory since $T \times L$ filter coefficients should be stored in the memory. Therefore, an alternative approach has been developed. This approach requires an intermediate step in order to ensure that the difference does not involve consecutive instances of $w_l(n)$ while calculating the derivative values. For this purpose, an intermediate element, $\bar{w}(n)$, is defined as

$$\bar{w}_l(n) \triangleq \begin{cases} w_l(n) & n = kL_m \\ \bar{w}_l(n-1) & n \neq kL_m \end{cases}, \tag{5.21}$$

where, L_m is the update period which defines the minimum time in order to obtain reliable derivative approximates. In order separation between $w_l(n)$ and $\bar{\bar{w}}_l(n)$ to be large enough, $\bar{w}_l(n)$ is delayed by L_m samples, $\bar{\bar{w}}_l(n) = \bar{w}_l(n - L_m)$, so that reasonable difference values are obtained. Then, $\bar{\bar{w}}_l(n)$ is defined as

$$\bar{\bar{w}}_l(n) \triangleq \begin{cases} \bar{w}_l(n) & n = kL_m \\ \bar{\bar{w}}_l(n-1) & n \neq kL_m \end{cases}. \tag{5.22}$$

Choice of L_m is a crucial issue that needs to be discussed in-depth. There are two major factors that affect the choice of L_m value. Firstly, it depends on the 'inherent' convergence speed of the algorithm, which, in turn, depends on the type of the input signal. As a rule of thumb, L_m should be selected to be smaller than the 'inherent' convergence time. If L_m is chosen to be comparable or larger than the 'inherent' convergence time then, effectively, derivative information as utilized in the proposed method will be redundant since the algorithm will converge before derivative information develops. Therefore, L_m value, for example, for an AR(1) type input signal should be smaller than that for a speech type input signal. The reason behind the use of smaller L_m value with AR(1) signals is that convergence is faster for AR(1) signals compared to speech input signals. If L_m value for AR(1) input signal is chosen to be close to the L_m value for a speech input then the algorithm may reach the steady state before the L_m^{th} iteration and; therefore, it becomes impossible to use derivative information during the convergence period. Therefore, the proposed algorithm shows almost the same performance as classical IPAPA for larger values of L_m. Consequently, it is reasonable to select relatively smaller L_m values for AR(1) input signals. On the other hand, larger L_m values can be selected for speech input signals to detect changes in the filter coefficients due to slow convergence of the adaptive filter.

Secondly, in the case of speech input, convergence of the filter coefficients has a non-uniform profile. If convergence characteristics of an individual filter coefficient is examined, it can be observed that filter coefficients do not move to their optimal values smoothly for a speech input signal in contrast to that encountered with an AR(1) signal. Therefore, in order to eliminate the effect of the non-uniform convergence, a larger block size is used in the calculation of the derivative values. In addition, using delayed instances of the filter coefficients helps to combat the non-uniform convergence. Figure 5.2 shows the evolution of a specific filter coefficient for both speech and AR(1) inputs. It can be seen that in the AR(1) case the filter coefficient moves to its optimal value in a smooth way. On the other hand, the filter coefficient value changes significantly slower and non-uniformly in the case of speech input signal.

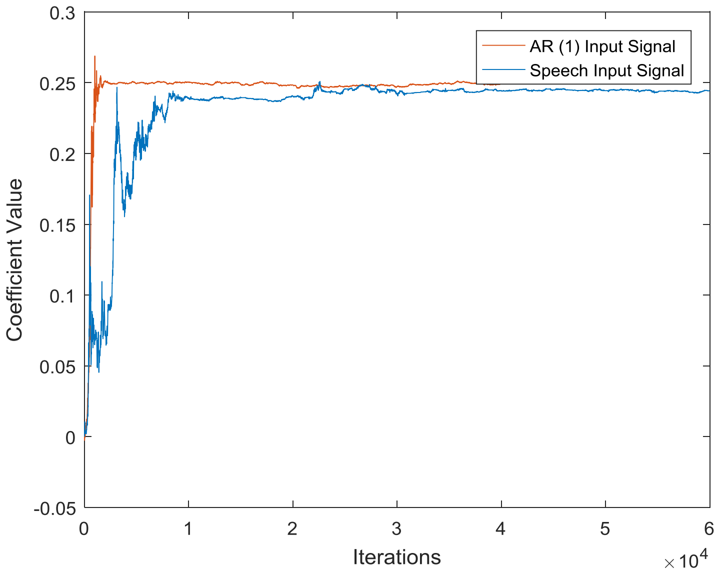

Figure 5.2 Coefficient evolutions for the same coefficient for AR(1) and Speech Input Signals for the proposed algorithm

Another fundamental issue of the proposed algorithm is the calculation of the normalization value of the proportionate factors. Other algorithms normalize the proportionate factors by the l_1-norm of the proportionate elements. In the proposed algorithm, a new normalization method is developed. This normalization method mainly aims to sense the changes in the filter coefficients in terms of their magnitudes. Therefore, the magnitude of the current filter estimate is a component of the normalization value which is effective in the steady-state. In the steady-state, changes in the filter coefficients are negligibly small compared to their magnitudes; hence, normalizing by the magnitude value, smaller step-sizes are assigned. Therefore, random fluctuations around the optimal value is avoided at the steady state. On the other hand, if a sudden change occurs in the filter coefficients than derivative values become comparable to magnitudes of the filter coefficients. Hence, in order to track the changes in the filter coefficients, larger step-sizes are applied to coefficients, which are subject to significant changes. However, during the transient period, derivative values might be significantly larger than magnitudes, which may lead to impulsive variations in the proportionate factors. In order to avoid such variations of the proportionate factors, magnitudes of the derivative values are used for the calculation of the normalization value. This forms the second part of the normalization value. Consequently, normalization value is dominated by the different components for different stages. Overall normalization value, $K_{mx}(n)$ is obtained as

$$W_{mx}(n) = \max\{|w_0(n)|, |w_1(n)|, \ldots, |w_{L-1}(n)|\}, \tag{5.23}$$

$$\nabla_{mx}(n) = \max\{\nabla_0(n), \nabla_1(n), \ldots, \nabla_{L-1}(n)\}, \tag{5.24}$$

$$K_{mx}(n) = W_{mx}(n) + \nabla_{mx}(n). \tag{5.25}$$

As a result, the proposed approach is integrated with the existing methods such that coefficient specific proportionate constant is calculated as

$$\omega_l(n) = \frac{1-\alpha}{2L} + \frac{(1+\alpha)\nabla_l(n)}{K_{mx}(n) + \epsilon}. \tag{5.26}$$

Rest of the procedure is the same with that of other algorithms, which employ proportionating. Therefore, the matrix containing coefficient specific step-sizes is formed as, $\mathbf{\Omega}(n) = diag[\omega_0(n), \omega_1(n), \ldots, \omega_{L-1}(n)]$, then the adaptation equation becomes,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu\mathbf{\Omega}(n)\boldsymbol{X}(n)[\boldsymbol{X}^{\boldsymbol{T}}(n)\mathbf{\Omega}(n)\boldsymbol{X}(n) + \delta\boldsymbol{I}]^{-1}. \tag{5.27}$$

The proposed method can be extended as a sign algorithm as

$$\omega_l(n) = \frac{1-\alpha}{2L} + \frac{(1+\alpha)\nabla_l(n)}{\nabla_{mx}(n) + \|\boldsymbol{w}(n)\|_1 + \epsilon}. \tag{5.28}$$

$$\boldsymbol{x}_{ds}(n) = \boldsymbol{X}(n)sgn\big(\boldsymbol{e}(n)\big), \tag{5.29}$$

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \frac{\mu \boldsymbol{\Omega}(n) \boldsymbol{x}_{ds}(n)}{\sqrt{\boldsymbol{x}_{ds}^T(n) \boldsymbol{\Omega}(n) \boldsymbol{x}_{ds}(n) + \delta}}. \tag{5.30}$$

Normalization value of the sign extension of the proposed approach is not the same as that of DB-IPAPA since smoother proportionate factors is needed in case of sign algorithms. However, (5.25) results in larger proportionate factors yielding significant changes in the filter coefficients which is undesired due to minimum disturbance constraint. Hence, proportionate factors are calculated as in (5.28) in order not to harm disturbance constraint. Normalization value in (5.28) provides smoother proportionate factors since l_1 norm of the filter coefficients is much larger than the difference values which gives smaller proportionate factors. However, $\nabla_{mx}(n)$ terms is kept in order to sustain stability in case of sudden changes of the impulse response.

Steady-state analysis of proportionate type algorithms is a difficult task. However, under some assumptions Duttweiler [11] has shown that the steady state misalignment of the PNLMS is the same as that of the NLMS algorithm. Nevertheless, without any assumption it can be observed that steady state misalignment of the proposed algorithm will be the same as the NLMS\APA since derivative values become zero at the steady state. Therefore, update equation of the proposed algorithm becomes identical to the update equation of NLMS\APA at the steady state.

Computational complexity of the DB-IPAPA is an important aspect to be considered. It can be noted that the proposed algorithm does not require additional matrix operations such as summation or multiplication. It only requires additional $2L$ comparisons compared to that of IPAPA. Therefore, it can be stated that DB-IPAPA achieves better performance without introducing extra computational burden. Similarly, proposed DB-IPAPSA requires additional L comparisons and L summations compared to IPAPSA.

5.4. Geometrical Interpretation of the Proportionate Type Algorithms

In section 2.8, geometrical interpretations for NLMS and APA are presented. In this section, proportionate approach will be investigated from a geometrical perspective as well. By using this perspective, convergence behavior of the proportionate type algorithms will be discussed and compared with the conventional algorithms. In the geometrical analysis, an impulse response with length 2 will be considered for simplicity. This impulse response is sparse such that one of the coefficient is zero and the other one is non zero, $\boldsymbol{h} = [h_0 \; 0]^T$. Assume that at the n^{th} iteration, the coefficients of the adaptive filter have values $\boldsymbol{w}(n) = [w_0(n), 0]$. Let $\widehat{\boldsymbol{w}}(\mathrm{n}+1)$ be the NLMS update of the $\boldsymbol{w}(n)$ and $\breve{\boldsymbol{w}}(n+1)$ be the DB-IPNLMS update of $\boldsymbol{w}(n)$. Proportionate matrix of DB-IPNLMS is $\boldsymbol{\Omega}(n) = \begin{bmatrix} \omega_0 & 0 \\ 0 & \omega_1 \end{bmatrix}$, where $\omega_0 \gg \omega_1$ and for any proportionate type NLMS (Pt-NLMS) $\omega_0 \approx 1$ and $\omega_1 \approx 0$. Furthermore, squares of the proportionate factors can be approximated as $\omega_0^2 \approx \omega_0$ and $\omega_1^2 \approx \omega_1$. Firstly, consider the NLMS update of $\boldsymbol{w}(n)$,

$$\widehat{\boldsymbol{w}}(\mathrm{n}+1) = \boldsymbol{w}(n) + \frac{\mu \boldsymbol{x}(n)}{\boldsymbol{x}^T(n)\boldsymbol{x}(n)} e(n). \tag{5.31}$$

Equation (5.31) can be written for each coefficient individually as

$$\widehat{w}_0(\mathrm{n}+1) = w_0(n) + \frac{\mu x(n)}{x^2(n) + x^2(n-1)} e(n), \tag{5.32}$$

$$\widehat{w}_1(\mathrm{n}+1) = w_1(n) + \frac{\mu x(n-1)}{x^2(n) + x^2(n-1)} e(n). \tag{5.33}$$

Now, consider the update equation of the Pt-NLMS of $\boldsymbol{w}(n)$,

$$\breve{\boldsymbol{w}}(\mathrm{n}+1) = \boldsymbol{w}(n) + \frac{\mu \boldsymbol{\Omega}(n)\boldsymbol{x}(n)}{\boldsymbol{x}^T(n)\boldsymbol{\Omega}(n)\boldsymbol{x}(n)} e(n). \tag{5.34}$$

Equation (5.34) can also be written for each coefficient individually,

$$\breve{w}_0(\mathrm{n}+1) = w_0(n) + \frac{\mu \omega_0 x(n)}{\omega_0 x^2(n) + \omega_1 x^2(n-1)} e(n), \tag{5.35}$$

$$\breve{w}_1(\mathrm{n}+1) = w_1(n) + \frac{\mu \omega_1 x(n-1)}{\omega_0 x^2(n) + \omega_1 x^2(n-1)} e(n). \tag{5.36}$$

By considering (5.34) and assumptions on ω_0 and ω_1, a modified input vector can be defined as

$$\breve{\boldsymbol{x}}(\mathrm{n}) = \begin{pmatrix} \omega_0 x_0(n) \\ \omega_1 x_1(n) \end{pmatrix}. \tag{5.37}$$

Remember that NLMS algorithm is a Newton's method based algorithm in which direction of the gradient is rotated by the autocorrelation matrix of the input signal. However, NLMS involves the approximation of the correlation matrices, which causes errors in the gradient calculation. Therefore, it can be stated that DB-IPNLMS and other proportionate type algorithms are developed to correct the correlation matrices such that adaptive filter follows a more direct path, which improves convergence rate as desired in the Newton's method by modifying the input matrix as in (5.37). Update process of both NLMS and DB-NLMS can be seen from Figure 5.3.

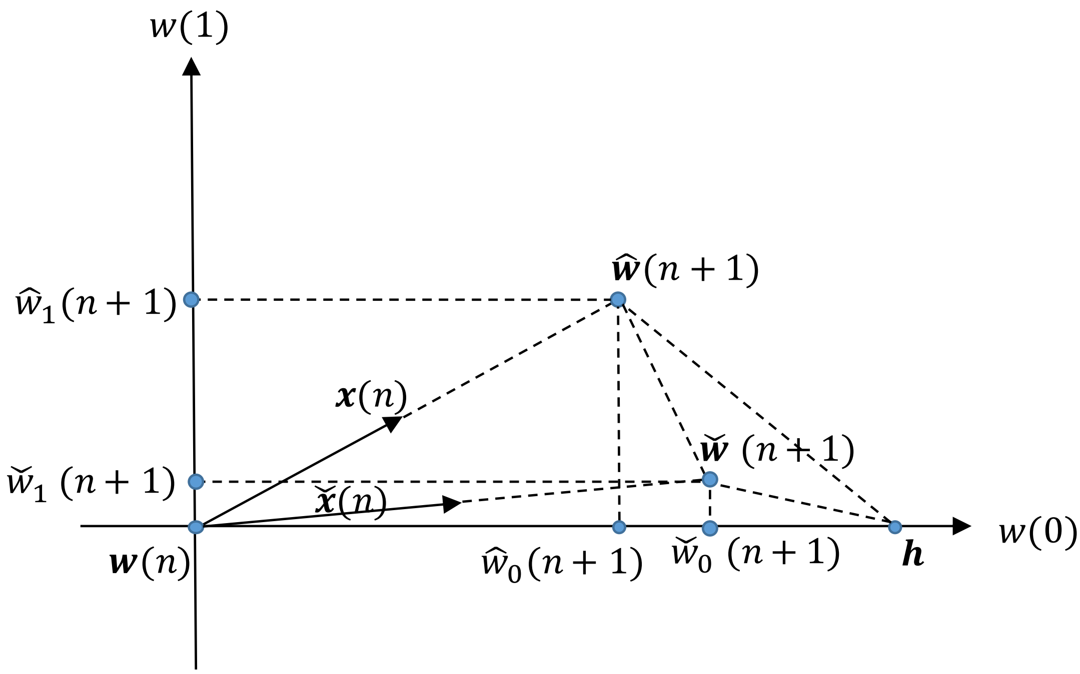

Figure 5.3 Geometrical illustration of update process of NLMS and DB-IPNLMS

Geometrically, misalignment is defined as the distance between the estimated and optimal filter coefficients. Therefore, misalignments of DB-IPNLMS and NLMS can be compared by considering the triangle with corners $\widehat{\boldsymbol{w}}(\mathrm{n}+1)$, $\breve{\boldsymbol{w}}(\mathrm{n}+1)$ and $\boldsymbol{h}$. It can be stated that if this triangle is an obtuse-angled triangle then misalignment at next iteration of DB-IPNLMS is guaranteed to be less than that of NLMS. Therefore, it should be shown that the following inequalities are satisfied to have obtuse-angled triangle,

$$|\breve{w}_0(\mathrm{n}+1)| \geq |\widehat{w}_0(\mathrm{n}+1)|, \tag{5.38}$$

$$|\breve{w}_1(\mathrm{n}+1)| \geq |\widehat{w}_1(\mathrm{n}+1)|. \tag{5.39}$$

In order to state that (5.38) holds, it should be shown that

$$\left|\frac{\mu\omega_0 x(n)}{\omega_0 x^2(n)+\omega_1 x^2(n-1)}e(n)\right| \geq \left|\frac{\mu x(n)}{x^2(n)+x^2(n-1)}e(n)\right|. \tag{5.40}$$

Left-hand side of the inequality (5.40) can be modified by using the assumption of $\omega_0 \gg \omega_1$ on the proportionate factors as

$$\left|\frac{\mu\omega_0 x(n)}{\omega_0 x^2(n)}e(n)\right| \geq \left|\frac{\mu x(n)}{x^2(n)+x^2(n-1)}e(n)\right|, \tag{5.41}$$

$$\left|\frac{\mu x(n)}{x^2(n)}e(n)\right| \geq \left|\frac{\mu x(n)}{x^2(n)+x^2(n-1)}e(n)\right|. \tag{5.42}$$

It is clear that this inequality holds since denominator of the left-hand side of the inequality is always greater than or equal to that of the right-hand side.

Then same procedure is carried out to show that (5.39) also holds,

$$\left|\frac{\mu\omega_1 x(n-1)}{\omega_0 x^2(n)+\omega_1 x^2(n-1)}e(n)\right| \leq \left|\frac{\mu x(n-1)}{x^2(n)+x^2(n-1)}e(n)\right|. \tag{5.43}$$

Since $\omega_1 \approx 0$, (5.43) is obvious. Therefore, it can be claimed that (5.39) holds. Consequently, by utilizing the fact that the triangle is an obtuse-angled triangle, following inequality can be written,

$$\|\boldsymbol{h}-\breve{\boldsymbol{w}}(n+1)\|_2 \leq \|\boldsymbol{h}-\widehat{\boldsymbol{w}}(n+1)\|_2. \tag{5.44}$$

Therefore, proportionate type algorithms achieve smaller misalignment than conventional algorithm at each step, which makes the convergence of the proportionate type algorithms faster. Consequently, by multiplying input vector with $\boldsymbol{\Omega}$ matrix, input vector is rotated such that gradient vector points to the optimal value.

CHAPTER 6

SIMULATION RESULTS

6.1. Introduction

In this chapter, simulation results are investigated in detail to evaluate the performances of the proposed algorithms. Misalignment curves of DB-IPAPA, IPAPA, D-IPAPA and GC-IPAPA are presented to compare the convergence performances of the algorithms for different configurations. Algorithms are tested for different values of the parameters step-size (μ), control parameter (α) and projection order (M). Misalignment curves are obtained for different types of input signals and SNR values. Proposed DB-IPAPSA is compared to the other proportionate type sign algorithms. Effects of double-talk interference on the convergence of both sign and non-sign algorithms are investigated. Lastly, time evolutions of assigned proportionate factors of the proposed approach are compared to those of the other approaches in order to understand the reasons of the performance differences.

6.2. Configuration of the Simulation Environment

In order the comparisons to be fair, simulation environment has to be constructed carefully. Selection of the simulation parameters is of crucial importance in evaluating the performances of the algorithms. Therefore, parameter values, which have been used in the literature, are also used in this thesis. The length of the unknown impulse response is taken as 512 as in [11], [12], [13], [19], [32]. Impulse responses with different sparseness of our choice and also from ITU-T G168 Recommendation [33] (padded with zeros) are considered in the simulations.

Selection of the step-size value is another important issue, which should be discussed. In order to select the optimal step-size, performances of all algorithms for different step-sizes are investigated. According to the obtained misalignment curves or different step-size values, it is concluded that step-size, $\mu = 0.2$, provides a good performance in terms of both convergence rate and steady-state misalignment for non-sign algorithms. This is consistent with the simulation results presented in [12], [13], [32]. In addition, as stated in [11], same step-size value yields same steady-state misalignment. Therefore, it is necessary to compare proportionate type algorithms for the same step-size value. However, in the case of sign algorithms, smaller step-sizes are selected when obtaining the misalignment curves of these algorithms as used in [21], [25]. Misalignment curves for different step-sizes are plotted in the subsequent sections to justify the selection of the step-size parameter.

Selection of the projection order is straightforward depending on the statistical properties of the input signal. As the correlation between input samples increases, projection order should also increase in order to improve convergence performance [10]. Therefore, projection order is selected for white Gaussian, AR(1), speech signals as $M = 1$, $M = 2$, $M = 8$ respectively. Control parameter, α, is selected to be 0 as used in the most of the studies [13], [19], [32]. However, in order to clarify this selection, performances of the algorithms are obtained for different α values.

Choice of block period L_m is also verified based on the simulation results presented in the subsequent section. Regularization parameter, δ, is selected as $\delta = \frac{10\sigma_x^2}{L}$, where σ_x^2 is the input signal power [34] and the other Regularization parameter, ϵ is set to 0.01 for all cases as in [13], [15],[19], [33].

6.3. Performance Comparisons of DB-IPNLMS and DB-IPAPA

Firstly, performance of the proposed DB-IPAPA is compared with that of IPAPA, D-IPAPA and GC-IPAPA with projection order $M = 1$. Input signal of the channel is a white Gaussian signal. A white Gaussian signal as the background noise is added to the output of the channel at an SNR of 30 dB.

Other parameters are selected as $\mu = 0.2$ and $\alpha = 0$ for all algorithms. Block period is selected as $L_m = L$ for the proposed algorithm and $P = 2L$ for D-IPNLMS.

6.3.1. Effect of Sparseness of the Channel

Results for the identification of a channel with sparseness $\xi = 0.7961$ is shown in Figure 6.1. It can be observed that proposed algorithm has the fastest convergence speed and both D-IPNLMS and GC-IPNLMS outperforms IPNLMS algorithm. Moreover, all algorithms have the same steady-state misalignment as stated in section 5.3.

In order to observe the effect of the channel sparseness, simulation results for channel with different sparseness levels are also shown. For this purpose, results for a channel with sparseness $\xi = 0.8896$ are also obtained which are shown in Figure 6.2. In Figure 6.2, it is observed that as the impulse response becomes sparser superiority of the DB-IPNLMS becomes more apparent. However, it is also observed that GC-IPNLMS has the performance degradation problem for such highly sparse channel.

Proposed algorithm is also compared with the other algorithms for a dispersive channel with sparseness $\xi = 0.5719$. In this case, the results of NLMS are also included in order to evaluate the performance of the proportionate type algorithms for dispersive channels. Results for the dispersive channel are shown in Figure 6.3. It can be seen that the convergence speeds of all proportionate type algorithms decrease and get close to each other since sparse channel assumption is no longer valid. Hence, it cannot be used to improve the convergence speed. However, proportionate type algorithms continue to have faster convergence speed compared to NLMS algorithm.

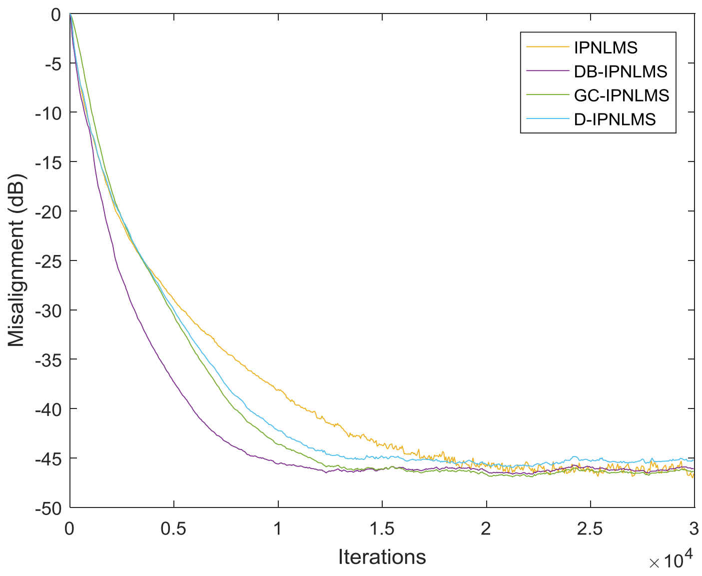

Figure 6.1 Performance comparison of proportionate type algorithms for $\xi = 0.7961$

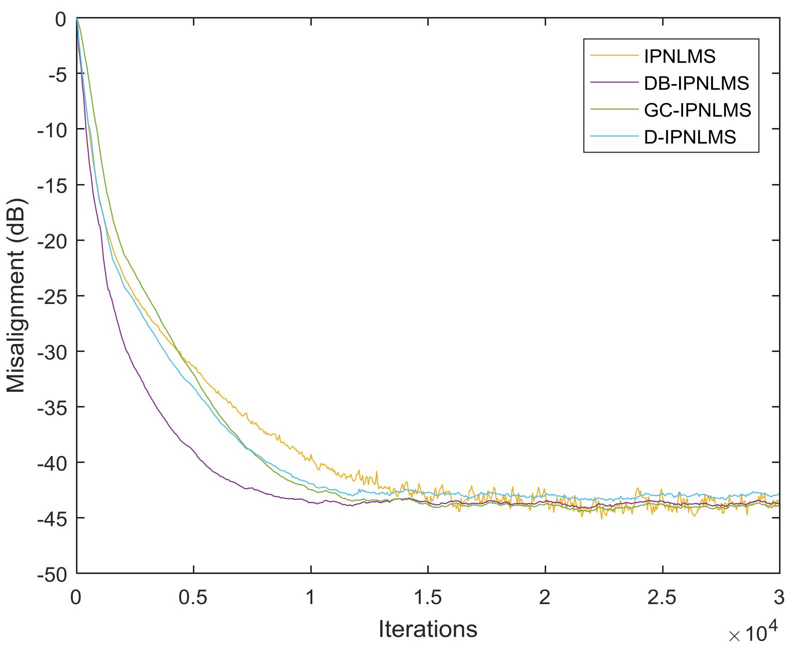

Figure 6.2 Performance comparison of proportionate type algorithms for $\xi = 0.8896$

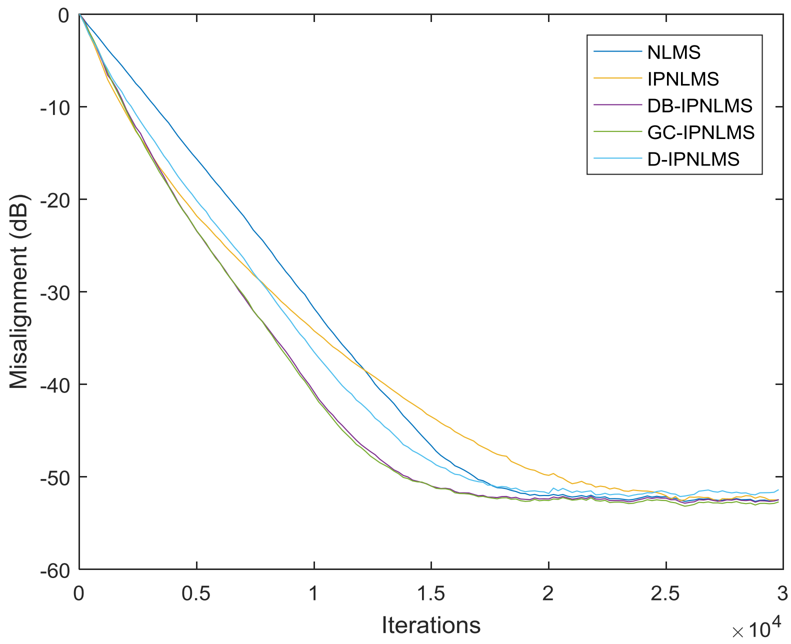

Figure 6.3 Performance comparison of proportionate type algorithms for $\xi = 0.5719$

6.3.2. Colored Input Signal Case

Proposed algorithm is compared to the other algorithms with colored input signals. For this purpose, a sparse channel, with $\xi = 0.8174$, is identified when the input signal is modeled as an AR (1) signal with a pole at 0.8. Since the input signal samples are correlated, algorithms with a higher projection order, $M = 2$, are considered. Proposed DB-IPAPA is compared with IPAPA, D-IPAPA and GC-IPAPA and the results are shown in Figure 6.4, which demonstrate the superior convergence performance of the proposed algorithm.

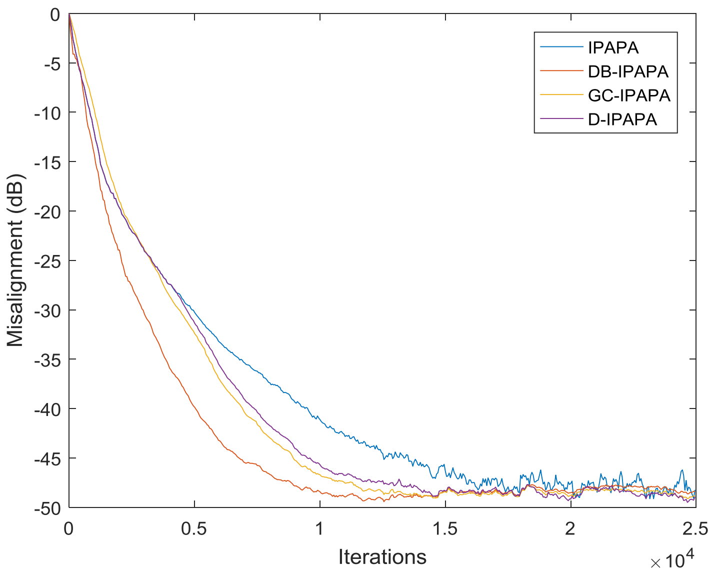

Figure 6.4 Performance comparison of proportionate type algorithms for $M = 2$

6.3.3. Tests for Network Echo Channels from ITU-T Standards

Channels, which have been identified so far have been randomly generated; however, it is more suitable to use channel models specified by the standards. Therefore, algorithms are also compared for echo path models described by ITU-T G68 Recommendations [33]. These echo path models are recommended to be used in tests of network echo cancellation applications on 4-wire telephone networks. In Figure 6.5 and Figure 6.6, performance comparisons of the proportionate type algorithms are shown for the echo path models EPM-1 and EPM-2 when the input signal is an AR (1) signal. Sparseness of EPM-1 is $\xi = 0.8970$ and that of EPM-2 is $\xi = 0.8031$. In all cases, proposed algorithm has superior performance compared to other algorithms. In addition to achieving faster convergence, proposed method reduces the gradient noise at the steady state since derivative terms become zero. This yields assignment of smaller steady-state step-sizes to the coefficients so that smaller fluctuations occur around the optimal value.

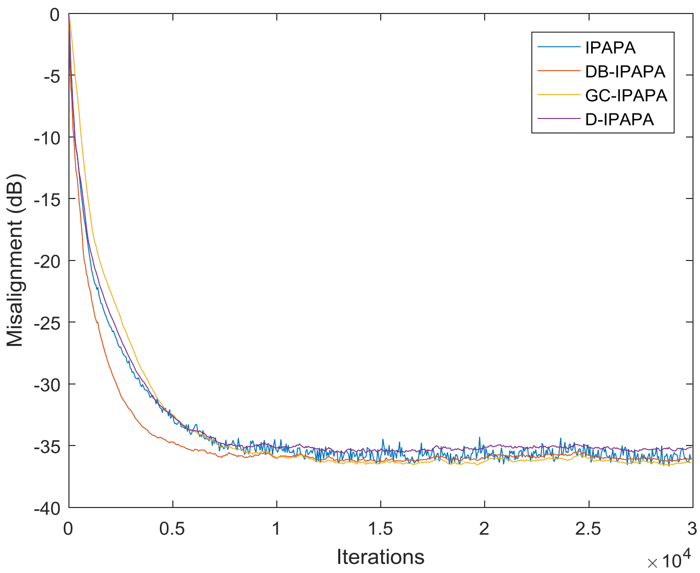

Figure 6.5 Performance comparison of proportionate type algorithms for EPM-1

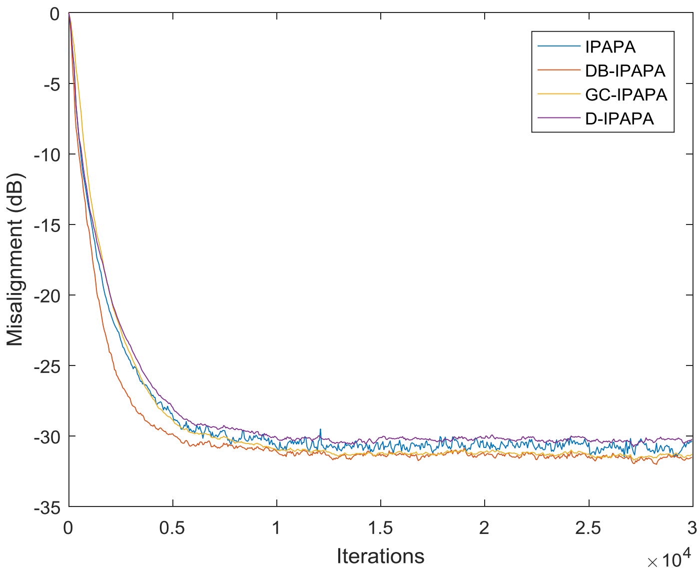

Figure 6.6 Performance comparison of proportionate type algorithms for EPM-2

In addition to an AR (1) signal, performances of these algorithms are also compared for speech input signal. Speech signal with sampling rate $8\,kHz$ is used in the simulations. Since speech signal samples are highly correlated and non-stationary, a higher projection order, $M = 8$, is used and block period is selected as $L_m = 4L$ for the proposed algorithm and $P = 8L$ for D-IPAPA. Figure 6.7 and Figure 6.8 show the results for EPM-1 and EPM-2 respectively. Proposed algorithm outperforms the other algorithms with speech input signal since it is able to detect the sudden changes in the filter coefficients, which occur due to non-stationary nature of the speech signal.

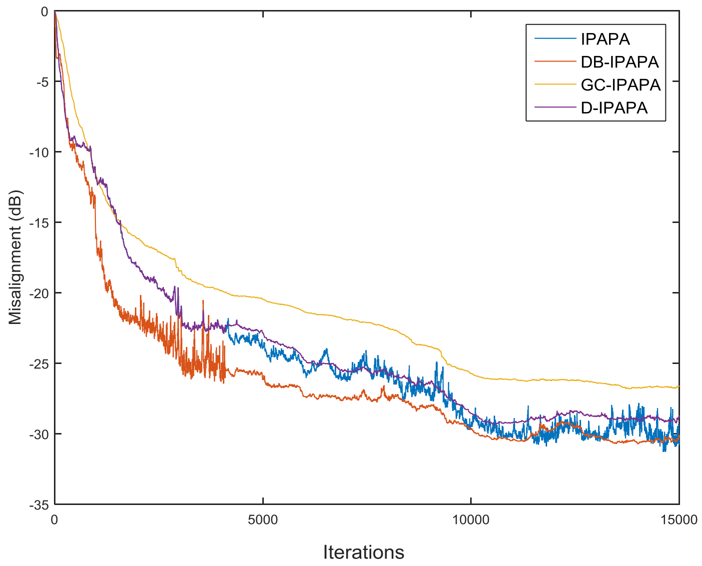

Figure 6.7 Performance comparison of proportionate type algorithms for EPM-1 for Speech Input

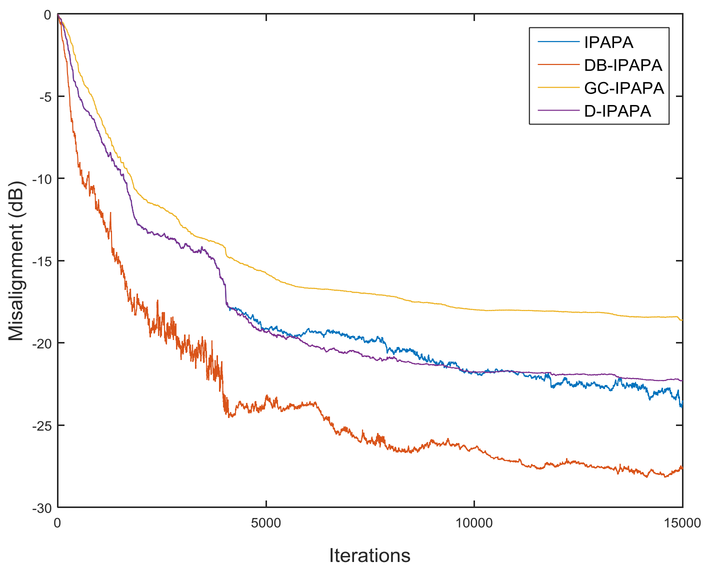

Figure 6.8 Performance comparison of proportionate type algorithms for EPM-2 for Speech Input

6.3.4. Tracking Performance

Tracking performance of the adaptive filtering algorithms is an important evaluation item since, as encountered in mobile channels, a sudden change in the echo path may occur during the adaptation. It is desired for the adaptive filter to recover as fast as possible. Hence, tracking performance of the proposed algorithm is investigated in the simulations. For this purpose, echo path is changed suddenly during the adaptation. Proposed algorithm also provides reasonable tracking ability compared to other proportionate type algorithms as shown in Figure 6.9. When the echo path changes suddenly, distance between current and optimal filter coefficients increases. This leads to fast variations in the filter coefficients yielding higher derivative values. Therefore, incorporation of derivative values in the formation of proportionate factors contributes to faster recovery of the filter after sudden changes. However, conventional IPAPA provide fast initial tracking since they store the information regarding the locations of the non-zero filter coefficients. Consequently, faster tracking is achieved by employing DB-IPAPA adaptation rule.

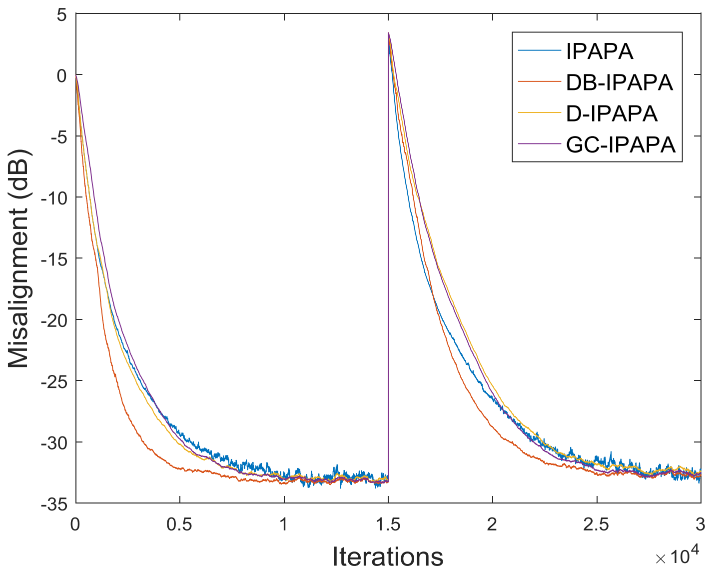

Figure 6.9 Tracking performance comparison of proportionate type algorithms

6.3.5. Effect of Step-size Parameter

Performance of the proposed algorithm is evaluated for different step-size values in order to experimentally determine the optimal value of the step-size. Figure 6.10 shows the misalignment curves of DB-IPAPA with a projection order of $M = 2$ for EPM-1 channel. An AR (1) signal is applied to the channel. Step-size is changed from $\mu = 0.1$ to $\mu = 0.5$ and it can be seen that as the step-size increases convergence speed increases; however, steady-state misalignment also increases. Therefore, one should make a trade-off between convergence rate and steady state misalignment. From Figure 6.10, it is observed that after $\mu = 0.2$, increase in the convergence rate decreases, on the other hand, steady-state misalignment continues to increase of a higher rate. Hence, step-size value, $\mu = 0.2$, is selected to be used in the simulations.

Moreover, results for larger step-sizes near the limit of divergence are also obtained. Figure 6.11 shows the results for step-sizes $\mu = 1.8$ to $\mu = 2.2$. Effect of step-size on the performance of IPAPA is also shown in Figure 6.12 to clarify the selection of step-size for classical proportionate algorithms. It can be seen that step-size $\mu = 0.2$ provides satisfactory steady-state and convergence speed performance.

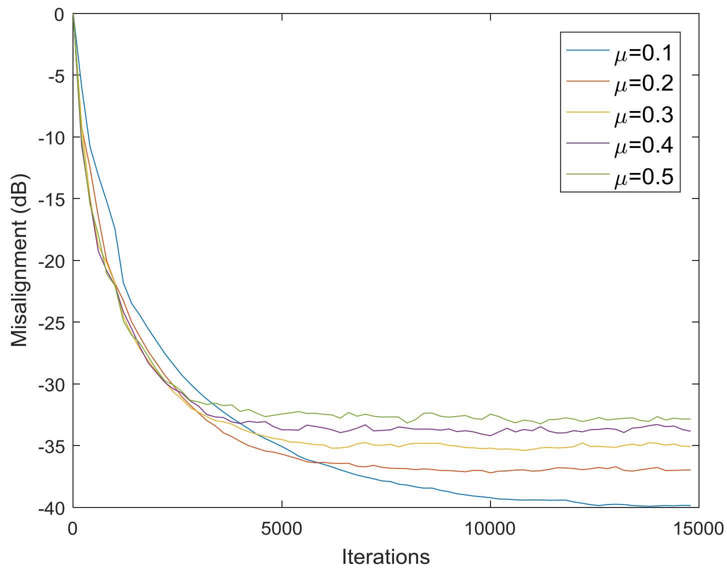

Figure 6.10 Misalignment curves of DB-IPAPA for different step-sizes $\mu = 0.1$ to $\mu = 0.5$

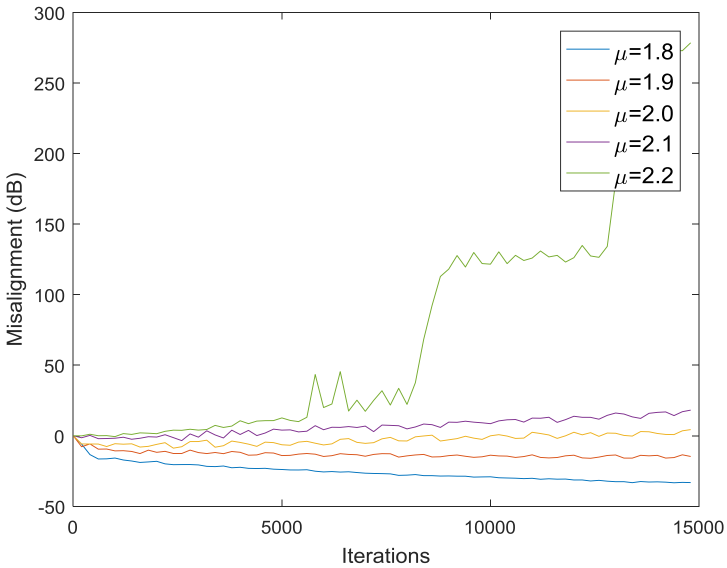

Figure 6.11 Misalignment curves of DB-IPAPA for different step-sizes $\mu = 1.8$ to $\mu = 2.2$

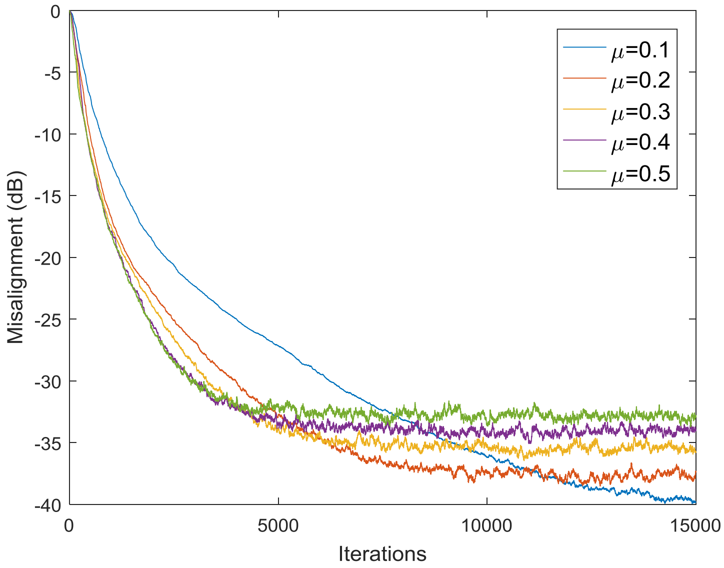

Figure 6.12 Misalignment curves of IPAPA for different step-sizes $\mu = 0.1$ to $\mu = 0.5$

6.3.6. Effect of α Parameter

Effects of the choice of the control variable α is investigated. Actually, α is a predetermined parameter depending on the sparseness of the impulse response. However, it is more reasonable to determine a value for α which yields considerably good performance for all cases. Therefore, a moderately sparse impulse response, EPM-2, is identified for this purpose and results are shown in Figure 6.13. It can be seen that performance of the DB-IPAPA is not significantly affected by the choice of α; hence, it is reasonable to select α as 0. Furthermore, different α values are considered for IPAPA algorithm. In Figure 6.14, it can be seen that performance of IPAPA clearly depends on the selection of α. Furthermore, it can be observed that $\alpha = 0$ provides best performance compared to other α values.

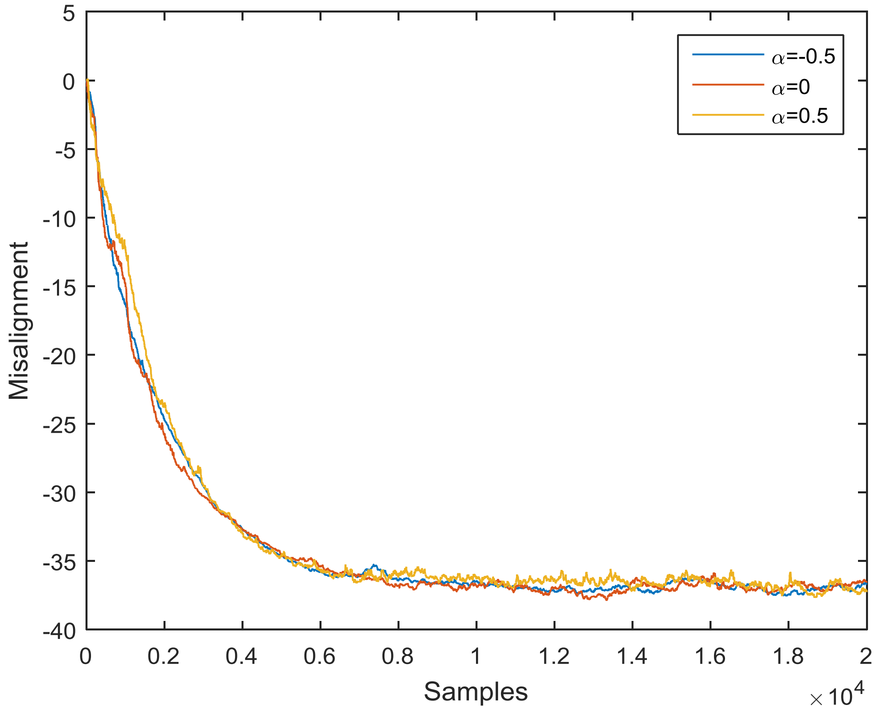

Figure 6.13 Misalignment curves of DB-IPAPA for different α values

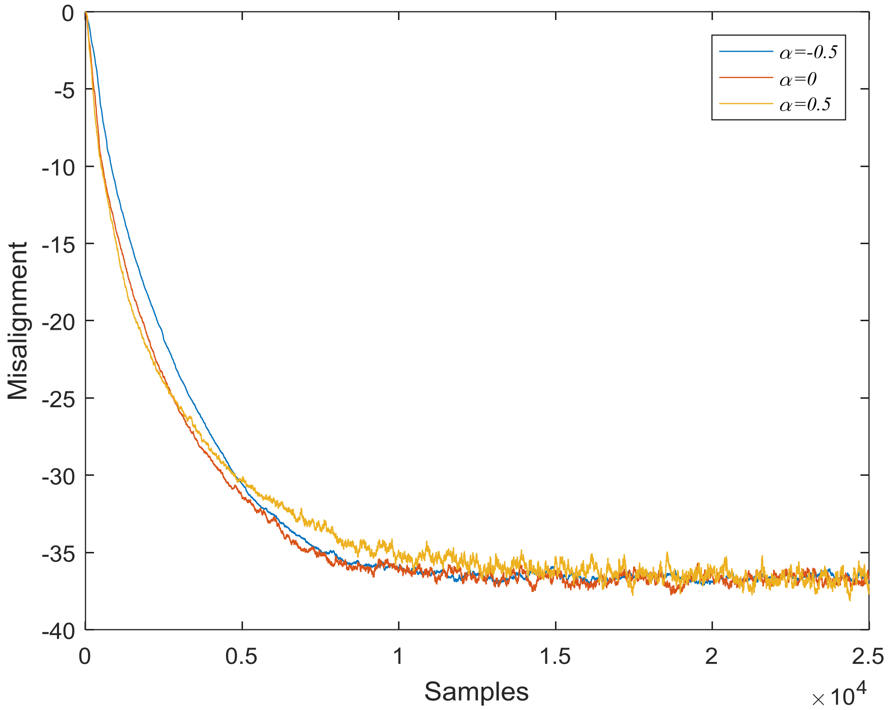

Figure 6.14 Misalignment curves of IPAPA for different α values

6.3.7. Effect of the Projection Order

Projection order is another important parameter whose effects should be studied. The effects of the projection order can be observed effectively for speech input signals. It is expected that as the projection order increases convergence speed of the proposed algorithm improves. However, increase in the projection order causes higher steady-state misalignment after $M = 8$. Consequently, projection order is selected as $M = 8$ for simulations. This convergence behavior for different M values, which is shown in Figure 6.15, is expected as stated in Section 2.6.

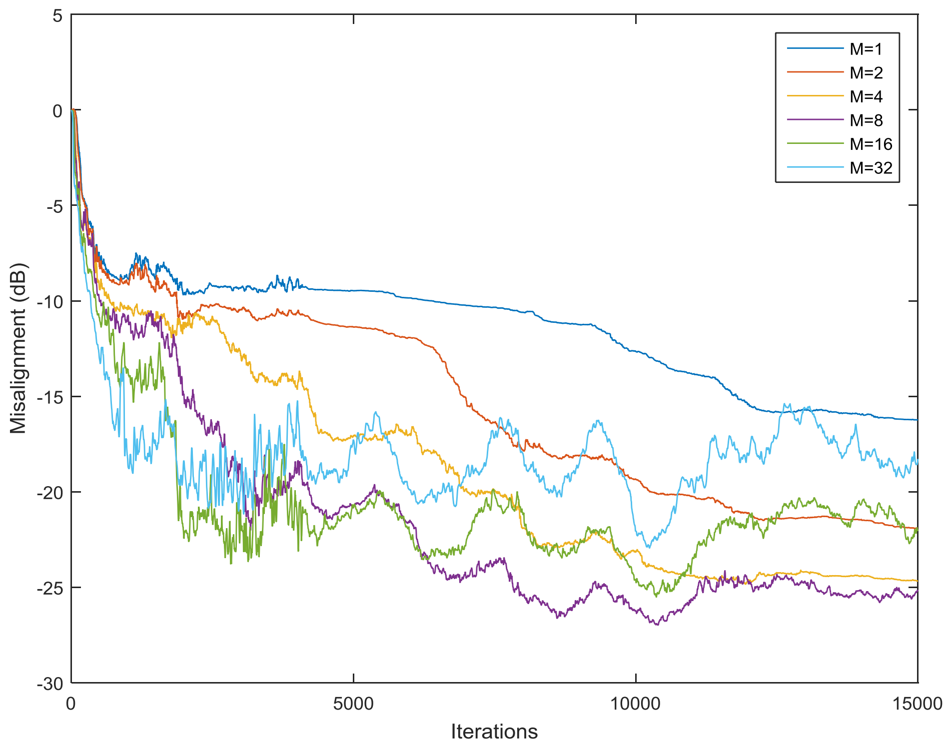

Figure 6.15 Misalignment curves of the proposed algorithm for different projection order

6.3.8. Effect of SNR

Misalignment curves of the DB-IPAPA is also obtained for different SNR values. Figure 6.16 shows that as SNR decreases steady state misalignment decreases. This result is consistent with misalignment expressions in [29], [11] which linearly depends on the noise power as

$$\sigma_r^2 \approx \sigma_v^2 \frac{\mu}{2-\mu}, \tag{6.1}$$

where σ_r^2 is the steady state misalignment and σ_v^2 is the noise power.

In addition to this analysis, DB-IPAPA is compared with the other algorithms for low SNR case. Figure 6.17 demonstrate the results for $SNR = 15\ dB$ and it can be seen that the superiority of the proposed algorithm remains. However, it can also be concluded that performances of the algorithms get close to each other compared to that of higher SNR case.

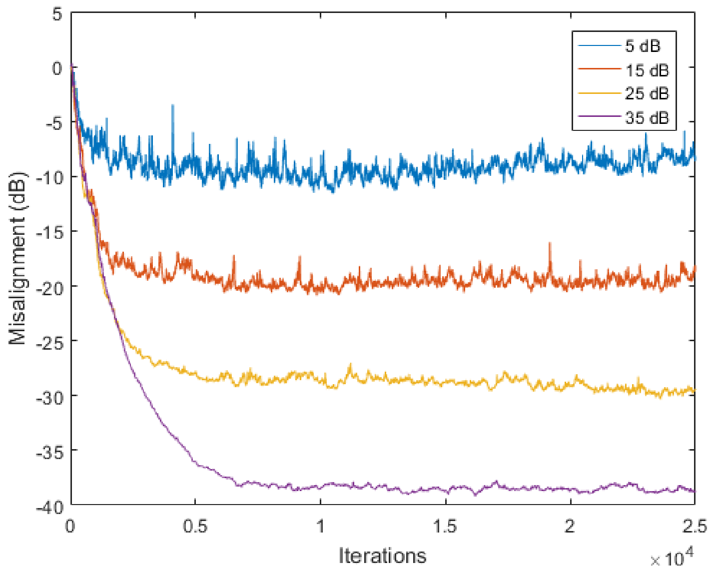

Figure 6.16 Misalignment curves of the proposed algorithm for different SNR values

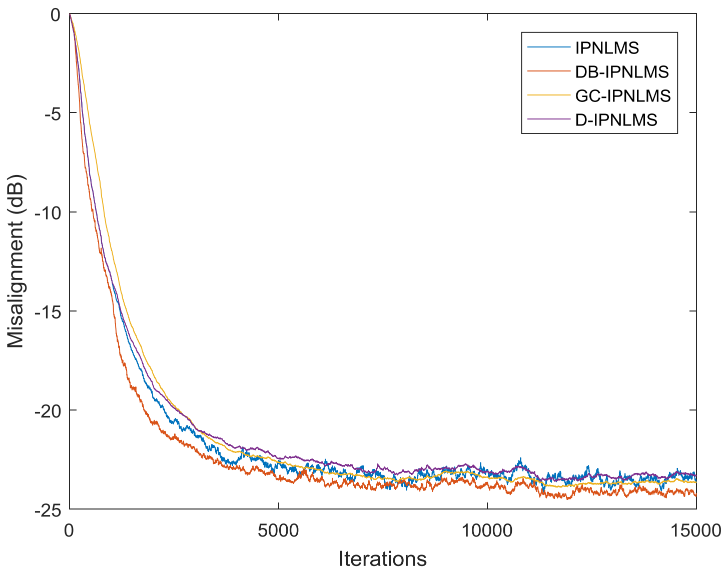

Figure 6.17 Performance comparison of the proportionate type algorithms for SNR=15 dB case

6.3.9. Selection of L_m value

In Section 5.3, it is stated that selection of L_m is an important issue. Therefore, performance of the proposed DB-IPAPA is evaluated for different L_m values. For this purpose both AR(1) and speech input signals are considered since choice of L_m value depends on the spectral characteristics of the input signal. Figure 6.18 shows misalignment curves for different L_m values for AR(1) input signal. Figure 6.18 illustrates that as the L_m increases performance of DB-IPAPA degrades since derivative information becomes redundant. Consequently, L_m is selected as L for AR(1) input signal. Figure 6.19 shows misalignment curves for different L_m values for speech input signal. In this case, higher $L_m$values provide superior performance up to a certain value. After that value performance degradation starts. According to Figure 6.19, L_m is selected as $4L$ for speech input signal.

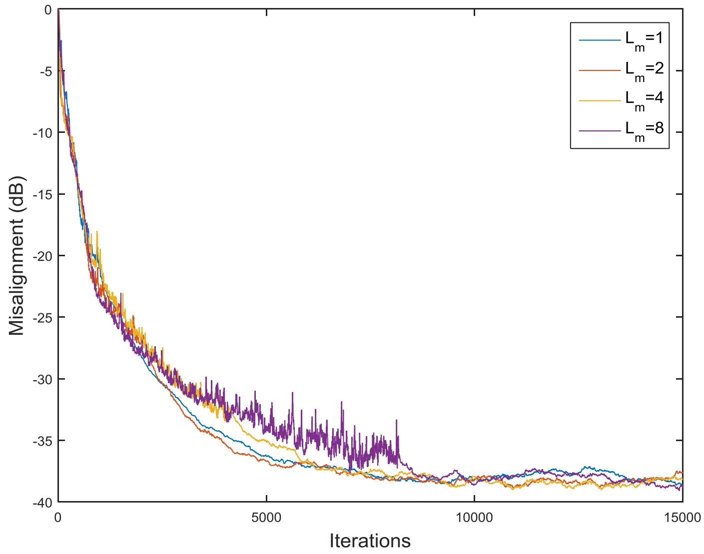

Figure 6.18 Performance of DB-IPAPA for different $L_m$values for AR(1) input signal

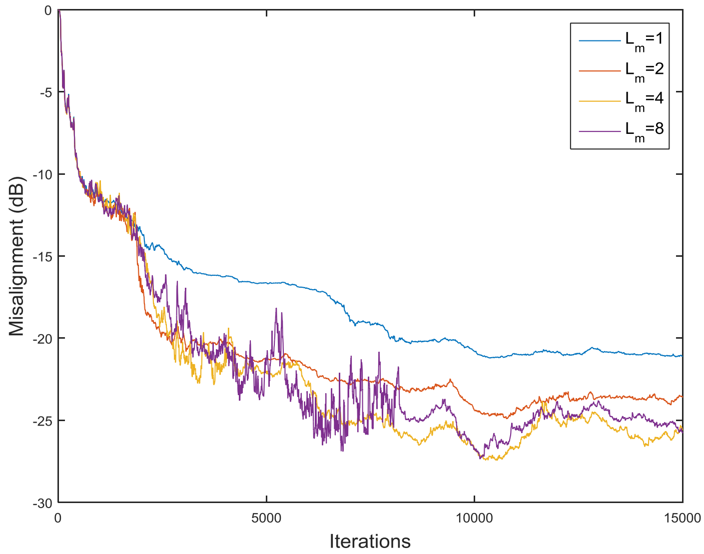

Figure 6.19 Performance of DB-IPAPA for different $L_m$values for speech input signal

6.3.10. Misalignment Performances of Proportionate Type Sign Algorithms

Simulations are also conducted for the cases where non-Gaussian interference dominates the background noise. Firstly, impulsive noise generated by a BG process is added to the system such that SIR is $-30\ dB$ and input signal of the channel is generated by AR (1) process. In addition to impulsive noise, a white Gaussian noise is added to the system to have 30 dB SNR. The same channel configuration is used as previous analysis, where channel is EPM-2. Performance of the proposed DB-IPAPSA is compared to that of PAPSA and IPAPSA algorithms. Step-sizes for the sign algorithms are selected as $\mu = 0.004$ for PAPSA and IPAPSA, $\mu = 0.04$ for DB-IPAPSA, projection order is $M = 2$ and block period for DB-IPAPSA is $L_m = L$. The other parameters are kept the same as previous analysis. As can be seen from Figure 6.20, DB-IPAPSA provides faster convergence compared to other algorithms.

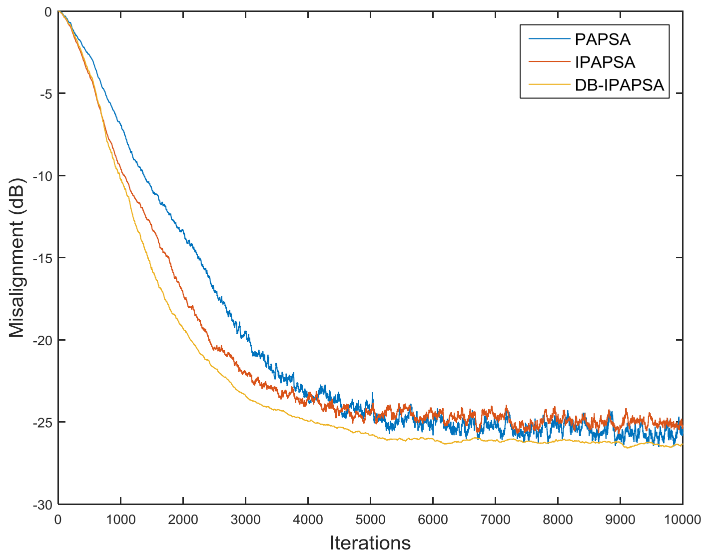

Figure 6.20 Performance comparisons of proportionate type sign algorithms for SIR=-30 dB

6.3.11. Selection of Step-size Parameter for DB-IPAPSA

Selection of the step-size parameter is also an important issue. Hence, effect of step-size parameter should also be studied in case of sign algorithms. Therefore, misalignment curves of DB-IPAPSA obtained as shown in Figure 6.21. As in the case of DB-IPAPA, steady-state misalignment is proportional to step-size value. According to simulation results, step-size is selected as $\mu = 0.04$ for simulations since it provides considerably good steady-state and convergence performance.

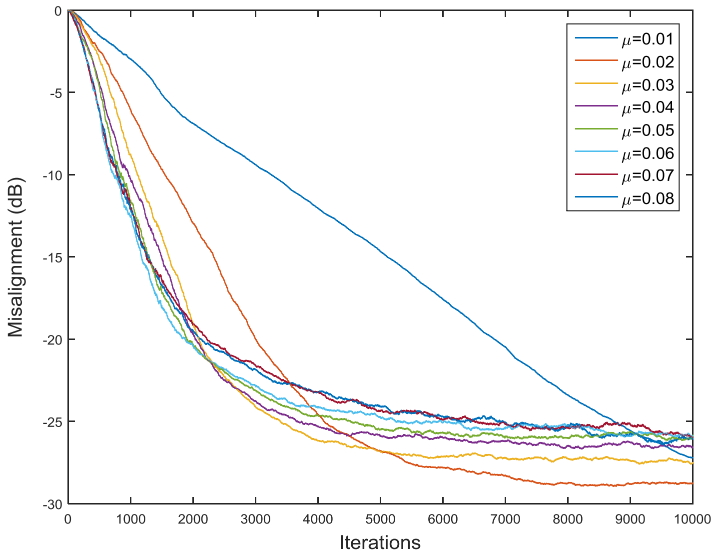

Figure 6.21 Misalignment curves for DB-IPAPSA for different step-sizes

Then, simulations are also conducted for the speech input signal. Since speech signal is used, a higher projection order, $M = 8$, is employed and block period for DB-IPAPSA is selected as $L_m = 4L$. In this case, instead of impulsive noise generated by the BG process, an actual speech signal is used as the double talk interference. Signal power to double talk power ratio is $6\ dB$ and double talk is active during samples of $[10{,}20] \times 10^4$. Figure 6.22 depicts that proposed algorithm provides robustness against double talk since there is no divergence trend in the misalignment curve. It is observed that misalignment of the proposed algorithm stays still during the double talk. In order to understand the benefit of the sign algorithms for double talk situations, Figure 6.23 shows the performances of non-sign algorithms when the desired signal is corrupted by double talk. It is clear that there is a divergence tendency of non-sign algorithms; therefore, they cannot be considered as robust in the presence of double talk.

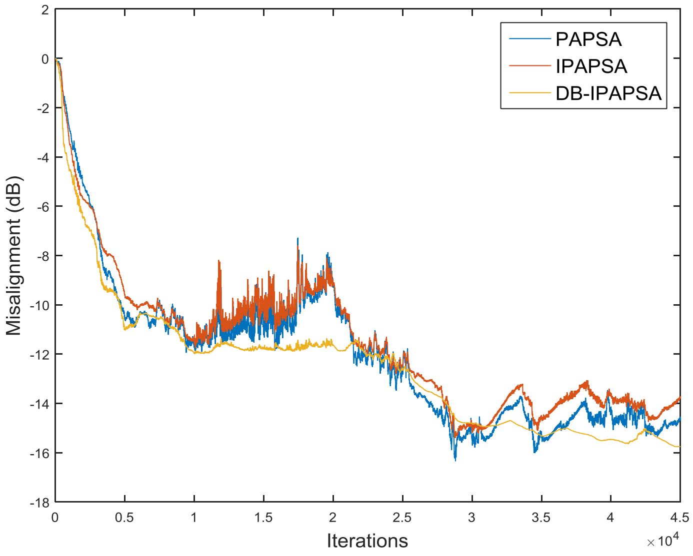

Figure 6.22 Performance comparisons of proportionate type sign algorithms when double talk occurs

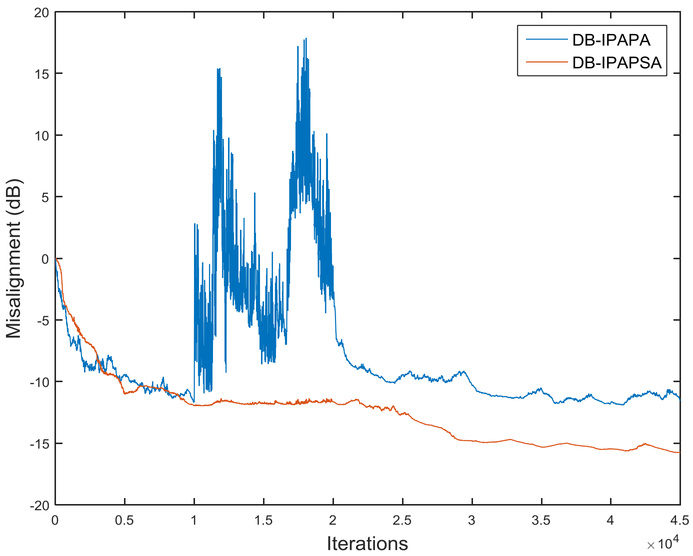

Figure 6.23 Performance comparisons of sign and non-sign algorithms when double talk occurs

6.3.12. Comparisons of Time Evolutions of Proportionate Factors of DB-IPAPA, D-IPAPA and IPAPA

In order to understand the properties of the proposed method, which makes it superior than the previously proposed algorithm, behavior of the proportionate factors should be investigated. Hence, proportionate factors of the proposed method are compared with that of IPAPA and D-IPAPA for both major minor coefficients.

Figure 6.24 shows the proportionate factors of DB-IPAPA and IPAPA for the same major coefficient. According to this figure, proportionate factor of DB-IPAPA rapidly increases to its maximum value and keeps that value for a while and decreases gradually. At the steady state, proportionate factor becomes zero. On the other hand, proportionate factor of IPAPA increases slower and after reaching steady-state it keeps its large value. Since proportionate factor of DB-IPAPA increases faster, DB-IPAPA acts to exploit the sparseness of the channels earlier. Therefore, convergence speeds of major filter coefficients are boosted in an earlier period of the adaptation. In addition, when a coefficient gets close to its optimal value, DB-IPAPA assigns smaller coefficient specific step-size to a major coefficient so that larger step-sizes can be assigned to the coefficients which are still far away from their optimal values. Consequently, convergence speeds of the smaller coefficients increase which yields improvement in the overall convergence speed. In addition, fluctuations around the optimal point are minimized due to assignment of smaller step-sizes at the steady-state. This reduces the misalignment variance at the steady state.

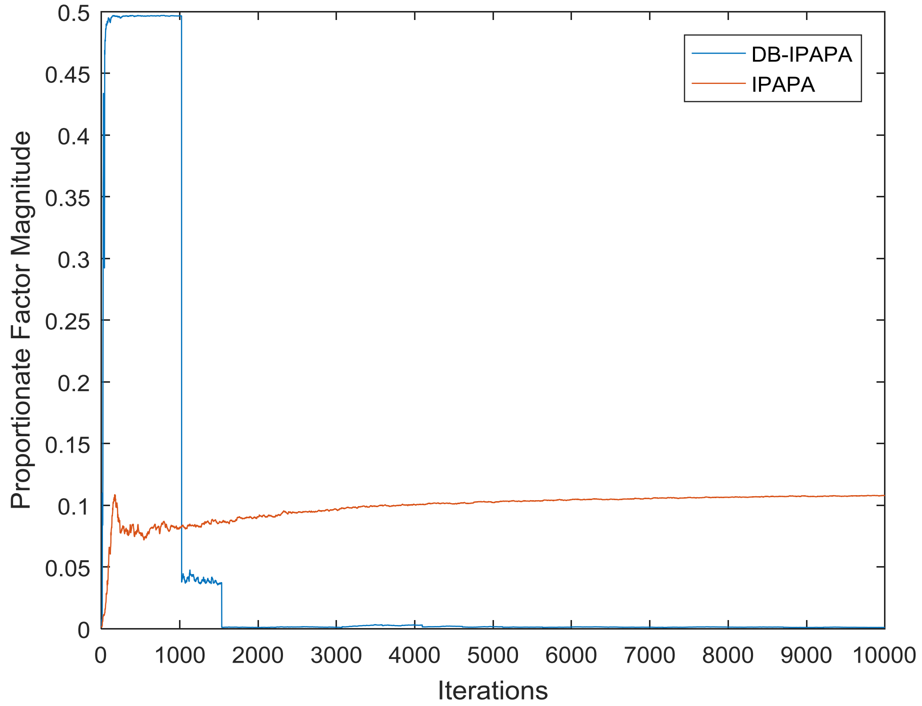

Figure 6.24 Proportionate factors of DB-IPAPA and IPAPA for a major coefficient

Figure 6.25 shows the proportionate factors of DB-IPAPA and IPAPA for the same minor coefficient. It is observed that proportionate factor of DB-IPAPA increases significantly at the beginning of the adaptation since there is no prior information regarding the locations of the major coefficients. Hence, at the initial stage DB-IPAPA behaves like conventional APA which causes higher derivative terms for minor coefficients. However, this stage lasts for only a short time interval so that it does not significantly damage the adaptation performance. In addition, even if the proportionate factor of DB-IPAPA for a minor coefficient is larger than that of IPAPA algorithm, it is still quite smaller than the proportionate factor for the major coefficient. Hence, effect of larger proportionate factor for minor coefficient at the initial stage is compensated.

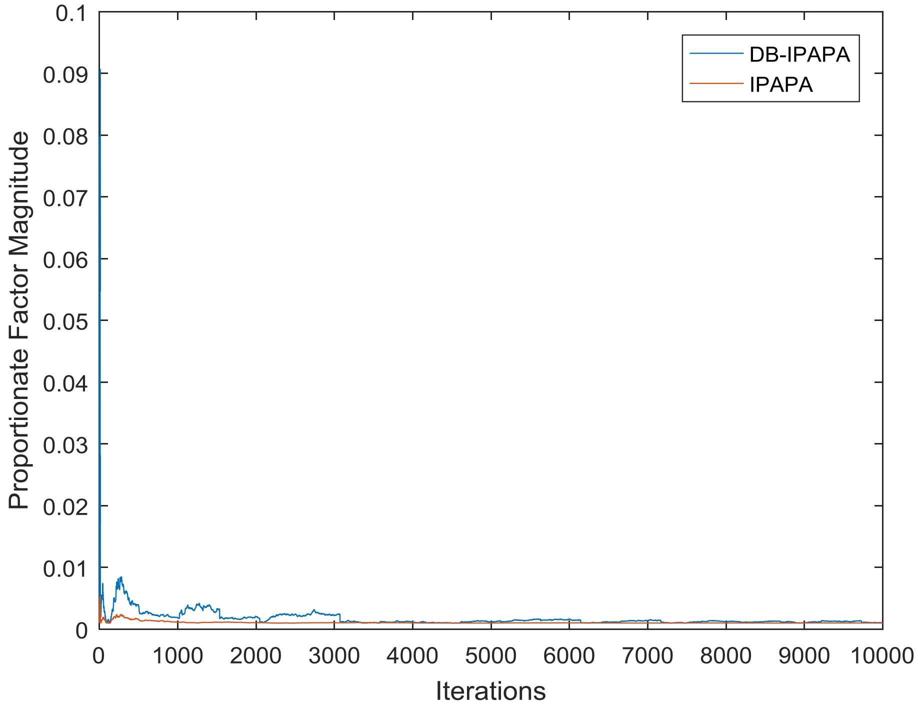

Figure 6.25 Proportionate factors of DB-IPAPA and IPAPA for a minor coefficient

Figure 6.26 shows the proportionate factors of DB-IPAPA and D-IPAPA for the same major coefficient. Expressions of DB-IPAPA and D-IPAPA may be similar; however, there are several crucial differences which makes the proposed algorithm much superior. Firstly, proportionate factor of D-IPAPA becomes zero at the beginning of each block and start to increase again as can be seen in Figure 6.26. This problem is solved by using the delayed instances of the filter coefficient such that the values of proportionate factors decrease gradually. This gradual decreasing profile is similar to the derivative profile of the filter coefficient. If the evolution of the filter coefficient is considered, there is a fast initial change in the filter coefficient, however, change becomes slower as the filter coefficient gets close to its optimal value, which is accompanied by smaller derivative terms. Consequently, by using more suitable derivative values, superior convergence performance is achieved by the proposed algorithm. Secondly, effect of the normalization technique of the proposed method can be observed from Figure 6.26. Since derivative terms become significantly smaller than the magnitude of the filter coefficients, proportionate factors of the DB-IPAPA is smaller than that of D-IPAPA at the steady-state. Therefore, steady state misalignment variance of the proposed DB-IPAPA becomes smaller.

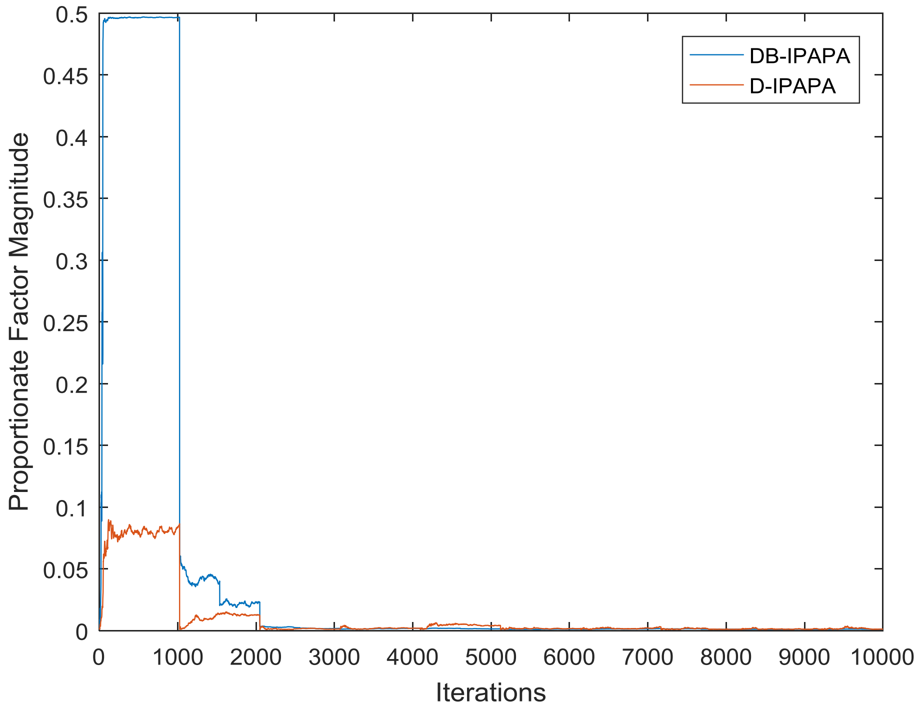

Figure 6.26 Proportionate factors of DB-IPAPA and D-IPAPA for a major coefficient

Figure 6.27 shows the proportionate factors of DB-IPAPA and D-IPAPA for the same minor coefficient. It is clear that proportionate factor of D-IPAPA is noisy and constantly larger compared to that of DB-IPAPA. This significantly degrades the performance of the D-IPAPA; therefore, performance of the D-IPAPA may be even worse than IPAPA for some cases.

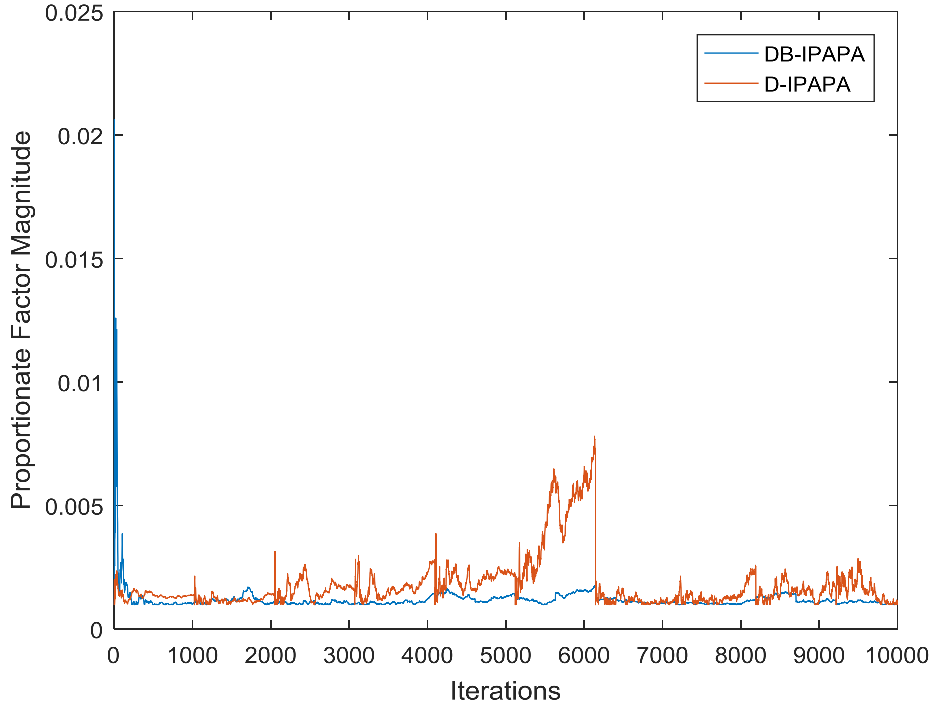

Figure 6.27 Proportionate factors of DB-IPAPA and D-IPAPA for a minor coefficient

6.3.13. A Benchmark for the Proportionate Type Algorithms

Lastly, a benchmark for the proportionate type adaptive filtering algorithms is introduced. In this case, it assumed that locations of the major coefficients are known and proportionate factors are assigned as 1 for these coefficients. For zero coefficients, proportionate factors are assigned as 0. Consequently, a limit for the convergence performance for the proportionate type algorithms is obtained. In Figure 6.28, performance of the proposed algorithm is compared with the benchmark. It can be seen that performance of the proposed algorithms is closed to the benchmark during the initial stage. However, as the adaptive filter gets close to its steady-state, its convergence speed decreases. Therefore, performance of the proposed algorithm can be improved further by speeding up the later stages of the adaptation.

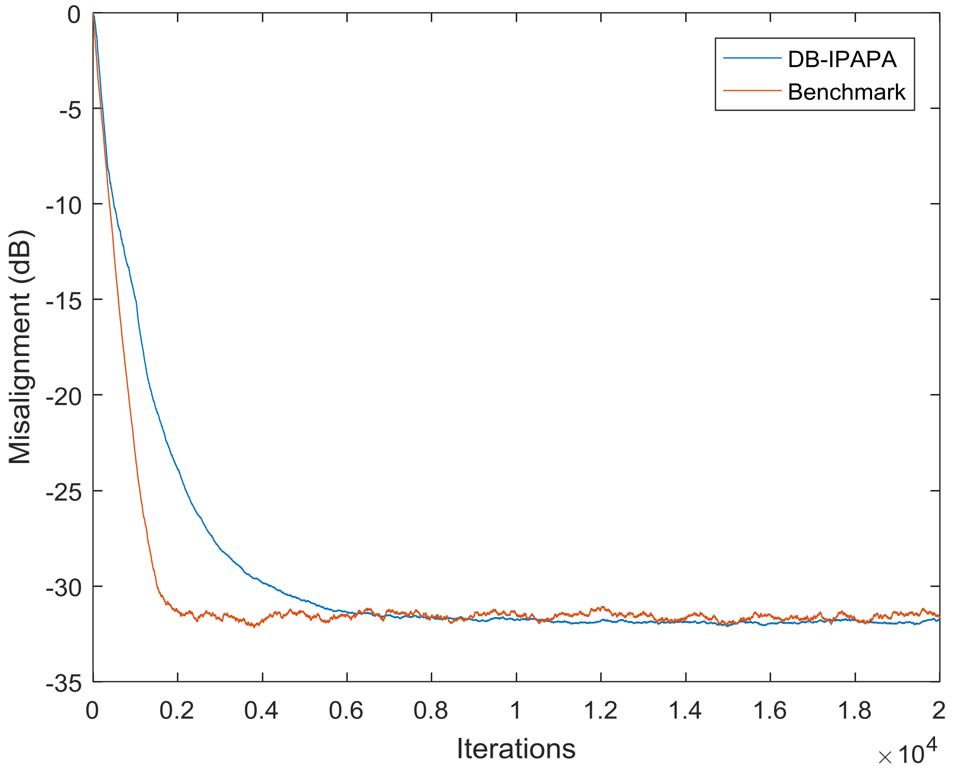

Figure 6.28 Misalignment curves of the proposed algorithm and the benchmark

CHAPTER 7

CONCLUSION AND FUTURE WORK

In this thesis, proportionate type algorithms for sparse impulse response identification have been studied. Some important proportionate type algorithms, PAPA, IPAPA, D-IPAPA and GC-IPAPA, were investigated in detail. In addition, sign algorithms, which are robust against impulsive noise were introduced. Sign extensions of the proportionate type algorithms namely, PAPSA and IPAPSA, were discussed. Lastly a “derivative” based proportionate approach was proposed in order to achieve faster convergence speed compared to the other proportionate type algorithms.

Development of the proposed method for sparse system identification was discussed in detail. Firstly, the intuitive idea behind the proposed “derivative” based approach was discussed. It was concluded that derivative information is an indicator of the distance between the current and optimal filter coefficients. This statement was also shown to be correct mathematically. Later, proposed method was applied to proportionate type algorithms to obtain DB-IPAPA and DB-IPAPSA. Finally, geometrical interpretation for the proportionate type algorithms was presented in order to geometrically show the improvement in the convergence rate.

Performances of the proposed algorithms were compared with that of previously proposed algorithms. Convergence rate and steady-state misalignment were considered while evaluating the performance of the algorithms. Simulations results have shown that proposed approach has better performance among all since it converges faster than the other algorithms. Performance comparisons were conducted for different configurations such as different sparseness levels, different input signal spectral characteristics, different SNR level, different parameter values etc.

For highly sparse channels, proposed algorithm improves the convergence speed significantly. However, as the channel impulse response becomes dispersive superiority of the proposed algorithm is not apparent as in the sparse channel case. In the case of speech input signal, proposed algorithm achieves significantly better performance compared to other proportionate type algorithms. Therefore, proposed algorithm can be considered as an attractive choice for the echo cancellation problems. Proposed algorithm also provides considerably good tracking performance. However, conventional proportionate type algorithms have better initial tracking response since they store the information about the structure of the impulse response. Although, conventional algorithms have faster initial tracking speed, proposed algorithm catches up these algorithms in the later stages of the adaptation. In addition, sign extension of the proposed approach has faster convergence speed compared to other proportionate type sign algorithms. Robustness of the proposed method to double talk situations is shown in the simulations. Therefore, proposed approach can be employed in the AEC applications which may involve double talk interference. Consequently, proposed algorithms can be used in echo cancellation applications in order to identify unknown echo path in a fast manner.

As a future work, adaptive selection of the block period, L_m can be studied to improve the convergence performance of the proposed algorithm further. Also, new methods for the calculation of the proportionate factors can be developed. Incorporation of the conventional proportionate type methods and the proposed method may provide better performance. Moreover, proposed algorithm can be applied to different applications such as equalization of sparse communication channels and non-linear sparse system identification.

REFERENCES

[1] Saberi, Ali, Anton A. Stoorvogel, and Peddapullaiah Sannuti. *Filtering theory: with applications to fault detection, isolation, and estimation*. Springer Science & Business Media, 2007.

[2] M. Bellanger, *Adaptive Digital Filters and Signal Analysis*. New York: Marcel Dekker, 1987

[3] Zaknich, A. (2006). *Principles of adaptive filters and self-learning systems*. Springer Science & Business Media.

[4] Haykin, S. S. (2008). *Adaptive filter theory*. Pearson Education India.

[5] Najim, M. (Ed.). (2013). *Digital filters design for signal and image processing*. John Wiley & Sons.

[6] Vega, L. R., and Rey, H. (2012). *A rapid introduction to adaptive filtering*. Springer Science & Business Media.

[7] Benesty, J., Gänsler, T., Morgan, D. R., Sondhi, M. M., and Gay, S. L. (2001). *Advances in network and acoustic echo cancellation*. Berlin: Springer.

[8] J. Liu and S. L. Grant, "Proportionate Adaptive Filtering for Block-Sparse System Identification," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 4, pp. 623-630, April 2016. .

[9] Nagumo, J. I., and Noda, A. (1967). A learning method for system identification. *IEEE Transactions on Automatic Control*, *12*(3), 282-287.

[10] Ozeki, K. (2016). *Theory of affine projection algorithms for adaptive filtering*. Springer Japan.

[11] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," in *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508-518, Sept. 2000 .

[12] T. Gansler, J. Benesty, S. L. Gay and M. M. Sondhi, "A robust proportionate affine projection algorithm for network echo cancellation," *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, Istanbul, Turkey, 2000, pp. II793-II796 vol.2.

[13] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Orlando, FL, 2002, pp. II-1881-II-1884.

[14] Hongyang Deng and M. Doroslovacki, "Improving convergence of the PNLMS algorithm for sparse impulse response identification," in *IEEE Signal Processing Letters*, vol. 12, no. 3, pp. 181-184, March 2005.

[15] F. d. C. de Souza, O. J. Tobias, R. Seara and D. R. Morgan, "A PNLMS Algorithm With Individual Activation Factors," in *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2036-2047, April 2010.

[16] Yilun Chen, Y. Gu and A. O. Hero, "Sparse LMS for system identification," *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, 2009, pp. 3125-3128.

[17] J. Benesty, C. Paleologu and S. Ciochina, "Proportionate Adaptive Filters From a Basis Pursuit Perspective," in *IEEE Signal Processing Letters*, vol. 17, no. 12, pp. 985-988, Dec. 2010.

[18] O. Hoshuyama, R. A. Goubran and A. Sugiyama, "A generalized proportionate variable step-size algorithm for fast changing acoustic environments," *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Que., 2004, pp. iv-iv.

[19] J. Yang and G. E. Sobelman, "A gradient-controlled proportionate technique for acoustic echo cancellation," *2013 Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, 2013, pp. 1941-194.

[20] L. Liu, "On improvement of proportionate adaptive algorithms for sparse impulse response," Ph.D. dissertation, Dept. Information Systems Engineering, Kochi University of Technology, Kochi, Japan, 2009.

[21] T. Shao, Y. R. Zheng and J. Benesty, "An Affine Projection Sign Algorithm Robust Against Impulsive Interferences," in *IEEE Signal Processing Letters*, vol. 17, no. 4, pp. 327-330, April 2010.

[22] Benesty, J., Paleologu, C., Gänsler, T., and Ciochină, S. (2011). *A perspective on stereophonic acoustic echo cancellation* (Vol. 4). Springer Science & Business Media.

[23] T. Gansler, S. L. Gay, M. M. Sondhi and J. Benesty, "Double-talk robust fast converging algorithms for network echo cancellation," in *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 656-663, Nov. 2000.

[24] O. Arikan, A. Enis Cetin and E. Erzin, "Adaptive filtering for non-Gaussian stable processes," in *IEEE Signal Processing Letters*, vol. 1, no. 11, pp. 163-165, Nov. 1994.

[25] Z. Yang, Y. R. Zheng and S. L. Grant, "Proportionate Affine Projection Sign Algorithms for Network Echo Cancellation," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2273-2284, Nov. 2011.

[26] Kay, S. M. (1993). *Fundamentals of statistical signal processing: Estimation Theory* (Vol. 1). Prentice-Hall.

[27] Wiener, N. Extrapolation, Interpolation, and Smoothing of Stationary Time Series. 1949. *New York*.

[28] B. Widrow and M. E. Hoff, Jr.. Adaptive switching circuits. *IRE WESCON Conv. Rec.*, Pt., 1960, pp. 96–104.

[29] Sayed, A. H. (2003). *Fundamentals of adaptive filtering*. John Wiley & Sons.

[30] Paleologu, C., Benesty, J., and Ciochina, S. (2010). Sparse adaptive filters for echo cancellation. *Synthesis Lectures on Speech and Audio Processing*, *6*(1), 1-124.

[31] Shao, Min, and Chrysostomos L. Nikias. "Signal processing with fractional lower order moments: stable processes and their applications." *Proceedings of the IEEE* 81.7 (1993): 986-1010.

[32] C. Paleologu, S. Ciochina and J. Benesty, "An Efficient Proportionate Affine Projection Algorithm for Echo Cancellation," in *IEEE Signal Processing Letters*, vol. 17, no. 2, pp. 165-168, Feb. 2010.

[33] Int. Telecommun. Union, ITU-T G.168 Std., 1997

[34] C. Paleologu, J. Benesty and F. Albu, "Regularization of the improved proportionate affine projection algorithm," *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, 2012, pp. 169-172.

[35] M. B. Salman and T. Çiloğlu, "Gradient controlled improved proportionate affine projection sign algorithm," *2018 26th Signal Processing and Communications Applications Conference (SIU)*, Izmir, 2018, pp. 1-4.

[36] J. Haupt, W. U. Bajwa, G. Raz and R. Nowak, "Toeplitz Compressed Sensing Matrices With Applications to Sparse Channel Estimation," in *IEEE Transactions on Information Theory*, vol. 56, no. 11, pp. 5862-5875, Nov. 2010.

[37] C. R. Berger, S. Zhou, J. C. Preisig and P. Willett, "Sparse Channel Estimation for Multicarrier Underwater Acoustic Communication: From Subspace Methods to Compressed Sensing," in *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1708-1721, March 2010.

[38] C. R. Berger, Z. Wang, J. Huang and S. Zhou, "Application of compressive sensing to sparse channel estimation," in *IEEE Communications Magazine*, vol. 48, no. 11, pp. 164-174, November 2010.

APPENDICES

Appendix A

STABILITY OF STEEPEST DESCENT

Update process of steepest descent algorithms is composed of a feedback loop; therefore, it may suffer from instability [4]. In order to provide a stable operation of steepest descent algorithms, step-size parameter should be chosen carefully since it controls the update gain of the system. Stability of the algorithms is ensured if the magnitude of each coefficient error monotonically decreases. Consider the update equation of steepest descent algorithm,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu\left[\boldsymbol{r}_{xy} - \boldsymbol{R}_x\boldsymbol{w}(n)\right], \tag{A.1}$$

then coefficient error vector at time $n+1$ becomes,

$$\widetilde{\boldsymbol{w}}(n+1) = \widetilde{\boldsymbol{w}}(n) - \mu[\boldsymbol{R}_x\boldsymbol{h} - \boldsymbol{R}_x\boldsymbol{w}(n)], \tag{A.2}$$

$$\widetilde{\boldsymbol{w}}(n+1) = \widetilde{\boldsymbol{w}}(n) - \mu\boldsymbol{R}_x\widetilde{\boldsymbol{w}}(n), \tag{A.3}$$

$$\widetilde{\boldsymbol{w}}(n+1) = [\boldsymbol{I} - \mu\boldsymbol{R}_x]\widetilde{\boldsymbol{w}}(n). \tag{A.4}$$

If input is colored input then $\boldsymbol{R}_x$ can be decomposed as,

$$\boldsymbol{R}_x = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^H, \tag{A.5}$$

which is called eigenvalue decomposition and $\boldsymbol{Q}$ is the unitary matrix, $\boldsymbol{Q}\boldsymbol{Q}^H = \boldsymbol{I}$ and $\boldsymbol{\Lambda}$ is diagonal matrix whose diagonal elements are the eigenvalues of $\boldsymbol{R}_x$. Then (A.4) can be written as,

$$\widetilde{\boldsymbol{w}}(n+1) = [\boldsymbol{I} - \mu \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^H]\widetilde{\boldsymbol{w}}(n), \tag{A.6}$$

$$\widetilde{\boldsymbol{w}}(n+1) = [\boldsymbol{Q}\boldsymbol{Q}^H - \mu \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^H]\widetilde{\boldsymbol{w}}(n), \tag{A.7}$$

$$\boldsymbol{Q}^H\widetilde{\boldsymbol{w}}(n+1) = [\boldsymbol{I} - \mu \boldsymbol{\Lambda}]\boldsymbol{Q}^H\widetilde{\boldsymbol{w}}(n). \tag{A.8}$$

It can be observed from (A.8) that $\widetilde{\boldsymbol{w}}$ is subject to a coordinate transformation. Transformed coefficient error vector is defined as, $\boldsymbol{v}(n) = \boldsymbol{Q}^H\widetilde{\boldsymbol{w}}(n+1)$, then (A.8) becomes,

$$\boldsymbol{v}(n+1) = [\boldsymbol{I} - \mu \boldsymbol{\Lambda}]\boldsymbol{v}(n). \tag{A.9}$$

Since $\boldsymbol{\Lambda}$ is a diagonal matrix (A.9) can be written for k^{th} coefficient,

$$v_k(n+1) = [1 - \mu\lambda_k]v_k(n), \tag{A.10}$$

λ_k is the k^{th} eigenvalue of $\boldsymbol{R}_x$. If the initial value of the error of k^{th} coefficient is $v_k(0)$, $v_k(n+1)$ can be written in terms of $v_k(0)$ as,

$$v_k(n+1) = [1 - \mu\lambda_k]^n v_k(0). \tag{A.11}$$

It is clear that in order to ensure the stability $1 - \mu\lambda_k$ term should satisfy,

$$-1 < 1 - \mu\lambda_k < 1, \tag{A.12}$$

which yields,

$$0 < \mu < \frac{2}{\lambda_k}, \tag{A.13}$$

Since eigenvalues of $\boldsymbol{R}_x$ are real and positive. (A.13) is valid only for a specific coefficient. These result can be generalized as,

$$0 < \mu < \frac{2}{\lambda_{max}}, \tag{A.14}$$

where λ_{max} is the maximum of the eigenvalues.

Appendix B

DERIVATION OF AFFINE PROJECTION ALGORITHM

Constraint optimization problem for APA is defined as,

$$\min_{\boldsymbol{w}(n+1)} \|\boldsymbol{w}(n+1) - \boldsymbol{w}(n)\|_2^2$$
$$\text{subject to } \boldsymbol{y}(n) = \boldsymbol{X}^T(n)\boldsymbol{w}(n+1). \tag{B.1}$$

This problem can be solved by using Langrange multipliers method. Hence, following cost function can be defined,

$$J = \|\boldsymbol{w}(n+1) - \boldsymbol{w}(n)\|_2^2 + \boldsymbol{\lambda}[\boldsymbol{y}(n) - \boldsymbol{X}^T(n)\boldsymbol{w}(n+1)], \tag{B.2}$$

where, $\boldsymbol{\lambda}$ is the $L \times M$ Lagrange multiplier matrix. (B.2) can be minimized by taking derivative and equating to zero,

$$\frac{\partial J}{\partial \boldsymbol{w}(n+1)} = 0 \tag{B.3}$$

$$\frac{\partial J}{\partial \boldsymbol{w}(n+1)} = 2\big(\boldsymbol{w}(n+1) - \boldsymbol{w}(n)\big) - \boldsymbol{X}(n)\boldsymbol{\lambda}^T = 0. \tag{B.4}$$

Then update equation becomes,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \frac{1}{2}\boldsymbol{X}(n)\boldsymbol{\lambda}^T. \tag{B.5}$$

In order to find $\boldsymbol{\lambda}$, (B.5) is inserted to constraint in (B.1),

$$\boldsymbol{y}(n) - \boldsymbol{X}^T(n)\boldsymbol{w}(n) + \frac{1}{2}\boldsymbol{X}^T(n)\boldsymbol{X}(n)\boldsymbol{\lambda}^T = \boldsymbol{0}, \tag{B.6}$$

$$\boldsymbol{e}(n) - \frac{1}{2}\boldsymbol{X}^T(n)\boldsymbol{X}(n)\boldsymbol{\lambda}^T = \boldsymbol{0}, \tag{B.7}$$

$$\boldsymbol{\lambda}^T = 2[\boldsymbol{X}^T(n)\boldsymbol{X}(n)]^{-1}\boldsymbol{e}(n). \tag{B.8}$$

Overall update equation is obtained by replacing the $\boldsymbol{\lambda}^T$ in (B.5) with the expression in (B.8),

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \boldsymbol{X}(n)[\boldsymbol{X}^T(n)\boldsymbol{X}(n)]^{-1}\boldsymbol{e}(n), \tag{B.9}$$

in order to avoid singularity in the matrix inversion a small constant is added to the matrix whose inverse is taken,

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \boldsymbol{X}(n)[\boldsymbol{X}^T(n)\boldsymbol{X}(n) + \delta\boldsymbol{I}]^{-1}\boldsymbol{e}(n). \tag{B.10}$$