FILTERED-X LEAST MEAN FOURTH FXLMF AND LEAKY-FXLMF ADAPTIVE ALGORITHMS: STOCHASTIC ANALYSIS AND SECONDARY PATH MODELING ERROR

BY

ALI MAHMOUD ALI ALOMOUR

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

MAY 2015

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **ALI MAHMOUD ALI ALOMOUR** under the direction his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN TELECOMMUNICATION ENGINEERING.**

azzedine zerguine

Dr. Azzedine Zerguine
(Advisor)

Dr. Ali Al-Shaikhi
Department Chairman

Dr. Abdelmalek Zidouri
(Member)

Dr. Salam A. Zummo
Dean of Graduate Studies

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

Dr. Tareq AlNaffouri
(Member)

13/6/15

Date

Dedicated to my parents, my brothers and sisters, my family, and my friends

ACKNOWLEDGMENTS

In the Name of Alláh, the Most Gracious, and the Most Merciful

All praises and thanks are due to Alláh, the Lord of all that exists. May the peace and blessings of Alláh be upon Muhammad, the Messenger of Alláh, and his Family and Companions and all who follow them in righteousness until the Day of Judgement.

It’s my privilege to release the feeling of my deepest gratitude and immeasurable appreciation and for the help and support are extended to the following persons who in some way or another have contributed in making this study possible.

My parents Mr. Mahmoud and Salma Al Omour, my brothers Amer and Adel, my sisters Asma, Israa, Zainab, and Waffa for their love, care, support, encouragement, patience and prayers, may Alláh blessings them with health and happiness

Dr Azzedine Zerguine my advisor for his support, guidance, advices, valuable comments, suggestions, sharing his knowledge and by giving his endless helped to finish this thesis.

Mr Mohannad Hijawi, Yazeed Al Omour , and Mr Mostafa Musleh my uncles my role model, for their immeasurable support and advice.

My professors and teachers, special mention for **Dr Abdelmalek Zidouri, Dr. Tareq Naffouri, Dr Raghda Abu Zaiton, Dr Maher Khammash, Dr Jamal Kharoosheh, Dr Waleed ba hamad, Mr Nomaan Naif, Mr Ali Abu Taleb, Mr Rafeeq Jaradat, Mr Jamal Mahameed.**

My friends, Ahmad Shawahna, Saif Najmaldeen, Shadi Abu abdo, Fadi Abu Samra, Ismail Manasra, Khaled Shaar, Shadi Abu Samra, Islam Othman, Hussain Abu baker, Abdulrahman Zebn, Khalid Assi, Ahmad AlFarra, Omar Sawafi, Mahmoud Shawahneh, Saber Barham, Mohammad Omari, Ahmad arman, Baker Abu Alhija, Nizar Ibrahim, Shouq Mohsen, Kamel Bshara, Mahmoud Janem, Muath Najjar, Amjad Abu Hassan, Amjad and Majdi Ayasih , Navid Iqbal, Motasim Zidan, Ahmad Murad, Bara Dwikat, Khalil Adham, Mohammad Omour, Abduallah and Ayman Manasra, Mahmoud Taraira, Ali Al Biladi, Mohammad T. Al Omour, Yousif Shanweir, Qadri Mayaleh, Anas Khateeb, Amro Ayad, Luqman Marabeh, Hammam Saeed, Ameen Ogool, Saeed Abdullah, Mohammd Jammal, Mohammad Barghothi, Haitham Hannon, Mr Sohaib Azzam, Mr Kamal Shannak, and to all friends and lovers

May the Almighty Alláh richly bless all of you.

Your sincerely

Ali Al Omour

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADC	:	Analog to Digital Convertor
ANC	:	Active Noise Control.
DAC	:	Digital to Analog Convertor.
DSP	:	Digital Signal Processing.
FXLMF	:	Filtered X Least Mean Fourth.
FXLMS	:	Filtered X Least Mean Square.
IID	:	Independent Identically Distributed.
IT	:	Independence theory.
LFXLMF	:	Leaky Filtered X Least Mean Fourth.
LFXLMS	:	Leaky Filtered X Least Mean Square.
LLMS	:	Leaky Least Mean Square.
LMF	:	Least Mean Fourth.
LMS	:	Least Mean Square.
MFXLMS	:	Modified Filtered X Least Mean Square.
MMSE	:	Minimum Mean Square Error.
MSE	:	Mean Square Error.

NLMF	:	Normalized Least Mean Fourth.
SNR	:	Signal to Noise Ratio.

ABSTRACT

Full Name : ALI MAHMOUD ALI AL OMOUR

Thesis Title : FILTERED-X LEAST MEAN FOURTH FXLMF AND LEAKY-FXLMF ADAPTIVE ALGORITHMS: STOCHASTIC ANALYSIS AND SECONDARY PATH MODELING ERROR

Major Field : TELECOMMUNICATION ENGINEERING

Date of Degree : May 2015

Adaptive Filtering Algorithms show promise for the improvement of the Active Noise Control (ANC) problem. Several adaptive algorithms have been developed and utilized for this purpose. Just to name a few, the Filtered-x Least Mean Square (FXLMS) algorithm, the Leaky version of FXLMS (LFXLMS) algorithm, and other modified LMS algorithms. On the other hand, the Least Mean Fourth (LMF) algorithm proves that it can outperform the LMS algorithm under special circumstances. In this work, we are proposing two new adaptive filtering algorithms, which are the Filtered-x Least Mean Fourth (FXLMF) algorithm and the Leakage-based variant (LFXLMF) of the FXLMF algorithm. The main target of this work is to derive the FXLMF and LFXLMF adaptive algorithms, study their convergence behaviors, examine their tracking and transient conduct, and analyze their performance for different noise environments. Moreover, a convex combination filter utilizing the proposed algorithm and algorithms robustness test is carried out.

Finally, several simulation results are conducted to validate the theoretical findings, and show the effectiveness for FXLMF and LFXLMF algorithms over other adaptive algorithms.

ملخص الرسالة

الاسم الكامل: علي محمود علي العمور

عنوان الرسالة: خوارزمية تصفية المدخل لأقل متوسط رباعي و خوارزمية ترشيح المدخل لأقل متوسط رباعي : تحليل احصائي و استخدام مسار ثانوية لنموذج الخطأ

التخصص:هندسة الإتصالات

تاريخ الدرجة العلمية:مايو 2015م

خوارزميات التصفية المتكيّفة أضهرت تقدماً واعداً في مشكلة مكافحة الضوضاء النشطة (ANC). وقد و وضعت عدة خوارزميات تصفية متكيفة و بغرض الأستفادة منها في هذه المشكلة, على سبيل المثال لا الحصر خوارزمية (FXLMS), و خوارزمية (LFXLMS) و خوارزمية (LMS) المحسنة. من ناحية أخرى فأن خوارزمية (LMF) أثبتت تفوقها على خوارزمية (LMS) في ظل ظروف خاصة. و في هذا العمل نقدم خوارزميتين جديدتين من خوارزميات التصفية المتكيفة و هما خوارزمية (FXLMF) و خوارزمية (LFXLMF) حيث أنه سيتم من خلال هذا العمل اشتقاقهما و دراسة سولكيات التقارب و دراسة تتبع السلوك المنتقل و تحليل أدائهما في ضل بيئات مختلفة من الضوضا. و علاوة على ذلك فانه سوف يستخدم مزيج من هذه الخوارزميات المقترحة و في تشكيل خوارزمية مشتركة و اختبارات المتانة لهم.

و أخيرا العديد من نتائج المحاكاة المحوسبة سوف يتم تطبيقها للتأكد من صلاحية النظرية و تظهر فاعلية الخوارزميات المقترحة على التكيف من الخوارزميات الاخرى.

1. CHAPTER 1

INTRODUCTION

In recent years, the evolution of technology and industrial growth opened the door for researchers around the word in most fields to enhance, and create new hardware, software, and discover more things in all spheres. However, this evolution created a dark side to human life, due to its pollution, wastes and noise. Fortunately, human stubbornness and the necessity to motivate minds to find solutions for these side effects of their progression continues.

In most communication systems, the signals suffer from the noise problem in different times at generation, transmission, reception and at the destination side analysis. Different types of noise signals, mostly statistically distributed, can harmfully affect the desired signal. Understanding the type of noise signal is the first step toward mitigating its effect. Furthermore, canceling of the noise was a great inspiration for researchers to develop control methods that led to reducing the noise to an acceptable level, depending on the type and effects of the noise, and the system environment itself. One of these controlling methods is the use of adaptive filtering techniques.

Noise cancelation is one of the most important issues that adaptive filtering algorithms strive to accomplish. Adaptive filtering algorithms have created an extremely wide range of interest. Diverse applications utilize this powerful tool for systems representations. Moreover, adaptive filtering algorithms promise great developments and improvements in most of the hot topics in communications, signal processing, estimation and control, to name a few, since they show their compatibility to be influenced by other techniques like compressed sensing, and by their ability to be extended even more due to technological developments.

The concept of adaptation can be explained as the ability of a system to adjust its parameters by responding to some phenomenon in the environment surrounding this system [1][3]. There is a variety of applications in communication systems which can be supported by an adaptive system, like channel estimation, noise cancellation, adaptive equalizations and more [1],[2]. When we say adaptive filtering, then we are describing a system that can be represented by a filter that is able to change its taps or weights according to a certain procedure to adapt a particular system, taking into account the effect of the environment surrounding that system.

These procedures, that are responsible for the adaptation process of the system, and the mathematical derivations which represent the filter structure, are the algorithm that rules this system. Proving and understanding the ability of the algorithm to represent this system brings about the need to study and examine the algorithm in different circumstances by clear verifications.

Some of these verifications which are utilized to understand the behavior of the algorithm can be summarized as follows: First, the ability of the algorithm to converge, and the required time that the algorithm needs before it reaches the target of the adaptation process, referred to as either the convergence time, or the algorithm speed of convergence. Second, the stability and robustness of the algorithm, specifying the factors that determine the range of stability of the algorithm. Finally, the ability of the algorithm on tracking the changes through the transient and steady state phases, and taking into consideration the algorithm's complexity performance and the availability of resources.

1.1 Background

1.1.1 Active Noise Control

There are two main types of noise that can be found in the environment. The first one is called broadband noise. In this type, the energy of the noise spreads almost equally over the frequency band which is totally random, such as the sound from jet planes or the impulse noise due to explosions, or white noise. The second type is narrowband noise. In this type of noise, the energy spreads over a particular range of frequencies, and it is characterized by a periodical behavior since it repeats itself, like the noise caused by rotating machines due to repetitive movement. This kind of noise can be considered as predictable [5].

In general there are two basic methods to control the noise. The first one uses passive control techniques, which aims to reduce the noise effect by using materials such as a combination of tubes, sound absorbing material, barriers, enclosures or silencers. This type of noise absorbing or dispersion techniques are good for broad frequency range; however, they are still large, expensive and insufficient for low frequencies [5].

The other technique is the Active Noise Control (ANC) technique. This method can be achieved by using an additional secondary source able to play an anti-noise signal having the same magnitude, and an opposite phase estimated from the primary signal using an adaptive algorithm and depending on the superposition principle [3].

Passive control techniques are effective in reducing the noise over broadband frequencies, but still they are expensive and sometimes they are difficult to apply, so they have limited noise attenuation, especially at low frequencies. Moreover, they show low noise mitigation when the passive silencer dimension is smaller than the acoustic wavelength. By contrast, the ANC is much easier to implement and qualified to adapt a diverse range of specific frequencies. The ANC has a distinctive feature compared to the other noise cancellation techniques, which is the existence of the transfer function located between the output of the adaptive filter and the error sensing node. This transfer function is called the secondary path, which is used to eliminate the noise from the primary signal.

Broadband noise can be controlled if we are able to predict the primary noise signal which is spread over all frequencies in the broadband case. This prediction is the same as having access to the original primary signal (which is not available in most cases); after this we can generate an inverted version of the acoustic noise, opposite in phase and having the same amplitude, to ultimately cancel the noise. By contrast, it is possible to predict the narrowband noise due to its periodic behavior, which gives us a kind of access to the input signal that makes it likely to be controlled. Figure 1-1shows the concept of an active noise cancelation technique. Note that when the estimated secondary noise matches the unwanted primary noise, the technique results in a small residual noise.

The ANC technique can be used for several applications, where all of these applications aim to attenuate the unavoidable noise sources at the end equipment, such as the noise from automotive vehicles, manufacturing and industrial operations, and equipment like engines, fans, transformers and other noise sources [3]-[5].

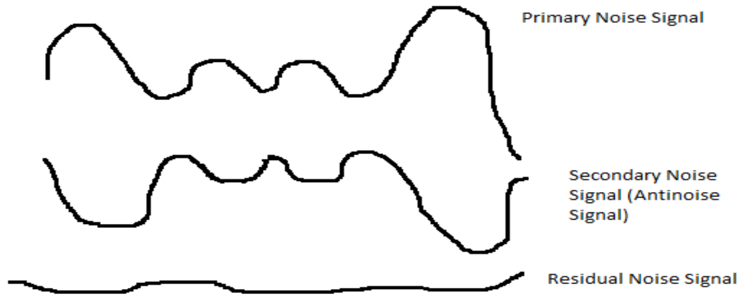
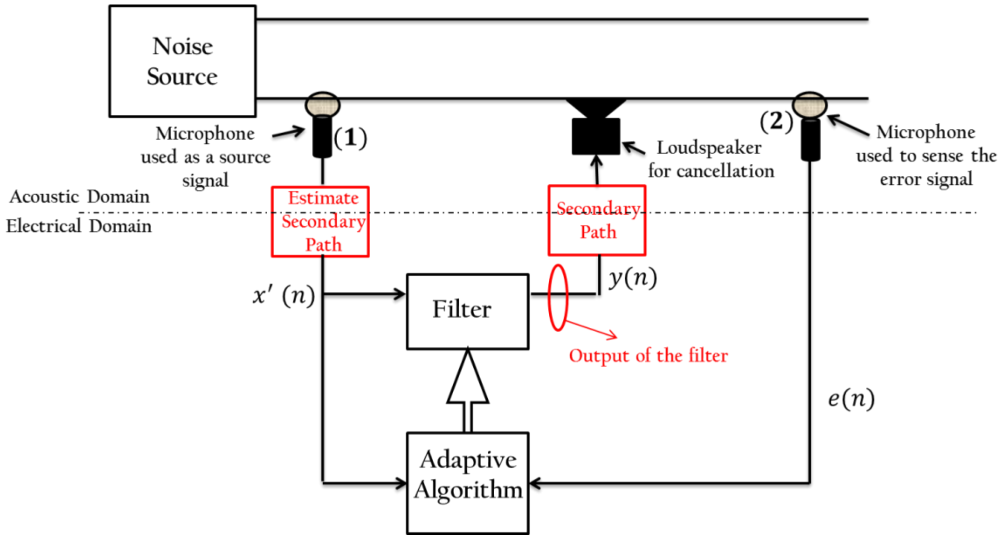


Figure 1-1: Concept of active noise cancelation

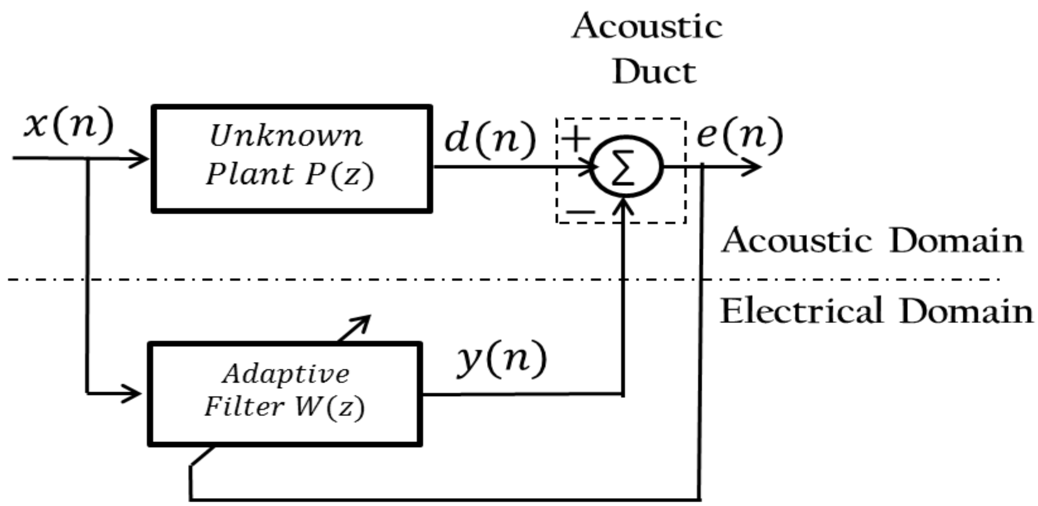
The effectiveness of ANC can be achieved by having exact precision in amplitude and phase for both the primary and the secondary noise, so they become like a mirror for each other as shown in Figure 1-1.With the new development and research in Digital Signal Processing (DSP) techniques, sampling and processing of the acoustic signal becomes easier , with precision and sufficient speed, to achieve noise cancellation effectively.

1.1.2 Filtered X Least Mean Square FXLMS Algorithm and Leaky FXLMS Algorithm

In an ANC system as mentioned previously, there is a secondary path located between the output of the adaptive filter and the error sensing node. This secondary path can be represented by a transfer function used as a superposition transformation point between the acoustic domain and electrical domain (the Adaptive filter) see Figure 1-2. This transfer function combines the DAC digital to analog convertor, reconstruction filter, power amplifier, loudspeaker, acoustic path from the loudspeaker to the error microphone, error microphone, preamplifier, anti-aliasing filter, and ADC analog to digital convertor.



(a)



(b)

Figure 1-2: (a) ANC system in duct, (b) ANC System

The presence of the secondary path after the adaptive filter requires a modification on the conventional LMS to ensure the convergence of the algorithm. The solution of this problem is to place a conformable filter to the secondary path between the reference signal and the weight update of the LMS algorithm. In other words, an estimated version of the secondary path is used to filter the reference input signal, which is later recognized as a Filtered-XLMS algorithm.

Later, the FXLMS algorithm is considered as the foundation and a direct application for ANC mitigation. The FXLMS is a modified version of the LMS algorithm. To ensure the convergence of the algorithm, and as a member of the LMS family, the FXLMS presents some problems, such as the problem of high noise level related to the low frequency resonances, which may cause nonlinear distortion by overloading the secondary source.

The solution of the overdriven problem was through the introduction of an output power constraint (called the leakage factor) for keeping the adaptive filter weights within bounds, and by limiting the power of the secondary source to avoid nonlinear distortion. Based on that, adding the leakage factor on the FXLMS algorithm, the so-called LFXLMS algorithm proposed a more robust and significant stability effect on the adaptive algorithm.

1.2 Literature Review

The importance of an adaptive filter system comes from its ability to adjust its filter parameters in response to the changes in the surrounding environment. Tracking the changes in the environment and continuously updating the filter parameters will guarantee the capability of the adaptive filter to simulate the unknown plant system [3].

Noise cancellation is one of the adaptive filtering applications which are widely used in everyday life. In addition, other diverse applications on Adaptive filtering theory, such as system identification, adaptive equalization, and plant modeling and more have been described in literature [1], [2]. Adaptive filtering provides improvement for ANC due to the filter's ability to change its weight adaptively with changes in the surrounding area to reduce the residual error, so we don't need an external source for control or physical modification. Furthermore, ANC offers potential benefits in volume, weight, cost, and size.

The first design for an ANC utilizing a microphone and loudspeaker which were energized electronically to generate the cancelling signal was proposed in 1936 by P. Lueg, [4] on his patent. The acoustic noise is characteristic of a time varying environment due to the continuous change in the phase, amplitude, frequency content and sound speed. These all required an ANC system to be adaptive [6], [7] in order to be able to handle the mentioned variations. Since then, a variety of adaptive algorithms have been developed and utilized for the purpose of ANC [3].

The most famous adaptive algorithm is the Least Mean Square (LMS) [10]. The LMS algorithm was derived by replacing the gradient vector in the steepest descent method [1]-[3]by statistical approximation of that gradient vector. The LMS algorithm states that the weight vector will converge to the optimal Winner filter if a stochastically input has been used. LMS has become the most widely adaptive algorithm used in different applications due its computational simplicity, ease of implementation, and robustness to signal statistics, and most of the adaptive algorithms arising later were modifications on the conventional LMS algorithm.

The LMS algorithm suffers from problems, such as, a degradation in the algorithm efficiency, due to the presence of a filter in the auxiliary or in the error path that. Also, slow convergence, instability of the algorithm, increased the residual noise power and lowered the convergence rate. Those constraints urged scientists to enhance the conventional LMS algorithm. Later on, in 1978, Mazo proved the independence theory experimentally [9] and then in 1984 Gardner employed the independence theory for an independent identically distributed (IID) white-input process. This proposal was a breakthrough for LMS, which makes the analysis of the LMS algorithm a short and fast process [8]-[9].

In 1981 Widrow derived the FXLMS algorithm when he was working on adaptive controls, while Burgess proposed the FXLMS algorithm for the ANC application [11]- [16], FXLMS was developed for ANC, and became the cornerstone algorithm for ANC applications; on this algorithm an identical copy of the secondary path (transfer function) is used to filter the input before it enters the algorithm in order to adjust the coefficient vector of the filter, as shown in Figure 1-3, where the main purpose of the secondary path is to solve the instability problem and to eliminate the noise from the primary signal [10].

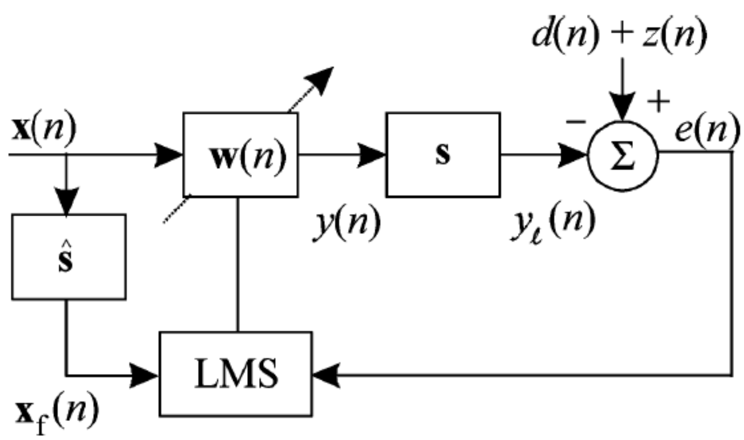


Figure 1-3: Block diagram for the FXLMS algorithm.

In the last decade, intensive research were introduced for the purpose of FXLMS algorithm modification, in literature [14] a new stochastic analysis for FXLMS algorithm was introduced, using an analytical model not based on the independence theory, to drive the first moment of the adaptive weight filter. The main assumption of this work was to ignore the correlation between the data vector and the weights and compared the correlation between data vectors, and preserved past and present data vector correlations. This model was valid for white and colored primary input signals and shows stability even when using large step sizes.

Usually, when designing adaptive algorithms, the input signal used is a stochastic signal, otherwise the algorithm will suffer from a non–Winner effect. Vicente and Masgrau [17] proposed a convergence analysis for a periodic or deterministic input signal by obtaining a strict upper boundary on the algorithm step size and applying the root locus theory to the transfer functions of the adaptive filter.

The FXLMS algorithm is preferred because of its inherent stability and simplicity, but sometimes the adaptive filter suffers from high noise levels caused by low frequency resonances, which may cause nonlinear distortion due to overloading the secondary source. This problem was solved by adding output power constraints to the cost function, as was proposed in the Leaky FXLMS algorithm LFXLMS[16][20], Moreover, the LFXLMS reduces the numerical error in the finite precision implementation and limits the output power of the secondary source to avoid nonlinear distortion; LFXLMS increases the algorithm stability, especially when a large source strength is used.

Another modification of the FXLMS algorithm[15] is the Modified FXLMS algorithm; MFXLMS proposes better convergence and reduces the computational operations, since FXLMS shows poor convergence performance.

LMS may suffer from divergence on the adaptive weight vector due to insufficient spectral excitation, like a sinusoid signal without noise, which may cause overflow for the weight vector during the updating process. The divergence problem can be solved by proposing the (Leakage) termed during the updated calculation of the weights vector [16] which results in a reduction in the adaptive filter performance, controlling the leakage factor necessary to balance between the lost performance, adding more complexity and robustness of the adaptive filter, as in the Leaky LMS.

The authors in [20], introduced a stochastic analysis for the Leaky FXLMS algorithms without employing the independence theory. Furthermore, they assumed an inexact estimation for the secondary path, which is the case in most practical implementations for the adaptive filter.

The Leaky LMS algorithm proposed in [16][18], aims to reduce the stalling, where the gradient estimate is too small to adjust the coefficients of the algorithm, due to a very low input signal. Moreover, the leakage term stabilized the LMS algorithm successfully. LLMS solved the problem of bursting in short distance telephones when we added the adaptive echo canceller [19].

A very important extension of the Windrow- Hoff LMS algorithm [10], is the Least Mean Fourth LMF algorithm [21], where the expected value of the error is raised to the power $2k$,where $k = 2,$ in the case of LMF, and $k = 1,2, \ldots,$ for higher order, where the cost function for LMF algorithm is given as the following:

$$J_{LMF}(\boldsymbol{w}) = E[{e_k}^4] \tag{1.1}$$

The LMF weights converge proportionally to the LMS weights. However, LMF outperform LMS in lower noise level for the weights at the same speed of convergence by a margin of 3-10 dB in favor of LMF under some circumstances, like the independence of the noise of the input signal; but this algorithm pays the penalty of computational complexity when compared to LMS.

As in the LMS family, many algorithms were proposed as a modified version of the LMF algorithm. The Leakage-based variant of LMF algorithm, proposed in [22], introduces a solution to reduce the weight drift problem, which appeared in the LMF algorithm. Another extension of the LMF family is the Normalized LMF algorithm [23], where NLMF shows a fast convergence and the lowest steady state, unlike LMS. Moreover, NLMF algorithm convergence is independent of the input correlation statistics.

1.3 Problem Statement

In the recent years, significant research effort was focused on ANC, trying to modify and improve it. This increase of interest derived from two major facts: the enormous development in the DSP, and the orientation on utilizing ANC to develop smart structures.

The most important fundamentals in ANC system are: s to be a fast adaptive system; in other words, the algorithm used needs to have a fast speed of convergence, to start controlling the noise immediately after predicting the input noise signal. Another necessity in the ANC system is to reduce the error between the input primary noise signal and the resulting one to an acceptable noise level, which can be achieved by insuring that the algorithms converge to the lowest weight noise level. These essential fundamentals can be verified by simulations and analytical models.

The effectiveness of ANC depends on the acoustical hardware implementation, and the signal processing of the adaptive algorithm used. Moreover, a reliable knowledge of the algorithm behavior is required for significant algorithm design. Most of the new works on this field are centered on developing the current techniques and proposing new ones.

Also, in general, there are other problems of algorithms facing adaptive filter research, such as: the algorithm's robustness, stability and validity for different types of statistically distributed noise and input signals, considerations of an acceptable level of error, and finally the hardware design and computational complexity.

1.4 Contributions

Adaptive filters received a lot of attention due to their ability to adjust the filter coefficients to minimize the error signal due to their simplicity. The Least mean-squared (LMS) algorithm family was the reference for such problems. One of the most popular algorithms in this family is the Filtered-X least mean square (FXLMS) algorithm, which is considered as the cornerstone for ANC systems. Furthermore, the LLMS and LFXLMF algorithms are well known proposed solutions for problems found in the LMS, and provide much robustness and stability for LMS algorithm, and both FXLMS and LFXLMS algorithms propose to be a great contribution for ANC.

In this work, we will propose new algorithms from the least Mean Fourth LMF algorithm family. More specifically, we are aiming to propose the filtered X least mean fourth adaptive algorithm, FXLMF, and the leakage based variant of the filtered X least mean fourth adaptive algorithm, LFXLMF. Those two algorithms are expected to have a high effectiveness on the ANC issue.

The necessity of having an algorithm with fast speed convergence and acceptable low weight noise level, as well as the promise of the outperformance of the LMF family over LMS in various aspects, are the main encouraging objectives behind our work, which will propose two new adaptive algorithms, the FXLMF and the LFXLMF adaptive algorithms. Moreover, we aim to carry out full studies for both the FXLMF and LFXLMF algorithms through analyzing the convergence behaviors, and examining the performance for both of them under different statistical input signals and noise, depending on secondary path modeling error using an energy conservation relation framework.

This work is organized into five chapters as follows: chapter one, an introduction to start the work, and then we moved to a background about the ANC and FXLMS algorithm. Later on is the literature review, followed by the problem statement of our work and the contributions we aim to achieve by the end of thesis. Next, in chapter two we will describe the methodology of our proposed work and the analytical derivations for the FXLMF and LFXLMF algorithms. In chapter three, we propose the convex combined FXLMF with the FXLMS algorithm. Simulation work done is presented in chapter four, and finally you will find in chapter five the conclusions and the future work.

2. CHAPTER 2

PROPOSED ALGORITHMS

In this chapter, the development and the derivations of the two proposed algorithms are considered. The two newly proposed algorithms are the Filtered X Least Mean Fourth FXLMF and Leaky Filtered X Least Mean Fourth LFXLMF algorithms. As can be seen from their acronyms, these two algorithms are related to their main algorithm, the LMF algorithm. Deriving from the development of these two algorithms, their respective block diagrams in the context of ANC are also designed. We will see in section 2.2 the block diagram Figure 2-2 which will help to better understand the mechanism of these algorithms. As a byproduct of these analyses, conditions to ensure stability with respect to their step size are also derived. First, we start by deriving the FXLMF algorithm and the LFXLMF algorithm will follow.

2.1 Methodology and Proposed Method

A direct and a clear example, that is usually used to explain the ANC system, is the air duct. This can be found in cars, planes, generators, just to name a few. An ANC system in the air duct can be presented as shown in Figure 2-1

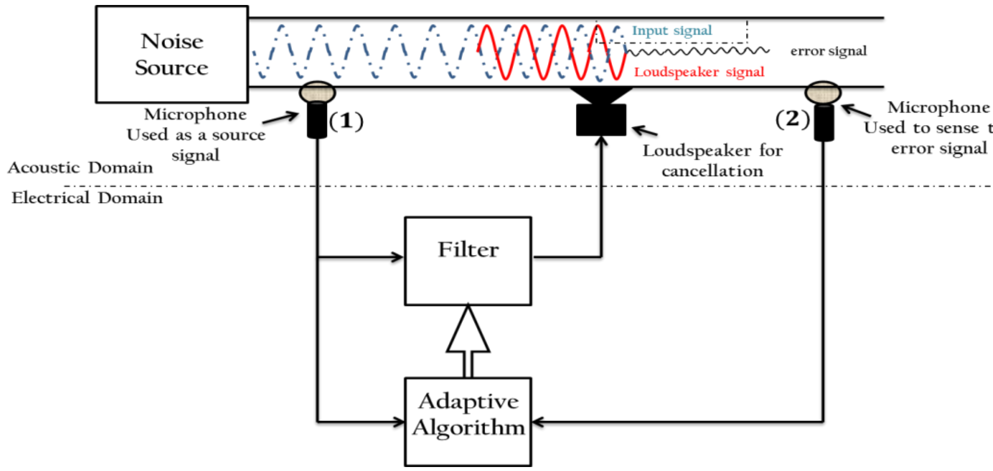


Figure 2-1: ANC system in the air duct

The primary path for this system is the path between the reference microphone (1) and the error microphone (2), while the two secondary paths are: the first one between the microphone (1) and the cancellation loudspeaker, and the second one between the loudspeaker and the microphone (2). The dashed line represents the undesirable noise signal which is sensed by microphone (1) and is used as the primary input signal to the adaptive filter. While the filter adjusts its weights depending on the adaptive algorithm used, the output of the adaptive filter is used to produce an inverted version of the input noise electronically by cancelling the loudspeaker through the duct. The primary input noise and the output of the loudspeaker should cancel each other. This occurs only when we have an exact estimation of magnitude and synchronization in time for both the primary noise and the output signal from the loudspeaker. Otherwise, an error signal is obtained. This signal continues through the duct to microphone (2); the error signal is sensed by the second microphone and used by the adaptive algorithm to adjust the filter weights, and ultimately to reduce the error to the lowest possible level.

The efficiency of the algorithm depends on how much time it will take before the adaptive algorithm converges and to what level of noise. In other words, how much time will it take before the resulting error becomes as low as possible. Moreover, the ability of the adaptive algorithm to converge when different types of stochastic input and noise signal used to energize the system.

2.2 Analysis

In this part, we will describe the general adaptive filtering system that will be used for both the FXLMF and the LFXLMF algorithms derivations. Figure 2-2 illustrates the block diagram of an ANC, and illustrates the location of the secondary path $\boldsymbol{S}$ and the estimated secondary path $\boldsymbol{\widehat{S}}$. The secondary path has a transfer function which can be represented by a group of a Digital to Analog (D/A) converter, a power amplifier, a cancelling loudspeaker, an error microphone and finally an (A/D) convertor.

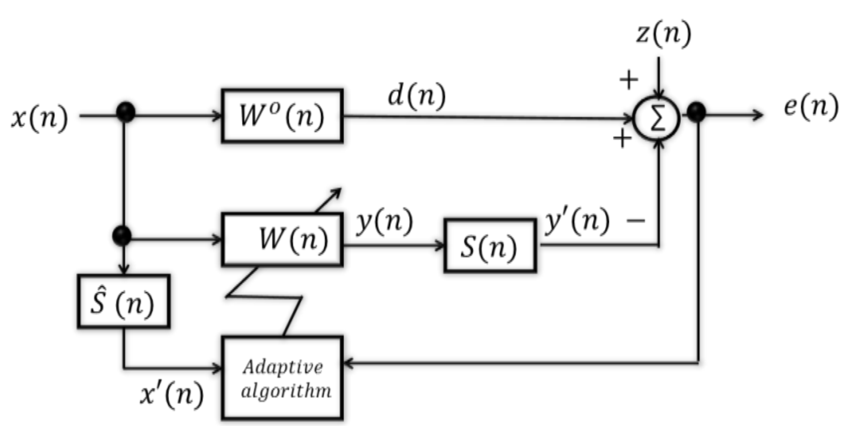


Figure 2-2: Block diagram of ANC.

The realization of the secondary path is usually obtained using a system identification technique; and for this work, the assumption used considers an inexact estimation for the secondary path which may cause errors on the number of coefficients, or on their values [20]; as a result, the values of the secondary path will be as $\widehat{\boldsymbol{S}} \neq \boldsymbol{S}$, and the filter coefficients taps number $\widehat{M} \neq M$.

Adaptive filter weights	$\boldsymbol{w}(n) = [w_0(n)\; w_1(n) \ldots\; w_{N-1}(n)]^T$
Stationary input signal	$\boldsymbol{x}(n) = [x(n)\; x(n-1) \ldots\; x(n-N+1)]^T$
Secondary path	$\boldsymbol{S} = [s_0\; s_1 \ldots\; s_{M-1}]^T$
Estimate of the secondary path	$\widehat{\boldsymbol{S}} = [\hat{s}_0\; \hat{s}_1 \ldots\; \hat{s}_{M-1}]^T$
Primary (desired) signal	$d(n)$
Stationary noise process	$z(n)$
Number of tap weight coefficients	N
Number of the secondary path coefficients	M

Table 2-1: Parameters and their descriptions used in Figure 2-2

Referring back to Figure 2-2, the error signal is given by

$$e(n) = d(n) - y'(n) + z(n) \tag{2.1}$$

where $d(n)$ is the desired response, and $y(n)$ is the output of the adaptive filter given by

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

$y'(n)$ is the output of the secondary path

$$y'(n) = \sum_{i=0}^{M-1} s_i y(n-i)$$

$$= \sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\underline{\boldsymbol{w}}(n-i) \tag{2.3}$$

and $z(n)$ is the active noise. Finally, the filtered input signal is given as

$$x'(n) = \sum_{i=0}^{\hat{M}-1} \hat{s}_i \boldsymbol{x}(n-i) \tag{2.4}$$

For the case of an exact approximation for the secondary path, that is $\boldsymbol{S} = \widehat{\boldsymbol{S}}$, then the input signal, $\boldsymbol{x}(n)$,will be filtered by $\boldsymbol{S}$.

2.3 Mathematical Background and Proposed Algorithms

Literature shows that adaptive filtering algorithms promise good solutions to noise control problems because of their ease of implementation, low cost, and other features (mentioned in Chapter 1). This work will add two new algorithms for ANC purposes. These are the FXLMF and LFXLMF algorithms.

The core of adaptive filtering techniques is the steepest decent method [1]-[2], for which the cost function $J(\boldsymbol{w})$ is a continuously differentiable function of the unknown $\boldsymbol{w}$, the weight vector of the adaptive filter. Each vector $\boldsymbol{w}$ is mapped by the cost function into a real number and the purpose is to find the optimum vector, $\boldsymbol{w}_o$, that satisfies the following

$$J(\boldsymbol{w}_o) \le J(\boldsymbol{w})\ for\ all\ \boldsymbol{w}. \tag{2.5}$$

In other words, the steepest decent method is an optimization algorithm, starting with an initial guess, $\boldsymbol{w}(0)$, which then generates a sequence of weight vectors $\boldsymbol{w}(1), \boldsymbol{w}(2), \ldots,$ to reduce the cost function iteratively where $J\big(\boldsymbol{w}(n+1)\big) < J\big(\boldsymbol{w}(n)\big)$. Based on this, the update weight vector equation can be written as

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) - \frac{1}{2}\mu\, g(n) \tag{2.6}$$

where the second term is called the weight adjustment term, $g(n)$ is the gradient vector of the cost function:

$$\begin{aligned} g(n) &= \nabla J\,(\boldsymbol{w}) \\ &= \frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}}, \end{aligned} \tag{2.7}$$

and μ is a positive constant called the step size parameter. The weight adjustment process is done by the algorithm on each iteration to update the weight vector $\boldsymbol{w}$ .Repeated updating of the weight vector for a large number of iterations will lead to the optimal weight vector $\boldsymbol{w_o}$ which satisfies $J(\boldsymbol{w_o}) = J_{min}$, that is

$$\lim_{n\to\infty} \boldsymbol{w}(n) = \boldsymbol{w_o}, \tag{2.8}$$

or,

$$\lim_{n\to\infty} J(n) = J_{min}. \tag{2.9}$$

The steepest decent method assumes that we have a priori information about the statistics of the input sequence, $\boldsymbol{x}(n)$, and the desired signal, $d(n)$. This is not the case in most of the adaptive filtering systems, and therefore we cannot apply it directly. To relax this situation Widrow proposed the LMS algorithm [6][7], the most famous adaptive filtering algorithm, where the instantaneous values of the autocorrected matrix of the input signal and the cross-correlation between the desired and the input signal are used instead, that is the update weight vector equation for the LMS algorithm is governed by the following recursion

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu \boldsymbol{x}(n) e(n) \tag{2.10}$$

where μ is the step size responsible for the convergence rate and controls the algorithm stability. The algorithm governed by (2.10) shows an unbiased behavior, so that the expected value of the weight vector converges to the optimal solution or the Wiener filter solution. Both LMS and FXLMS algorithms have the same update equation (2.10), but the difference is the error equation for each one of them, so the update in each of them will be different.

In 1984 Widrow and Walach proposed the Least Mean Fourth (LMF) algorithm and its family [21]. They showed that the general form this family updates is governed by

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu k \, \boldsymbol{x}(n) e^{2k-1}(n) \tag{2.11}$$

where $k = 1,2, \ldots$, and the LMS algorithm and the LMF algorithm are achieved when $k = 1$, and $k = 2$ respectively. The LMF algorithm has an interesting behavior since it shows substantially lower weight noise level comparing to the LMS algorithm under some circumstances. As it was the case for LMS and FXLMS algorithms, the weight update equation for the LMF and proposed FXLMF algorithms is the same as in equation (2.11) when $k = 2$.

In section 1.1.2 we mentioned the problem of the weight vector overflow during the updating process, because of an insufficient spectral excitation for the LMS algorithm; also we said that the solution was achieved through adding a power constraint to the weight update equation, which is called gamma γ , the leakage factor [16]. The function of the leakage factor is to retain the unconstrained growth in the weights during the update process. The leaky LMS algorithm is given as follows:

$$\boldsymbol{w}(n+1) = (1-\gamma\mu)\boldsymbol{w}(n) + \mu\boldsymbol{x}(n)e(n) \tag{2.12}$$

and the value $\upsilon = (1-\gamma\mu)$, where $0 < \upsilon \leq 1$ adds more stability and robustness to the adaptive algorithm at the expense of degradation in the algorithm's performance.

2.4 Development of FXLMF Algorithm

Using the block diagram in Figure 2-2 the cost function for the FXLMF algorithm is given by the following relation

$$J_{FXLMF}(\boldsymbol{w}) = E\,[e^4(n)] \tag{2.13}$$

where the error signal, $e(n)$, is given by (2.1), as the difference between the output signal from the secondary path and the primary signal, that is,

$$e(n) = d(n) - \sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{w} + z(n) \tag{2.14}$$

Deriving the curves of the derivations, we will resort to the same assumptions, used in the literature [17]-[22], to simplify our algorithms. These are the following:

Assumption A1: $\boldsymbol{x}(n)$ is the input signal, a zero mean wide-stationary Gaussian process with variance ${\sigma_x}^2$, and $\boldsymbol{R_{i,j}} = E[\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)] > 0$ is a positive definite autocorrelation matrix of the input vector.

Assumption A2: $z(n)$ is the measurement noise, an independent and identically distributed random (i.i.d) variable with zero mean and variance ${\sigma_z}^2 = E[z^2(n)]$, and there is no correlation between the input signal and the measurement noise. In other words, the sequence $z(n)$ is independent of $\boldsymbol{x}(n)$ & $\boldsymbol{w}(n)$. The measurement is assumed to have an even probability density function $f_z(z) = f_z(-z)$

Now, substituting (2.14) in (2.13), and assuming that the vector $\boldsymbol{w}$ is fixed, then the cost function look like the following:

$$J_{FXLMF} = E\left[\left(d(n) - \sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{w} + z(n)\right)^4\right] \tag{2.15}$$

Expanding equation (2.15) will lead to the following

$$J_{FXLMF} =$$

$$\left\{\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{M-1}\sum_{l=0}^{M-1} s_i s_j s_k s_l \, E[\boldsymbol{x}^T(n-l)\boldsymbol{x}(n-k)\,\boldsymbol{x}^T(n-j)\boldsymbol{x}(n-i)]\right)\right\}\|\underline{\boldsymbol{w}}\|^4$$

$$-4\left\{\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{M-1} s_i s_j s_k \, E[d(n)\boldsymbol{x}(n-k)\,\boldsymbol{x}^T(n-j)\boldsymbol{x}(n-i)]\right)\right\}\|\underline{\boldsymbol{w}}\|^3$$

$$+6\left\{\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1} s_i s_j \, E[d^2(n)\,\boldsymbol{x}(n-j)\,\boldsymbol{x}^T(n-i)]\right)\right.$$

$$\left. + \sigma_z{}^2\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1} s_i s_j \, E[\boldsymbol{x}(n-j)\,\boldsymbol{x}^T(n-i)]\right)\right\}\|\underline{\boldsymbol{w}}\|^2$$

$$-4\left\{\left(\sum_{i=0}^{M-1} s_i \, E[d^3(n)\,\boldsymbol{x}^T(n-i)]\right) - \sigma_z{}^2\left(\sum_{i=0}^{M-1} s_i \, E[d(n)\boldsymbol{x}^T(n-i)]\right)\right\}\|\underline{\boldsymbol{w}}\|$$

$$+\{(E\,[d^4(n)] + \sigma_z{}^2 - 4\,E\,[z^3(n)]E\,[d(n)] + 6\,\sigma_d{}^2\sigma_z{}^2)\} \tag{2.16}$$

The cost function in (2.16) is in the 4th order, and to obtain the optimal weight vector for the cost function we need to find its derivative of equation (2.16) with respect to $\boldsymbol{w}$ and equate it to zero. This done as follows and it looks like:

$$\frac{\partial J_{FXLMF}}{\partial \boldsymbol{w}} =$$

$$4\left\{\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{M-1}\sum_{l=0}^{M-1} s_i s_j s_k s_l\, E[\boldsymbol{x}^T(n-l)\boldsymbol{x}(n-k)\, \boldsymbol{x}^T(n-j)\boldsymbol{x}(n-i)]\right)\right\}\|\boldsymbol{w}\|^3$$

$$-12\left\{\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{M-1} s_i s_j s_k\, E[d(n)\boldsymbol{x}(n-k)\, \boldsymbol{x}^T(n-j)\boldsymbol{x}(n-i)]\right)\right\}\|\boldsymbol{w}\|^2$$

$$+12\left\{\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1} s_i s_j\, E[d^2(n)\, \boldsymbol{x}(n-j)\, \boldsymbol{x}^T(n-i)]\right)\right.$$

$$\left. + \sigma_z{}^2 \left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1} s_i s_j\, E[\boldsymbol{x}(n-j)\, \boldsymbol{x}^T(n-i)]\right)\right\}\|\boldsymbol{w}\|$$

$$+4\left\{\left(\sum_{i=0}^{M-1} s_i\, E[d^3(n)\, \boldsymbol{x}^T(n-i)]\right) - \sigma_z{}^2\left(\sum_{i=0}^{M-1} s_i\, E[d(n)\boldsymbol{x}^T(n-i)]\right)\right\}. \qquad (2.17)$$

or in a compact form; after multiplying (2.17) by $\left(\frac{\widetilde{\boldsymbol{R}}_{s^4}{}^{-1}}{4}\right)$, equation (2.17) leads to

$$\frac{\partial J_{FXLMF}(n)}{\partial \boldsymbol{w}(n)} =$$

$$\|\boldsymbol{w}\|^3 - 3\left(\widetilde{\boldsymbol{R}}_{s^4}{}^{-1}\widetilde{\boldsymbol{P}}_{d,s^3}\right)\|\boldsymbol{w}\|^2 + 3\widetilde{\boldsymbol{R}}_{s^4}{}^{-1}\left(\widetilde{\boldsymbol{P}}_{d^2,s^2} + \sigma_z{}^2\widetilde{\boldsymbol{R}}_{s^2}\right)\|\boldsymbol{w}\|$$

$$- \widetilde{\boldsymbol{R}}_{s^4}{}^{-1}\left(\widetilde{\boldsymbol{P}}_{d^3,s} - \sigma_z{}^2\widetilde{\boldsymbol{P}}_{d,s}\right) \qquad (2.18)$$

where notations in (2.18) are defined as follows:

$$\widetilde{\boldsymbol{R}}_{s^2} = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} s_i s_j \, E\left[\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)\right]$$

$$= \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} s_i s_j \, \boldsymbol{R_{i,j}}$$

$\boldsymbol{R_{i,j}} = E\left[\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)\right]$ Is the input autocorrelation matrix.

$$\widetilde{\boldsymbol{P}}_{d,s} = \sum_{j=0}^{M-1} s_j \, E\left[d(n)\boldsymbol{x}(n-j)\right]$$

$$= \sum_{j=0}^{M-1} s_j P_{d,j}$$

$\boldsymbol{P}_{d,j} = E\left[d(n)\boldsymbol{x}(n-j)\right]$ Is the cross correlation between the input and the primary signals.

$$\widetilde{\boldsymbol{R}}_{s^4} = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} \sum_{k=0}^{M-1} \sum_{l=0}^{M-1} s_i s_j s_k s_l \, E\left[\boldsymbol{x}^T(n-l)\boldsymbol{x}(n-k)\, \boldsymbol{x}^T(n-j)\boldsymbol{x}(n-i)\right]$$

$$= \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} \sum_{k=0}^{M-1} \sum_{l=0}^{M-1} s_i s_j s_k s_l \, \boldsymbol{R_{i,j,k,l}} \,.$$

$$\widetilde{\boldsymbol{P}}_{d,s^3} = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} \sum_{k=0}^{M-1} s_i s_j s_k \, E\left[d(n)\boldsymbol{x}(n-k)\, \boldsymbol{x}^T(n-j)\boldsymbol{x}(n-i)\right]$$

$$= \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} \sum_{k=0}^{M-1} s_i s_j s_k \, \boldsymbol{P}_{d,i,j,k}.$$

$$\widetilde{\boldsymbol{P}}_{d^2,s^2} = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} s_i s_j \, E\left[d^2(n)\, \boldsymbol{x}(n-j)\, \boldsymbol{x}^{\boldsymbol{T}}(n-i)\right]$$

$$= \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} s_i s_j \boldsymbol{P}_{d^2,i,j}.$$

$$\widetilde{\boldsymbol{P}}_{d^3,s} = \sum_{i=0}^{M-1} s_i\, E[d^3(n)\, \boldsymbol{x}^T(n-i)]$$

$$= \sum_{i=0}^{M-1} s_i\ \boldsymbol{P}_{d^3,i}$$

discarding the noise $z(n)$, equation (2.18) now becomes:

$$\frac{\partial J_{FXLMF}(n)}{\partial \boldsymbol{w}(n)} =$$

$$\|\boldsymbol{w}\|^3 - 3\left(\widetilde{\boldsymbol{R}}_{s^4}{}^{-1}\widetilde{\boldsymbol{P}}_{d,s^3}\right)\|\boldsymbol{w}\|^2 + 3\widetilde{\boldsymbol{R}}_{s^4}{}^{-1}\left(\widetilde{\boldsymbol{P}}_{d^2,s^2}\right)\|\boldsymbol{w}\| - \widetilde{\boldsymbol{R}}_{s^4}{}^{-1}\left(\widetilde{\boldsymbol{P}}_{d^3,s}\right) \qquad (2.19)$$

Equation (2.19) is a cubic equation with respect to the unknown weight vector. This equation has three solutions and we need to find the solution with the global minimum, which represents the value of the weight vector which gives the lowest error. Since it's not a unique solution, it will be difficult to solve it mathematically. Instead, interestingly we found that the optimal solution for the LMS algorithm [1]-[3] is a valid solution for the (2.19). To prove this, we find that (2.19) is equal to zero for this solution, as can be seen from the following:

$$\boldsymbol{w_o} = \widetilde{\boldsymbol{R}}_{s^2}{}^{-1}\widetilde{\boldsymbol{P}}_{d,s} \qquad (2.20)$$

substituting (2.20) in (2.19) as below

$$\frac{\partial \boldsymbol{J}_{FxLMF}(n)}{\partial \boldsymbol{w}(n)} = \left\| \widetilde{\boldsymbol{R}}_{s^2}{}^{-1} \widetilde{\boldsymbol{P}}_{d,s} \right\|^3 - 3\left(\widetilde{\boldsymbol{R}}_{s^4}{}^{-1} \widetilde{\boldsymbol{P}}_{d,s^3} \right) \left\| \widetilde{\boldsymbol{R}}_{s^2}{}^{-1} \widetilde{\boldsymbol{P}}_{d,s} \right\|^2$$

$$+ 3\widetilde{\boldsymbol{R}}_{s^4}{}^{-1} \left(\widetilde{\boldsymbol{P}}_{d^2,s^2} \right) \left\| \widetilde{\boldsymbol{R}}_{s^2}{}^{-1} \widetilde{\boldsymbol{P}}_{d,s} \right\| - \widetilde{\boldsymbol{R}}_{s^4}{}^{-1} \widetilde{\boldsymbol{P}}_{d^3,s}$$

$$= 0$$

$$\rightarrow = \left(\frac{\widetilde{\boldsymbol{P}}_{d,s}}{\widetilde{\boldsymbol{R}}_{s^2}} \right)^3 - 3\left(\frac{\widetilde{\boldsymbol{P}}_{d,s^3}}{\widetilde{\boldsymbol{R}}_{s^4}} \right) \left(\frac{\widetilde{\boldsymbol{P}}_{d,s}}{\widetilde{\boldsymbol{R}}_{s^2}} \right)^2 + 3\left(\frac{\widetilde{\boldsymbol{P}}_{d^2,s^2}}{\widetilde{\boldsymbol{R}}_{s^4}} \right) \left(\frac{\widetilde{\boldsymbol{P}}_{d,s}}{\widetilde{\boldsymbol{R}}_{s^2}} \right) - \left(\frac{\widetilde{\boldsymbol{P}}_{d^3,s}}{\widetilde{\boldsymbol{R}}_{s^4}} \right) = 0$$

also by using long division for (2.19) we can extract it as the following:

$$\frac{\partial \boldsymbol{J}_{FxLMF}(n)}{\partial \boldsymbol{w}(n)} = \|\boldsymbol{w}\|^3 - 3\left(\widetilde{\boldsymbol{R}}_{s^4}{}^{-1} \widetilde{\boldsymbol{P}}_{d,s^3} \right) \|\boldsymbol{w}\|^2 + 3\widetilde{\boldsymbol{R}}_{s^4}{}^{-1} \left(\widetilde{\boldsymbol{P}}_{d^2,s^2} \right) \|\boldsymbol{w}\|$$

$$- \widetilde{\boldsymbol{R}}_{s^4}{}^{-1} \left(\widetilde{\boldsymbol{P}}_{d^3,s} \right)$$

$$= \left(\|\boldsymbol{w}\|^2 - 2\widetilde{\boldsymbol{R}}_{s^4}{}^{-1} \widetilde{\boldsymbol{P}}_{d,s^3} \|\boldsymbol{w}\| + \widetilde{\boldsymbol{R}}_{s^2}{}^{-1} \widetilde{\boldsymbol{P}}_{d^2,s} \right) \left(\|\boldsymbol{w}\| - \widetilde{\boldsymbol{R}}_{s^2}{}^{-1} \widetilde{\boldsymbol{P}}_{d,s} \right)$$

$$= \left(\|\boldsymbol{w}\| - \widetilde{\boldsymbol{R}}_{s^2}{}^{-1} \widetilde{\boldsymbol{P}}_{d,s} \right) \left(\|\boldsymbol{w}\| - \widetilde{\boldsymbol{R}}_{s^2}{}^{-1} \widetilde{\boldsymbol{P}}_{d,s} \right) \left(\|\boldsymbol{w}\| - \widetilde{\boldsymbol{R}}_{s^2}{}^{-1} \widetilde{\boldsymbol{P}}_{d,s} \right) \qquad (2.21)$$

Cardano’s method [25] is the general method to solve cubic equations. In this method the solution can be found in one of three scenarios. The first scenario, states that there are three real roots and at least two are equal, so that in this work the optimal solution will be as in (2.20), the same solution as in the first scenario in Cardano’s method.

2.4.1 Mean Behavior for the FXLMF Algorithm

The block diagram shown in Figure 2-3 is used to drive the expression for the mean weight behavior for FXLMF algorithm. The FXLMF algorithm is governed by the following recursion:

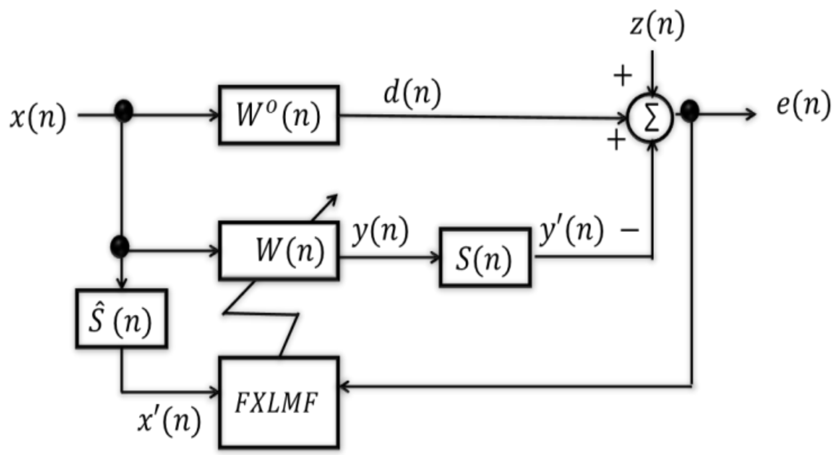


Figure 2-3: Block Diagram of the FXLMF algorithm

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) - \frac{\mu}{4}\,\frac{\partial J_{FXLMF}(n)}{\partial \boldsymbol{w}(n)} \tag{2.22}$$

where the instantaneous gradient can be approximated as :

$$\frac{\partial \hat{J}_{FXLMF}(n)}{\partial \boldsymbol{w}(n)} \approx -4\, e^3(n) \sum_{i=1}^{\widehat{M}-1} \hat{s}_i \boldsymbol{x}^T(n-i) \tag{2.23}$$

due the absence of the exact knowledge of the secondary path. Substituting equations (2.1)-(2.4) and (2.23) in (2.22), the adaptive weight vector update expression is given by:

$$\boldsymbol{w}(n+1) = \boldsymbol{w}\,(n) +$$

$$\mu \sum_{i=0}^{\widehat{M}-1} \hat{s}_i\, d^3(n)\boldsymbol{x}(n-i) - 3\,\mu \left(\sum_{i=0}^{M-1} \sum_{j=0}^{\widehat{M}-1} s_i \hat{s}_j\, d^2(n)\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)\,\boldsymbol{w}(n-i) \right)$$

$$+3\ \mu \left(\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} \sum_{k=0}^{\widehat{M}-1} s_i s_j \hat{s}_k\, d(n)\boldsymbol{x}(n-k)\boldsymbol{x}^T(n-j)\ \boldsymbol{x}(n-i)\,\boldsymbol{w}(n-j)\ \boldsymbol{w}^T(n-i) \right)$$

$$-6\ \mu \left(\sum_{i=0}^{M-1} \sum_{j=0}^{\widehat{M}-1} s_i \hat{s}_j\, z(n)d(n)\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)\,\boldsymbol{w}(n-i) \right)$$

$$-\mu \left(\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} \sum_{k=0}^{M-1} \sum_{l=0}^{\widehat{M}-1} s_i s_j s_k \hat{s}_l\, \boldsymbol{x}(n-l)\boldsymbol{x}^T(n-k)\ \boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)\,\boldsymbol{w}(n-k)\ \boldsymbol{w}^T(n-j)\boldsymbol{w}(n-i) \right)$$

$$+3\ \mu \left(\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} \sum_{k=0}^{\widehat{M}-1} s_i s_j \hat{s}_k\, z(n)\boldsymbol{x}(n-k)\boldsymbol{x}^T(n-j)\ \boldsymbol{x}(n-i)\,\boldsymbol{w}(n-j)\ \boldsymbol{w}^T(n-i) \right)$$

$$-3\,\mu \left(\sum_{i=0}^{M-1} \sum_{j=0}^{\widehat{M}-1} s_i \hat{s}_j\, z^2(n)\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)\,\boldsymbol{w}(n-i) \right) + \mu \left(\sum_{i=0}^{\widehat{M}-1} \hat{s}_i\, z^3(n)\boldsymbol{x}(n-i) \right)$$

$$+3\,\mu \left(\sum_{i=0}^{\widehat{M}-1} \hat{s}_i\, d(n)z^2(n)\boldsymbol{x}(n-i) \right)$$

$$+3\,\mu \left(\sum_{i=0}^{\widehat{M}-1} \hat{s}_i\, d^2(n)z(n)\boldsymbol{x}(n-i) \right) \tag{2.24}$$

The first moment or the mean weight of the adaptive weight vector for the FXLMF algorithm can be found by taking the expectation for both sides of equation(2.24). Since the proposed adaptive algorithm builds on the stochastic model, we have to rely on the independence theory [9]-[10], and use the same assumptions as in [21]-[22], to find the expectations for the terms in the right side, following suggested assumptions:

Assumption A3: Independence theory IT states that the taps of the input vector $\boldsymbol{x}(n-i), i = 0,1,2 \ldots$ are statistically independent, so that $E[\boldsymbol{x}(n-i)\boldsymbol{x}^T(n-j)] = E[\boldsymbol{x}(n-i)\boldsymbol{x}^T(n-j)\boldsymbol{x}(n-k)] = E[\boldsymbol{x}(n-l)\boldsymbol{x}^T(n-k)\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)] = 0,$ for any $i \neq j, \neq j \neq k$, and $i \neq j \neq k \neq l$ respectively.

Assumption A4: Take into consideration the correlation between $\boldsymbol{x}(n-i), \boldsymbol{x}(n-j), \boldsymbol{x}(n-k)$ and $\boldsymbol{x}(n-l), \forall\, i.j,k,l,$ and ignore the correlation between $\boldsymbol{w}(n-v)$ and $\boldsymbol{x}(n-i)$ or, $\boldsymbol{x}(n-k)$ or $\boldsymbol{x}(n-j), \forall\, i.j,k$ and as a result,

Following assumptions (***A1***- ***A4***), then the mean weight of the adaptive weight vector for FXLMF algorithm is expressed as:

$$E[\boldsymbol{w}(n+1)] = E[\boldsymbol{w}(n)] + \ \mu \sum_{i=0}^{\widehat{M}-1} \hat{s}_i \, E[d^3(n)\boldsymbol{x}(n-i)]$$

$$-3\,\mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{\widehat{M}-1} s_i\hat{s}_j \, E[d^2(n)\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)]E[\boldsymbol{w}\,(n-i)]\right)$$

$$+3\,\mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{\widehat{M}-1} s_i s_j \hat{s}_k \, E[d(n)\boldsymbol{x}(n-k)\boldsymbol{x}^T(n-j)\,\boldsymbol{x}(n-i)]E[\boldsymbol{w}(n-j)\boldsymbol{w}^T(n-i)]\right)$$

$$-\,\mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{M-1}\sum_{l=0}^{\widehat{M}-1} s_i s_j s_k \hat{s}_l \, E[\boldsymbol{x}(n-l)\boldsymbol{x}^T(n-k)\,\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)]\,E[\boldsymbol{w}(n-k)\boldsymbol{w}^T(n-j)\boldsymbol{w}(n-i)]\right)$$

$$-3\,\mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{\widehat{M}-1} s_i\hat{s}_j \, E[z^2(n)]E[\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)]E[\boldsymbol{w}\,(n-i)]\right)$$

$$+3\,\mu\,\left(\sum_{i=0}^{\widehat{M}-1} \hat{s}_i E[z^2(n)]\,E[d(n)\boldsymbol{x}(n-i)]\right) \tag{2.24}$$

Rewriting (2.24)using the simplified notations gives

$$E[\boldsymbol{w}(n+1)] =$$

$$E\ [\boldsymbol{w}\ (n)] + \ \mu \sum_{i=0}^{\widehat{M}-1} \hat{s}_i \ \boldsymbol{P}_{d^3,i} - 3\,\mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{\widehat{M}-1} s_i\hat{s}_j \ \ \boldsymbol{P}_{d^2,i,j} E[\boldsymbol{w}\,(n-i)]\right)$$

$$+3\,\mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{\widehat{M}-1} s_i s_j \hat{s}_k \ \boldsymbol{P}_{d,i,j,k} \, E[\boldsymbol{w}(n-j)\boldsymbol{w}^T(n-i)]\right)$$

$$-\mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{M-1}\sum_{l=0}^{\widehat{M}-1} s_i s_j s_k \hat{s}_l \ \boldsymbol{R}_{\boldsymbol{i,j,k,l}} \, E[\boldsymbol{w}(n-k)\boldsymbol{w}^T(n-j)\boldsymbol{w}(n-i)]\right)$$

$$-3\,\mu\,{\sigma_z}^2\left(\sum_{i=0}^{M-1}\sum_{j=0}^{\widehat{M}-1} s_i\hat{s}_j \ \boldsymbol{R}_{\boldsymbol{i,j}} \, E[\boldsymbol{w}\,(n-i)]\right) + 3\,\mu\,{\sigma_z}^2\left(\sum_{i=0}^{\widehat{M}-1} \hat{s}_i \ \boldsymbol{P}_{d,i}\right) \tag{2.25}$$

Now using the independent assumption results in

$$E[\boldsymbol{w}(n+1)] =$$

$$E[\boldsymbol{w}(n)] + \mu \, \hat{s}_0 \, E[d^3(n)\boldsymbol{x}^{\boldsymbol{T}}(n)]$$

$$-3\,\mu\left(\sum_{i=0}^{\min(\widehat{M},M)-1} s_i\hat{s}_i \, E[d^2(n)\boldsymbol{x}(n-i)\boldsymbol{x}^T(n-i)]E[\boldsymbol{w}\,(n-i)]\right)$$

$$+3\,\mu\left(\sum_{i=0}^{\min(\widehat{M},M)-1} s_i s_i\hat{s}_i \, E[d(n)\boldsymbol{x}(n-i)\boldsymbol{x}^T(n-i)\,\boldsymbol{x}(n-i)]E[\boldsymbol{w}(n-i)\boldsymbol{w}^T(n-i)]\right)$$

$$-\,\mu\left(\sum_{i=0}^{\min(\widehat{M},M)-1} s_i s_i s_i\hat{s}_i E[\boldsymbol{x}(n-i)\boldsymbol{x}^T(n-i)\,\boldsymbol{x}(n-i)\boldsymbol{x}^T(n-i)]\,E[\boldsymbol{w}(n-i)\boldsymbol{w}^T(n-i)\boldsymbol{w}(n-i)]\right)$$

$$-3\,\mu\left(\sum_{i=0}^{\min(\widehat{M},M)-1} s_i\hat{s}_i \, E[z^2(n)]E[\boldsymbol{x}(n-i)\boldsymbol{x}^T(n-i)]E[\boldsymbol{w}\,(n-i)]\right)$$

$$+3\,\mu\,(\hat{s}_0 E[z^2(n)]\,E[d(n)\boldsymbol{x}(n)])$$

(2.26)

Using the simplified notation then (2.26) will be as the following:

$$
\begin{aligned}
&E[\boldsymbol{w}(n+1)] = \\
&E[\boldsymbol{w}(n)] + \mu\,\hat{s}_0\,E[d^3(n)\boldsymbol{x}^{\boldsymbol{T}}(n)] \\
&-3\,\mu\left(\sum_{i=0}^{\min(\widehat{M},M)-1} s_i\hat{s}_i\,\boldsymbol{P}_{d^2,i,j}E[\boldsymbol{w}\,(n-i)]\right) \\
&+3\,\mu\left(\sum_{i=0}^{\min(\widehat{M},M)-1} s_i s_i\hat{s}_i\,\boldsymbol{P}_{d,i,j,k}E[\boldsymbol{w}(n-i)\boldsymbol{w}^T(n-i)]\right) \\
&-\mu\left(\sum_{i=0}^{\min(\widehat{M},M)-1} s_i s_i s_i\hat{s}_i\boldsymbol{R}_{\boldsymbol{i,j,k,l}}\,E[\boldsymbol{w}(n-i)\boldsymbol{w}^T(n-i)\boldsymbol{w}(n-i)]\right) \\
&-3\,\mu{\sigma_z}^2\left(\sum_{i=0}^{\min(\widehat{M},M)-1} s_i\hat{s}_i\boldsymbol{R}_{\boldsymbol{i,j}}E[\boldsymbol{w}\,(n-i)]\right) + 3\,\mu\,{\sigma_z}^2\hat{s}_0\,E[d(n)\boldsymbol{x}(n)]
\end{aligned}
\tag{2.27}
$$

Since in a practical situation an exact modeling for the secondary path can't be achieved, which may lead to incorrect number of tap weights, such as $\widehat{M} < M$ or the values of the taps $\widehat{M} \neq M$, we have to build our work over estimation for the secondary path as was the case for (2.23). Moreover, to study the steady state condition we can assume that the optimal solution of tap weights that the algorithm converges to can be reached if (n) goes to infinity as $\lim_{n\to\infty} E[\boldsymbol{w}(n+1)] = \lim_{n\to\infty} E[\boldsymbol{w}(n)] = \boldsymbol{w}_\infty$ as a result then:

$$
\begin{aligned}
\boldsymbol{w}_\infty &\approx \boldsymbol{w}_o \\
&= {\widetilde{\boldsymbol{R}}_{s^2}}^{-1}\widetilde{\boldsymbol{P}}_{d,s}
\end{aligned}
\tag{2.28}
$$

2.4.2 Second Moment Analysis for FXLMF Algorithm.

In this section the performance analysis for the mean square error, $E[e^2(n)]$, for the FXLMF algorithm is carried out, where the error is updated according to the adaptive weight update equation (2.11). Using equations (2.1)-(2.4) to attain the Mean Square Error (MSE) for FXLMF, as in the following:

$$MSE_{FXLMF}(n) = E\ [e^2(n)]$$

$$= E[d^2(n)] - 2\sum_{i=0}^{M-1} s_i\, E[d(n)\boldsymbol{x}(n-i)]E[\boldsymbol{w}(n-i)]$$

$$+\sum_{i=0}^{M-1}\sum_{j=0}^{M-1} s_i s_j\, E[\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)]E[\boldsymbol{w}\ (n-i)\boldsymbol{w}^T\ (n$$

$$-\ i)] + E[z^2(n)].$$

$$= {\sigma_d}^2 - 2\sum_{i=0}^{M-1} s_i\, \boldsymbol{P}_{d,i}\, E[\boldsymbol{w}(n-i)]$$

$$+\sum_{i=0}^{M-1}\sum_{j=0}^{M-1} s_i s_j\, \boldsymbol{R_{i,j}}\, E[\boldsymbol{w}\ (n-i)\boldsymbol{w}^T\ (n-i)] + {\sigma_z}^2 \qquad (2.29)$$

Then to find the Minimum Mean Square Error (MMSE) [1][3]we need to substitute the optimal solution of the FXLMF algorithm (2.28), $\boldsymbol{w}_o$,in (2.29), to get:

$$MMSE_{FXLMF} = E\left[\left(d(n) - \sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{w}_o\right)\left(d(n) - \sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{w}_o\right)^*\right]$$

Relying on the orthogonality principle [1]-[3], then the input signal will be orthogonal to the error, that is,

$$E\left[\left(d(n)-\sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{w}_o\right)\left(\sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\right)^*\right]=0 \tag{2.30}$$

as a result then the MMSE for the FXLMF algorithm is

$$MMSE_{FXLMF}=E\left(\left(d(n)-\sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{w}_o\right)\big(d(n)\big)^*\right)$$

$$=\sigma_d{}^2-\widetilde{\boldsymbol{P}}_{d,s}{}^*\widetilde{\boldsymbol{R}}_{s^2}\widetilde{\boldsymbol{P}}_{d,s} \tag{2.31}$$

2.4.3 FXLMF Algorithm Stability

In the previous section, the update weight of the recursion is governed by the step size μ parameter, which is responsible for the algorithm stability and its convergence rate. Choosing the right value of step size ensures that the algorithm will converge. The FXLMF algorithm's weight update equation is given by:

$$\boldsymbol{w}(n+1)=\ \boldsymbol{w}(n)+\mu\, e^3(n)\, x'(n) \tag{2.32}$$

In the moment the algorithm converges, then $E[\boldsymbol{w}(n+1)]=E[\boldsymbol{w}(n)]$; in other words the weight adjustment term will zero.

$$\mu\, E[e^3(n)\, x'(n)]=0$$

$$or \tag{2.33}$$

$$\mu\, E\left[\left(d(n)-\sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{w}+z(n)\right)^3\left(\sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\right)\right]=0$$

since,

$$\begin{aligned} e(n) &= d(n)-\sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{w}+z(n) \\ &= \sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{w_o} - \sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{w}+z(n) \end{aligned} \tag{2.34}$$

The weight error vector is defined as

$$\boldsymbol{v}(n)=\boldsymbol{w}(n)-\boldsymbol{w_o} \tag{2.35}$$

hence

$$e(n)=z(n)-\sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{v}(n) \tag{2.36}$$

Using (2.35) and (2.36) in (2.32), then:

$$\boldsymbol{v}(n+1)=\boldsymbol{v}(n)+\mu\left(z(n)-\sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{v}(n)\right)^3\left(\sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\right) \tag{2.37}$$

The value of $\boldsymbol{v}\,(\boldsymbol{n})$ approaches zero when the algorithm converges, so that we can ignore the high order terms of $\boldsymbol{v}(n)$ and as a result the weight error update equation can be written as:

$$\boldsymbol{v}(n+1) \cong$$

$$\boldsymbol{v}(n) + \mu\left(z^3(n) - 3\sum_{i=0}^{M-1} s_i\, z^2(n)\boldsymbol{x}^T(n-i)\boldsymbol{v}(n)\right)\left(\sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\right) \tag{2.38}$$

To find the mean weight error, we need to take the expectation of (2.38), and relying on the assumptions on (***A1- A4***), we can assume that the noise is independent of the input signal and also independent of the weight error vector. This results then into:

$$E[\boldsymbol{v}(n+1)] = E[\boldsymbol{v}(n)] - \mu\left\{3{\sigma_z}^2\sum_{i=0}^{M-1}\sum_{j=0}^{M-1} s_i\, E[\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)]\, E[\boldsymbol{v}(n)]\right\}$$

$$E[\boldsymbol{v}(n+1)] = \left(I - \mu\left(3{\sigma_z}^2\widetilde{\boldsymbol{R}}_{s^2}\right)\right) E[\boldsymbol{v}(n)] \tag{2.39}$$

From the last equation (2.39), since we assume before a positive definite of autocorrelation matrix $\boldsymbol{R}_{\boldsymbol{i},\boldsymbol{j}} > 0$, then the range of the step size for FXLMF algorithm can be shown to be given by:

$$0 < \mu < \frac{2}{3{\sigma_z}^2\lambda_{max}\left(\widetilde{\boldsymbol{R}}_{s^2}\right)} \tag{2.40}$$

where $\lambda_{max}\left(\widetilde{\boldsymbol{R}}_{s^2}\right)$ represents the maximum eigenvalue of $\widetilde{\boldsymbol{R}}_{s^2}$

2.5 Development of LFXLMF Algorithm.

In this section we are developing the leaky version of the FXLMF algorithm. Most of the derivations and analysis are repeated, as the assumptions proposed in the FXLMF algorithm derivation will be repeated for LFXLMF algorithm development. Using the block diagram in Figure 2-2, the cost function for the LFXLMF algorithm will be as follows:

$$\boldsymbol{J}_{LFXLMF}(\boldsymbol{w}) = E\,[e^4(n)] + \gamma\, \boldsymbol{w}^T \boldsymbol{w} \tag{2.41}$$

where, γ is the leakage factor $\gamma \geq 0$. In the case where $\gamma = 0$, then the cost function will be for the FXLMF algorithm. The error signal, $e(n)$, is given by using (2.1)-(2.3) as the difference between the output signal from the secondary path and the primary signal.

$$e(n) = d(n) - \sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{w} + z(n) \tag{2.42}$$

For the LFXLMF algorithm, we are analyzing stochastic input and noise signals and the assumptions are used (***A1- A4).***Now, using (2.42) in (2.41), and assuming that the vector $\boldsymbol{w}$ is fixed, then the cost function will be as the following:

$$J_{LFXLMF} = E\left[\left(d(n) - \sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{w} + z(n)\right)^4\right] + \gamma\, \boldsymbol{w}^T \boldsymbol{w} \tag{2.43}$$

Expanding equation (2.43) will lead to the following:

$$
\begin{aligned}
&J_{LFXLMF} = \\
&\left\{\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{M-1}\sum_{l=0}^{M-1} s_i s_j s_k s_l\, E[\boldsymbol{x}^T(n-l)\boldsymbol{x}(n-k)\,\boldsymbol{x}^T(n-j)\boldsymbol{x}(n-i)]\right)\right\}\|\boldsymbol{w}\|^4 \\
&-4\left\{\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{M-1} s_i s_j s_k\, E[d(n)\boldsymbol{x}(n-k)\,\boldsymbol{x}^T(n-j)\boldsymbol{x}(n-i)]\right)\right\}\|\boldsymbol{w}\|^3 \\
&+6\left\{\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1} s_i s_j\, E[d^2(n)\,\boldsymbol{x}(n-j)\,\boldsymbol{x}^T(n-i)]\right)\right. \\
&\qquad\qquad\left. +\,\sigma_z{}^2\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1} s_i s_j\, E[\boldsymbol{x}(n-j)\,\boldsymbol{x}^T(n-i)]\right)+\frac{\gamma}{6}I\right\}\|\boldsymbol{w}\|^2 \\
&-4\left\{\left(\sum_{i=0}^{M-1} s_i\, E[d^3(n)\,\boldsymbol{x}^T(n-i)]\right)-\sigma_z{}^2\left(\sum_{i=0}^{M-1} s_i\, E[d(n)\boldsymbol{x}^T(n-i)]\right)\right\}\|\boldsymbol{w}\| \\
&+\{(E\,[d^4(n)]+\sigma_z{}^2-4\,E\,[z^3(n)]E\,[d(n)]+6\,\sigma_d{}^2\sigma_z{}^2)\}
\end{aligned}
\tag{2.44}
$$

where $\boldsymbol{I}$ is the Identity matrix. Now to obtain the optimal weight vector for the cost function for LFXLMF, we have to derive equation (2.44) with respect to $\boldsymbol{w}$ and then equate the derivative to zero. The next equation (2.45) is the derivative of the cost function (2.44) with respect to the weight vector $\boldsymbol{w}$.

$$\frac{\partial J_{LFXLMF}}{\partial \boldsymbol{w}} =$$

$$4\left\{\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{M-1}\sum_{l=0}^{M-1} s_i s_j s_k s_l\, E[\boldsymbol{x}^T(n-l)\boldsymbol{x}(n-k)\,\boldsymbol{x}^T(n-j)\boldsymbol{x}(n-i)]\right)\right\}\|\boldsymbol{w}\|^3$$

$$-12\left\{\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{M-1} s_i s_j s_k\, E[d(n)\boldsymbol{x}(n-k)\,\boldsymbol{x}^T(n-j)\boldsymbol{x}(n-i)]\right)\right\}\|\boldsymbol{w}\|^2$$

$$+12\left\{\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1} s_i s_j\, E[d^2(n)\,\boldsymbol{x}(n-j)\,\boldsymbol{x}^T(n-i)]\right)\right.$$

$$\left. + {\sigma_z}^2\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1} s_i s_j\, E[\boldsymbol{x}(n-j)\,\boldsymbol{x}^T(n-i)]\right) + \frac{\gamma}{6}I\right\}\|\boldsymbol{w}\|$$

$$+4\left\{\left(\sum_{i=0}^{M-1} s_i\, E[d^3(n)\,\boldsymbol{x}^T(n-i)]\right) - {\sigma_z}^2\left(\sum_{i=0}^{M-1} s_i\, E[d(n)\boldsymbol{x}^T(n-i)]\right)\right\} \tag{2.45}$$

using the simplified notations, (2.45) as the following.

$$\frac{\partial J_{LFXLMF}(n)}{\partial \boldsymbol{w}(n)} =$$

$$\|\boldsymbol{w}\|^3 - 3\left({\widetilde{\boldsymbol{R}}_{s^4}}^{-1}\widetilde{\boldsymbol{P}}_{d,s^3}\right)\|\boldsymbol{w}\|^2 + 3{\widetilde{\boldsymbol{R}}_{s^4}}^{-1}\left(\widetilde{\boldsymbol{P}}_{d^2,s^2} + {\sigma_z}^2\widetilde{\boldsymbol{R}}_{s^2} + \frac{\gamma}{2}I\right)\|\boldsymbol{w}\|$$

$$- {\widetilde{\boldsymbol{R}}_{s^4}}^{-1}\left(\widetilde{\boldsymbol{P}}_{d^3,s} - {\sigma_z}^2\widetilde{\boldsymbol{P}}_{d,s}\right) \tag{2.46}$$

to find the optimal weight vector we need to discard the noise $z(n)$, so the last equation (2.46) will be as the following:

$$\frac{\partial J_{LFXLMF}(n)}{\partial \boldsymbol{w}(n)} = \|\boldsymbol{w}\|^3 - 3\left({\widetilde{\boldsymbol{R}}_{s^4}}^{-1}\widetilde{\boldsymbol{P}}_{d,s^3}\right)\|\boldsymbol{w}\|^2 + 3{\widetilde{\boldsymbol{R}}_{s^4}}^{-1}\left(\widetilde{\boldsymbol{P}}_{d^2,s^2} + \frac{\gamma}{2}I\right)\|\boldsymbol{w}\|$$

$$- {\widetilde{\boldsymbol{R}}_{s^4}}^{-1}\widetilde{\boldsymbol{P}}_{d^3,s} \tag{2.47}$$

Equation (2.47) is a cubic equation with respect to the unknown weight vector $\boldsymbol{w}$. This equation also has three solutions, and we need to find this global minimum which represents the value of the weight vector which gives the lowest error, but the case here in not as with the FXLMF algorithm; the leakage term makes the solution indirect as was the case in the previous algorithm.

2.5.1 Mean Behavior of the Adaptive Weight Vector for LFXLMF Algorithm

The block diagram shown in Figure 2-4 is for LFXLMF algorithm, and it can be seen that the weight update equation for LFXLMF algorithm is given as

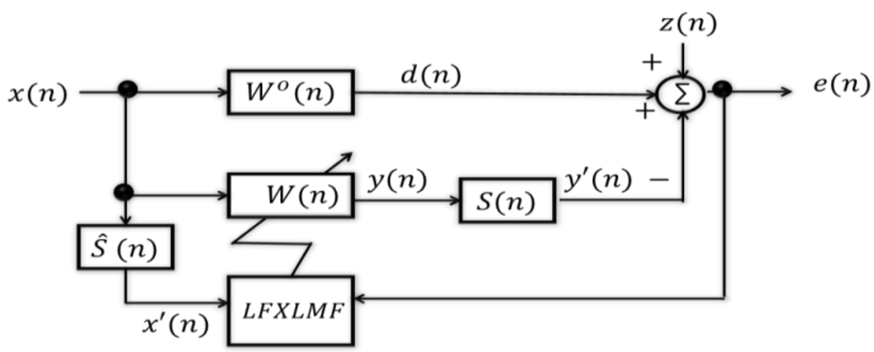


Figure 2-4: Block Diagram of the LFXLMF algorithm

$$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) - \frac{\mu}{4}\,\frac{\partial J_{LFXLMF}(n)}{\partial \boldsymbol{w}(n)}$$

$$= \left(1 - \mu\frac{\gamma}{2}\right)\boldsymbol{w}(n) + \mu\, e^3(n)\, x'(n) \tag{2.48}$$

where the instantaneous gradient can be approximated as follows:

$$\frac{\partial \hat{\boldsymbol{J}}_{LFXLMF}(n)}{\partial \boldsymbol{w}(n)} \approx -4\, e^{3}(n) \sum_{i=1}^{\hat{M}-1} \hat{s}_i \boldsymbol{x}^T(n-i) + 2\gamma \boldsymbol{w}(n) \tag{2.49}$$

Since we don't have exact knowledge of the secondary path, we can substitute equations (2.1)-(2.4) and (2.49) in (2.48) to get the adaptive weight vector update expression as follows:

$$\boldsymbol{w}(n+1) = \left(1-\mu\frac{\gamma}{2}\right)\boldsymbol{w}\,(n) +$$

$$\mu \sum_{i=0}^{\hat{M}-1} \hat{s}_i\, d^3(n)\boldsymbol{x}(n-i) - 3\,\mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{\hat{M}-1} s_i\hat{s}_j\, d^2(n)\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)\,\boldsymbol{w}(n-i)\right)$$

$$+3\ \mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{\hat{M}-1} s_i s_j \hat{s}_k\, d(n)\boldsymbol{x}(n-k)\boldsymbol{x}^T(n-j)\,\boldsymbol{x}(n-i)\,\boldsymbol{w}(n-j)\,\boldsymbol{w}^T(n-i)\right)$$

$$-6\ \mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{\hat{M}-1} s_i\hat{s}_j\, z(n)d(n)\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)\,\boldsymbol{w}(n-i)\right)$$

$$-\mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{M-1}\sum_{l=0}^{\hat{M}-1} s_i s_j s_k \hat{s}_l\, \boldsymbol{x}(n-l)\boldsymbol{x}^T(n-k)\,\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)\,\boldsymbol{w}(n-k)\,\boldsymbol{w}^T(n-j)\boldsymbol{w}(n-i)\right)$$

$$+3\ \mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{\hat{M}-1} s_i s_j \hat{s}_k\, z(n)\boldsymbol{x}(n-k)\boldsymbol{x}^T(n-j)\,\boldsymbol{x}(n-i)\,\boldsymbol{w}(n-j)\,\boldsymbol{w}^T(n-i)\right)$$

$$-3\,\mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{\hat{M}-1} s_i\hat{s}_j\, z^2(n)\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)\,\boldsymbol{w}(n-i)\right)$$

$$+\mu\left(\sum_{i=0}^{\hat{M}-1} \hat{s}_i\, z^3(n)\underline{\boldsymbol{x}}(n-i)\right) + 3\,\mu\,\left(\sum_{i=0}^{\hat{M}-1} \hat{s}_i\, d(n)z^2(n)\boldsymbol{x}(n-i)\right)$$

$$+3\,\mu\left(\sum_{i=0}^{\hat{M}-1} \hat{s}_i\, d^2(n)z(n)\boldsymbol{x}(n-i)\right) \tag{2.50}$$

Following assumptions (***A1- A4***), then the mean weight of the adaptive weight vector for LFXLMF algorithm is as expressed in the following:

$$
\begin{aligned}
E[\boldsymbol{w}(n+1)] = &\left(1-\mu\frac{\gamma}{2}\right)E[\boldsymbol{w}(n)] + \mu \sum_{i=0}^{\hat{M}-1} \hat{s}_i\, E[d^3(n)\boldsymbol{x}(n-i)] \\
&-3\,\mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{\hat{M}-1} s_i\hat{s}_j\, E[d^2(n)\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)]E[\boldsymbol{w}\,(n-i)]\right) \\
&+3\,\mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{\hat{M}-1} s_i s_j \hat{s}_k\, E[d(n)\boldsymbol{x}(n-k)\boldsymbol{x}^T(n-j)\,\boldsymbol{x}(n-i)]E[\boldsymbol{w}(n-j)\boldsymbol{w}^T(n-i)]\right) \\
&-\mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{M-1}\sum_{l=0}^{\hat{M}-1} s_i s_j s_k \hat{s}_l\, E[\boldsymbol{x}(n-l)\boldsymbol{x}^T(n-k)\,\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)]\, E[\boldsymbol{w}(n-k)\boldsymbol{w}^T(n-j)\boldsymbol{w}(n-i)]\right) \\
&-3\,\mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{\hat{M}-1} s_i\hat{s}_j\, E[z^2(n)]E[\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)]E[\boldsymbol{w}\,(n-i)]\right) \\
&+3\,\mu\left(\sum_{i=0}^{\hat{M}-1} \hat{s}_i E[z^2(n)]\, E[d(n)\boldsymbol{x}(n-i)]\right) \qquad (2.51)
\end{aligned}
$$

It’s the same as the one for the FXLMF algorithm but with an addition to the power constraint term.

Using the simplified notations, we can rewrite the previous equation (51) as the following:

$$E[\boldsymbol{w}(n+1)] =$$

$$\left(1-\mu\frac{\gamma}{2}\right)E\ [\boldsymbol{w}\ (n)]$$

$$+\ \mu\ \sum_{i=0}^{\widehat{M}-1}\hat{s}_i\ \boldsymbol{P}_{d^3,i} - 3\ \mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{\widehat{M}-1}s_i\hat{s}_j\ \ \boldsymbol{P}_{d^2,i,j}E[\boldsymbol{w}\ (n-i)]\right)$$

$$+3\ \mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{\widehat{M}-1}s_i s_j\hat{s}_k\ \boldsymbol{P}_{d,i,j,k}\ E[\boldsymbol{w}(n-j)\boldsymbol{w}^T(n-i)]\right)$$

$$-\mu\left(\sum_{i=0}^{M-1}\sum_{j=0}^{M-1}\sum_{k=0}^{M-1}\sum_{l=0}^{\widehat{M}-1}s_i s_j s_k\hat{s}_l\ \boldsymbol{R}_{\boldsymbol{i},\boldsymbol{j},\boldsymbol{k},\boldsymbol{l}}\ E[\boldsymbol{w}(n-k)\boldsymbol{w}^T(n-j)\boldsymbol{w}(n-i)]\right)$$

$$-3\ \mu\ {\sigma_z}^2\left(\sum_{i=0}^{M-1}\sum_{j=0}^{\widehat{M}-1}s_i\hat{s}_j\ \boldsymbol{R}_{\boldsymbol{i},\boldsymbol{j}}\ E[\boldsymbol{w}\ (n-i)]\right)$$

$$+3\ \mu\ {\sigma_z}^2\left(\sum_{i=0}^{\widehat{M}-1}\hat{s}_i\ \boldsymbol{P}_{d,i}\right) \tag{2.52}$$

To write the mean weight of the adaptive weight vector for the LFXLMF algorithm depending on the IT**,** the expression will be written as:

$$E[\boldsymbol{w}(n+1)] = \left(1 - \mu\frac{\gamma}{2}\right)E[\boldsymbol{w}(n)] + \mu\,\hat{s}_0\,E[d^3(n)\boldsymbol{x}^{\boldsymbol{T}}(n)]$$

$$-3\,\mu\left(\sum_{i=0}^{\min(\widehat{M},M)-1} s_i\hat{s}_i\,E[d^2(n)\boldsymbol{x}(n-i)\boldsymbol{x}^T(n-i)]E[\boldsymbol{w}\,(n-i)]\right)$$

$$+3\,\mu\left(\sum_{i=0}^{\min(\widehat{M},M)-1} s_i s_i\hat{s}_i\,E[d(n)\boldsymbol{x}(n-i)\boldsymbol{x}^T(n-i)\,\boldsymbol{x}(n-i)]E[\boldsymbol{w}(n-i)\boldsymbol{w}^T(n-i)]\right)$$

$$-\,\mu\left(\sum_{i=0}^{\min(\widehat{M},M)-1} s_i s_i s_i\hat{s}_i E[\boldsymbol{x}(n-i)\boldsymbol{x}^T(n-i)\,\boldsymbol{x}(n-i)\boldsymbol{x}^T(n-i)]\,E[\boldsymbol{w}(n-i)\boldsymbol{w}^T(n-i)\boldsymbol{w}(n-i)]\right)$$

$$-3\,\mu\left(\sum_{i=0}^{\min(\widehat{M},M)-1} s_i\hat{s}_i\,E[z^2(n)]E[\boldsymbol{x}(n-i)\boldsymbol{x}^T(n-i)]E[\boldsymbol{w}\,(n-i)]\right)$$

$$+3\,\mu\,(\hat{s}_0 E[z^2(n)]\,E[d(n)\boldsymbol{x}(n)]) \qquad (2.53)$$

and by using the notations then

$$E[\boldsymbol{w}(n+1)]=$$

$$\left(1-\mu\frac{\gamma}{2}\right)E[\boldsymbol{w}(n)]+\ \mu\,\hat{s}_0\ E[d^3(n)\boldsymbol{x}^{\boldsymbol{T}}(n)]$$

$$-3\,\mu\left(\sum_{i=0}^{\min(\widehat{M},M)-1} s_i\hat{s}_i\ \boldsymbol{P}_{d^2,i,j}E[\boldsymbol{w}\ (n-i)]\right)$$

$$+3\,\mu\left(\sum_{i=0}^{\min(\widehat{M},M)-1} s_is_i\hat{s}_i\ \boldsymbol{P}_{d,i,j,k}E[\boldsymbol{w}(n-i)\boldsymbol{w}^T(n-i)]\right)$$

$$-\,\mu\left(\sum_{i=0}^{\min(\widehat{M},M)-1} s_is_is_i\hat{s}_i\boldsymbol{R}_{\boldsymbol{i},\boldsymbol{j},\boldsymbol{k},\boldsymbol{l}}\ E[\boldsymbol{w}(n-i)\boldsymbol{w}^T(n-i)\boldsymbol{w}(n-i)]\right)$$

$$-3\,\mu{\sigma_z}^2\left(\sum_{i=0}^{\min(\widehat{M},M)-1} s_i\hat{s}_i\ \boldsymbol{R}_{\boldsymbol{i},\boldsymbol{j}}E[\boldsymbol{w}\ (n-i)]\right)$$

$$+3\,\mu\ {\sigma_z}^2(\hat{s}_0\ E[d(n)\boldsymbol{x}(n)]) \qquad (54)$$

2.5.2 Second Moment Analysis for LFXLMF

In this section, the performance analysis for the mean square error, $E[e^2(n)]$, for the LFXLMF algorithm is carried out, where the error is updated according to the adaptive weight update equation (50). Using equations (2.1)-(2.4) to attain the MSE for the LFXLMF algorithm we get the following:

$$MSE_{LFXLMF}(n) = E\ [e^2(n)]$$

$$= E[d^2(n)] - 2\sum_{i=0}^{M-1} s_i\, E[d(n)\boldsymbol{x}(n-i)]E[\boldsymbol{w}(n-i)]$$

$$+ \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} s_i s_j\, E[\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)]E[\boldsymbol{w}\,(n-i)\boldsymbol{w}^T\,(n$$

$$-\, i)] + E[z^2(n)].$$

$$= {\sigma_d}^2 - 2\sum_{i=0}^{M-1} s_i\, \boldsymbol{P}_{d,i}\, E[\boldsymbol{w}(n-i)]$$

$$+ \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} s_i s_j\, \boldsymbol{R_{i,j}}\, E[\boldsymbol{w}\,(n-i)\boldsymbol{w}^T\,(n-i)] + {\sigma_z}^2 \tag{2.55}$$

The expression of the MMSE for the LFXLMF algorithm will be the same as the one for the FXLMF algorithm, since the MMSE depends on the error $e(n)$ as in equation (2.42), and it doesn't depend on the leakage factor directly, unlike the weight update equation of the LFXLMF algorithm (50), which is determined by the leakage factor. The MMSE for the LFXLMF algorithm will be as the following:

$$MMSE_{FXLMF} = E\left(\left(d(n) - \sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{w}_o\right)\left(d(n)\right)^*\right)$$

$$= \sigma_d{}^2 - \widetilde{\boldsymbol{P}}_{d,s}{}^*\widetilde{\boldsymbol{R}}_{s^2}\widetilde{\boldsymbol{P}}_{d,s} \tag{2.56}$$

2.5.3 LFXLMF Algorithm Stability

In this section, the effect of leakage factor γ on the stability of the LFXLMF algorithm is discussed. Back in section (2.3) we talked about the reason behind introducing the power constraint or the leakage factor, which it was to solve the problem of weight vector overflow, usually the value of γ determined by the filter designer by trial and error. For this work, the range of the leakage factor can be given with respect to the step size μ.To do that, first let us start with the LFXLMF algorithm weight update (50):

$$\boldsymbol{w}(n+1) = \left(1 - \mu\frac{\gamma}{2}\right)\boldsymbol{w}(n) - \frac{\mu}{4}\ \frac{\partial \boldsymbol{J}_{LFXLMF}(n)}{\partial \boldsymbol{w}(n)}$$

$$= \left(1 - \mu\frac{\gamma}{2}\right)\ \boldsymbol{w}(n) + \mu\, e^3(n)\, x'(n) \tag{2.57}$$

The algorithm converges, when $E[\boldsymbol{w}(n+1)] = E[\boldsymbol{w}(n)]$. In other words, the weight adjustment term will be zero.

$$\mu\, E[\, e^3(n)\, x'(n)] = 0$$

$$E\left[\left(d(n) - \sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{w} + z(n)\right)^3 \left(\sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\right)\right] = 0 \tag{2.58}$$

But,

$$e(n) = d(n) - \sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{w} + z(n)$$

$$= \sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{w_o} - \sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{w} + z(n) \tag{2.59}$$

assuming fixed $\boldsymbol{w}$**,** then we can define the weight error vector

$$\boldsymbol{v}(n) = \boldsymbol{w}(n) - \boldsymbol{w_o} \tag{2.60}$$

hence

$$e(n) = z(n) - \sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{v}(n) \tag{2.61}$$

Using (2.60) and (2.61) in (2.57), then:

$$\boldsymbol{v}(n+1) =$$

$$\left(1 - \mu\frac{\gamma}{2}\right)\boldsymbol{v}(n) + \mu\left(z(n) - \sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\boldsymbol{v}(n)\right)^3 \left(\sum_{i=0}^{M-1} s_i\, \boldsymbol{x}^T(n-i)\right) \tag{2.62}$$

The value of $\boldsymbol{v(n)}$ approaches zero when the algorithm converges, so that we can ignore the high order terms of $\boldsymbol{v}(n)$ and as a result the weight error update equation can be written as:

$$\boldsymbol{v}(n+1) \cong \left(1-\mu\frac{\gamma}{2}\right)\boldsymbol{v}(n)$$
$$+\mu\left(z^3(n) - 3\sum_{i=0}^{M-1} s_i\, z^2(n)\boldsymbol{x}^T(n-i)\boldsymbol{v}(n)\right)\left(\sum_{i=0}^{M-1} s_i\,\boldsymbol{x}^T(n-i)\right) \tag{2.63}$$

To find the mean weight error, we need to take the expectation of (2.63), and relying on the assumptions on (***A1- A4***), we can assume that the noise is independent of the input signal, and is also independent of the weight error vector. Equation (2.64) shows the mean of the weight error vector:

$$E[\boldsymbol{v}(n+1)] = \left(1-\mu\frac{\gamma}{2}\right)E[\boldsymbol{v}(n)]$$
$$-\mu\left\{3{\sigma_z}^2\sum_{i=0}^{M-1}\sum_{j=0}^{M-1} s_i\, E[\boldsymbol{x}(n-j)\boldsymbol{x}^T(n-i)]\, E[\boldsymbol{v}(n)]\right\}$$
$$E[\boldsymbol{v}(n+1)] = \left(\left(1-\mu\frac{\gamma}{2}\right)I - \mu\left(3{\sigma_z}^2\widetilde{\boldsymbol{R}}_{s^2}\right)\right)E[\boldsymbol{v}(n)] \tag{2.64}$$

From the last equation (2.64), since we assume before a positive definition of autocorrelation matrix $\boldsymbol{R_{i,j}} > 0$, then the range of the leakage factor γ for LFXLMF algorithm is the following:

$$\frac{3{\sigma_z}^2\lambda_{max}\left(\widetilde{\boldsymbol{R}}_{s^2}\right)-1}{\frac{3}{2}{\sigma_z}^2\lambda_{max}\left(\widetilde{\boldsymbol{R}}_{s^2}\right)} < \gamma < \frac{2}{\mu} \tag{2.65}$$

where $\lambda_{max}\left(\widetilde{\boldsymbol{R}}_{s^2}\right)$ represent the maximal eigenvalue of $\widetilde{\boldsymbol{R}}_{s^2}$

3. CHAPTER 3

ALGORITHMS CONVEX COMBINATION

In this chapter we will examine the behavior of our algorithm through the convex combination of the proposed algorithms and other algorithms, the Convex Combination with the FXLMF Algorithm. The method of combining two algorithms is an interesting proposal which aims to mix the output of each filter, and highlights the best features of each individual algorithm, then utilizes the features in the overall equivalent filter to improve the performance of the adaptive filter [34]-[37]. In this section we will examine our proposed algorithms with members from the LMS and LMF families.

Figure 3-1 is the proposed block diagram for the convex combination of two filtered input signals, where the output of the overall combined filter can be given as in [34][35] by the following equation:

$$y(n) = \lambda(n) {y'}_1(n) + [1 - \lambda(n)] {y'}_2(n) \tag{3.1}$$

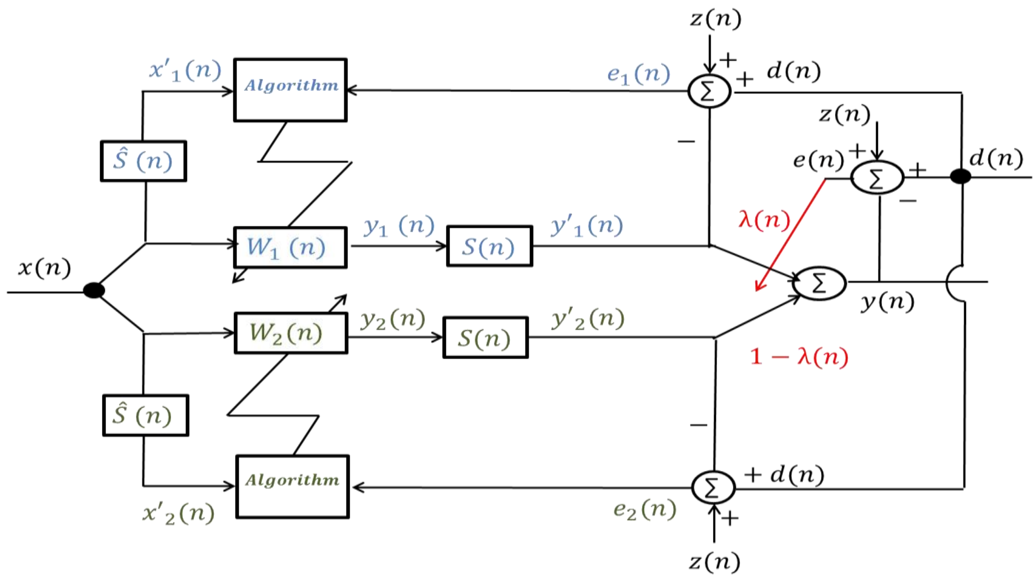


Figure 3-1: Block diagram of adaptive convex combination for two filtered input signal algorithms.

where $y'_1(n)$ and $y'_2(n)$ are the output of the two filters, and $\lambda(n)$ is the contribution or mixing parameter, where $0 \leq \lambda(n) \leq 1$. This parameter shows the percentage of involvement for each algorithm in the overall filter output. Therefore, the combined filter will extract the best features for each filter $\boldsymbol{w}_1(n)$ and $\boldsymbol{w}_2(n)$ individually. Assuming both filters $\boldsymbol{w}_1(n)$ and $\boldsymbol{w}_2(n)$ have the same size M, then the weight vector of the overall filter can be given as:

$$\boldsymbol{w}(n) = \lambda(n)\boldsymbol{w}_1(n) + [1-\lambda(n)]\boldsymbol{w}_2(n) \tag{3.2}$$

Each filter is updated individually, depending by its own error $e_1(n)$ or $e_2(n)$, and the overall weight vector is updated according to the total error $e(n) = [\,d(n) - y(n) + z(n)]$ which adapts the mixing parameter $\lambda(n)$. Using the gradient descent method we can minimize the fourth order $e^4(n)$ and the second order $e^2(n)$ error for the overall filter. Based on that, we can use the convex combined filter over two scenarios.

First, we will do the minimization for the quadratic error $e^2(n)$, where $\lambda(n)$ is the sigmoidal function given as:

$$\lambda(n) = \frac{1}{1+e^{-a(n)}}. \tag{3.3}$$

and instead of doing the update equation with respect to $\lambda(n)$, we will define the update equation with respect to the changing value $a(n)$ as following:

$$\begin{aligned} a(n+1) &= a(n) - \frac{\mu_{a^2}}{2}\frac{\partial e^2(n)}{\partial a(n)} \\ &= \ a(n) - \frac{\mu_{a^2}}{2}\frac{\partial e^2(n)}{\partial \lambda(n)}\frac{\partial \lambda(n)}{\partial a(n)} \\ &= a(n) + \mu_{a^2} e(n)\big[y'_1(n) - y'_2(n)\big]\lambda(n)[1-\lambda(n)] \end{aligned} \tag{3.4}$$

The second scenario is to conduct the minimization for the fourth order error of the overall filter; then the updated equation with respect to $a(n)$ will be as the following:

$$\begin{aligned} a(n+1) &= a(n) - \frac{\mu_{a^4}}{4}\frac{\partial e^4(n)}{\partial a(n)} \\ &= \ a(n) - \frac{\mu_{a^4}}{4}\frac{\partial e^4(n)}{\partial \lambda(n)}\frac{\partial \lambda(n)}{\partial a(n)} \\ &= a(n) + \mu_{a^4} e^3(n)\big[y'_1(n) - y'_2(n)\big]\lambda(n)[1-\lambda(n)] \end{aligned} \tag{3.5}$$

where μ_{a^2} and μ_{a^4} are the step size for the overall filter, for the quadratic and fourth order error respectively. For this work we will study the mean square performance for the convex combined filter utilized filtered input signal. Since the range of $\lambda(n)$ is between one and zero, we need to insure that the combined filter keeps adapting and that we don’t stick with one algorithm all the time. For this purpose, we have to reduce the interval of the mixing parameter by limiting the value of $a(n)$ inside $[1 - a^{+}, a^{+}]$; then the range of the mixing parameter will be between $1 - \lambda^{+} \leq \lambda(n) \leq \lambda^{+}$, as the following:

$$\lambda(n) = \begin{cases} 0.998, & a(n) > a^{+} \\ \lambda(n), & a^{+} \geq a(n) \geq -a^{+} \\ 0.002, & a(n) < -a^{+} \end{cases} \tag{3.6}$$

Simulations in section 4.2 will investigate in four cases, where the comparison will be done by using FXLMF and FXLMS algorithms, as the two transversal filters are used in the convex combination, according to the second error order minimization.

4. CHAPTER 4

SIMULATION & COMPARISON

The simulations in this chapter are divided into two main sections. Section 4.1 examines the proposed algorithms in the mean square error and mean weight context. The simulation has been done for FXLMF and LFXLMF algorithms under some conditions and environments. On the other hand, Section 4.2 tests the concept of convex combinations over the FXLMF and FXLMS algorithms. Furthermore, comparisons with other algorithms are carried out to show under which circumstances the new proposed algorithms outperform. The plant vector used to filter the input signal is $\boldsymbol{w}p$ with 9 taps where $\boldsymbol{w}p =$ $[0.0179 \quad 0.1005 \quad 0.2795 \quad 0.4896 \quad 0.5860 \quad 0.4896 \quad 0.2795 \quad 0.1005 \quad 0.0179]$

And for simplicity we assumed the secondary path and the estimated secondary path are equal $\boldsymbol{S} = \widehat{\boldsymbol{S}} = [0.7756 \quad 0.5171 \quad -0.3620]$. In the simulation we examine the effect of the step size, leakage factor, the type and value of the measurement noise, on the algorithms convergence and speed of convergence, using our proposed assumptions and the independence theory.

4.1 FXLMF and LFXLMF algorithms: Mean Square Error Performance and Mean of Weight Vector

In this section we examine our proposed algorithms with members from the LMS family. Each figure has it is own descriptions and a table of parameters is used to accomplish the simulation.

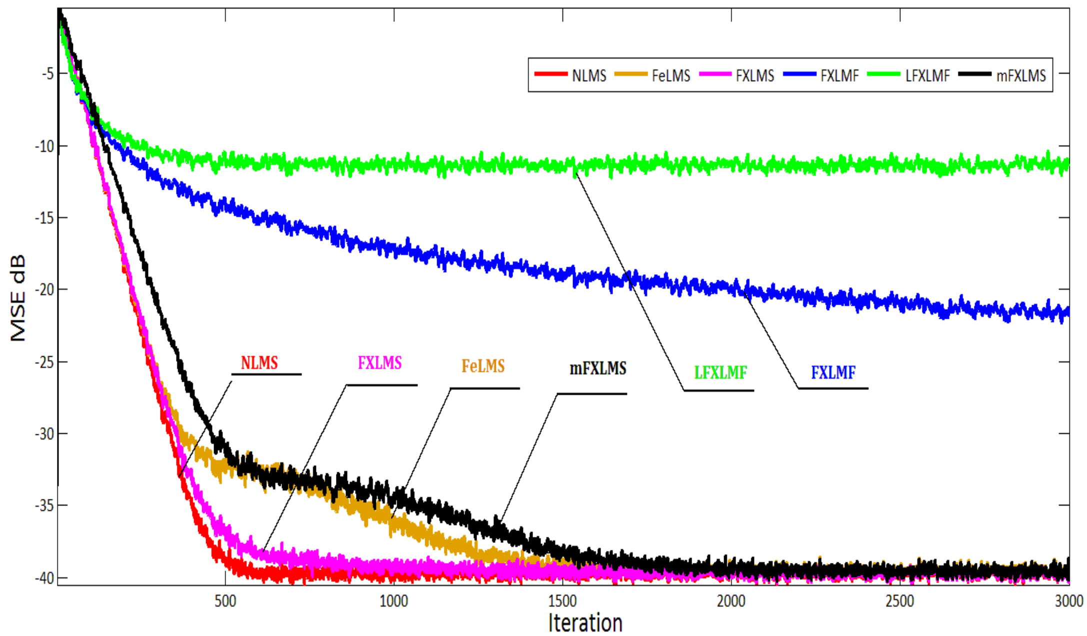


Figure 4-1: Comparison over MSE for FXLMF and LFXLMF with other algorithms using variable step size and high SNR.

Parameter	Value
Number of iterations	3000
Averaging over	500
Type of Noise	White Gaussian Noise
Variance of Measurement Noise $\sigma_z{}^2$ ↔SNR	$\sigma_z{}^2 = 0.0001$ ↔ 40 dB
Step size μ	Variable
Leakage Factor γ	γ= 0.05

Table 4-1 : Parameter used for simulation in Figure 4-1

Figure 4-1 above shows a comparison of the mean square error MSE behavior for different algorithms. It can be shown that the FXLMF algorithm converges and it will reach the weight noise level after a large number of iterations. For the LFXLMF algorithm it reaches the steady state level faster than the others, and after almost 500 iterations, but it converges to a higher weight noise level at almost 12dB.

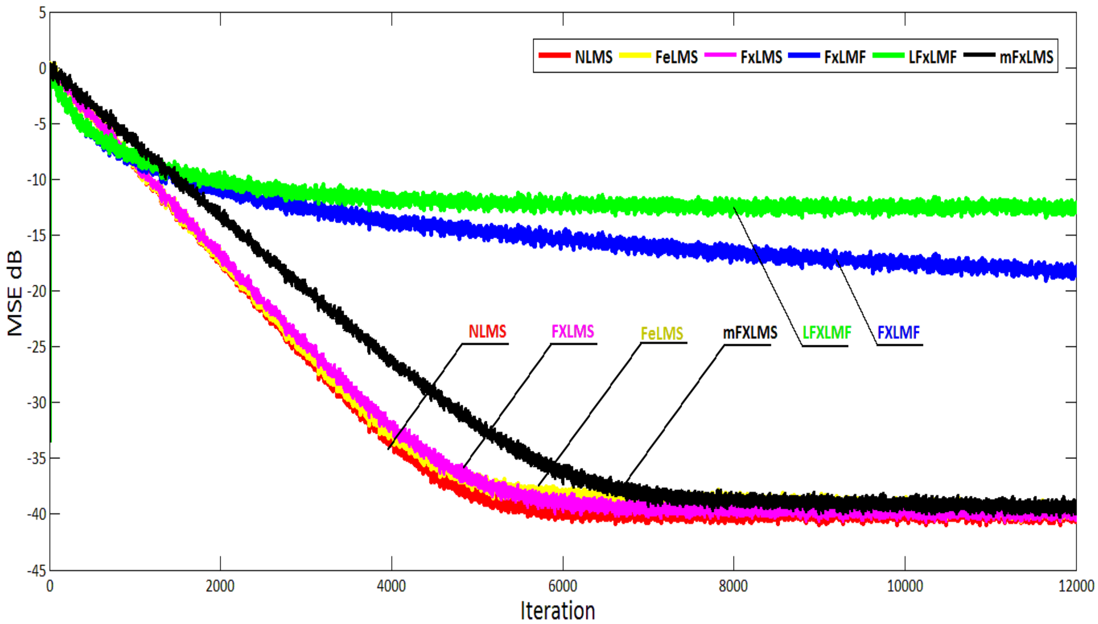


Figure 4-2: Comparison over MSE for FXLMF and LFXLMF with other algorithms using fixed step size and high SNR

Parameter	Value
Number of iterations	15000
Averaging over	500
Type of Noise	White Gaussian Noise
Variance of Measurement Noise ${\sigma_z}^2$ ↔SNR	${\sigma_z}^2 = 0.0001$ ↔ 40 dB
Step size μ	$\mu = 0.001$
Leakage Factor γ	γ= 0.05

Table 4-2: Parameter used for simulation in Figure 4-2

Figure 4-2 above shows a comparison of the mean square error MSE behavior for different algorithms for fixed step size. We can see that changing the type of step size to a fixed value instead of variable step size will increase the convergence time to more than 5000 iterations, and all algorithms have almost the same convergence behavior as in the previous example. Using a larger step size μ may lead the algorithm to diverge.

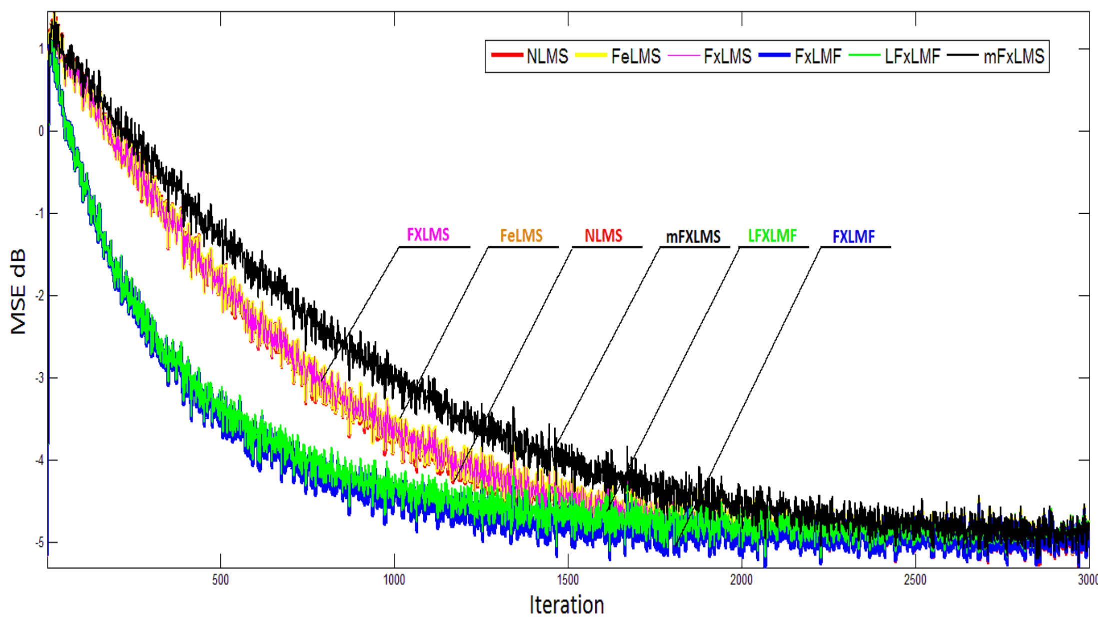


Figure 4-3: Comparison over MSE for FXLMF and LFXLMF with other algorithms using variable step size on low SNR.

Parameter	Value
Number of iterations	3000
Averaging over	1500
Type of Noise	White Gaussian Noise
Variance of Measurement Noise ${\sigma_z}^2$ ↔SNR	${\sigma_z}^2 = 0.3163$ ↔ 5 dB
Step size μ	Variable
Leakage Factor γ	γ= 0.05

Table 4-3: Parameter used for simulation in Figure 4-3

Figure 4-3 above shows a comparison of the mean square error MSE behavior for different algorithms for variable step size, but this time for high SNR with a value of 5dB, and we can clearly notice that the FXLMF and LFXLMF algorithms outperform other LMS family algorithms in speed of convergence, in advantage to the LMF family with almost 500 iterations, both FXLMF and LFXLMF almost have identical curves because we are using small leakage factor γ.

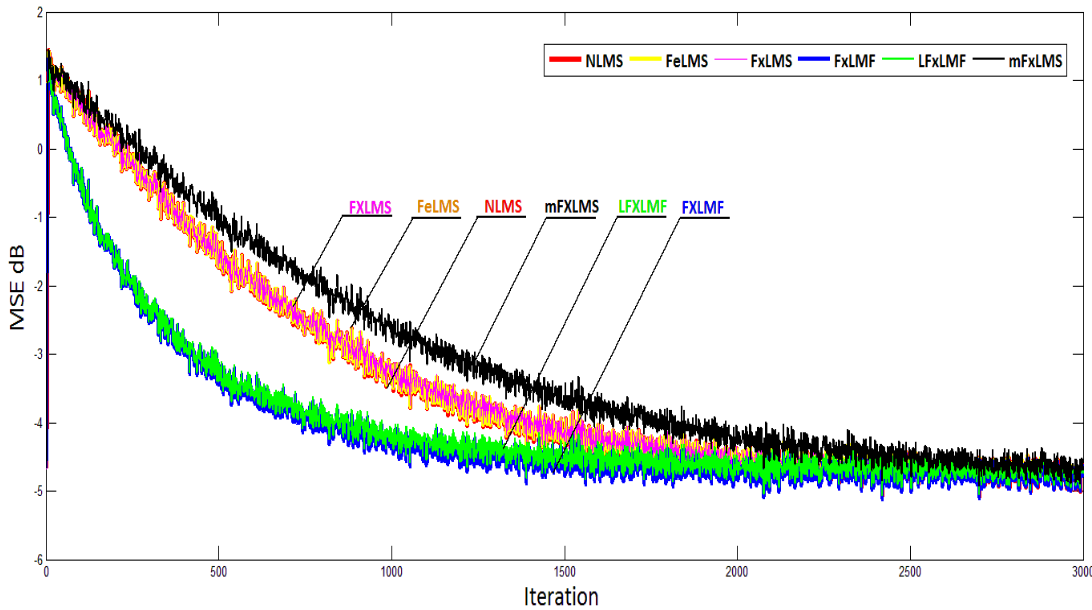


Figure 4-4: Comparison over MSE for FXLMF and LFXLMF algorithms with other algorithms using fixed step size on low SNR

Parameter	Value
Number of iterations	3000
Averaging over	1000
Type of Noise	White Gaussian Noise
Variance of Measurement Noise $\sigma_z{}^2$ ↔SNR	$\sigma_z{}^2 =$ 0.3163 ↔ 5 dB
Step size μ	$\mu = 0.001$
Leakage Factor γ	γ= 0.05

Table 4-4: Parameter used for simulation in Figure 4-4

Figure 4-4, shows the same result as Figure 4-3, but we can see that using a fixed step size will reduce the speed of convergence of the algorithm. Also the FXLMF and LFXLMF algorithms shows faster convergence compared to other algorithms at low Signal to Noise Ratio, SNR.

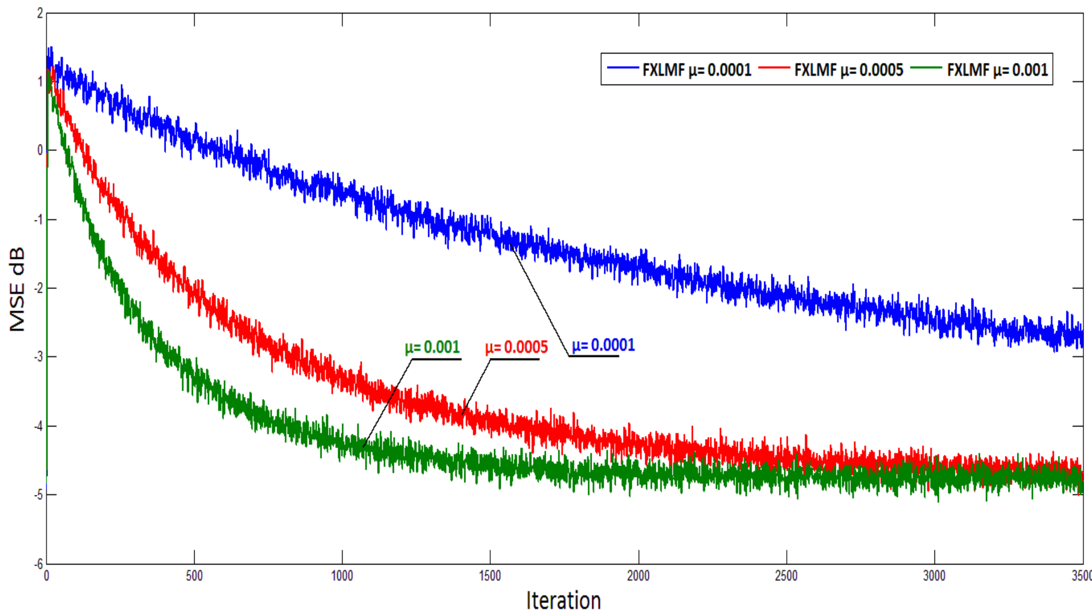


Figure 4-5: Comparison over MSE for FXLMF algorithms using different fixed step size and low SNR

Parameter	Value
Number of iterations	3500
Averaging over	1500
Type of Noise	White Gaussian Noise
Variance of Measurement Noise ${\sigma_z}^2$ ↔SNR	${\sigma_z}^2 = 0.3163$ ↔ 5dB
Step size μ	$\mu = [0.001, 0.0005, 0.0001]$

Table 4-5: Parameter used for simulation in Figure 4-5

Figure 4-5 shows the effect of changing the step size on the FXLMF algorithm convergence. From the figure we can see that increasing the step size μ will lead to faster convergence. However, step size should not go over the range given in equation (2.40), otherwise the algorithm will diverge.

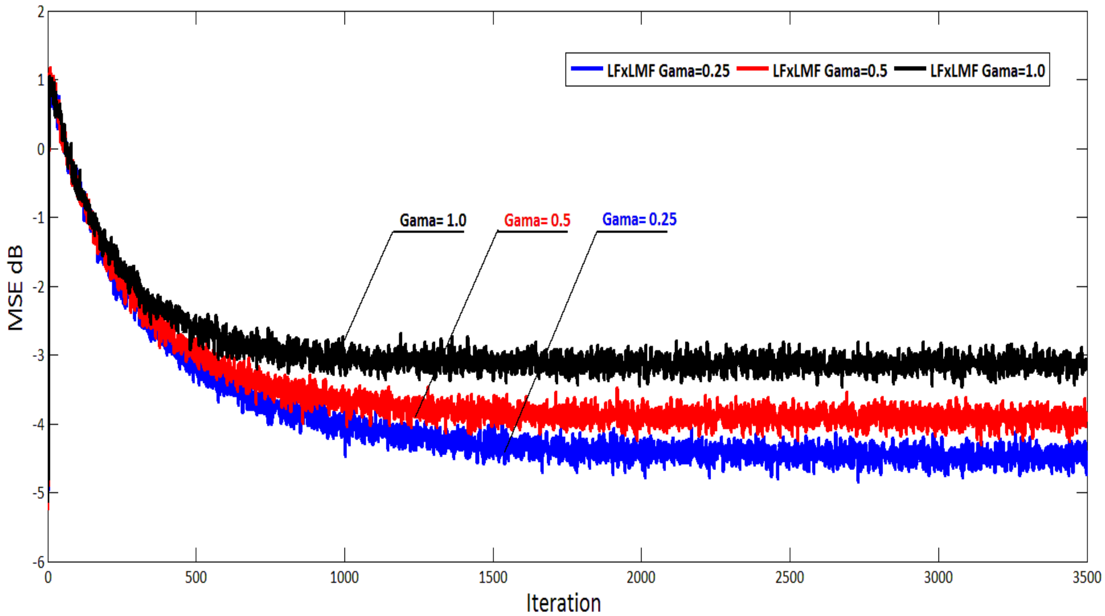


Figure 4-6: Comparison over MSE for LFXLMF algorithm using different fixed leakage factors and fixed step size and low SNR

Parameter	Value
Number of iterations	3500
Averaging over	1500
Type of Noise	White Gaussian Noise
Variance of Measurement Noise ${\sigma_z}^2$ ↔SNR	${\sigma_z}^2 = 0.3163$ ↔ 5 dB
Step size μ	$\mu = 0.001$
Leakage Factor γ	$\gamma = [0.250, 0.50, 1]$

Table 4-6: Parameter used for simulation in Figure 4-6

Figure 4-6 shows the effect of changing the leakage factor γ on the LFXLMF algorithm convergence, and it's clear that increasing the γ , leads to converging the algorithm to a lower noise level, while reducing γ makes the algorithm look like the FXLMF; boundaries of the leakage factor were shown before in section (2.5.2).

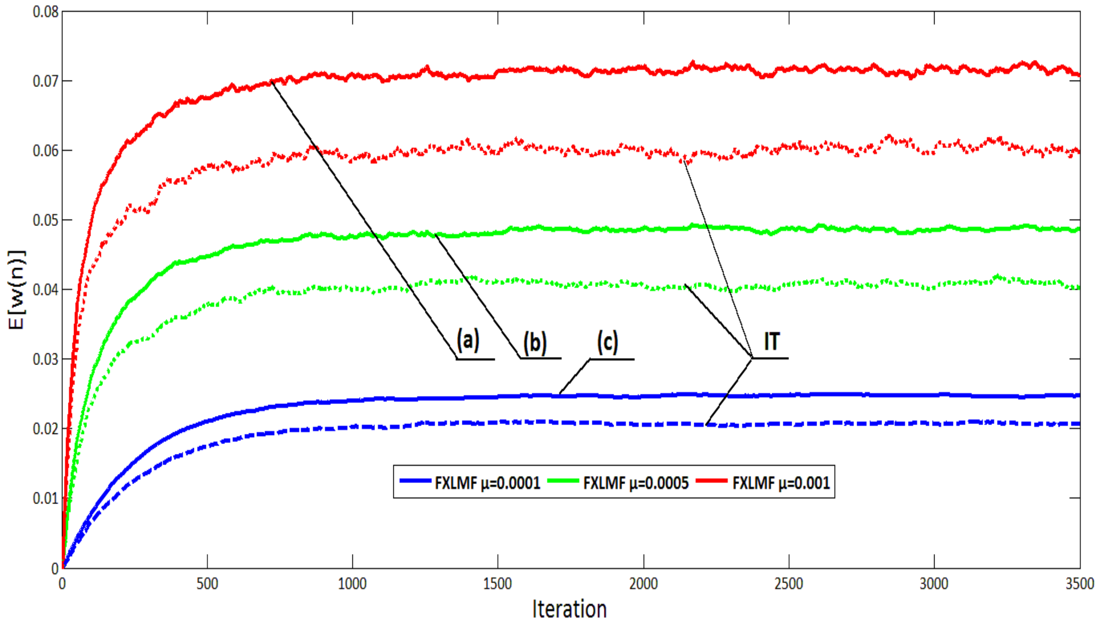


Figure 4-7: Comparison over mean weight vector for FXLMF algorithms using different fixed step size and low SNR solid line: proposed model (a), (b), and (c). Dashed line: IT model

Parameter	Value
Number of iterations	3500
Averaging over	1000
Type of Noise	White Gaussian Noise
Variance of Measurement Noise $\sigma_z{}^2$ ↔SNR	dB
Step size μ	$\mu = [0.001, 0.0005, 0.0001]$

Table 4-7: Parameter used for simulation in Figure 4-7

Figure 4-7 shows the effect of changing the step size on the mean weight vector of the FXLMF algorithm; when we increase the values of the step size the algorithm converges faster to the larger mean of the weight. Moreover, using assumption 0 as in section 2.4.1 makes the algorithm converge to a higher mean weight level.

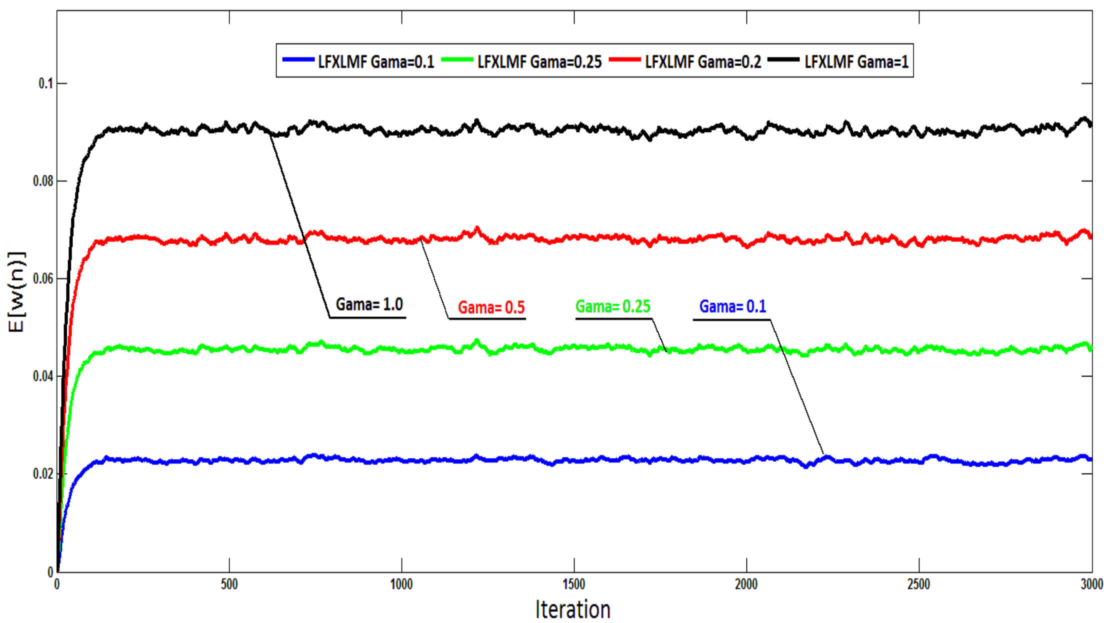


Figure 4-8: Comparison over mean weight vector for LFXLMF algorithms using different fixed leakage factors, and fixed step size on low SNR.

Parameter	Value
Number of iterations	3500
Averaging over	1500
Type of Noise	White Gaussian Noise
Variance of Measurement Noise ${\sigma_z}^2$ ↔SNR	${\sigma_z}^2 = 0.3163$ ↔ 5 dB
Step size μ	$\mu = 0.001$
Leakage Factor γ	$\gamma = [0.1, 0.250, 0.50, 1]$

Table 4-8: Parameter used for simulation in Figure 4-8

Figure 4-8, shows the effect of changing the leakage factor on the mean weight of the LFXLMF algorithm. We can see that increasing the value of the leakage factor will increase the mean weight of the LFXLMF algorithm, and it does not affect the speed of convergence.

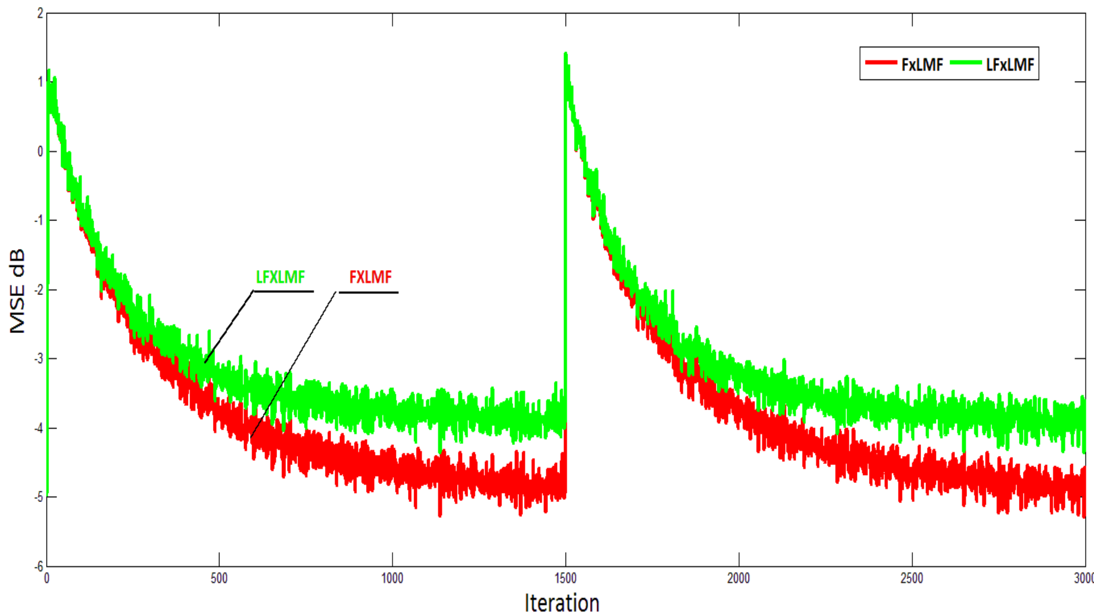


Figure 4-9: MSE for FXLMF and LFXLMF algorithm robustness at low SNR and Gaussian noise.

Parameter	Value
Number of iterations	3000
Averaging over	1500
Type of Noise	White Gaussian Noise
Variance of Measurement Noise ${\sigma_z}^2$ ↔SNR	${\sigma_z}^2 = 0.3163$ ↔ 5 dB
Step size µ	$\mu = 0.00125$
Leakage Factor γ	$\gamma = [\,0.50]$

Table 4-9: Parameter used for simulation in Figure 4-9

Figure 4-9 shows the robustness of the proposed algorithms FXLMF and LFXLMF at low SNR and using Gaussian noise.

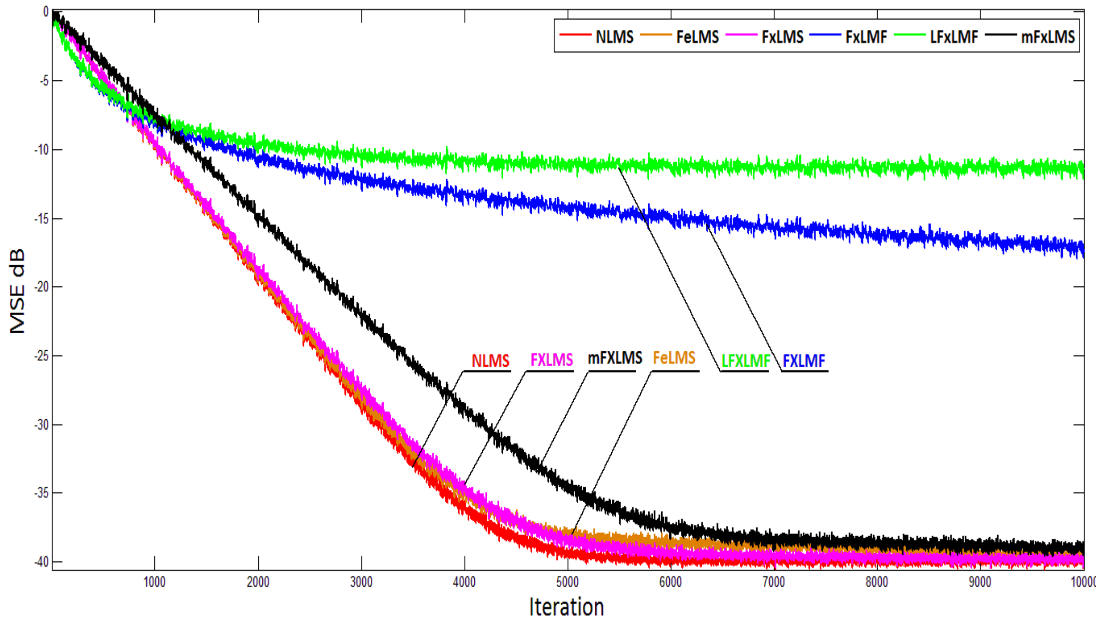


Figure 4-10: Comparison over MSE for FXLMF and LFXLMF with other algorithms using variable step size and high SNR.

Parameter	Value
Number of iterations	10000
Averaging over	500
Type of Noise	Uniform Noise
Variance of Measurement Noise $\sigma_z{}^2$ ↔SNR	$\sigma_z{}^2 = 0.0001 \leftrightarrow$ 40dB
Step size μ	Variable
Leakage Factor γ	γ= 0.05

Table 4-10: Parameter used for simulation in Figure 4-10

On Figure 4-10, we are studying the effect of changing the type of noise to use uniform noise instead of Gaussian, using the same conditions as we used before in Figure 4-1. As we can see we have almost the same result, since both the FXLMF and LFXLMF algorithms converge, where the first one keeps converging while the second one reaches the steady state faster. The difference is that we need to use more iterations than with the Gaussian case before the algorithms converge.

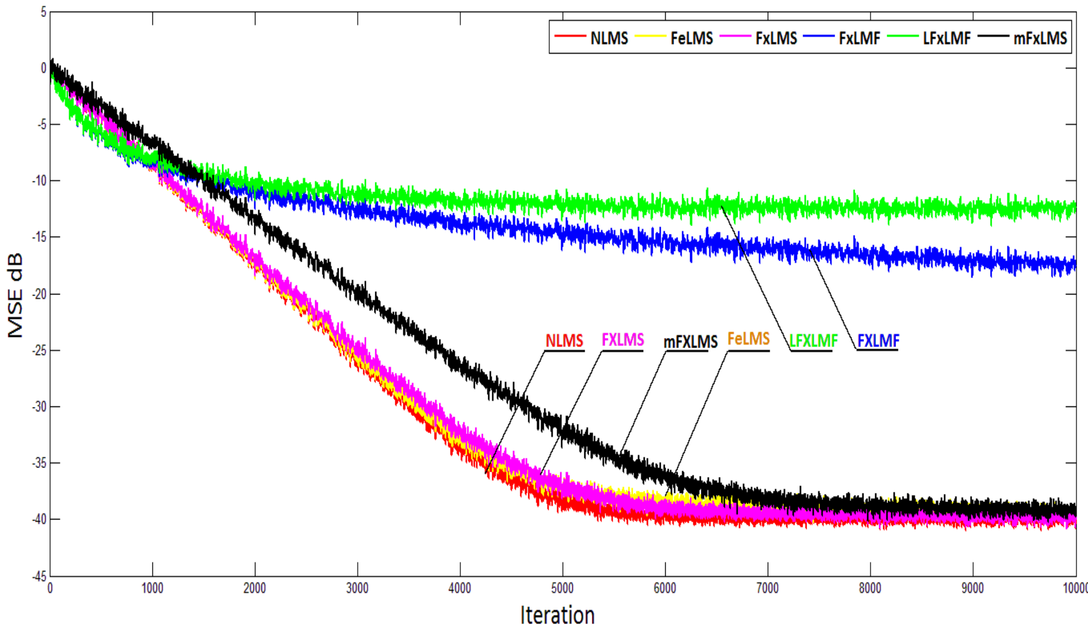


Figure 4-11: Comparison over MSE for FXLMF and LFXLMF with other algorithms using fixed step size and high SNR.

Parameter	Value
Number of iterations	100000
Averaging over	500
Type of Noise	Uniform Noise
Variance of Measurement Noise ${\sigma_z}^2$ ↔SNR	${\sigma_z}^2 = 0.0001$ ↔ 40 dB
Step size μ	$\mu = 0.001$
Leakage Factor γ	γ= 0.05

Table 4-11: Parameter used for simulation in Figure 4-11

Figure 4-11, shows the same behavior under the same conditions for Figure 4-2, but with uniform noise instead of Gaussian.

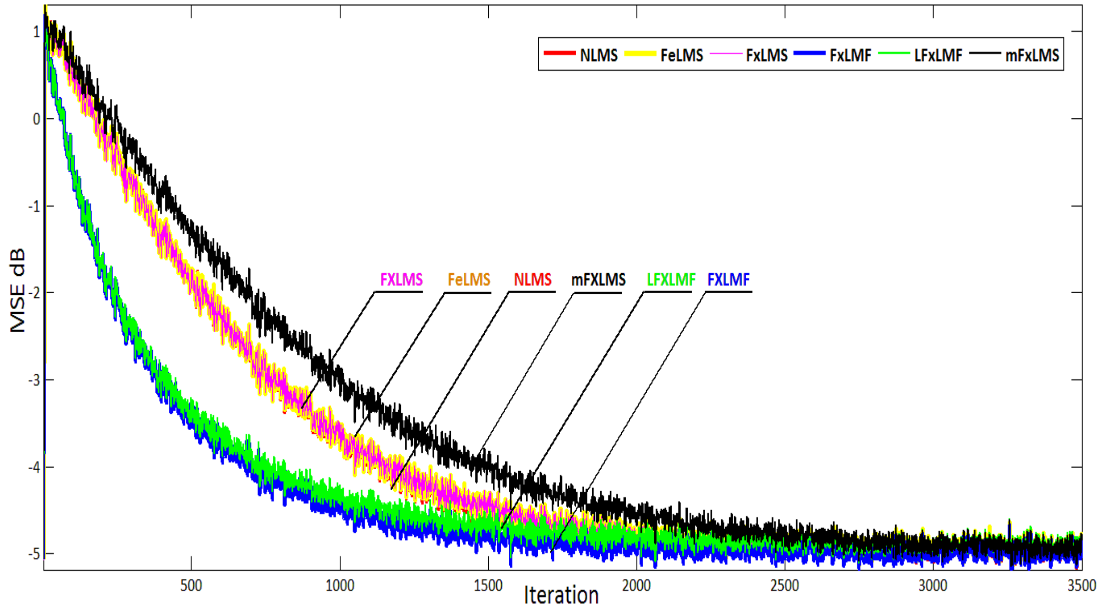


Figure 4-12: Comparison over MSE for FXLMF and LFXLMF with other algorithms using variable step size and low SNR.

Parameter	Value
Number of iterations	3500
Averaging over	1500
Type of Noise	Uniform Noise
Variance of Measurement Noise $\sigma_z{}^2$ ↔SNR	$\sigma_z{}^2 = 0.3163 \leftrightarrow 5$dB
Step size μ	Variable
Leakage Factor γ	$\gamma = 0.05$

Table 4-12: Parameter used for simulation in Figure 4-12

Figure 4-12, shows that our FXLMF and LFXLMF algorithms also outperform the LMS family under low SNR using uniform noise, in the same environment as in Figure 4-3

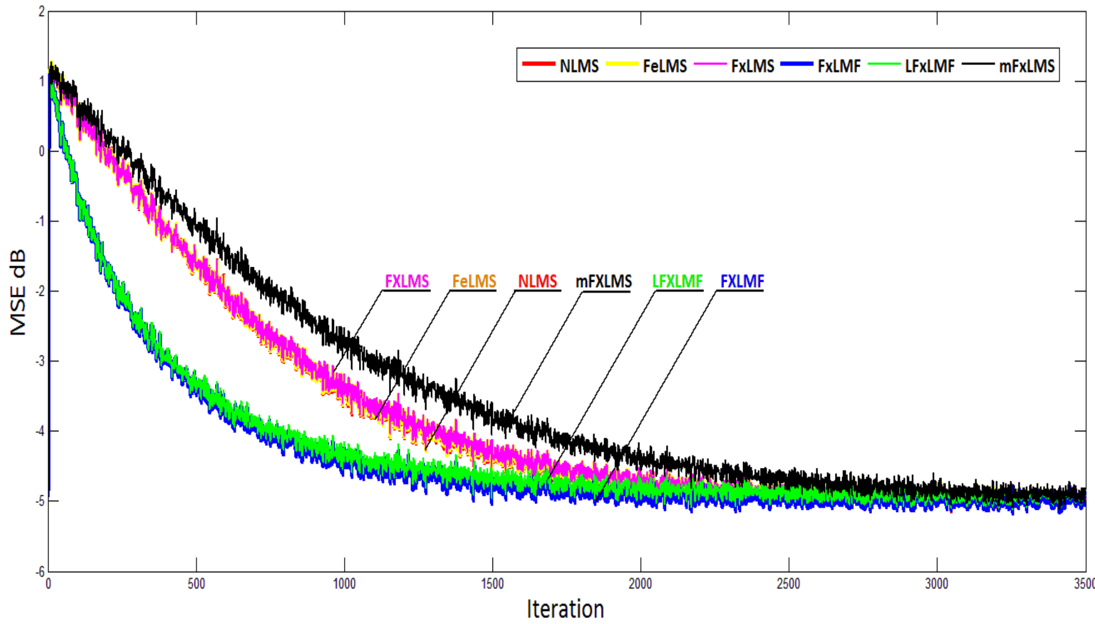


Figure 4-13: Comparison over MSE for FXLMF and LFXLMF algorithms with other algorithms using fixed step size and low SNR.

Parameter	Value
Number of iterations	3500
Averaging over	1500
Type of Noise	Uniform Noise
Variance of Measurement Noise ${\sigma_z}^2$ ↔SNR	${\sigma_z}^2 = 0.3163$ ↔ 5 dB
Step size μ	$\mu = 0.001$
Leakage Factor γ	γ= 0.05

Table 4-13: Parameter used for simulation in Figure 4-13

Figure 4-13 is the same as Figure 4-4, using fixed step size and uniform noise. Also, the FXLMF and LFXLMF algorithms beat the LMS family in speed of convergence.

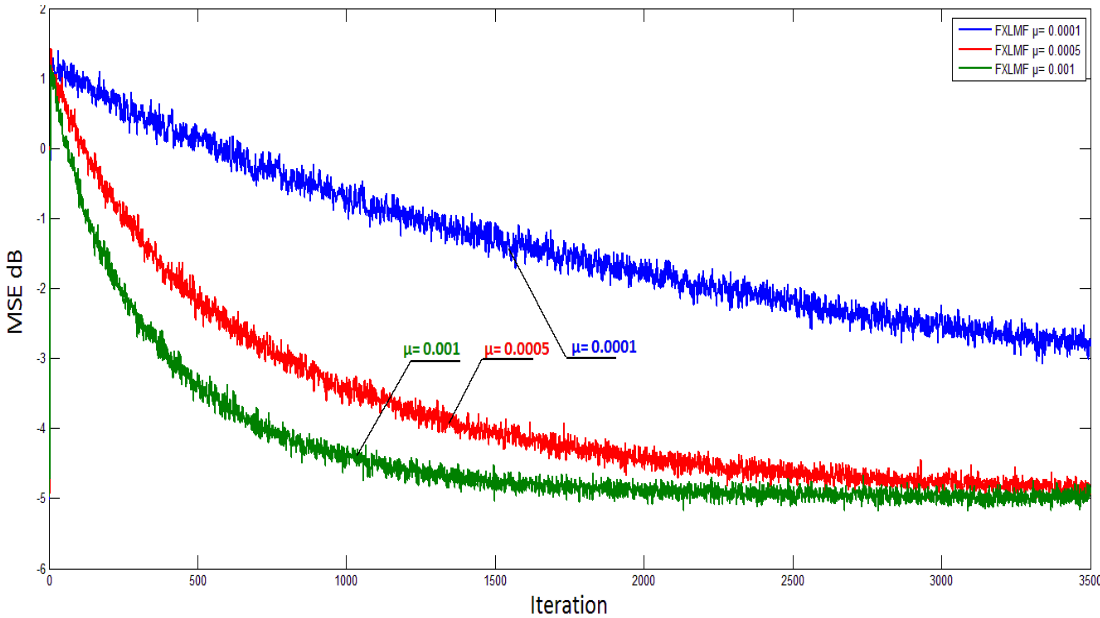


Figure 4-14: Comparison over MSE for FXLMF algorithms using different fixed step size and low SNR.

Parameter	Value
Number of iterations	3500
Averaging over	1500
Type of Noise	Uniform Noise
Variance of Measurement Noise ${\sigma_z}^2$ ↔SNR	${\sigma_z}^2 = 0.3163 \leftrightarrow$ 5dB
Step size μ	$\mu = [0.001, 0.0005, 0.0001]$

Table 4-14: Parameter used for simulation in Figure 4-14

Figure 4-14 shows the effect of changing the step size on the LFXLMF algorithm using uniform noise, and the algorithm has the same behavior as in Figure 4-5.

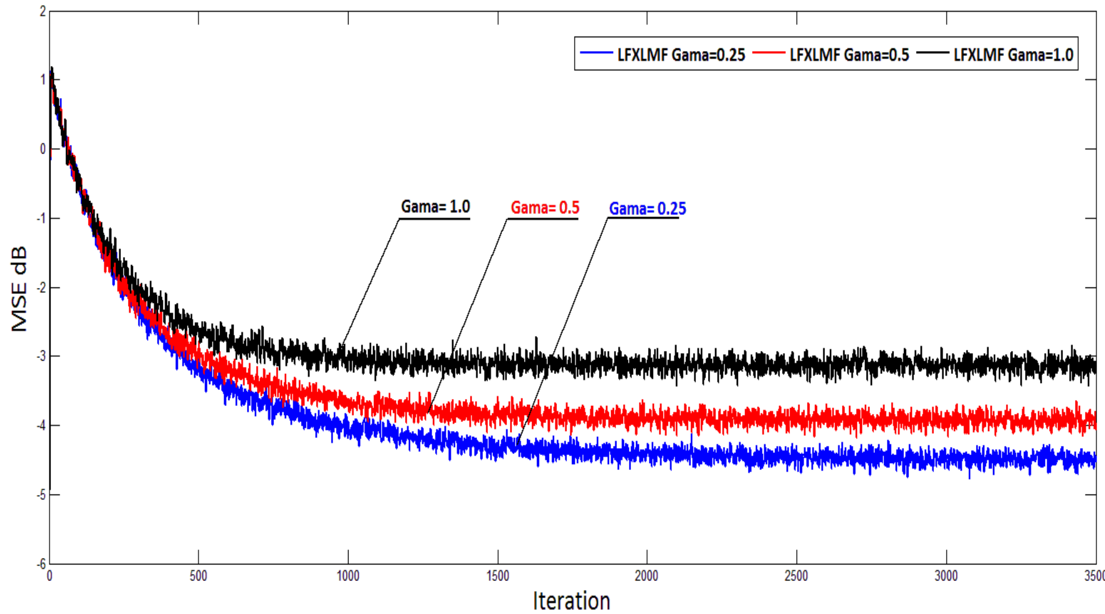


Figure 4-15: Comparison over MSE for LFXLMF algorithm using different fixed leakage factors and fixed step size and low SNR.

Parameter	Value
Number of iterations	3500
Averaging over	1500
Type of Noise	Uniform Noise
Variance of Measurement Noise $\sigma_z{}^2$ ↔SNR	$\sigma_z{}^2 = 0.3163 \leftrightarrow 5$ dB
Step size µ	$\mu = 0.001$
Leakage Factor γ	$\gamma = [0.250, 0.50, 1]$

Table 4-15: Parameter used for simulation in Figure 4-15

Figure 4-15 shows the effect of changing the leakage factor value on the LFXLMF algorithm convergence using uniform noise, and we can see it also has the same behavior as in Figure 4-6.

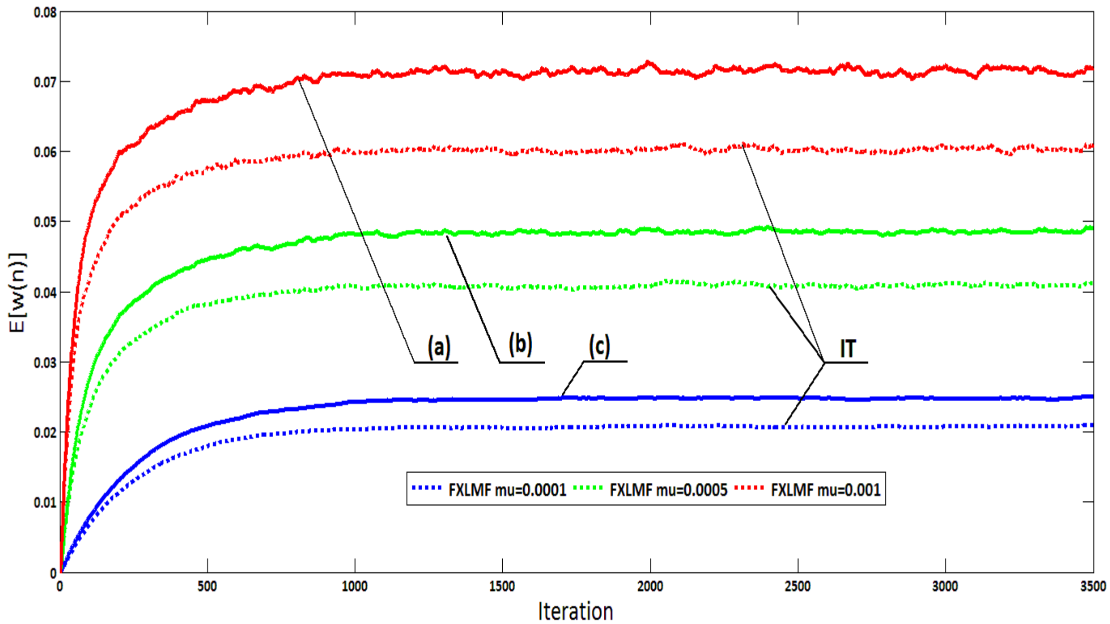


Figure 4-16: Comparison over mean weight vector for FXLMF algorithms using different fixed step size on low SNR.

Parameter	Value
Number of iterations	3500
Averaging over	1000
Type of Noise	Uniform Noise
Variance of Measurement Noise $\sigma_z{}^2$ ↔SNR	$\sigma_z{}^2 = \ 0.3163 \leftrightarrow 5$ dB
Step size μ	$\mu = [0.001,0.0005,0.0001]$

Table 4-16 : Parameter used for simulation in Figure 4-16

Figure 4-16 shows the effect of changing the step size on the mean weight vector of the FXLMF algorithm; as in Figure 4-7 the algorithm converges faster as we increase the step size using uniform noise.

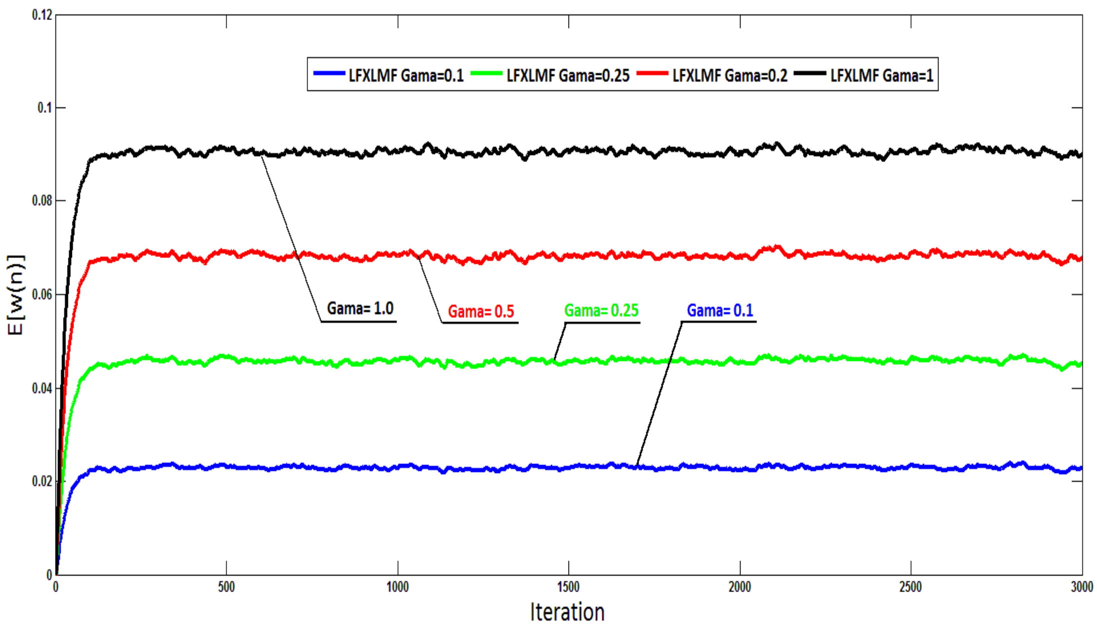


Figure 4-17: Comparison over mean weight vector for LFXLMF algorithms using different fixed leakage factors, and fixed step size on low SNR.

Parameter	Value
Number of iterations	3500
Averaging over	1000
Type of Noise	Uniform Noise
Variance of Measurement Noise ${\sigma_z}^2$ ↔SNR	${\sigma_z}^2 = 0.3163 \leftrightarrow 5$ dB
Step size µ	$\mu = 0.001$
Leakage Factor γ	$\gamma = [0.1, 0.250, 0.50, 1]$

Table 4-17: Parameter used for simulation in Figure 4-17

Finally, Figure 4-17, shows the effect of changing the leakage factor on the mean weight of the LFXLMF algorithm, and as seen in Figure 4-8, increasing the value of the leakage factor will increase the mean weight of the LFXLMF algorithm.

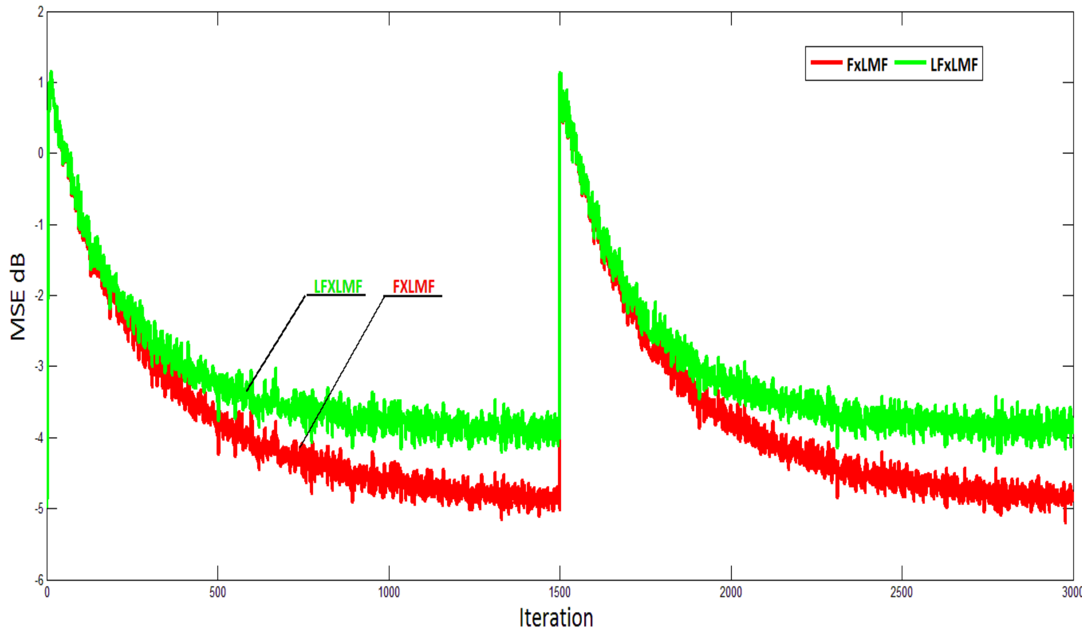


Figure 4-18: MSE for FXLMF and LFXLMF algorithm robustness at low SNR and Uniform noise.

Parameter	Value
Number of iterations	3000
Averaging over	1000
Type of Noise	Uniform Noise
Variance of Measurement Noise $\sigma_z{}^2$ ↔SNR	$\sigma_z{}^2 = 0.3163$ ↔ 5 dB
Step size μ	$\mu = 0.001$
Leakage Factor γ	$\gamma = [0.50]$

Table 4-18: Parameter used for simulation in Figure 4-18

Figure 4-18 shows the robustness of the proposed algorithms FXLMF and LFXLMF at low SNR and using uniform noise.

4.2 Mean Square Performance for Convex Combination with FXLMF Algorithm

Back in Section 0, we discussed the involvement of our proposed FXLMF algorithm, in a convex combination system, then we derived the mixing parameter equation according to the second and fourth order of the error equation. The following simulations show the scenarios where we examine the FXLMF algorithm.

First we used both of the transversal filters to have the same adaptive algorithm, but with different step size, then we did a comparison between the FXLMF and FXLMS algorithms at low and high SNR. All previous simulations were done using the minimization for quadratic error equation.

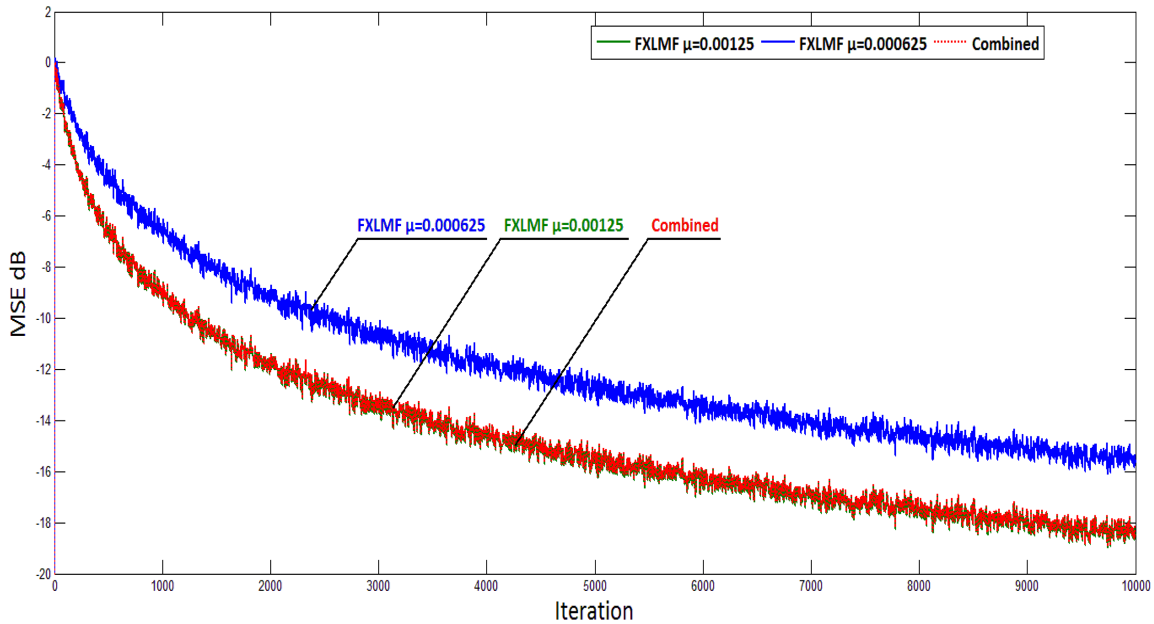


Figure 4-19: MSE for Combined FXLMF for two different step sizes at high SNR.

Parameter	Value
Number of iterations	10000
Averaging over	700
Type of Noise	White Gaussian Noise
Variance of Measurement Noise ${\sigma_z}^2$ ↔SNR	${\sigma_z}^2 = 0.0001$ ↔ 40 dB
Step size μ	$\mu = 0.00125$ & 0.000625

Table 4-19: Parameter used for simulation in Figure 4-19

Figure 4-19 shows the behavior of the convex combined filter when we use the same FXLMF algorithm with two different step sizes at high level of SNR. We can see that the combined filter followed the FXLMF with the larger step size as expected, since the FXLMF algorithm with larger step size is faster and has a lower level of noise at any iteration n.

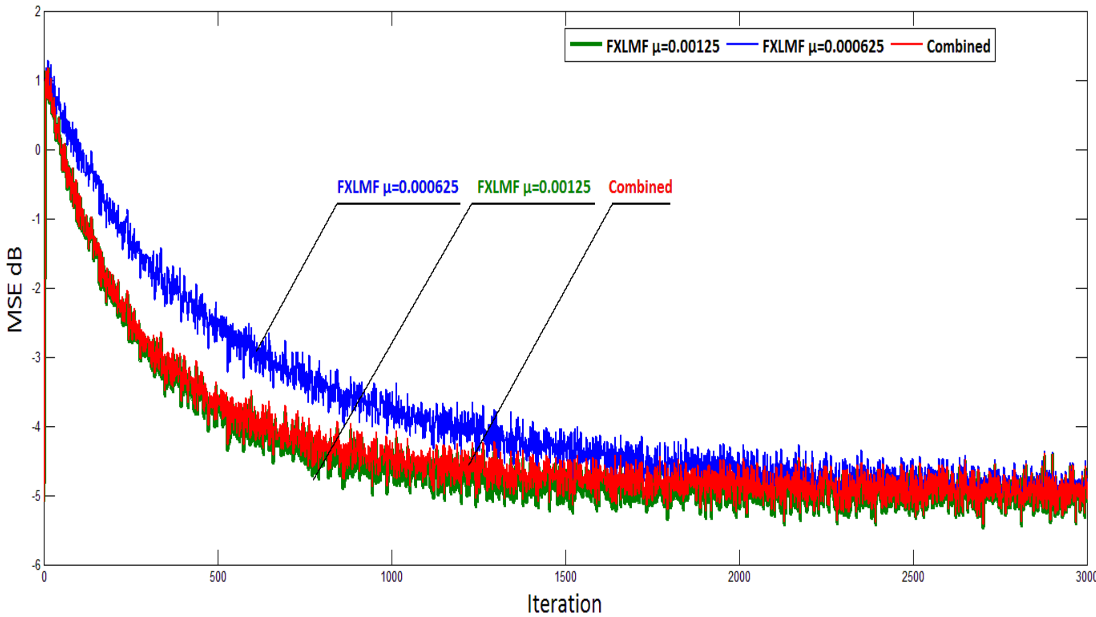


Figure 4-20: MSE for Combined FXLMF for two different step sizes at Low SNR.

Parameter	Value
Number of iterations	3000
Averaging over	1500
Type of Noise	White Gaussian Noise
Variance of Measurement Noise ${\sigma_z}^2$ ↔SNR	${\sigma_z}^2 = 0.3163$ ↔ 5 dB
Step size μ	μ = 0.00125 & 0.000625

Table 4-20: Parameter used for simulation in Figure 4-20

The same behavior for the convex combined filter of the FXLMF algorithm for low SNR.

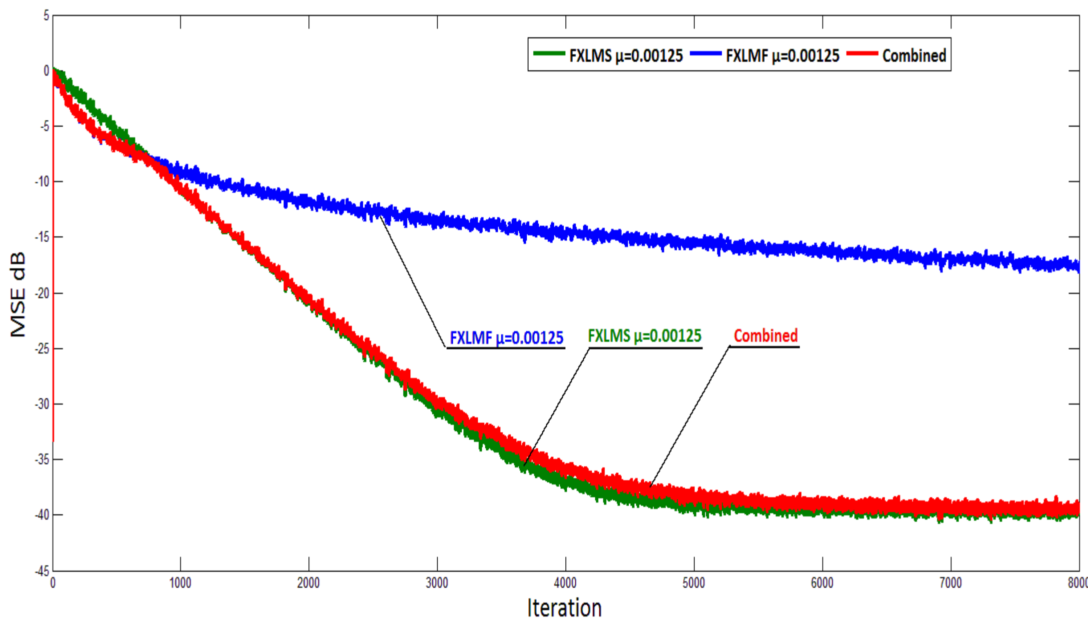


Figure 4-21: MSE for Combined FXLMF and FXLMS at high SNR.

Parameter	Value
Number of iterations	8000
Averaging over	500
Type of Noise	White Gaussian Noise
Variance of Measurement Noise ${\sigma_z}^2$ ↔SNR	${\sigma_z}^2 = 0.0001$ ↔ 40 dB
Step size μ	$\mu = 0.00125$

Table 4-21: Parameter used for simulation in Figure 4-21

Figure 4-21 illustrates the behavior of the convex combined filter of FXLMS and FXLMF algorithms; we can see at the beginning the combined filter followed the FXLMF algorithm since it has a faster speed of convergence. After that the combined filter moved to the FXLMS algorithm, which showed better convergence at high SNR.

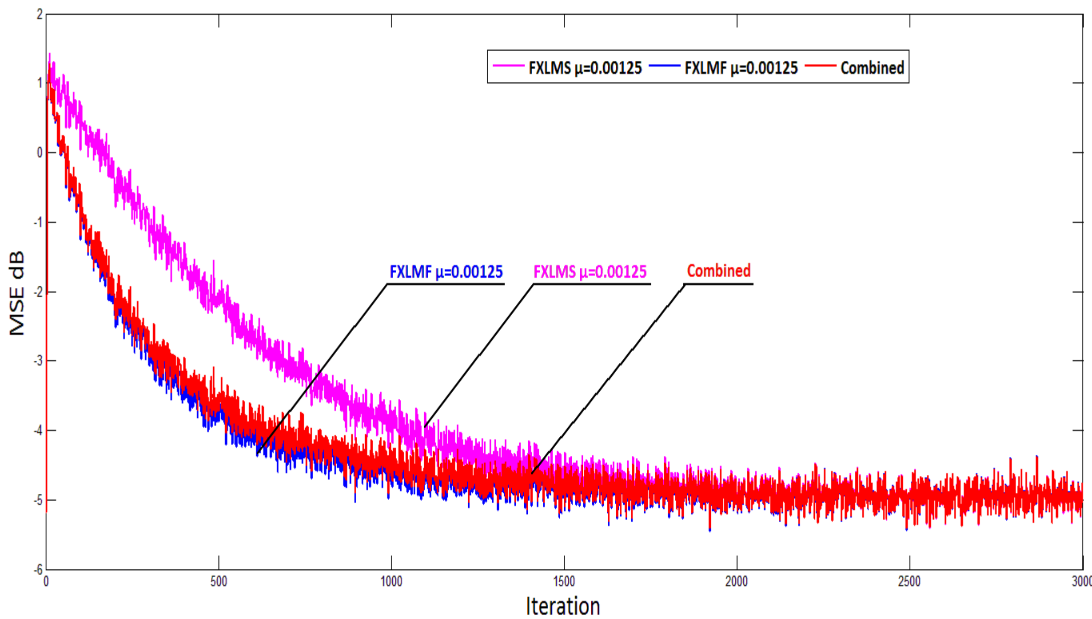


Figure 4-22: MSE for Combined FXLMF and FXLMS at Low SNR.

Parameter	Value
Number of iterations	3000
Averaging over	500
Type of Noise	White Gaussian Noise
Variance of Measurement Noise $\sigma_z{}^2$ ↔SNR	$\sigma_z{}^2 = 0.3163 \leftrightarrow 5$ dB
Step size μ	$\mu = 0.00125$

Table 4-22: Parameter used for simulation in Figure 4-22

The same environment as in Figure 4-21 but with low SNR, where the FXLMF algorithm outperformed the FXLMS algorithm, and the combined filter completely followed the FXLMF algorithm, which shows better performance.

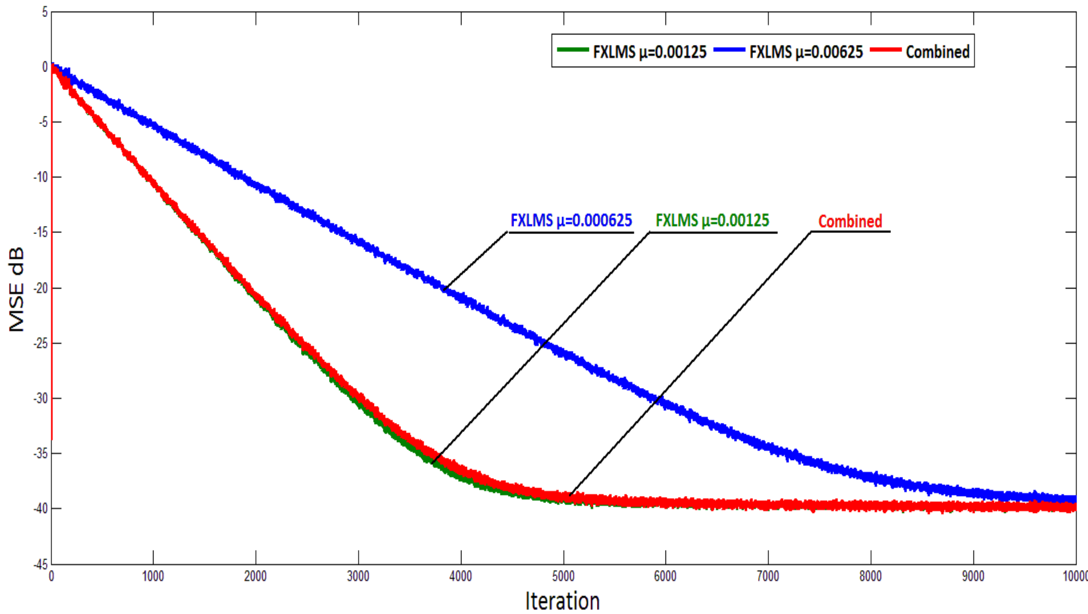


Figure 4-23: MSE for Combined FXLMS for two different step sizes at high SNR.

Parameter	Value
Number of iterations	10000
Averaging over	1500
Type of Noise	White Gaussian Noise
Variance of Measurement Noise ${\sigma_z}^2$ ↔SNR	${\sigma_z}^2 = 0.0001$ ↔ 40 dB
Step size μ	μ = 0.00125 & 0.000625

Table 4-23: Parameter used for simulation in Figure 4-23

For a different step size of the FXLMS algorithm at high SNR, a larger step size outperformed. We can clearly see that the convex combined filter followed the FXLMS algorithm with a larger step size.

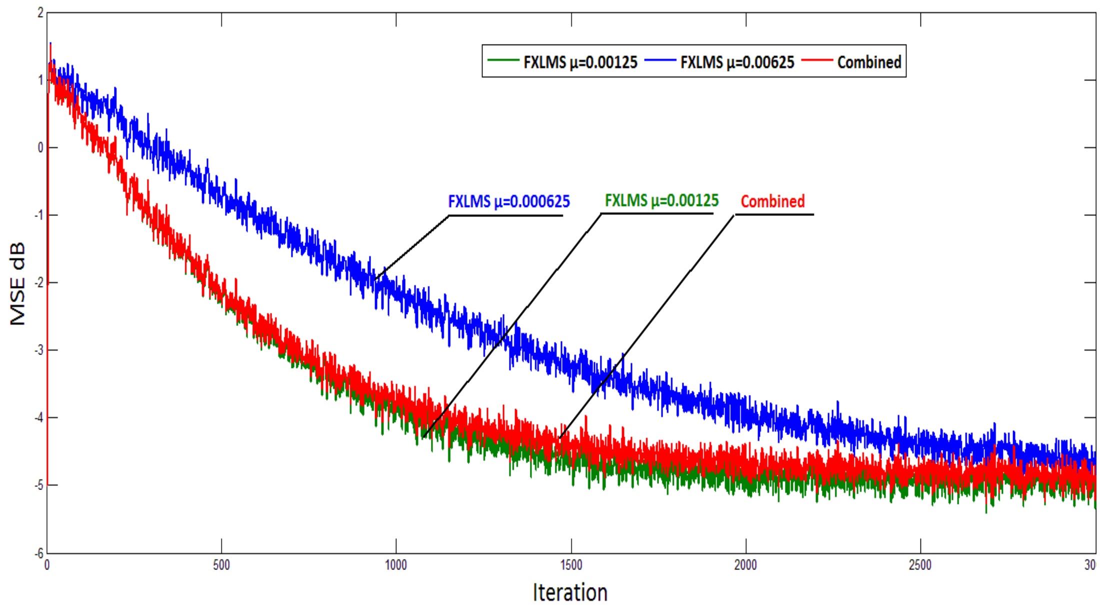


Figure 4-24: MSE for Combined FXLMS for two different step sizes at Low SNR.

Parameter	Value
Number of iterations	3000
Averaging over	500
Type of Noise	White Gaussian Noise
Variance of Measurement Noise ${\sigma_z}^2$ ↔SNR	${\sigma_z}^2 = 0.3163 \leftrightarrow 5$ dB
Step size μ	μ = 0.00125 & 0.000625

Table 4-24: Parameter used for simulation in Figure 4-24

The same as the previous figure, but with low SNR. The combined filter followed the FXLMS with a larger step size, which has better performance.

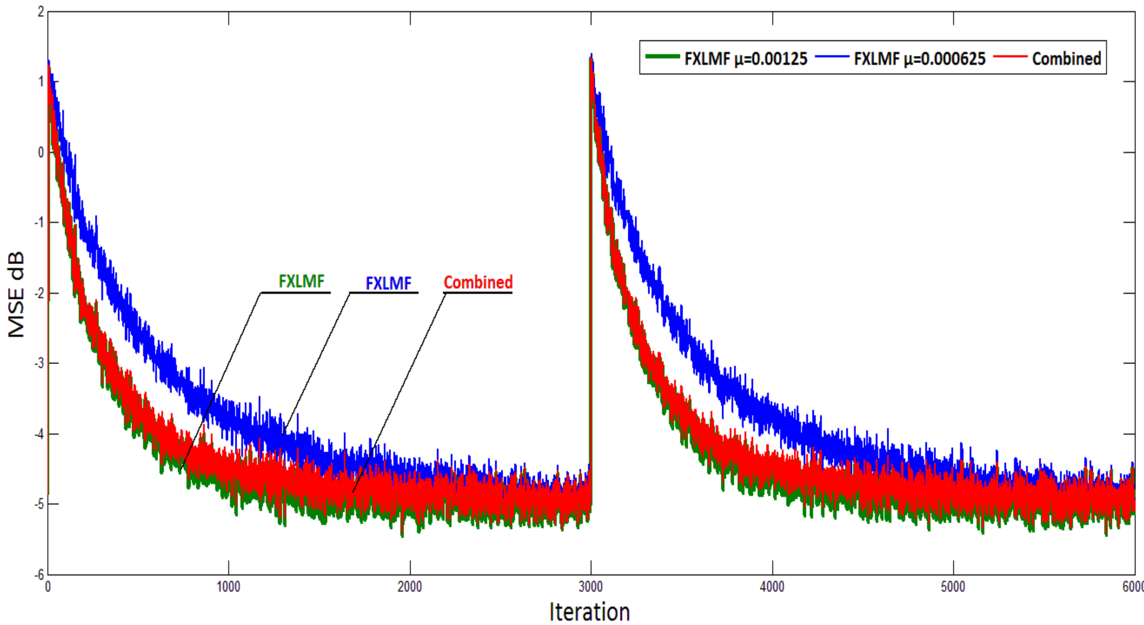


Figure 4-25: MSE for combined FXLMF algorithm robustness at low SNR and Gaussian noise

Parameter	Value
Number of iterations	6000
Averaging over	1500
Type of Noise	White Gaussian Noise
Variance of Measurement Noise σ_z^2 ↔SNR	$\sigma_z^2 = 0.3163 \leftrightarrow 5$ dB
Step size μ	μ = 0.00125 & 0.000625

Table 4-25: Parameter used for simulation in Figure 4-25

Figure 4-25 shows the robustness of the convex combined filter of FXLMF for two different step sizes at low SNR and using Gaussian noise. We can clearly see that the combined filter followed the one with larger step size, which already shows better performance.

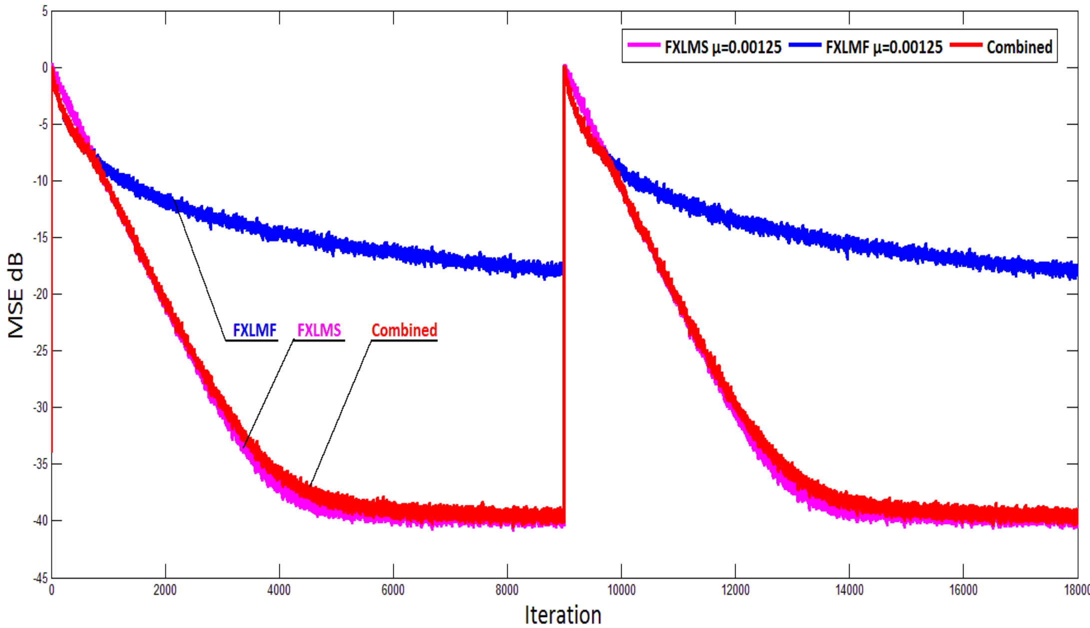


Figure 4-26: MSE for combined FXLMF and FXLMS algorithm robustness test at high SNR and Gaussian noise.

Parameter	Value
Number of iterations	18000
Averaging over	500
Type of Noise	White Gaussian Noise
Variance of Measurement Noise $\sigma_z{}^2$ ↔SNR	$\sigma_z{}^2 = 0.0001$ ↔ 40 dB
Step size μ	μ = 0.00125

Table 4-26: Parameter used for simulation in Figure 4-26

Figure 4-26 shows the robustness of the convex combined filter of FXLMF and FXLMS algorithms at high SNR and using Gaussian noise. We can clearly see that the combined filter followed the FXLMF algorithm at the beginning, then turned into the FXLMS algorithm, which shows better performance at high SNR.

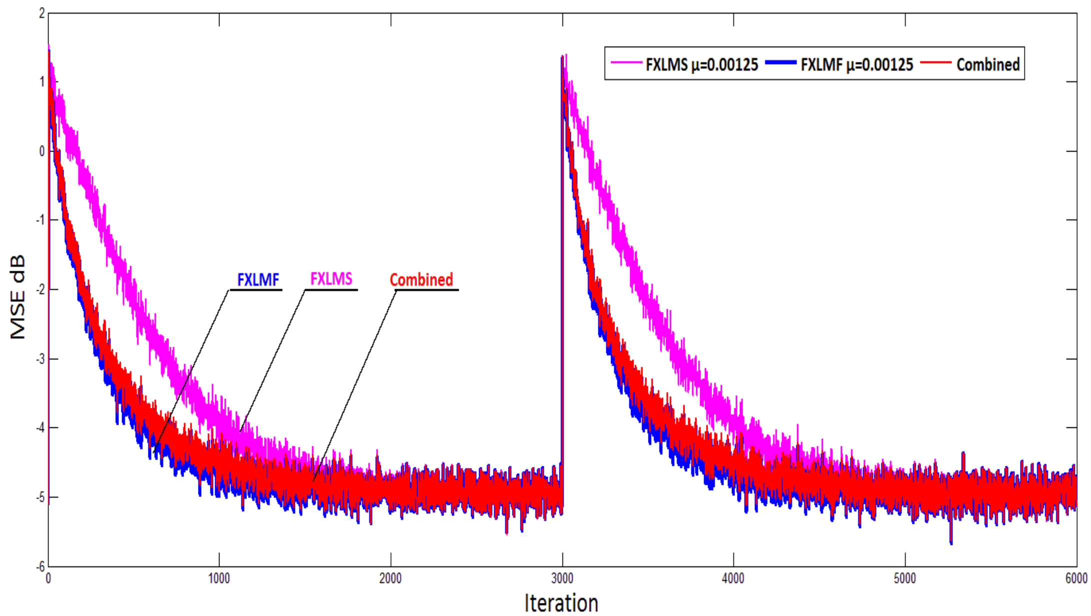


Figure 4-27: MSE for combined FXLMF and FXLMS algorithm robustness test at low SNR and Gaussian noise.

Parameter	Value
Number of iterations	6000
Averaging over	1500
Type of Noise	White Gaussian Noise
Variance of Measurement Noise σ_z^2 ↔SNR	$\sigma_z^2 = 0.3163 \leftrightarrow 5$ dB
Step size μ	$\mu = 0.00125$

Table 4-27: Parameter used for simulation in Figure 4-27

Figure 4-27 shows the robustness of the convex combined filter of FXLMF and FXLMS algorithms at low SNR and using Gaussian noise. We can clearly see that the combined filter followed the FXLMF algorithm all the time since it shows better performance than the FXLMS algorithm at low SNR.

5. CHAPTER 5

CONCLUSION

5.1 Conclusions

FXLMF and LFXLMF algorithms were proposed by this work, an analytical study and mathematical derivations for the mean weight adaptive vector and the mean square error for both algorithms have been done, and moreover the step size and the leakage factor bound ranges were investigated.

From the literature we received a good sense about proposing new algorithms to the LMF family, as was proposed before in the LMS. The FXLMF and LFXLMF algorithms successfully converge under a large range of SNR. Moreover, they distinctly outperform the LMS algorithm in speed of convergence under a low SNR. Furthermore, we see the ability of both algorithms to converge under different environments of noise: Gaussian, uniform distributed noise, and colored noise. However, the LMF family requires more computational complexity; our proposed algorithms were faster in convergence than members of the LMS family under some circumstances.

From the simulations in chapter four, we see that both algorithms converge well under relatively high SNR, but they converge faster under low SNR. Also, we see that using a variable step size requires fewer iterations before the algorithms converge, compared to the fixed step size values. In addition, using a step size near the upper boundary will guarantee less time to converge; however, working close to the upper boundary of the step size ensures faster convergence but we have to take the risk of algorithm divergence. Also, we see that a larger step size will increase the mean of the weight vector. Step size under the upper boundary is given in equation (2.40).

The leakage factor in the LFXLMF algorithm reduces the performance of the algorithm, as was expected from the literature, at the expense of adding more stability to the algorithm. The leakage factor boundaries were derived in section (2.5.3).

The convex combination is an interesting proposal to get the best features of two or more adaptive algorithms. We were able to successfully apply it using the FXLMF and FXLMS algorithms with different step sizes as the two main transversal filters. In the other scenario we applied the combination over the FXLMS and FXLMF algorithms, and we noticed that the convex combined filter, at every iteration, followed the algorithm best.

A robustness test was done for all the scenarios used, to ensure that the proposed algorithms are able to adapt in case of sudden change of tap weights of the filter, either in the transient or steady state stage.

5.2 Contributions

This work successfully proposed two new algorithms, the FXLMF and LFXLMF algorithms. An analytical study by mathematical derivations and MATLAB simulation was conducted for both of them, and a comparison was made of the proposed algorithms with previous work in LMS family. Furthermore we conducted analysis in two noise environments and the algorithms proved their convergence and stability.

We performed a means square error performance study for the convex combination of FXLMF and FXLMS algorithms with themselves at different step sizes, and with each other at different SNR. We saw that the combined filter followed the algorithm with the best features at every iteration to ensure we have the best overall behavior of the convex combined filter.

5.3 Recommendations and Future Work

Adaptive filtering is a rich land for a researcher to invest his time. This hot and renewable topic in digital signal processing keeps promising more development and progression in the communication and control world.

Future work on these two algorithms is too big and a lot of fields need to be investigated. The following are some areas of future work which can be done as a supplement for our proposed algorithm:

- Examining different types of statistical input data, like the human voice and sinusoidal signals.
- Apply the proposed algorithm, using the Lattice or IIR filter.

REFERENCES

[1] Haykin, S.: ‘Adaptive filter theory’ (Prentice-Hall, Upper Saddle River, NJ, USA, 1996)

[2] Sayed, A.H.: ‘Adaptive filters’ (Wiley, NJ, USA, 2008.

[3] Farhang- Boroujeny: ‘Adaptive Filters Theory and Applications’ (Wiley & Sons 2013), pp551-583.

[4] P. Lueg, “Process of silencing sound oscillations.” U.S. Patent No. 2043416, 1936.

[5] S.M. Kuo, and D.R. Morgan: “Active Noise control systems, algorithms and DSP implementation functions,” New York, Wiley 1996.

[6] Widrow, and D. Shur, and Shaffer: ‘On Adaptive Inverse Control’ in Proc. 15th Asilomar Conf., 1981, pp185-189.

[7] B.Widrow and S.D.Stearns, “Adaptive Signal Processing”, Prentice-Hall, 1985.

[8] W. A. Gardner, “Learning characteristics of stochastic-gradient-descent algorithms: a general study, analysis and critique”. Signal Processing, 6:113–133, 1984.

[9] J. E. Mazo, “On the independence theory of equalizer convergence,” Bell Syst. Tech. J., vol. 58, pp. 963–993, May 1979.

[10] B.Widrow and M.E.Hoff, “Adaptive switching circuits,” Proc. Of WESCON Conv. Rec., part 4, pp.96-140, 1960.

[11] E. Bjarnason: “Analysis of the filtered-X LMS algorithm,” IEEE Trans. Speech Audio Process., pp. 504–514, Nov. 1995.

[12] O. J. Tobias, J. C. M. Bermudez, N. J. Bershad, and R. Seara, "Mean weight behavior of the filtered-X LMS algorithm," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 1998, pp. 3545–3548.

[13] O. J. Tobias, "Stochastic Analysis of the Filtered-X LMS Algorithm," Ph.D dissertation, Federal Univ. Santa Catarina, Santa Catarina, Brazil, Oct. 1999.

[14] O. J. Tobias, J. C. M. Bermudez, and N. J. Bershad, "Mean weight behavior of the filtered-X LMS algorithm," IEEE Trans. Signal Process., vol. 48, no. 4, pp. 1061–1075, Apr. 2000.

[15] M. Rupp, A. H. Sayed. : 'Modified FXLMS algorithm with improved convergence performance' in IEEE Proc. of Asilomar-29, pp1255-1259, 1996.

[16] R. P. Gitlin et al., "The tap-leakage algorithm: An algorithm for the stable operation of a digitally implemented fractionally spaced equalizer," Bell Syst. Tech. J., vol. 61, no. 8, pp. 1817–1839, Oct. 1982.

[17] Vicente, L. and Masgrau,. "Novel FxLMS Convergence Condition with Deterministic Reference". IEEE Trans. Signal Processing, vol. 54, 3768-3774 (2006).

[18] K. Mayyas and T. Aboulnasr, "Leaky LMS algorithm: MSE analysis for Gaussian data," IEEE Trans. Signal Process., vol. 45, no. 4, pp. 927–934, Apr. 1997.

[19] J. R. Gonzal et al., "The dynamics of bursting in simple adaptive feedback systems with leakage," IEEE Trans. Circuits Syst., vol. 38, pp. 475–488, May 1991.

[20] O. J. Tobias, and R. Seara. : 'Leaky-FXLMS Algorithm: Stochastic Analysis for Gaussian Data and Secondary Path Modeling Error' IEEE Trans. 'On Speech and Audio Processing', VOL. 13, NO. 6, NOV. 2005

[21] E. Walach, and B. Widrow,: 'The Least Mean Fourth (LMF) Adaptive Algorithm and Its Family' IEEE Trans. on Information Theory, VOL. IT-30, NO. 2, MARCH 1984.

[22] O. Khattak, and A. Zerguine,: ' Leaky Least Forth Adaptive Algorithm' IET Signal Processing, Vol. 7 Iss.2, pp.134-145, 2013.

[23] Azzedine Zerguine : 'Convergence Behavior of the Normalized Least Mean Fourth Algorithm' IEEE 2000.

[24] Nascimento, V.H., Bermudez, J.C.M.: 'Probability of divergence for the least-mean fourth (LMF) algorithm', IEEE Trans. Signal Process., 2006, 54, pp. 1376–1385.

[25] Katz, Victor (2004). A History of Mathematics. Boston: Addison Wesley. p. 220. ISBN 9780321016188.

[26] Papoulis, A., Pilai, S.: 'Probability, random variables and stochastic processes' (McGraw Hill, New York, 2002).

[27] S. Haykin, Adaptive Filter Theory, 4th ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[28] V. Solo and X. Kong, Adaptive Signal Processing Algorithms: Stability and Performance. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[29] B. Pathak, and P P. Hirave. 'FXLMS Algorithm for Feed forward Active Noise Cancellation' Proc. of the Intl. Conf. on Advances in Computer, Electronics and Electrical Engineering, 2012.

[30] Bermudez, J.C.M., Hubscher, P.I., Nascimento, V.H.: ‘A mean-square stability analysis of the least mean fourth (LMF) adaptive algorithm’, IEEE Trans. Signal Process., 2007, 55, pp. 4018–4028

[31] Hubscher, P.I., Bermudez, J.C.: ‘An improved statistical analysis of the least mean fourth algorithm’, IEEE Trans. Signal Process., 2003, 51, (3), pp. 664–671.

[32] Zerguine, A., Chan, M.K., Al-Naffouri, T.Y., Moinuddin, M., Cowan, C.F.N.: ‘Convergence and tracking analysis of a variable normalized LMF (XE-NLMF) algorithm’, Signal Process. 2009, 89, (5), pp. 778–790.

[33] J. C. Burgess, “Active adaptive sound control in a duct: A computer simulation,” J. Acoust. Soc. Am., 70, 715-726, Step. 1981.

[34] P. Anderson, “Adaptive forgetting in recursive identification through multiple models,” Int. J. Contr., vol. 42, pp. 1175–1193, 1985.

[35] S. S. Kozat and A. C. Singer, “Multi-stage adaptive signal processing algorithms,” in Proc. 2000 IEEE Sensor Array Multichannel SignalWork- shop, Cambridge, MA, 2000, pp. 380–384.

[36] A. C. Singer and M. Feder, “Universal linear prediction by model order weighting,” IEEE Trans. Signal Process., vol. 47, pp. 2685–2700, 1999.

[37] M. Niedz`wiecki, “Multiple-model approach to finite memory adaptive filtering,” IEEE Trans. Signal Process., vol. 40, pp. 470–473, 1992.

VITAE

- Ali Mahmoud Ali Al Omour.
- Born in Riyadh, Saudi Arabia on May 23, 1990.
- Received Bachelor of Engineering (BE) degree in Electrical Engineering from An Najah National University, Nablus, Palestine August 2011.
- Joined National Palestinian Company (NPC- Coca Cola) as a Technical Support Engineer January 2012.
- Joined King Fahd University of Petroleum and Minerals KFUPM in September 2012 as a Full Time Student in Electrical Engineering Department to continue graduate studies in Telecommunication Engineering.
- Contact Details:

- Present Address: Department of Electrical Engineering, KFUPM, Dhahran 31261, Saudi Arabia.
- E-mail Address: aalomour@gmail.com, g201204440@kfupm.edu.sa .
- Mobile: +966 552782728 KSA, +970 42440145 PAL.
- Permanent Address: Bader quarter, Rummanah, Jenin, Palestine.