Frequency-Response-Shaped Least Mean Square Adaptive Channel Estimation

Mohammad Mustafa Shukri AHMAD

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

Master of Science
in
Electrical and Electronic Engineering

Eastern Mediterranean University
June, 2009

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ayhan Bilsel
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of
Master of Science in Electrical and Electronic Engineering

Prof. Dr. Derviş Z. Deniz
Chairman

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

Assoc. Prof. Dr. Aykut Hocanın
Supervisor

Prof. Dr. Osman Kükrer
Cosupervisor

Examining Committee

1. Assoc. Prof. Dr. Aykut Hocanın

2. Prof. Dr. Osman Kükrer

3. Assoc. Prof. Dr. Hüseyin Bilgekul

4. Assoc. Prof. Dr. Hüseyin Özkaramanlı

5. Asst. Prof. Dr. Hasan Demirel

ACKNOWLEDGEMENTS

I would like to give my sincere gratitude to my supervisor Assoc. Prof. Dr. Aykut Hocanin and my co-supervisor Prof. Dr. Osman Kukrer for their continuous support, great guidance, endless help, knowledge and huge confidence they gave me. Many thanks to my department and fellow associates for their help and support during my course of study. My warm regards to the my friend Hasan AbuHilal for his presence, as it enhanced my motivation by making me feel at home. Finally, words will not be enough to thank my family for their infinite support and patience, thanks to you; especially my parents and my brother Nael.

ABSTRACT

Frequency-Response-Shaped Least Mean Square (FRS-LMS) Adaptive Channel Estimation

Keywords: Adaptive Filters, LMS algorithm, Leaky LMS algorithm, Modified Leaky LMS, FRS-LMS, Correlated Noise, Impulsive noise, Channel estimation.

In this thesis, Mean Square Error (MSE) performance of the standard Least Mean Square (LMS), Normalized LMS (NLMS), Leaky LMS, Modified Leaky LMS (ML-LMS) and Frequency Response Shaped Least Mean Square (FRS-LMS) algorithms have been investigated in Additive White Gaussian Noise (AWGN) and correlated Gaussian noise environments. The FRS-LMS algorithm has been shown to have superior performance in terms of MSE or speed of convergence compared to the other algorithms. The performance of the FRS-LMS adaptive algorithm in estimating a sinusoidal signal in impulsive and correlated noise is further studied. The algorithm does not require a priori knowledge about the nominal Gaussian process and is able to adapt to changes in the environment. The performance of the FRS-LMS is compared to that of the Leaky-LMS algorithms in terms of MSE and convergence speed. The results indicate that the FRS-LMS provides superior performance in impulsive and correlated noise environments. The performance gain is due to the frequency shaping and outlier reduction properties of the algorithm.

An LMS type adaptive channel estimation algorithm is proposed. The proposed algorithm employs the FRS-LMS algorithm which outperforms the standard LMS and the

ML-LMS algorithms in a fading channel with correlated Gaussian noise. The algorithm is based on shaping the frequency response of the transversal filter by inclusion of a leakage factor in matrix form. The results show that the FRS-LMS algorithm converges to a lower MSE compared with the ML-LMS in correlated Gaussian noise leading to a substantial decrease in bit error rate. In a fading channel with AWGN, the proposed algorithm has a lower performance gain relative to the ML-LMS algorithm but provides a significant performance advantage over the standard LMS algorithm.

ÖZET

Frekans-Tepki-Şekillendirmeli Enküçük Ortalama Kare (FTŞ-EOK) Algoritması Kullanarak Uyarlanır Kanal Kestirimi

Anahtar Sözcükler: Uyarlanır Süzgeçler, EOK Algoritması, Sızdırmalı EOK, Değiştirilmiş Sızdırmalı EOK, FTŞ-EOK, İlintili Gürültü, Dürtün Gürültü, Kanal Kestirimi.

Bu tezde, standard EOK, Sızdırmalı EOK (S-EOK), Değiştirilmiş Sızdırmalı EOK (DS-EOK), Düzgelenmiş EOK (D-EOK) ve FTŞ-EOK algoritmaları incelenmiş ve Toplanır Beyaz Gauss Gürültüsü (TBGG) ve ilintili gürültü ortamlarındaki başarımları karşılaştırılmıştır. FTŞ-EOK algoritmasının Ortalama Karesel Hata (OKH) ve yakınsama hızı kriterlerine göre daha yüksek başarıma sahip olduğu gösterilmiştir. FTŞ-EOK algoritmasının, bir sinüs işaretinin TBGG ve ilintili gürültü ortamında kestirimi konusu detaylı olarak incelenmiş ve sonuçlar sunulmuştur. Söz konusu algoritma herhangi bir ön bilgiye gereksinim duymamakta ve uygulandığı ortama hızlı uyum sağlayabilmektedir. Dürtün gürültünün bulunduğu ortamda, FTŞ-EOK algoritması aşırı değerleri azaltmakta ve frekans tepkisini de şekillendirerek yüksek başarım sağlamaktadır.

FTŞ-EOK tabanlı yeni bir uyarlanır kanal kestirim algoritması önerilmiştir. Önerilen algoritma, sönümlemeli iletişim kanallarında, standard EOK ve DS-EOK algoritmalarından daha yüksek başarım sağlamaktadır. Başarım kazancı kanaldaki gürültünün ilintili olduğu durumda daha fazladır. Algoritma, bir matris yardımıyla sızdırma sağlamakta ve ilintili gürültü ortamında süzgecin frekans şekillendirmesine olanak tanımaktadır. Algoritmanın kullanımı OKH'yı düşürüp Bit Hata Oranını (BHO) azaltmasına yol açmaktadır. FTŞ-

EOK tabablı algoritma, TBGG'ye sahip sönümlemeli kanallarda, DS-EOK'ya göre daha az başarım kazancı sağlamakla birlikte, standard EOK'ya göre çok daha başarılıdır.

For my parents

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

ϵ	Impulsive noise frequency
κ	Impulsive noise strength
γ	Leakage factor
λ	Eigenvalue
λ_w	Wavelength
μ	Convergence rate
σ^2	Variance
σ_τ	rms delay spread
τ	Time delay
$\overline{\tau}$	Mean excess delay
B_c	Coherence bandwidth
B_D	Doppler spread
f_c	Carrier frequency
f_d	Doppler frequency shift
f_m	Maximum Doppler frequency shift
N	Filter Length
p	Cross-correlation matrix
R	Correlation matrix
T_c	Coherence time
w	Tap weights vector

AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CLT	Central Limit Theorem
DSP	Digital Signal Processing
EMSE	Ensemble Mean Square Error
FIR	Finite Impulse Response
FRS-LMS	Frequency Response Shaped Least Mean Square
IIR	Infinite Impulse Response
ISI	Inter-Symbol Interference
Leaky LMS	Leaky Least Mean Square
LMS	Least Mean Square
LOS	Line-of-Sight
MAP	Maximum A posteriori Probability
ML	Maximum Likelihood
ML-LMS	Modified Leaky Least Mean Square
MLSE	Maximum Likelihood Sequence Estimation
MSE	Mean Square Error
NLMS	Normalized Least Mean Square
NLOS	Non-Line-of-Sight
pdf	probability density function
QPSK	Quadrature Phase Shift Keying
SNR	Signal to Noise Ratio
W-CDMA	Wide Code Division Multiple Access

CHAPTER 1

ADAPTIVE FINITE IMPULSE RESPONSE (FIR) FILTERS

1.1. Introduction

Digital Signal Processing (DSP) is the major technology that can be applied to noise filtering, system identification, and voice prediction. Standard DSP techniques are not enough to solve these problems quickly and obtain acceptable results. Adaptive filtering techniques must be implemented to obtain accurate solutions with timely convergence.

1.2. Adaptive Filtering System Configurations

Adaptive filter had first established its engineering use in 1960s. It was applied as an equalizer to combat the effect of Inter-Symbol Interference (ISI) of data transmission in telephone channels [1]. Since then, adaptive filter was modified into different forms and applied in many different areas such as; signal processing, communication systems,etc.

There are four major types of adaptive filtering configurations; adaptive system identification, adaptive noise cancellation, adaptive linear prediction, and adaptive inverse system. All of the above systems are similar in the implementation of the algorithm, but different in system configuration. All four systems have the same general components; an input signal $\mathbf{x}(n)$, a desired result $d(n)$, an output $y(n)$, an adaptive transfer function $\mathbf{w}(n)$, and an error signal $e(n)$ which is the difference between the desired output $d(n)$

and the actual output $y(n)$. In addition to these, the system identification and the inverse system configurations have an unknown linear system $\mathbf{u}(n)$ that can produce a linear output to the given input [1].

1.2.1. Adaptive System Identification Configuration

The adaptive system identification is primarily responsible for determining a discrete estimation of the transfer function for an unknown digital or analog system. The same input $\mathbf{x}(n)$ is applied to both the adaptive filter and the unknown system from which the outputs are compared, as shown in Fig. 1.1. The output of the adaptive filter $y(n)$ is subtracted from the output of the unknown system resulting in a desired signal $d(n)$. The resulting difference is an error signal $e(n)$ which is used to manipulate the filter coefficients of the adaptive system. After convergence, the error signal tends toward zero.

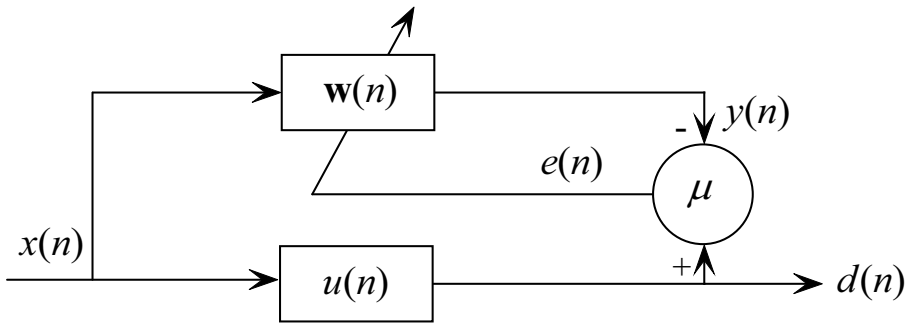

Figure 1.1. Adaptive System Identification Configuration [1].

After a number of iterations of this process, the adaptive filter's transfer function will converge to, or near to, the unknown system's transfer function. For this configuration, the error signal does not have to go to zero (although convergence to zero is the ideal situation) to closely approximate the given system. There will, however, be a difference

between the adaptive filter transfer function and the unknown system transfer function if the error is nonzero and the magnitude of that difference will be directly related to the magnitude of the error signal.

1.2.2. Adaptive Noise Cancellation Configuration

The second configuration is the adaptive noise cancellation configuration as shown in Fig. 1.2. In this configuration, the input $\mathbf{x}(n)$ (a noise source $N_1(n)$), is compared with a desired signal $d(n)$, which consists of a signal $s(n)$ corrupted by another noise signal $(N_0(n))$. The adaptive filter coefficients adapt to cause the error signal to be a noiseless version of the signal $s(n)$.

Both of the noise signals for this configuration need to be uncorrelated to the signal $s(n)$. In addition, the noise sources must be correlated to each other in some way, preferably equal, to get the best results [2].

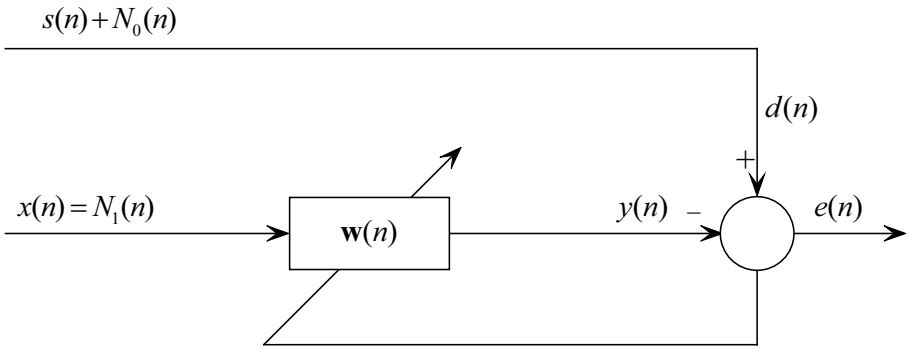

Figure 1.2. Adaptive Noise Cancellation Configuration [1].

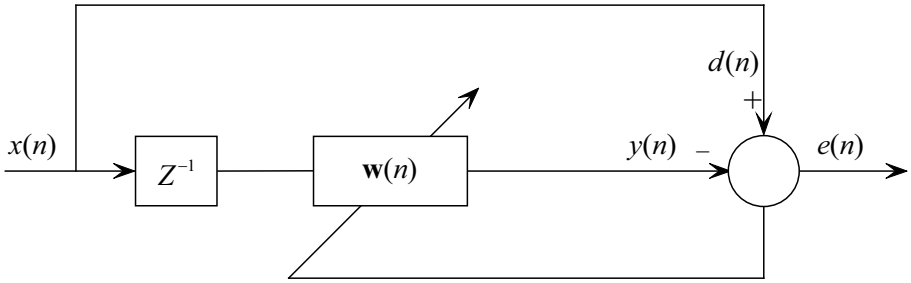

Figure 1.3. Adaptive Linear Prediction Configuration [1].

1.2.3. Adaptive Linear Prediction Configuration

Adaptive linear prediction is the third type of adaptive configuration as shown in Fig. 1.3. This configuration essentially performs two operations: The first operation, is linear prediction; if the output is taken from the error signal $e(n)$. The adaptive filter coefficients are being trained to predict, from the statistics of the input signal $\mathbf{x}(n)$, what the next input signal will be. The second operation, is a noise filter similar to the adaptive noise cancellation outlined in the previous section; if the output is taken from $y(n)$, .

In the case of noise filtering, as outlined in the previous section, $y(n)$ will converge to the noiseless version of the input signal.

1.2.4. Adaptive Inverse System Configuration

The final filter configuration is the adaptive inverse system configuration as shown in Fig. 1.4. The goal of the adaptive filter here is to model the inverse of the unknown system $u(n)$. This is particularly useful in adaptive equalization where the goal of the filter is to eliminate any spectral changes that are caused by a prior system or transmission line. The way this filter works is as follows; the input $\mathbf{x}(n)$ is sent through the unknown

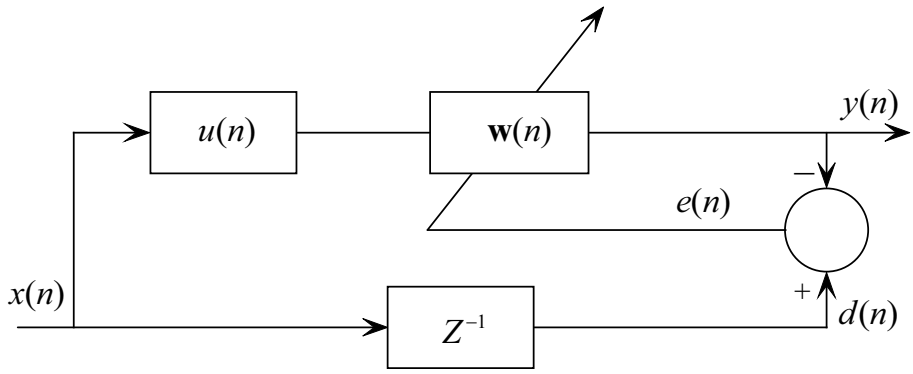

Figure 1.4. Adaptive Inverse System Configuration [1].

system $u(n)$ and then through the adaptive filter resulting in an output $y(n)$. The input is also sent through a delay to attain $d(n)$. As the error signal is converging to zero, the adaptive filter coefficients $\mathbf{w}(n)$ are converging to the inverse of the unknown system $u(n)$.

For this configuration, the error can theoretically go to zero. This is only true if the unknown system consists only of a finite number of poles or the adaptive filter is an Infinite Impulse Response (IIR) filter. If neither of these conditions is true, the system will converge only to a constant due to the limited number of zeroes available in a Finite Impulse Response FIR system [1].

1.3. Performance Measures in Adaptive Systems

Some important measures will be discussed in the following sections; *convergence rate*, *minimum mean square error*, *computational complexity*, *stability*, *and filter length* [2].

1.3.1. Convergence Rate

The *Convergence rate* determines the rate at which the filter converges to its resultant state. Usually a faster convergence rate is a desired characteristic of an adaptive system. Convergence rate is not independent of all the other performance characteristics. There is usually a tradeoff, with convergence rate and other performance criteria.

1.3.2. Mean Square Error

The *Mean Square error* (MSE) is a metric indicating how much a system can adapt to a given solution. A small MSE is an indication that the adaptive system has accurately modeled, predicted, adapted and/or converged to a solution for the system. There are a number of factors which will help to determine the MSE including, but not limited to; *quantization noise*, *order of the adaptive system*, *measurement noise*, and *error of the gradient due to the finite step size* [2].

1.3.3. Computational Complexity

Computational complexity is particularly important in real time adaptive filter applications. When a real time system is being implemented, there are hardware limitations that may affect the performance of the system. A highly complex algorithm will require much greater hardware resources than a simplistic algorithm.

1.3.4. Stability

Stability is probably the most important performance measure for the adaptive system. By the nature of the adaptive system, there are very few completely asymptotically

stable systems that can be realized. In most cases, the systems that are implemented are marginally stable, with the stability determined by the initial conditions, transfer function of the system and the step size of the input [2].

1.3.5. Filter Length

The *Filter Length* of the adaptive system is inherently tied to many of the other performance measures. The length of the filter specifies how accurately a given system can be modeled by the adaptive filter. In addition, the filter length affects the convergence rate, by increasing or decreasing computation time, it can affect the stability of the system, at certain step sizes, and it affects the MSE. If the filter length of the system is increased, the number of computations will increase, decreasing the maximum convergence rate [2].

CHAPTER 2

LEAST MEAN SQUARE (LMS) FILTERING ALGORITHMS

Before starting the discussion of the LMS algorithms family, an old optimization technique known as the *steepest descent method* will be presented.

2.1. Steepest-Descent Method

This method is recursive in the sense that starting form some initial (arbitrary) value for the tap-weight vector, it improves with the increased number of iterations. The important point to note is that the method of steepest descent is descriptive of *multiparameter closed-loop deterministic control system* that finds the minimum point of the ensemble-averaged error-performance surface without knowledge of the surface itself [1].

Consider a transversal filter with *tap inputs* $u(n), u(n-1), \ldots, u(n-N+1)$ and a corresponding set of *tap weights* $w_0(n), w_1(n), \ldots, w_{N-1}(n)$. The tap inputs represent samples drawn from a wide-sense stationary stochastic process of zero mean and correlation matrix $\mathbf{R} = \mathbf{u}(n)\mathbf{u}^H(n)$. Also the filter is supplied with a *desired response* $d(n)$ that provides a frame of reference for the optimum filtering action; this is illustrated clearly in Fig. 2.1.

The vector of tap inputs at time n is denoted by $\mathbf{u}(n)$, and the corresponding *estimate* of the desired response at the filter output is denoted by $\hat{d}(n|U_n)$, where U_n is the space

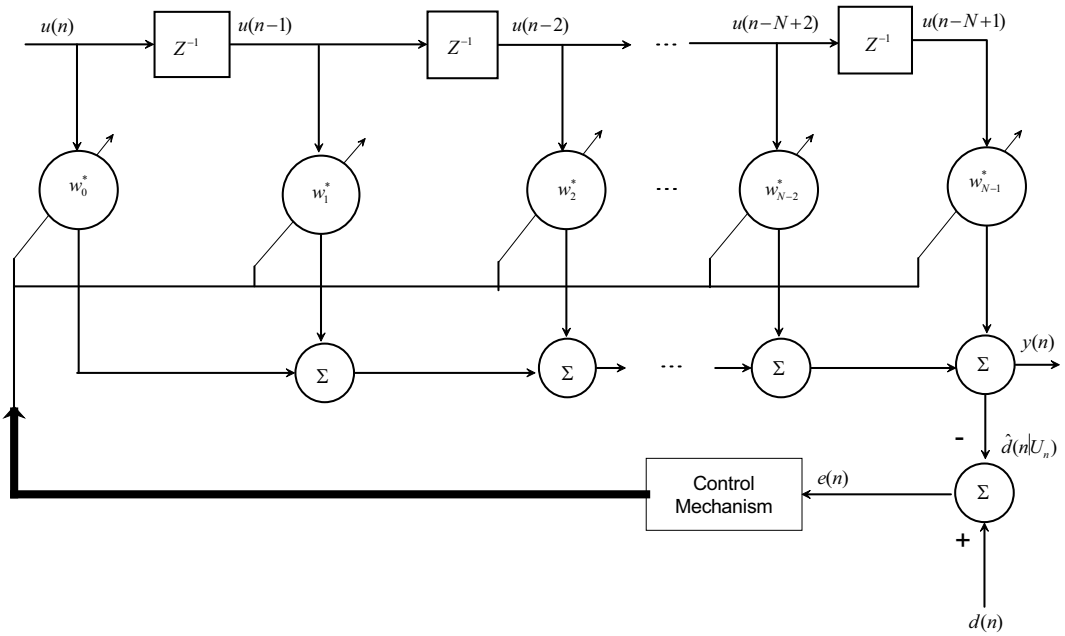

Figure 2.1. Structure of adaptive transversal filter.

spanned by the tap inputs $u(n), u(n-1), \ldots, u(n-N+1)$. By comparing this with the actual desired response $d(n)$, we produce an *estimation error* denoted by $e(n)$.

$$e(n) = d(n) - \hat{d}(n|U_n) = d(n) - \mathbf{w}^H(n)\mathbf{u}(n), \tag{2.1}$$

where the superscript "*H*" is the Hermitian Transpose, the term $\mathbf{w}^H(n)\mathbf{u}(n)$ is the inner product of the tap-weight vector $\mathbf{w}(n)$ and the tap input vector $\mathbf{u}(n)$. The tap-weight vector, the tap-input vector and the cost function are denoted by:

$$\mathbf{w}(n) = [w_0(n)\ w_1(n)\ \ldots\ w_{N-1}(n)]^T, \tag{2.2}$$

$$\mathbf{u}(n) = [u(n)\ u(n-1)\ \ldots\ u(n-N+1)]^T, \tag{2.3}$$

$$J(n) = E\{|e(n)|^2\}, \tag{2.4}$$

If the tap-input vector $\mathbf{u}(n)$ and the desired $d(n)$ are jointly stationary, then the mean-squared error or cost function $J(n)$ at time n could be written as:

$$J(n) = \sigma_d^2 - \mathbf{w}^H(n)\mathbf{p} - \mathbf{p}^H\mathbf{w}(n) + \mathbf{w}^H(n)\mathbf{R}\mathbf{w}(n), \tag{2.5}$$

where;

σ^2= variance of the desired response $d(n)$.

p = cross-correlation vector between the tap-input vector $\mathbf{u}(n)$ and the desired response $d(n)$.

R = correlation matrix of the tap-input vector $\mathbf{u}(n)$.

Equation (2.5) defines the mean-squared error that would result if the tap-weight vector in the transversal filter was fixed at the value $\mathbf{w}(n)$. Since $\mathbf{w}(n)$ varies with time n, it is only natural that the mean-squared error varies with time n in a corresponding fashion, hence, the use of $J(n)$ for the mean-squared error in that equation. The variation of the mean-squared error $J(n)$ with time n signifies the fact that the estimation error process $e(n)$ is nonstationary [1].

We visualize the dependence of the mean-squared error $J(n)$ on the elements of the

tap-weight vector $\mathbf{w}(n)$ as a bowl-shaped surface with a unique minimum. This is called as the error-performance surface of the adaptive filter. This occurs when the tap-weight vector takes on the optimum value $\mathbf{w}_0$ [1]. We define:

$$\mathbf{R}\mathbf{w}_0 = \mathbf{p}, \tag{2.6}$$

and the minimum mean-squared error is:

$$J_{min} = \sigma_d^2 - \mathbf{p}^H \mathbf{w}_0, \tag{2.7}$$

The *Steepest-Descent Algorithm* [1] [2] is relatively straightforward; nevertheless, it presents serious computational difficulties, especially when the filter contains a large number of tap-weights and when the input data rate is high.

This implies that we can use the Steepest-Descent method to find the minimum value of the mean-squared error J_{min} as follows:

1. Start with an initial value $\mathbf{w}(0)$ for the tap-weight vector, which is chosen arbitrarily. The value $\mathbf{w}(0)$ provides an initial guess as to where the minimum point of the error-performance surface may be located. Usually, $\mathbf{w}(0)$ is set equal to the null vector.

2. Using this assumption, we compute the *gradient vector*, the real and imaginary parts of which are defined as the derivative of the mean-squared error $J(n)$, evaluated with respect to the real and imaginary parts of the tap-weight vector $\mathbf{w}(n)$ at time n.

3. Compute the next guess of the tap-weight vector by changing the present guess in

a direction opposite to that of the gradient vector.

4. Go back to step 2 and repeat the process.

Let $\nabla(J(n))$ denote the value of the *gradient vector* at time n. Let $\mathbf{w}(n)$ denote the value of the tap-weight vector at time $n+1$, computed by using the simple recursive relation:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{1}{2}\mu[-\nabla(J(n))], \tag{2.8}$$

Where μ is a positive real-valued constant, and

$$\nabla(J(n)) = \begin{pmatrix} \frac{\partial J(n)}{\partial a_0(n)} & j\frac{\partial J(n)}{\partial b_0(n)} \\ \frac{\partial J(n)}{\partial a_1(n)} & j\frac{\partial J(n)}{\partial b_1(n)} \\ \frac{\partial J(n)}{\partial a_2(n)} & j\frac{\partial J(n)}{\partial b_2(n)} \\ \vdots & \vdots \\ \frac{\partial J(n)}{\partial a_{N-1}(n)} & j\frac{\partial J(n)}{\partial b_{N-1}(n)} \end{pmatrix} = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}(n), \tag{2.9}$$

Where $\frac{\partial J(n)}{\partial a_k(n)}$ and $\frac{\partial J(n)}{\partial b_k(n)}$ are partial derivatives of the cost function $J(n)$ with respect to real part $a_k(n)$ and the imaginary part $b_k(n)$ of the k^{th} tap weight $w_k(n)$, respectively. For the application of the steepest-descent algorithm, we assume that in (2.9), the correlation matrix $\mathbf{R}$ and the cross-correlation vector $\mathbf{p}$ are known, so we may compute the gradient vector$\nabla(n)$for a given value of the tap-weight vector $\mathbf{w}(n)$. Substituting (2.9) in (2.8) we will get the updated value of the tap-weight vector by using the simple recursive relation:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(n)] \quad n = 1, 2, 3, \ldots \tag{2.10}$$

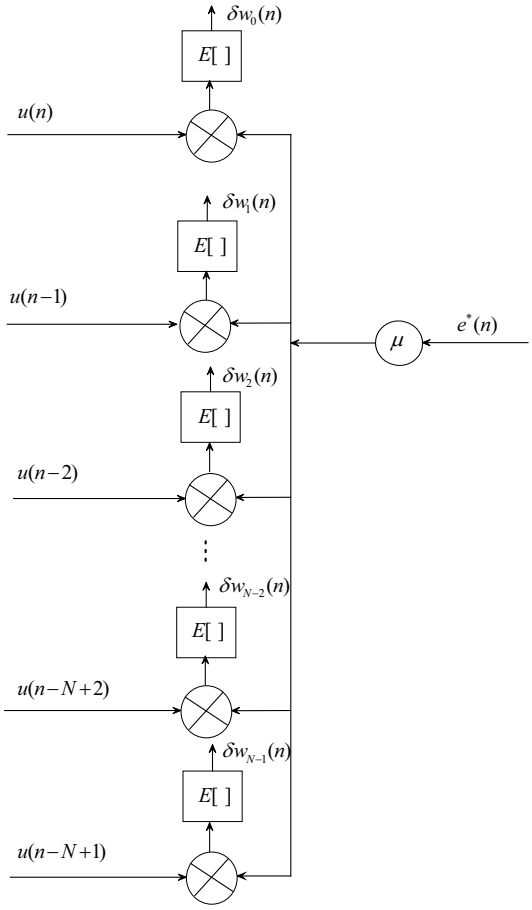

Figure 2.2. Bank of cross-correlators for computing the corrections of the elements of the tap-weight vector at $n + 1$, [1].

We observed that the parameter μ controls the size of the incremental correction applied to the tap-weight vector as we proceed from one iteration cycle to the other. We call μ the *step-size parameter* or *weighting constant*. Equation (2.10) describes the mathematical formulation of the steepest-descent algorithm.

According to (2.10), the correction $\delta\mathbf{w}(n) = \mathbf{w}(n+1) - \mathbf{w}(n)$ applied to the tap-weight vector at time $n + 1$ is equal to $\mu[\mathbf{p} - \mathbf{R}\mathbf{w}(n)]$. This correction may be expressed as μ times the expectation of the inner product of the input vector $\mathbf{u}(n)$ and the estimation error $e(n)$. This suggests using a bank of cross-correlators to compute the correction

$\delta\mathbf{w}(n)$ applied to the tap-weight vector $\mathbf{w}(n)$ as in Fig. 2.2. Another point is that we may view the steepest-descent algorithm of (2.10) as a feedback model.

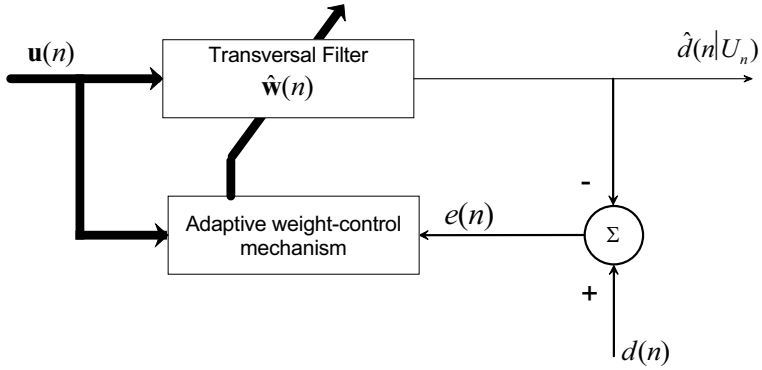

Figure 2.3. Block Diagram of Adaptive Transversal Filter [1].

The operation of the *least-mean-square* (LMS) *algorithm* is descriptive of *a feedback control system*. Basically, it consists of a combination of two basic processes:

1. An adaptive process, which involves the automatic adjustment of a set of tap weights.

2. A filtering process, which involves forming the inner product of a set of tap weights emerging from the adaptive process to produce an estimate of a desired response, and generating an estimation error by comparing this estimate with the actual value of the desired response; which in turn (the estimation error) is used to actuate the adaptive process, thereby closing the feedback loop.

We are going to identify the two basic components in the structural constitution of the LMS algorithm as in Fig. 2.3, which has a transversal filter with LMS algorithm (for filtering process), and a mechanism for adaptive control process on the tap weights of the transversal filter.

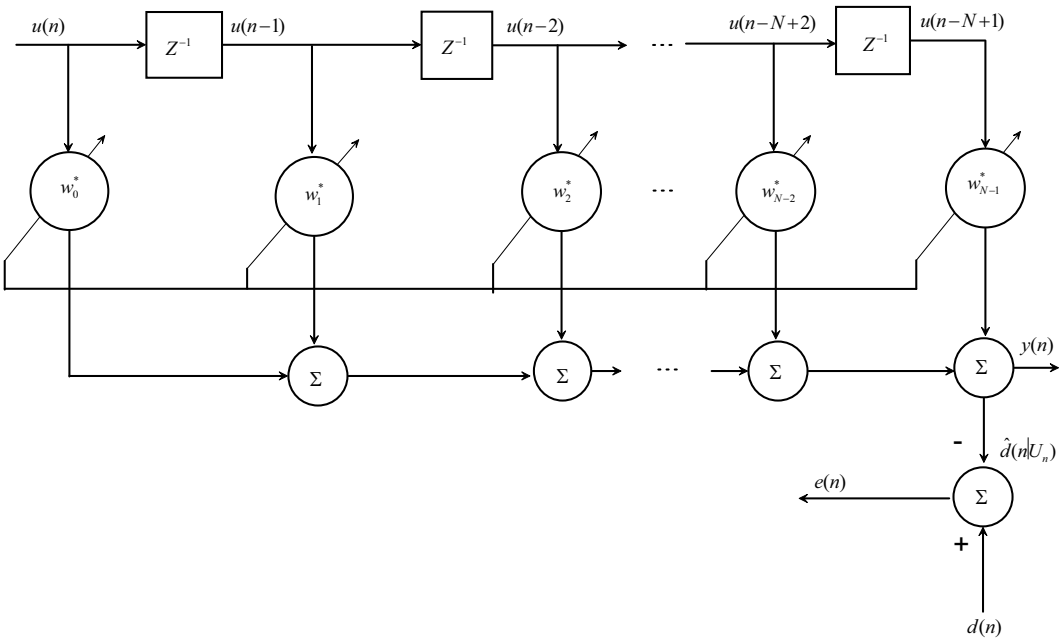

Figure 2.4. Detailed Structure of the Transversal Filter Component, [1].

During the filtering process the *desired response* $d(n)$ is supplied for processing, alongside the tap-input vector $\mathbf{u}(n)$. With this input the transversal filter produces an output $\hat{d}(n|U_n)$ used as an *estimate* of the desired response $d(n)$. Also we may set up an *estimation error* $e(n)$ as the difference between the desired response and the filter output, as in Fig. 2.4. Both $e(n)$ and $\mathbf{u}(n)$ are applied to the control mechanism, and the feedback loop around the tap weights is thereby closed.

Figure 2.5 presents details of the *adaptive weight-control mechanism*. Specifically, a scaled version of the *inner product of the estimation error* $e(n)$ and tap-input $u(n-k)$ is computed for $k = 0, 1, \ldots, N-1$. The obtained result defines the *correction* $\delta\hat{w}_k(n)$ applied to the tap weight $\hat{w}_k(n)$ at time $n+1$. The scaling factor μ is called the *adaptation constant or step-size parameter* (as mentioned previously).

Comparing Fig. 2.5 and Fig. 2.2 we see that the LMS algorithm uses the inner

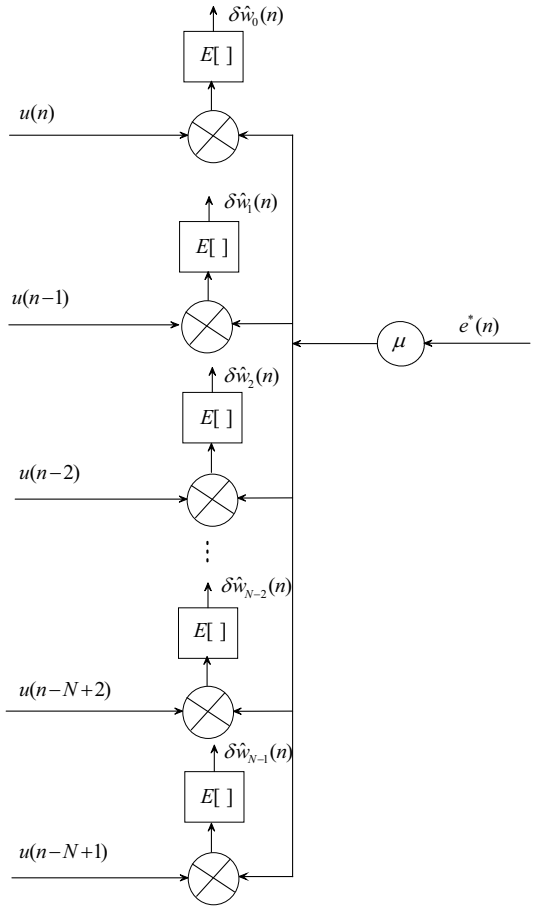

Figure 2.5. Detailed Structure of the Adaptive Weight-Control Mechanism.

product $u(n-k)e^*(k)$ as an estimator of element k in the gradient vector $\nabla(J(n))$ that characterizes the steepest-descent method. The recursive computation of each tap weight in the LMS algorithm suffers from *gradient noise*.

The tap-weight vector $\hat{\mathbf{w}}(n)$ computed by the LMS algorithm executes a *random motion* around the minimum point of the error-performance surface. This random motion gives rise to two forms of convergence behavior of the LMS algorithm.

1. Convergence in the mean.

2. Convergence in the mean square.

2.2. Least-Mean-Square Adaptation Algorithm

If it were possible to make exact measurement of the gradient vector $\nabla(J(n))$ at each iteration, if the step-size parameter μ is suitably chosen, then the tap-weight vector computed by using the steepest-descent algorithm would indeed converge.

Exact measurements of the gradient vector are, in reality, not possible since this would require prior knowledge of both the correlation matrix **R** of the tap input and the cross-correlation vector **p** between the tap inputs and the desired response.

Consequently, the gradient vector must be estimated from the available data. That means the tap-weight vector according to an algorithm *adapts to the incoming data* (Least-Mean-Square (LMS) Algorithm). A significant feature of the LMS algorithm is its simplicity; it does not require measurements of the pertinent correlation functions, and it does not require matrix inversion [1].

To develop an estimate of the gradient vector $\nabla(J(n))$, we substituted estimates of the correlation matrix **R** and the cross-correlation vector **p** in (2.9).

$$\nabla(J(n)) = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}(n), \tag{2.11}$$

the simplest choice of estimator for **R** and **p** is to use instantaneous estimates that are based on sample values of the tap input vector and desired response.

$$\hat{\mathbf{R}}(n) = \mathbf{u}(n)\mathbf{u}^H(n), \tag{2.12}$$

$$\hat{\mathbf{p}} = \mathbf{u}(n)d^*(n), \tag{2.13}$$

Corresponding, the instantaneous estimate of the gradient vector is:

$$\hat{\nabla}(J(n)) = -2\mathbf{u}(n)d^*(n) + 2\mathbf{u}(n)\mathbf{u}^H(n)\hat{\mathbf{w}}(n), \tag{2.14}$$

Generally, this estimate is biased because the tap-weight estimate vector $\hat{\mathbf{w}}(n)$ is a random vector that depends on the tap-input vector $\mathbf{u}(n)$. Noting that the estimate $\hat{\nabla}(J(n))$ may also be viewed as the gradient operator ∇ applied to instantaneous squared error $|e(n)|^2$.

Substituting the estimate of (2.14) for the gradient vector $\nabla(J(n))$ in the steepest descent algorithm as described in (2.8), we get a new recursive relation for updating the tap-weight vector:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu\mathbf{u}(n)[d^*(n) - \mathbf{u}^H(n)\hat{\mathbf{w}}(n)] \tag{2.15}$$

Here we have used the "cap" over the symbol of the tap-weight vector to distinguish it from the value obtained by the steepest-descent algorithm. We may write the result in three basic relations:

1. Filter Output.

$$y(n) = \hat{\mathbf{w}}^H(n)\mathbf{u}(n), \tag{2.16}$$

2. Estimation Error.

$$e(n) = d(n) - y(n), \tag{2.17}$$

3. Tap-Weight Adaptation.

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu e^*(n)\mathbf{u}(n), \tag{2.18}$$

Equations (2.16), (2.17) define the estimation error $e(n)$, the computation of which is based on the *current estimate* of the tap-weight vector, $\hat{\mathbf{w}}(n)$. Note that the second term, $\mu e^*(n)\mathbf{u}(n)$ on the right side of (2.18) represents the *correction* that is applied to the current estimate of the tap-weight vector, $\hat{\mathbf{w}}(n)$. The iterative procedure is started with the initial guess $\hat{\mathbf{w}}(0)$. A convenient choice for this initial guess is the null vector, we may thus set $\hat{\mathbf{w}}(0) = 0$.

The algorithm described by (2.16) to (2.18) is the *complex* form of the adaptive LMS algorithm [1]. In the LMS algorithm the allowed set of directions along which we "step" from one iteration cycle to the next is quite random and cannot therefore be thought of as being gradient directions.

2.3. Normalized Least Mean Squares (NLMS) Algorithm

In "Normalized" LMS, the gradient step factor μ is normalized by the energy of the data vector. Normalization has several interpretations [3]:

1. Corresponds to the 2^{nd}-order convergence bound.
2. Makes the algorithm independent of signal scaling.
3. Adjusts $\mathbf{w}(n+1)$ to give zero error with current input.

So the NLMS tap weight adaptation vector equation has the form:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \frac{\mu e^*(n)\mathbf{u}(n)}{\epsilon + \mathbf{u}^H(n)\mathbf{u}(n)} \tag{2.19}$$

Where ϵ is a very small value to avoid dividing by zero.

NLMS usually converges much more quickly than LMS at very little extra cost; NLMS is very commonly used in some applications such as acoustic echo cancellation

problems [4].

2.4. Leaky LMS Adaptive Algorithm

The Leaky adaptive filter can be implemented either directly or by adding random white noise to the input signal of the LMS adaptive filter [5].

The problem of bias accumulation in the coefficients of the adaptive filter is generally solved by using a "leakage" technique. In this way, the accumulation of bias in the coefficients is limited. The basic idea of this technique is to add a zero-mean white noise to the input signal of the adaptive filter. It is a good solution since it will decrease the input eigenvalue spread. The new eigenvalue spread is smaller than the original eigenvalue spread; this means that the leaky algorithm's worst-case transient performance will be better than that of the standard LMS algorithm.

The problem is solved by introducing a leakage factor $(1 - \mu\gamma)$ into the coefficient adjustment algorithm of the adaptive algorithm. If this technique is used on the LMS algorithm then the leaky LMS algorithm will be obtained [6] [7] and its coefficient adjustment algorithm is given by

$$\hat{\mathbf{w}}(n+1) = (1 - \mu\gamma)\hat{\mathbf{w}}(n) + \mu e^*(n)\mathbf{u}(n), \tag{2.20}$$

where γ is a real positive parameter which satisfies the condition $0 \leq \gamma \leq \frac{2}{\mu}$ [28].

2.5. Modified Leaky LMS Adaptive Algorithm

The Modified Leaky LMS is the same as the Leaky LMS algorithm, except that the leakage factor is introduced here as a variable matrix based on the knowledge of the noise

variance. This implies that this algorithm is almost blind, the only parameter it requires is the noise variance as mentioned before. We may write the result in three basic relations [8]:

1. Filter Output.

$$y(n) = \hat{\mathbf{w}}^H(n)\mathbf{u}(n), \tag{2.21}$$

2. Estimation Error.

$$e(n) = d(n) - y(n), \tag{2.22}$$

3. Tap-Weight Adaptation.

$$\hat{\mathbf{w}}(n+1) = [\mathbf{I} - \mu\hat{\mathbf{A}}(n)]\hat{\mathbf{w}}(n) + \mu e^*(n)\mathbf{u}(n), \tag{2.23}$$

where $\hat{\mathbf{A}}(n)$ in (2.23) is an $N \times N$ matrix consisting of all zeros except for the $(1,1)^{th}$ element which is given by $\alpha(n) = \frac{\sigma^2}{w_0(n)}$ where $w_0(n)$ is the first element of $\mathbf{w}(n)$.

2.6. Frequency Response Shaped (FRS) LMS Adaptive Algorithm

The FRS-LMS algorithm was recently introduced [12] and it is based on shaping the frequency response of the transversal FIR filter. This shaping is performed on-line by the inclusion of an additional term similar to the leakage factor in Leaky LMS. This term's introduction is aimed at suppressing specific bands of the frequency response where the noise is more prominent. The FRS-LMS algorithm provides the leakage factor in a matrix form where its elements are determined based on the noise power spectrum [12].

2.6.1. Derivation of the FRS-LMS Algorithm

Consider the output of the adaptive transversal filter as:

$$y(n) = \mathbf{w}^H(n)\mathbf{u}(n), \tag{2.24}$$

where **w(**n) is the adaptive weight vector,

$$\mathbf{w}(n) = [w_0(n) \quad w_1(n) \quad \dots \quad w_{N-1}(n)]^T, \tag{2.25}$$

where N is the filter length, and $\mathbf{u}(n)$ is the discrete time input signal given by

$$\mathbf{u}(n) = [u(n) \quad u(n-1) \quad \dots \quad u(N-n+1)]^T. \tag{2.26}$$

The update equation of the weight vector in the standard LMS algorithm is given as [1]:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e^*(n)\mathbf{u}(n), \tag{2.27}$$

where $e(n)$ denotes the error, * denotes complex conjugation and μ is the step size. $e(n)$ is given by:

$$e(n) = d(n) - y(n), \tag{2.28}$$

where $d(n)$ is the desired response of the filter, $y(n)$ is the output of the filter. In order to minimize the error, we start by establishing the cost function as:

$$J(n) = e^2(n), \tag{2.29}$$

The cost function in Leaky LMS [1] is modified by adding a filter norm term to (2.29),

$$J_{leaky}(n) = e^2(n) + \gamma \mathbf{w}^H(n)\mathbf{w}(n), \tag{2.30}$$

where γ stands for the leakage factor. In FRS-LMS, the leakage factor γ becomes a matrix [12]. Consider the total weighted noise power of the filter output at time step n:

$$G(n) = \int_0^{\pi} w_0(\omega)|W_n(\omega)|^2 d\omega, \tag{2.31}$$

where

$$W_n(\omega) = \sum_{k=0}^{N-1} w_k(n)e^{-j\omega k}, \tag{2.32}$$

is the frequency response of the filter at time n, $w_0(\omega)$ is a properly chosen weight function from Fig. 2.6. If $w_0(\omega)$ becomes equal to the noise power spectral density (psd), then (2.31) becomes the total correlated noise power at the output of the filter. It is also assumed that $G(n)$ is constrained by an upper limit, P_0.

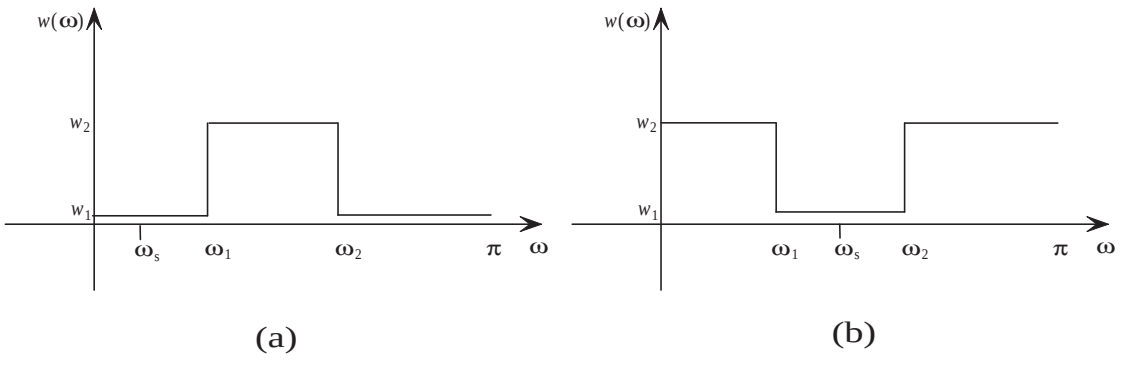

Figure 2.6. Weight Functions.

Combining (2.31), (2.32) and by using the method of Lagrange multipliers, the cost

function to be minimized is obtained as:

$$J_{FRS}(n) = e^2(n) + \zeta(G(n) - P_0), \tag{2.33}$$

and

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{1}{2}\mu\nabla J_{FRS}(n), \tag{2.34}$$

where $\nabla J_{FRS}(n)$ is the gradient of the cost function with respect to the vector $\mathbf{w}(n)$ and is given by:

$$\nabla J_{FRS}(n) = -2e^*(n)\mathbf{u}(n) + 2\zeta\mathbf{F}_0\mathbf{w}(n), \tag{2.35}$$

where $G(n)$ is a matrix given by:

$$G(n) = \mathbf{w}^H(n)\mathbf{F}_0\mathbf{w}(n), \tag{2.36}$$

In (2.36), $\mathbf{F}_0$ is a matrix with elements determined as:

$$f_0(m,k) = \int_0^\pi w_0(\omega)cos[(m-k)\omega]d\omega, \tag{2.37}$$

Finally, using (2.34) and (2.35), the filter coefficient vector update equation becomes:

$$\mathbf{w}(n+1) = [\mathbf{I} - \mu\mathbf{F}]\mathbf{w}(n) + \mu e^*(n)\mathbf{u}(n), \tag{2.38}$$

where $\mathbf{F} = \zeta\mathbf{F}_0$.

For the weight functions in Fig. 2.6, the elements of the matrix $\mathbf{F}$ are given by:

$$f(n,m)=\begin{cases}\frac{w_2-w_1}{n-m}[\sin((n-m)\omega_2)-\sin((n-m)\omega_1)], & n\neq m\\ (w_2-w_1)(\omega_2-\omega_1)+w_1\pi, & n=m\end{cases} \tag{2.39}$$

2.6.2. Convergence Analysis

The weight error vector is defined as:

$$\epsilon(n) = \mathbf{w}_o - \mathbf{w}(n), \tag{2.40}$$

In (2.40) $\mathbf{w}_o$ is the solution of the Wiener-Hopf equation ($(\mathbf{R}_u u + \mathbf{F})\mathbf{w}_o = \mathbf{p}$), where $\mathbf{R}_u u = E\{\mathbf{u}(n)\mathbf{u}^T(n)\}$ and $\mathbf{p} = E\{d(n)\mathbf{u}(n)\} = [p_0\ p_1\ \ldots\ p_{N-1}]^T$. Substituting (2.40) and (2.28) in (2.38) we obtain:

$$\epsilon(n+1) = [\mathbf{I} - \mu(\mathbf{F} + \mathbf{u}(n)\mathbf{u}^T(n))]\epsilon(n) + \mathbf{f}_0(n), \tag{2.41}$$

where the force vector is:

$$\mathbf{f}_0(n) = \mu(\mathbf{F}\mathbf{w}_o - \widetilde{e}_o(n)\mathbf{u}(n)), \tag{2.42}$$

and the estimation error produced by the optimum Wiener filter is:

$$\tilde{e}_o(n) = d(n) - \mathbf{u}^T(n)\mathbf{w}_o(n), \tag{2.43}$$

It should be noted that the force vector can also be written as:

$$\mathbf{f}_0(n) = \mu[\mathbf{F} + \mathbf{u}(n)\mathbf{u}^T(n)]\mathbf{w}_o(n) - \mu\mathbf{u}(n)d(n), \tag{2.44}$$

so that $E\{\mathbf{f}_0(n)\} = \mathbf{0}$. Using the direct averaging method [1] with a small step-size assumption, (2.41) becomes,

$$\epsilon_0(n+1) = [\mathbf{I} - \mu\mathbf{R}]\epsilon_0(n) + \mathbf{f}_0(n), \tag{2.45}$$

where $\mathbf{R} = \mathbf{F} + \mathbf{R}_{uu}$ and $\epsilon_0(n)$ is the zero-order solution [1] of (2.41). Taking the expected value of (2.45), it is clear that $\lim_{k\to\infty} E\{\epsilon_0(n)\} = \mathbf{0}$ provided that the eigenvalues of $\mathbf{R}$ satisfy

$$\frac{2}{\lambda_{max}(\mathbf{R})}, \tag{2.46}$$

To establish the weight-error convergence in the mean-square sense, the difference equation in (2.45) is transformed by the unitary transformation,

$$\mathbf{Q}^T\mathbf{R}\mathbf{Q} = \Lambda, \tag{2.47}$$

where $\mathbf{Q}$ is the matrix of eigenvectors of $\mathbf{R}$, and $\Lambda = diag\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$ is the matrix of eigenvalues of $\mathbf{R}$. The transformed error and force vectors are given by $\mathbf{v}(n) = \mathbf{Q}^T\epsilon_0(n)$ and $\Phi(n) = \mathbf{Q}^T\mathbf{f}_0(n)$, respectively. Application of this transformation yields:

$$\mathbf{v}(n+1) = [\mathbf{I} - \mu\Lambda]\mathbf{v}(n) + \Phi(n), \tag{2.48}$$

The steady state solution of the expected value of the square of the k^{th} scalar equation of (2.48) is:

$$\lim_{k\to\infty} E\{v_k^2(n)\} = \frac{E\{\phi_k^2(n)\}}{\mu\lambda_k(2-\mu\lambda_k)}, \tag{2.49}$$

Noting that $E\{\phi_k^2(n)\}$ can be written as [12]:

$$E\{\phi_k^2(n)\} = \mu^2 J_{min}[\lambda_k - \frac{1}{2}\sum_{m=1}^{M}\hat{w}_m(\mathbf{e}_m^T\mathbf{q}_k\mathbf{q}_k^T\mathbf{e}_m^* + \mathbf{e}_m^H\mathbf{q}_k\mathbf{q}_k^T\mathbf{e}_m)], \tag{2.50}$$

where $\hat{w}_m = \frac{\pi w(\omega_m)}{M}$ and ω_m are evenly spaced frequencies in $[0, \pi]$ [12], $\mathbf{q}_k$ is the eigenvector corresponding to λ_k. Substituting (2.50) in (2.49) we obtain the steady state variance of the k^{th} component of the error vector as:

$$E\{v_k^2(\infty)\} = \frac{\mu^2 J_{min}}{2-\mu\lambda_k}(1 - \frac{1}{\lambda_k}\sum_{m=1}^{M}\hat{w}_m|\mathbf{q}_k^T\mathbf{e}_m|^2, \tag{2.51}$$

Using (2.51) it is possible to obtain the steady state variance of the error vector as:

$$E\{\|\epsilon_0(\infty)\|^2\} = \sum_{k=1}^{N} E\{v_k^2(\infty)\} = \sum_{k=1}^{N}\frac{\mu J_{min}}{2-\mu\lambda_k} - \mu J_{min}\sum_{k=1}^{N}\frac{\sum_{m=1}^{M}\hat{w}_m|\mathbf{q}_k^T\mathbf{e}_m|^2}{\lambda_k(2-\mu\lambda_k)}, \tag{2.52}$$

from (2.52) the upper-bound of the variance can be obtained as:

$$E\{\|\epsilon_0(\infty)\|^2\} \leq \mu J_{min}(\sum_{k=1}^{N}\frac{1}{2-\mu\lambda_k} - \frac{N}{c_1}\sum_{m=1}^{M}\hat{w}_m), \tag{2.53}$$

where

$$\sum_{k=1}^{N}\mathbf{q}_k\mathbf{q}_k^T = \mathbf{I}, \tag{2.54}$$

and

$$c_1 = \sup_{1\leq k\leq N}\{\lambda_k(2-\mu\lambda_k)\} = \begin{cases} \lambda_{max}(2-\mu\lambda_{max}), & when\ \mu < 1/\lambda_{max}, \\ 1/\mu, & otherwise. \end{cases} \tag{2.55}$$

In order to be able to compare the error vector variance to that of the standard LMS al-

gorithm, it is necessary to express the eigenvalues of the modified autocorrelation matrix $\mathbf{R}$ in terms of those of the original matrix $\mathbf{R}_u u$. The eigenvalues of a symmetric autocorrelation matrix for large N are given by [12]:

$$\lambda_k = \tilde{R}(\omega_k), \qquad \omega_k = \frac{(k-1)\pi}{N};\ k = 1, \ldots, N, \tag{2.56}$$

where $\tilde{R}(\omega)$ is the power spectral density of the signal. Therefore, the eigenvalues of $\mathbf{R} = \mathbf{R}_u u + \mathbf{F}$ can be approximately written as:

$$\lambda_k = \lambda_{ok} + S_g(\omega_k),\ k = 1, \ldots, N, \tag{2.57}$$

where λ_{ok} are the eigenvalues of $\mathbf{R}_u u$ and $S_g(\omega)$ is the Fourier transform of the sequence defined as:

$$g_q = \begin{cases} \frac{w_2 - w_1}{q}[sin(q\omega_2) - sin(q\omega_1)], & -(N-1) \leq q \leq (N-1);\ q \neq 0, \\ (w_2 - w_1)(\omega_2 - \omega_1) + w_1\pi, & q = 0. \end{cases} \tag{2.58}$$

It should be noted that the elements of the matrix $\mathbf{F}$ can be generated from the symmetric sequence g_q. Using (2.57) the following approximation can be obtained for the values of $\mu \ll 2/\lambda_k$.

$$\frac{1}{2 - \mu\lambda_k} \simeq \frac{1}{2 - \mu\lambda_{ok}}(1 + \frac{\mu S_g(\omega_k)}{2 - \mu\lambda_{ok}}), \tag{2.59}$$

Substituting (2.59) in (2.53) gives:

$$E\{\|\epsilon_0(\infty)\|^2\} \leq E\{\|\epsilon_0^{(o)}(\infty)\|\} - \mu J_{min}(\frac{N}{c_1}\sum_{m=1}^{M}\hat{w}_m - \mu\sum_{k=1}^{N}\frac{S_g(\omega_k)}{(2-\mu\lambda_{ok})^2}), \quad (2.60)$$

In (2.60) the first term in the right-hand side of the inequality is the variance of the original LMS algorithm. The summation in the second term can be in (2.60) upper-bound as:

$$\sum_{k=1}^{N}\frac{S_g(\omega_k)}{(2-\mu\lambda_{ok})^2} \leq \frac{1}{c_2}\sum_{k=1}^{N}S_g(\omega_k), \quad (2.61)$$

where

$$c_2 = \inf_{1\leq k\leq N}(2-\mu\lambda_{ok})^2 = \begin{cases} (2-\mu\lambda_{o,max})^2, & when\ \lambda_{o,max} < 2/\mu, \\ 0, & otherwise. \end{cases} \quad (2.62)$$

Using the approximation [12]:

$$N\sum_{m=1}^{M}\hat{w}_m \simeq \sum_{k=1}^{N}S_g(\omega_k), \quad (2.63)$$

Equation (2.60) can be simplified to obtain:

$$E\{\|\epsilon_0(\infty)\|^2\} \leq E\{\|\epsilon_0^{(o)}(\infty)\|\} - \mu J_{min}(\frac{1}{c_1} - \frac{\mu}{c_2}) \sum_{k=1}^{N} S_g(\omega_k), \qquad (2.64)$$

Using (2.55) and (2.62) it can be easily shown that the second term in the right-hand side of (2.64) is always positive if μ is chosen less than $1/\lambda_{max}$ [12]. Therefore, the error vector variance of the proposed algorithm can be made less than that of the original LMS.

2.6.3. Implementation Issues

2.6.3.1. Fast Computation. The extra term $\mathbf{Fw}(n)$ in (2.35) requires N^2 multiplications and $N(N-1)$ additions, which makes the algorithm inefficient in its present form. A fast calculation of this product is possible if it is observed that the elements of the matrix $\mathbf{F}$ can be generated from the symmetric sequence g_q given in (2.58) [12].

The n^{th} element of the vector $\mathbf{w}_f(n)$ can be written as:

$$w_{f,k}(n) = \sum_{m=1}^{N} f_{k,m} w_{m-1}(n), \quad k = 1, \ldots, N. \qquad (2.65)$$

Rewriting (2.65) in terms of the sequence given in (2.58) gives:

$$w_{f,k}(n) = \sum_{m=1}^{N} g_{n-m-1} w_m(n), \quad k = 1, \ldots, N. \qquad (2.66)$$

which is in the form of a convolution. Now, taking $(2N-1)$-point DFT of both sides at time n,

$$W_{fe}(l) = G(l)W_e(l), \quad l = 1, \ldots, 2N-1. \tag{2.67}$$

where $W_{fe}(l)$ is the DFT of the zero-padded sequence $\{w_{f_{e,k}}(n);\ k = 1, \ldots, 2N-1.\}$:

$$w_{f_{e,k}}(n) = \begin{cases} w_{f,k}(n), & k = 1, \ldots, N, \\ 0, & k = N+1, \ldots, 2N-1. \end{cases} \tag{2.68}$$

and $W_e(l)$ is the DFT of $\mathbf{w}_e(n) = [\underline{0}\ \ \mathbf{w}(n)]^T$, where $\underline{0}$ is an $(N-1)$-dimensional zero vector. The sequence $\{w_{f,k}(n);\ k = 1, \ldots, N\}$ can be recovered from the inverse DFT of $W_{fe}(l)$.

2.6.3.2. Weight Function. In real life problems, the frequency of the signal changes most of the time, due to this, the weight function must also be consequently updated. A signal adaptive weight function can be generated by first estimating the signal frequency from its periodogram. Then, the edge frequencies of the weight function can be calculated as:

$$\omega_1 = \omega_s - \frac{\omega_b}{2}; \qquad \omega_2 = \omega_s + \frac{\omega_b}{2}, \tag{2.69}$$

where $\omega_b = \omega_2 - \omega_1$ is the window's width, ω_s is the frequency of the signal which is estimated from the periodogram of the signal. This procedure may be repeated at regular intervals (e.g., 30 or 50 samples) and the sequence $g_q; \quad q = -(N-1), \ldots, (N-1)$ is updated in order to be compatible with the sinusoid to be recovered. choosing the weight

function depends on the criteria; the frequency of the signal is in the noise band or outside the noise band, the weights w_1 and w_2 are selected proportional to the noise power [12], and the width of the window is to be selected large enough to guarantee not suppressing any part of the signal and small enough to guarantee suppressing most of the noise [12]. Therefore, the smallest width that would not cause the gain at the signal frequency to decrease should be chosen.

This algorithm will be used in adverse channel environments to reduce the effects of impulsive noise, correlated noise and fading.

CHAPTER 3

MOBILE RADIO CHANNEL

Communication Systems have become very important in our lives. Each communication system mainly consists of three parts; *Transmitter*, *Channel* and *Receiver* as shown in Fig. 3.1. The medium connecting the transmitter and receiver is called the *channel*.

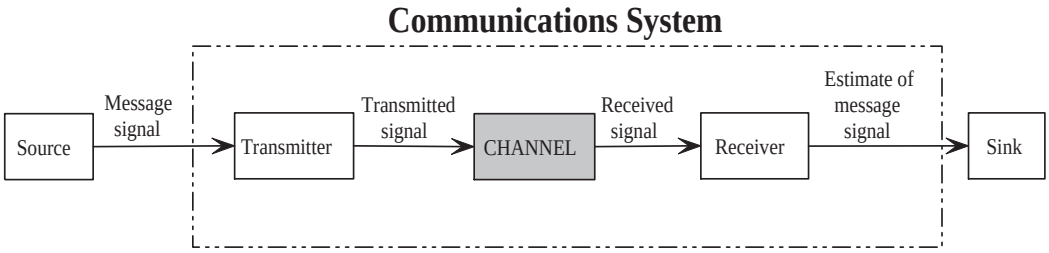

Figure 3.1. Block diagram of a communications system.

3.1. Additive Gaussian Noise Channel

The simplest communication channel model is the additive noise channel, this model is illustrated in Fig. 3.2. The transmitted signal $s(k)$ is corrupted by an additive random noise process $n(k)$. This noise process may arise from the interference during the motion in the propagation medium (in the case of wireless communications), or from the electronic components and the amplifiers at the receiver in the communication system (due to the electron random motion). Due to this corruption, the received signal $r(k)$ can be

expressed as,

$$r(k) = s(k) + n(k), \tag{3.1}$$

If the noise is primarily introduced by the electronic components and amplifiers at the receiver, it may be characterized as *thermal noise* which is statistically characterized as Additive Gaussian Noise (AGN) process [13].

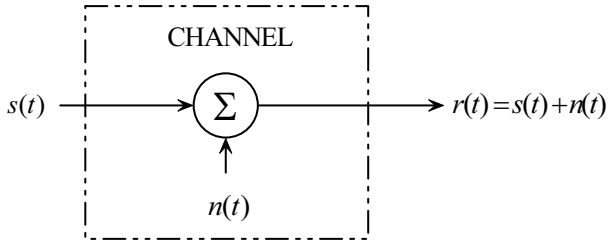

Figure 3.2. Additive Gaussian Noise Channel.

Even though AWGN channels are usually used as reference channel models in communication systems, they are insufficient to describe communication channels in real life, since there are different noise sources that may corrupt the transmitted signal.

3.2. Impulsive Noise Channel

According to the Central Limit Theorem (CLT), the noise which results from the addition of many sources is usually modeled as Gaussian noise. However, in real life, because of man-made noise, underwater acoustic noise, etc. [14] [15], these exhibit non-Gaussian distribution, which can be modeled as impulsive noise [16]. The probability density function (pdf) of an impulsive noise model can be described using the Gaussian mixture model [17] [18]:

$$f = (1-\epsilon)N(0,\sigma_n^2) + \epsilon N(0,\kappa\sigma_n^2), \tag{3.2}$$

and the total noise variance is given by [19]:

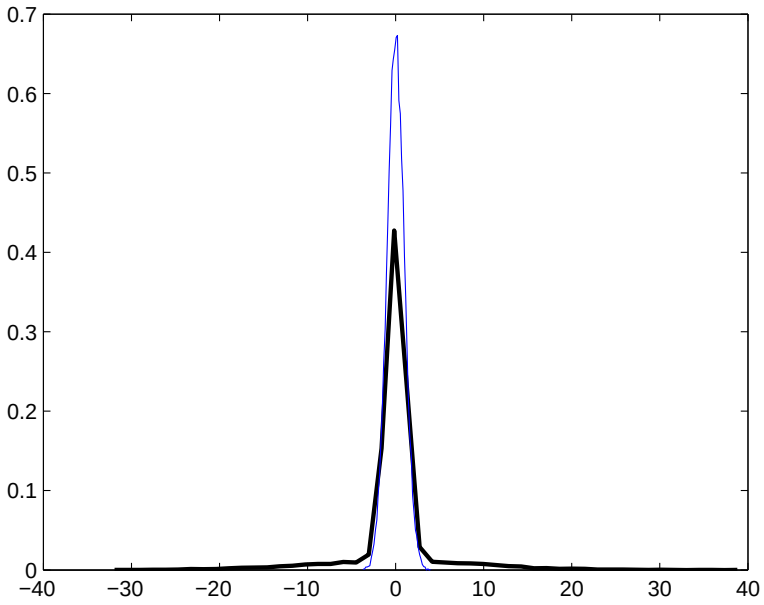

Figure 3.3. Thick line: Impulsive noise pdf ($\epsilon = 0.2$, $\kappa = 100$), thin line: AWGN pdf ($\mu = 0$, $\sigma^2 = 1$).

$$\sigma^2 = (1-\epsilon)\sigma_n^2 + \epsilon\kappa\sigma_n^2, \tag{3.3}$$

where $N(0,\sigma_n^2)$ is a Gaussian pdf with zero mean and variance σ_n^2, representing the nominal background noise. $N(0,\kappa\sigma_n^2)$ represents the impulsive component where ϵ is the probability to have an impulsive component and it is $0 < \epsilon < 1$, κ is the strength of the impulsive noise and $\kappa \geq 1$. The received signal corrupted by impulsive noise is given by

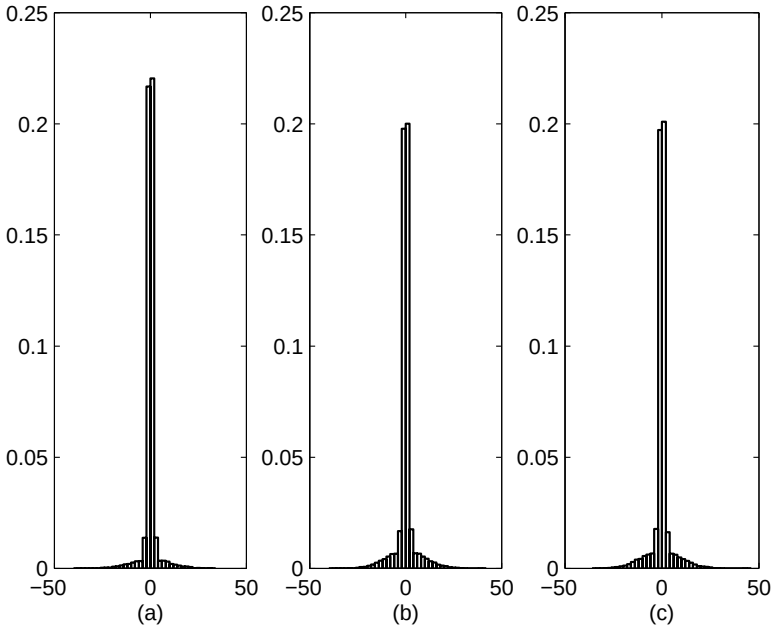

Figure 3.4. The impulsive noise histograms for various values of ϵ and $\kappa = 100$: (a) $\epsilon = 0.1$, (b) $\epsilon = 0.2$, (c) $\epsilon = 0.3$.

(3.4), where $s(k)$ is the desired response and $n(k)$ is the impulsive noise component with algebraic tails which are significantly heavier than the exponential tail of the Gaussian distribution [20] as shown in Fig. 3.3.

$$x(k) = s(k) + n(k), \tag{3.4}$$

As ϵ increases, the impulsiveness also increases as depicted in Fig. 3.4. The tails of the impulsive noise distribution do not go to zero, but they spread apart.

3.3. Fading Channels

In mobile communications, the transmitted electromagnetic waves do not reach directly the receiver due to obstacles; mountains, trees, buildings, moving objects, ..., etc, that block the Line-of-Sight (LOS) path. In addition to the LOS path, these obstacles result in reflected, diffracted, scattered and LOS signals that are vectorially summed to give one signal at the receiver; this effect is usually called *multipath propagation*. A typical scenario for a mobile radio channel is depicted in Fig. 3.5. Due to multipath propagation, the received signal consists of the sum of attenuated, delayed and phase-shifted multi-replicas of the transmitted signal. This addition can be constructive or destructive, depending on the phase shift of each replica [21].

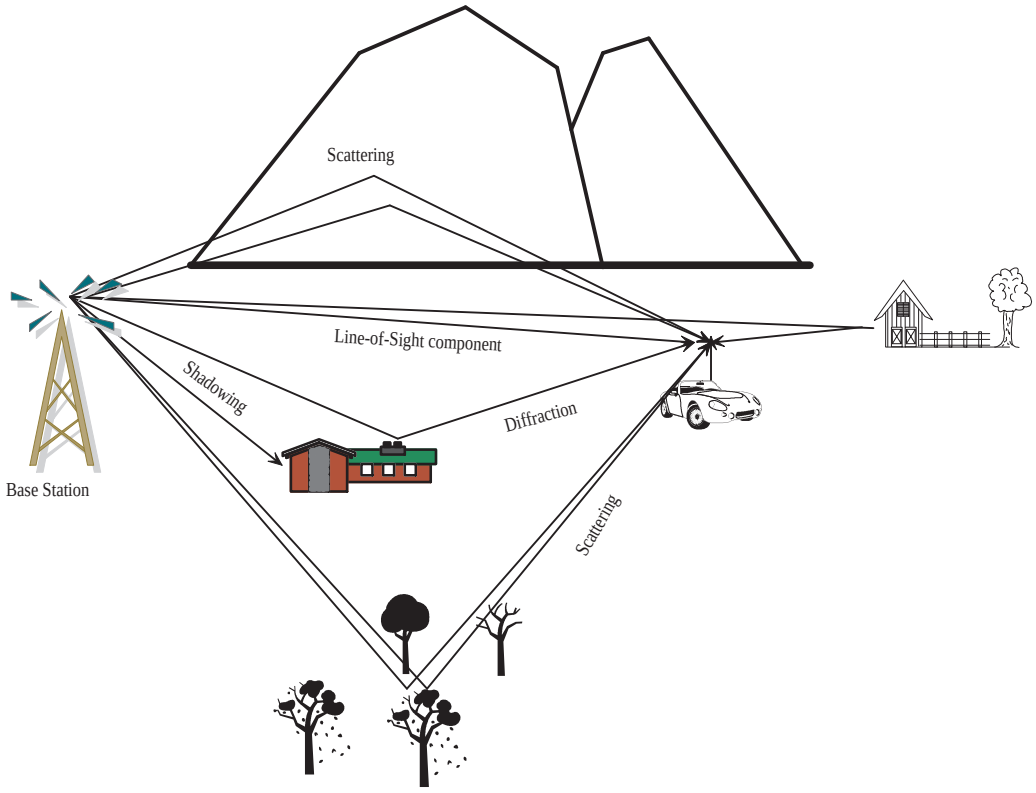

Figure 3.5. Multipath propagation in a mobile radio channel, [21].

3.3.1. Parameters Characterizing Fading Channels

Multipath channels are characterized by mobile channel parameters. Many channel parameters are derived from the power delay profile which describes how much power arrives within a certain delay interval [21]. Power delay profiles are derived by averaging the instantaneous power delay profile, given in (3.5), measured in a local area. These power delay profiles are generally represented as plots of relative received power as a function of excess delay with respect to a fixed time delay reference. In order to characterize and compare these multipath channels, some of its parameters have to be discussed; *time dispersion parameters*, *Doppler spread*, *coherence time*, and *coherence bandwidth*.

$$|r(t_0)|^2 = \sum_{k=0}^{M-1} a_k^2(t_0). \tag{3.5}$$

M is the total number of multipath components and $a_k^2(t_0)$ represents the received signal power in each path.

3.3.1.1. Time Dispersion Parameters. The *mean excess delay* ($\overline{\tau}$), *rms delay spread* (σ_τ), and *excess delay spread* (X dB) are multipath channel parameters that can be determined from the power delay profile of that channel. The mean excess delay ($\overline{\tau}$) is the first moment of the power delay profile [21] and is given by (3.6),

$$\overline{\tau} = \frac{\sum_k a_k^2 \tau_k}{\sum_k a_k^2}, \tag{3.6}$$

The rms delay spread is the square root of the second central moment of the power delay profile [21], given by (3.7)

$$\sigma_\tau = \sqrt{\overline{\tau^2} - (\overline{\tau})^2}, \tag{3.7}$$

where

$$\overline{\tau^2} = \frac{\sum_k a_k^2 \tau_k^2}{\sum_k a_k^2}, \tag{3.8}$$

The maximum excess delay (X dB) of the power delay profile [21] is defined to be the time delay during which multipath energy falls to X dB below the maximum. Maximum excess delay is defined as $\tau_X - \tau_0$, where τ_0 is the first arriving signal and τ_X is the maximum delay at which a multipath component is within X dB of the strongest arriving multipath signal.

3.3.1.2. Doppler Spread and Coherence Time. Doppler spread and coherence time are parameters which describe the time varying nature of the channel in a small-scale region [22]. Doppler spread B_D is essentially the range of frequencies over which the received Doppler spectrum is non-zero. When a signal with carrier frequency (f_c) is transmitted, the Doppler spectrum of the received signal will be in the range $f_c - f_d$ to $f_c + f_d$, where f_d is the Doppler frequency shift and is measured as,

$$f_d = \frac{v}{\lambda_w} \cos\theta, \tag{3.9}$$

where v is the velocity of the receiver towards or away from the transmitter, λ_w is the wavelength and θ is the measured angle between the direction of motion of the receiver and direction of arrival of the scattered waves [21], as shown in Fig. 3.6. *For very far distances between the transmitter and receiver*; θ *is assumed to be zero*. If the baseband signal bandwidth is much greater than B_D, the effects of Doppler spread are negligible at the receiver, which is called a *slow fading channel*.

Coherence time T_c is the time domain dual of Doppler spread and is used to charac-

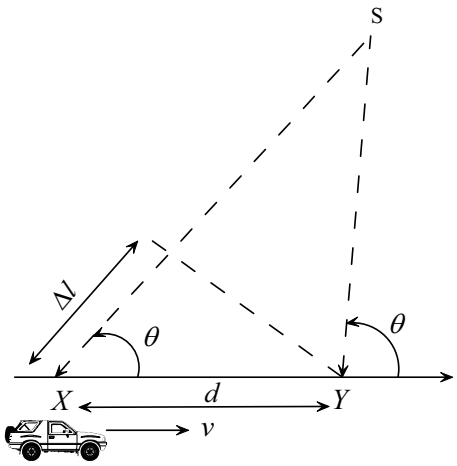

Figure 3.6. Illustration of Doppler effect, [21].

terize the time varying nature of the frequency dispersiveness of the channel in the time domain, and is related to the maximum Doppler shift as its reciprocal [21] as given in (3.10),

$$T_c \approx \frac{1}{f_m}, \tag{3.10}$$

3.3.1.3. Coherence Bandwidth. Coherence bandwidth is a measure of the range of frequencies over which the channel can be considered *flat* [22]. In other words, it is the range of frequencies over which two frequency components have a strong potential for amplitude correlation [21]. If the coherence bandwidth is defined as the bandwidth over which the frequency correlation is above 0.9 [21], then the coherence bandwidth is given as (3.11)

$$B_c \approx \frac{1}{50\sigma_\tau}, \tag{3.11}$$

But if the coherence bandwidth is defined as the bandwidth over which the frequency correlation is above 0.5 [21], then the coherence bandwidth is given as (3.12)

$$B_c \approx \frac{1}{5\sigma_\tau}, \tag{3.12}$$

From (3.11) and (3.12) we note that there is no exact relation between the coherence bandwidth and the rms delay spread.

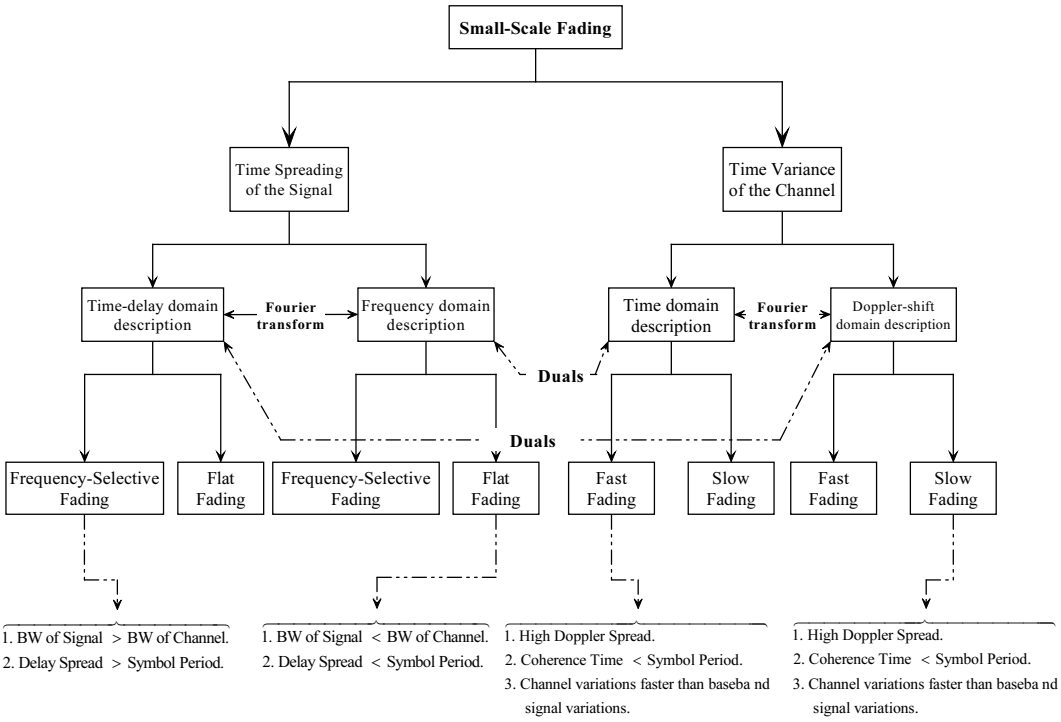

Figure 3.7. Types of small-scale fading, [23].

3.3.2. Types of Small-Scale Fading

The type of fading experienced by a signal propagating through a mobile radio channel depends on the relation between the transmitted signal parameters and the channel parameters. These will lead to four types of fading as depicted in Fig. 3.7: *flat fading*, *frequency-selective fading*, *slow fading*, and *fast fading* [23].

3.3.2.1. Flat Fading and Frequency-Selective Fading. A channel is said to be flat or frequency nonselective, if the mobile radio channel bandwidth is greater than the bandwidth of the transmitted signal. The amplitude of the received signal changes with time, due to the fluctuations in the gain of the channel. The distribution of the amplitude of a flat fading channel is important. The most common amplitude distribution is Rayleigh distribution [21]; its pdf is given in (3.13) and is depicted in Fig. 3.8.

$$p(x) = \begin{cases} \frac{x}{\sigma^2} e^{(-\frac{x^2}{2\sigma^2})} & , \quad 0 \leq x \leq \infty \\ 0 & , \quad \text{otherwise} \end{cases} \tag{3.13}$$

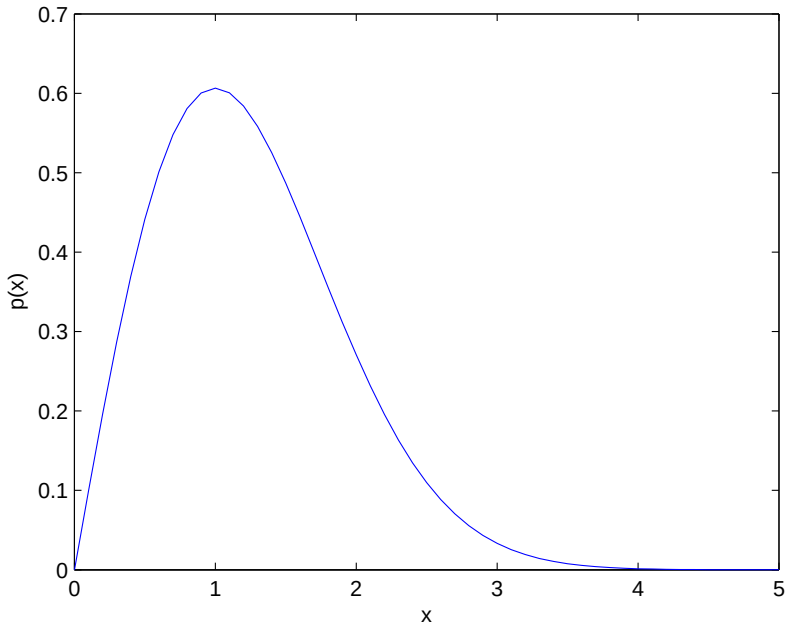

Figure 3.8. Rayleigh probability density function.

A channel is said to be frequency-selective if the mobile channel bandwidth is smaller than the bandwidth of the transmitted signal. In this case, the channel impulse response has a delay spread greater than the symbol period on the transmitted signal. In a frequency selective fading channel, Inter-Symbol Interference (ISI) occurs due to the time

dispersion of the transmitted symbols within the channel.

3.3.2.2. Fast Fading and Slow Fading. A channel is said to be a fast fading channel if its impulse response changes rapidly within the symbol period. When this happens, coherence time of the channel will be smaller than the symbol period of the transmitted signal. Similarly, a channel is said to be slow fading channel, if its impulse response changes much slower than the transmitted signal; Doppler spread of the channel is less than the bandwidth of the transmitted signal.

3.4. Channel Equalization

An equalizer is an input estimator. Since we are interested in making correct decisions, it is natural to choose an input estimator that minimizes the probability of making an error; when the estimated signal is not equal to the desired one ($\hat{d}(t) \neq d(t)$) [31]. Such an estimator is optimal under the so-called MAP-criterion (*Maximum A posteriori Probability*). If all values in the symbol alphabet are equally probable, this criterion is equivalent to the more familiar ML-criterion (*Maximum Likelihood*). If the noise is white and Gaussian, this optimal estimator computes the quantity in (3.14) for all possible input sequences, and chooses the sequence which results in the smallest J.

$$J = \sum_{t} \mid y(t) - d(t) \mid^2, \tag{3.14}$$

Generally, equalizers are divided into two types based on their linear properties; i.e. *Linear* and *Nonlinear* equalizers. one of the most important points to notice is that all the equalizers in their basic structure make use of the Least Square (LS) algorithms. This is shown in Fig. 3.9 [24].

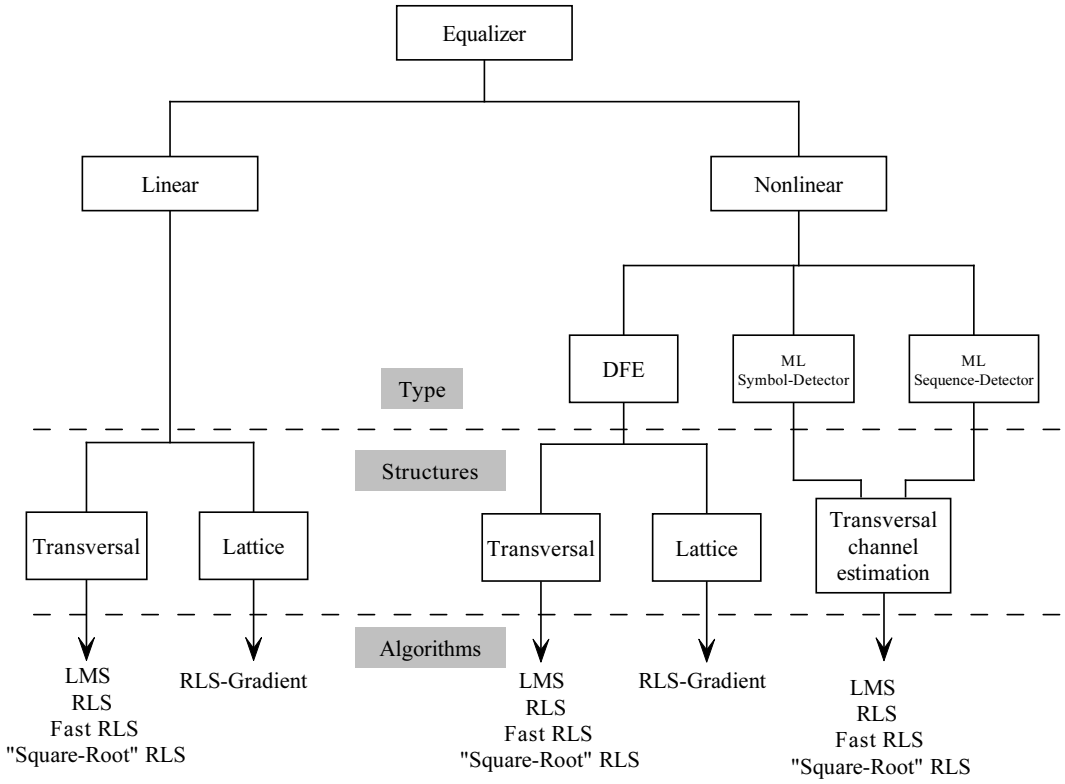

Figure 3.9. Comparison of equalizer structures, [24].

Among the many used structures for equalization the simplest is the transversal (tapped delay-line or nonrecursive) equalizer shown in Fig. 2.4. In such an equalizer, the current and past values of the received signal are linearly weighted by equalizer coefficients (tap gains; $w_n,\ n = 0, 1, \ldots, N-1$) and summed to produce the output. If the delays and tap-gain multipliers are analog, the continuous output of the equalizer $y(t)$ is sampled at the symbol rate and the samples go to the decision device. In the universally used digital implementation, samples of the received signal at the symbol rate are stored in a digital shift register (or memory), and the equalizer output samples (sums of products) are computed digitally, once per symbol.

The LMS-based equalizer [1] is a robust one. The equalizer coefficients are chosen to minimize the mean-square error-the sum of squares of all the ISI terms plus the noise

power at the output of the equalizer. Therefore, the LMS-based equalizer maximizes the signal-to-distortion ratio at the equalizer output within the constraints of the equalizer length and delay [1].

The delay introduced by the equalizer depends on the position of the main or reference tap of the equalizer. Typically, the tap gain corresponding to the main tap is the largest.

If the values of the channel impulse response at the sampling instants are known, the N coefficients of the LMS based equalizers can be obtained by solving a set of N linear simultaneous equations for each case.

In the next chapter, we will try to estimate the reciprocal channel impulse response in different channel environments and by using different LMS-based equalizer types.

CHAPTER 4

SIMULATION RESULTS

As mentioned in the previous chapter, there are three main channel types; AGN channels, impulsive noise channels, and fading channels. In this chapter, the discussion will be based on the noise type; white noise channels, and correlated noise channels.

In white noise channels case, the eigenvalue spread ($\frac{\lambda_{max}}{\lambda_{min}}$) of the autocorrelation matrix (**R**) of the input signal is unity. Whereas, in the case of correlated noise this eigenvalue spread will become $\gg 1$ depending on the response of the filter used to have correlated noise.

In order to study the effect of the noise distribution, we will discuss the Gaussian noise and the constrained impulsive noise environments. In the case of impulsive noise channels, increasing the impulsive component strength, or the probability of the outliers, will increase the noise power. This increment will affect the transmitted signal more.

In fading channels, attenuating the signal's amplitude leads to poor performance in detecting the signal at the receiver. Simulations will discuss those effects in detail.

In this thesis *MATLAB* Software Package is used for the simulation of the standard LMS, NLMS, L-LMS, ML-LMS and FRS-LMS algorithms. Simulations discuss the performances of these algorithms in AWGN, correlated Gaussian noise, white and correlated constrained impulsive noise, white and correlated impulsive noise, and mobile

fading channels.

Simulations of the FRS-LMS algorithm were done using the fast computation method in [12] with the weight function as shown in Fig. 2.6(b).

4.1. Additive Gaussian Noise

Suppressing noise is a major area of research. As mentioned in the previous chapter, due to the Central Limit Theorem (CLT), this noise is usually modeled as Gaussian noise. Therefore, this section discusses the performance of the standard LMS, NLMS, Leaky LMS, Modified Leaky LMS and FRS-LMS algorithms in additive Gaussian noise environments. The signal model can be described as [1]:

$$x(k) = s(k) + n(k), \tag{4.1}$$

4.1.1. Additive White Gaussian Noise

In order to study the effect of AWGN, two cases will be discussed; $s(k)$ has a fixed frequency (ω_s), and when an abrupt frequency change occurs in $s(k)$.

4.1.1.1. Single Tone Signals. In this section, an additive white Gaussian noise process is generated and added to the received signal $s(k)$. $s(k)$ is a sinusoidal signal with frequency $\omega_s = \pi/6$ and unit amplitude. Simulations were done using the parameters; $N = 32$ taps, SNR$= 10$dB. For the FRS-LMS algorithm, $\mu = 0.017$, window size ($\omega_b = \omega_2 - \omega_1$), $\omega_b = \pi/7$, $w_1 = 20$ and $w_2 = 0.002$. For the NLMS algorithm, $\mu = 0.017$. For the L-LMS algorithm the parameters used were; $\mu = 0.0008$, $\gamma = 0.001$. For the LMS algorithm $\mu = 0.0007$. Fig. 4.1 shows the performances of the mentioned algorithms.

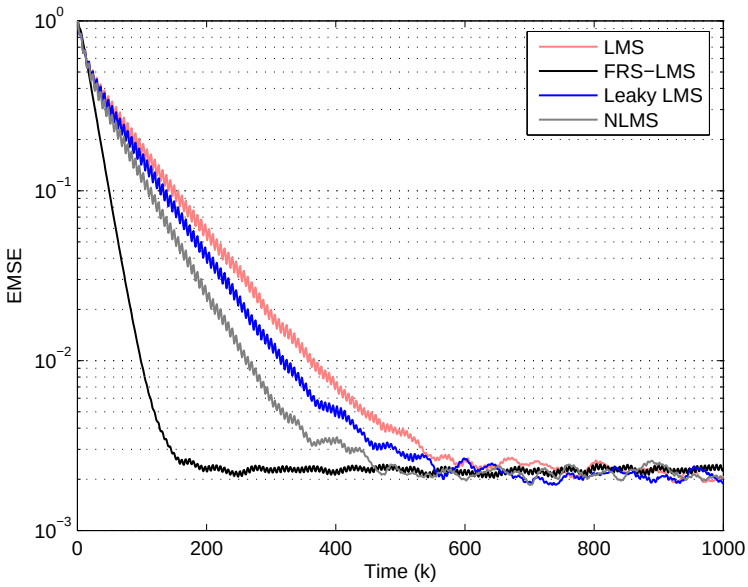

Figure 4.1. Ensemble average of the MSE of FRS-LMS, NLMS, L-LMS, and LMS with AWGN: $N = 32$, SNR$= 10$dB, $\omega_s = \pi/6$, FRS-LMS: $\mu = 0.017$, $w_1 = 20$, $w_2 = 0.002$, $\omega_b = \pi/7$, NLMS: $\mu = 0.017$, L-LMS: $\mu = 0.0008$, $\gamma = 0.001$, LMS: $\mu = 0.0007$.

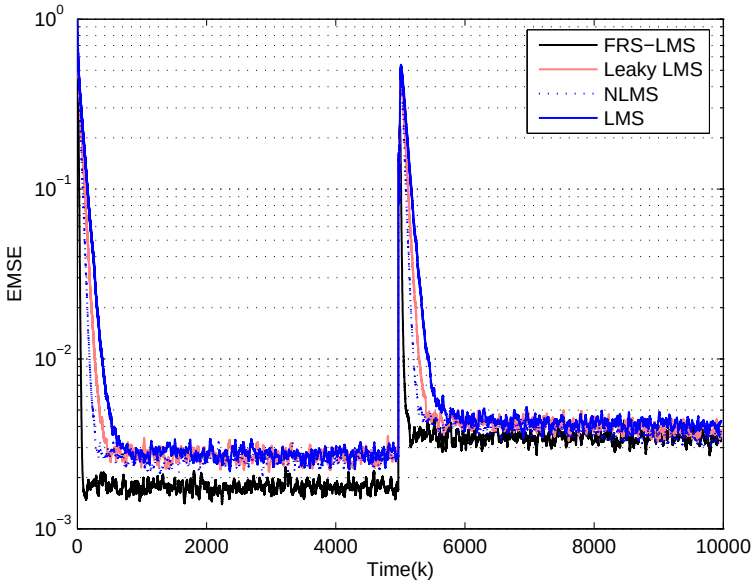

Figure 4.2. Ensemble average of the MSE of FRS-LMS, NLMS, L-LMS, and LMS with AWGN when abrupt change in frequency occurs: from $\omega_s = 1.57$ rad to $\omega_s = 1.67$ rad, $N = 32$, SNR= 10dB. FRS-LMS: $\mu = 0.0134$, $w_1 = 20$, $w_2 = 0.002$, $\omega_b = \pi/6$, NLMS: $\mu = 0.025$, L-LMS: $\mu = 0.001$, $\gamma = 0.001$, LMS: $\mu = 0.0007$.

It can be observed from the figure that they have similar performances in terms of MSE (MSE=0.0025), but the FRS-LMS converges much faster than the other algorithms (FRS-LMS converges after 160 time samples, NLMS converges after 460 time samples, Leaky-LMS converges after 570 time samples and the standard LMS converges after 600 time samples). The previous parameters were used in order to have the same steady state MSE for all the algorithms. But, it may be noted that, decreasing the step-size of the FRS-LMS to be like that of the L-LMS for example, will give the same MSE's convergence rate, but most probably lower MSE.

4.1.1.2. Abrupt Change in Frequency. In this part, an additive white Gaussian noise process is added to the received signal $s(k)$. $s(k)$ is a sinusoidal signal with initial fre-

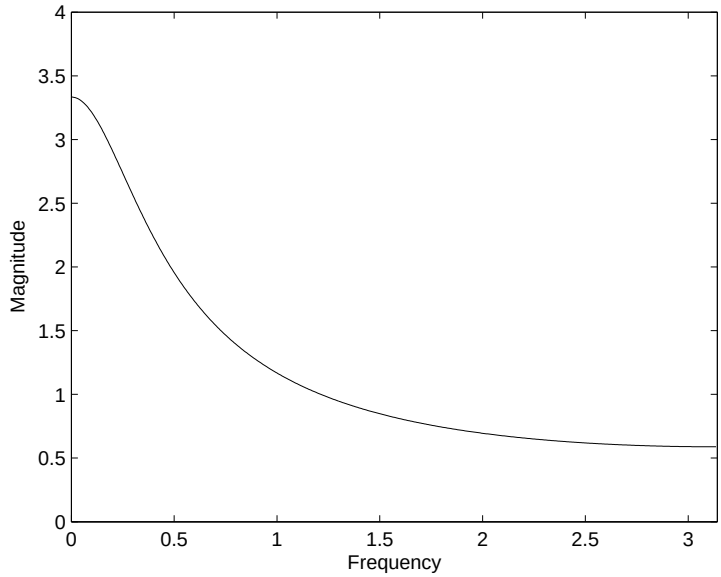

Figure 4.3. Filter response of first-order autoregressive process (AR(1)) with autocorrelation coefficient $\rho = 0.7$.

quency $\omega_s = 1.57$ rad, the frequency abruptly changes to $\omega_s = 1.67$ rad at the 5000^{th} time samples. Simulations were done using the parameters; $N = 32$ taps, SNR$= 10$dB. For the FRS-LMS algorithm the parameters were; $\mu = 0.0134$, $\omega_b = \pi/6$ and $w_1 = 20$, $w_2 = 0.002$. For NLMS algorithm, $\mu = 0.025$. For the L-LMS algorithm the parameters used were; $\mu = 0.001$, $\gamma = 0.001$, the LMS algorithm $\mu = 0.0007$. Fig. 4.2 shows the performance comparison between these algorithms. It shows that they have similar performances in terms of MSE (MSE=0.0035) after the frequency changes, but the FRS-LMS converges faster than the other algorithms (FRS-LMS converges 200 time samples before the NLMS, NLMS

converges 180 time samples before the L-LMS, and L-LMS converges 220 time samples before the standard LMS algorithm). We note that after the abrupt change in the signal's frequency occurs, the FRS-LMS algorithm again converges very fast but with a

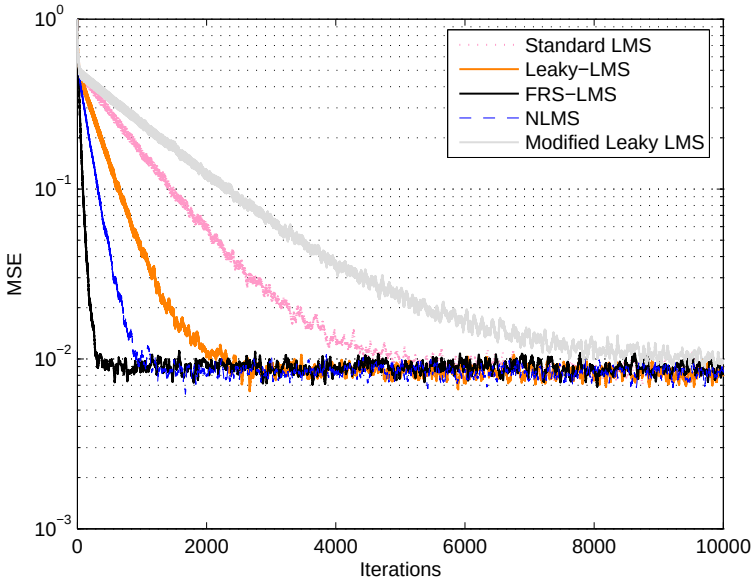

Figure 4.4. Ensemble average of the MSE of FRS-LMS, NLMS, L-LMS, ML-LMS, and LMS with AR(1) Gaussian noise process: $\rho = 0.7$, $N = 32$, SNR$= 5$dB, $\omega_s = 0.3\pi$. FRS-LMS: $\mu = 0.0015$, $w_1 = 10$, $w_2 = 0.001$, $\omega_b = \pi/6$, NLMS: $\mu = 0.007$, L-LMS: $\mu = 0.00016$, $\gamma = 0.0001$, ML-LMS: $\mu = 0.000045$, LMS: $\mu = 0.00007$.

slightly higher MSE (0.0035) than the MSE before the frequency change (0.0018). This results from the fact that, increasing the signal frequency with the same used parameters may lead to a higher MSE or even to a slower tendency to convergence.

4.1.2. Additive Correlated Gaussian Noise (ACGN)

To study the effect of correlated noise, two examples are discussed in this section. The first example discusses the case when the correlated noise is created by AR(1) process, and the second example discusses the case when the correlated noise is created by a FIR filter with frequency response given in Fig. 4.5.

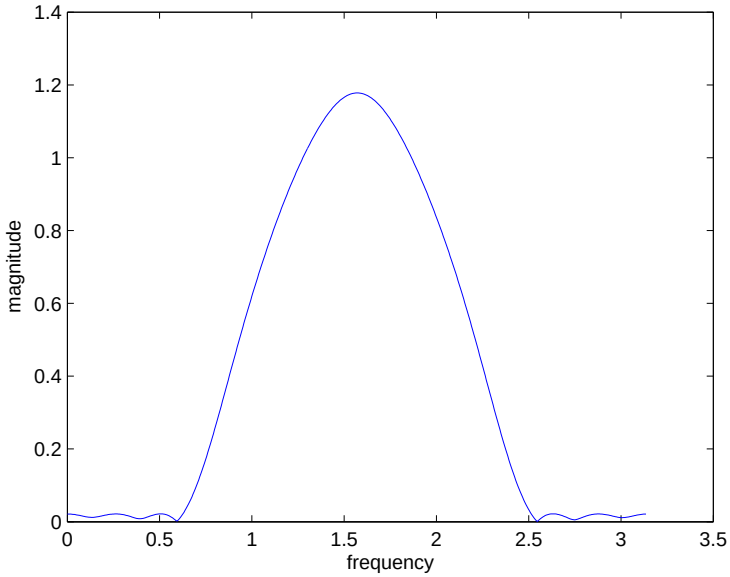

Figure 4.5. Magnitude frequency response of the noise filter.

4.1.2.1. Additive Correlated Gaussian Noise using First-Order Autoregressive Process.

In this section, $s(k)$ is a sinusoidal signal with frequency $\omega_s = 0.3\pi$ and unit amplitude which is corrupted by a first-order autoregressive noise (AR(1)), generated using the model $v(k+1) = \rho v(k) + v_0(k)$, where $v_0(k)$ is an additive white Gaussian noise process and ρ is the correlation parameter. The filter response of such a process with the $\rho = 0.7$ is given in Fig. 4.3. Simulations were done with the parameters; $N = 32$ taps, SNR= 5dB. For the FRS-LMS algorithm, $\mu = 0.0015$, $\omega_b = \pi/6$ and $w_1 = 10$, $w_2 = 0.001$. For NLMS algorithm, $\mu = 0.007$. For the L-LMS algorithm the parameters used were; $\mu = 0.00016$, $\gamma = 0.0001$, for the ML-LMS algorithm $\mu = 0.000045$, and for the LMS algorithm $\mu = 0.00007$. Fig. 4.4 shows the performances of the mentioned algorithms. It can be observed from the figure that they have similar performances in terms of MSE (MSE=0.009), but the FRS-LMS converges much faster than the other algorithms (FRS-LMS converges after 300 time samples, NLMS converges after 950 time

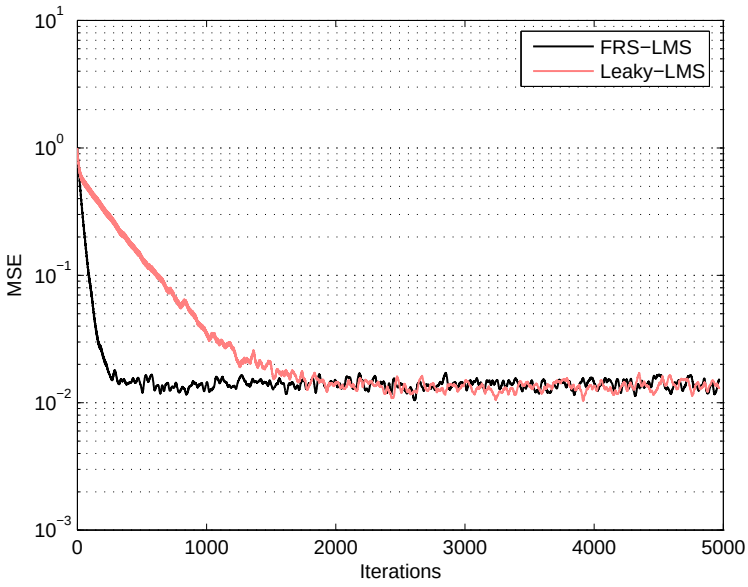

Figure 4.6. The ensemble MSE for FRS-LMS and L-LMS in ACGN. $N = 32$, SNR$= 5$dB, $\omega_s = 0.3\pi$. L-LMS: $\mu = 0.00016$, $\gamma = 0.0001$. FRS-LMS: $\mu = 0.0015$, $\omega_b = \pi/6$, $w_1 = 10$, $w_2 = 0.001$.

samples, L-LMS converges after 2350 time samples, the standard LMS converges after 4900 time samples and the ML-LMS converges after 9850 time samples). This good performance of the FRS-LMS algorithm refers to that shaping the frequency response of the FIR filter and giving less weights around the signal's frequency gives more accurate MSE. Also, giving higher weights outside the frequency band makes the algorithm tends to the MSE steady state faster than the other algorithms.

4.1.2.2. Additive Correlated Gaussian Noise using FIR Filter. In this part, $s(k)$ is a sinusoidal signal with frequency $\omega_s = 0.3\pi$ and unit amplitude which is corrupted by a correlated Gaussian noise process which is generated by filtering an AWGN process with an FIR filter having the magnitude response characteristic shown in Fig. 4.5. Simulation parameters for FRS-LMS are similar with $\mu = 0.0015$, $\omega_b = \pi/6$ and $w_1 = 10$, $w_2 = 0.001$, SNR$= 5$dB. The Leaky-LMS algorithm has $\mu = 0.00016$, $\gamma = 0.0001$. Fig. 4.6 shows that for the same MSE (MSE=0.014) (the FRS-LMS converges after about

300 time samples, whereas the Leaky-LMS converges after 1800 time samples). Here, the FRS-LMS is able to shape the frequency response in order to suppress correlated noise more effectively.

4.2. Impulsive Noise Model

As mentioned in the previous section, due to the Central Limit Theorem (CLT), the noise is usually modeled as Gaussian noise. However, in real life, this noise can be modeled as impulsive noise [16] [20] [32].

Due to the ability of FRS-LMS algorithm in suppressing the correlated Gaussian noise (as shown in the previous section), we investigate its ability in suppressing white and correlated impulsive noise, and we compare its performance with the L-LMS algorithm performance.

4.2.1. Input Signal with Constrained Impulsive Noise

In order to study the effects of the noise distribution, a constrained impulsive noise is added to the signal $s(k)$.

4.2.1.1. Constrained White Impulsive Noise. In this part, a constrained white impulsive noise process with $\epsilon = 0.2$, $\kappa = 100$ is generated and added to the received signal $s(k)$. $s(k)$ is a sinusoidal signal with frequency $\omega_s = \pi/6$ and unit amplitude. Simulations were done using the parameters; $N = 32$ taps, SNR$= 5$dB. For the FRS-LMS algorithm the parameters used were; $\mu = 0.017$, $\omega_{=}\pi/9$, $w_1 = 20$, $w_2 = 0.002$. For the L-LMS algorithm the parameters used were; $\mu = 0.001$, $\gamma = 0.001$. Fig. 4.7 shows that they have similar performances in terms of MSE (MSE=0.0024), but the FRS-LMS converges

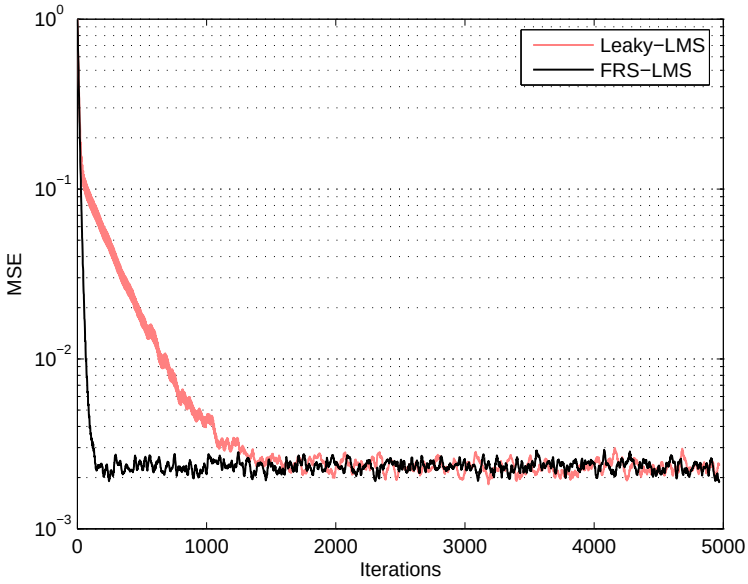

Figure 4.7. The ensemble MSE for FRS-LMS and L-LMS in constrained white impulsive noise. $\epsilon = 0.2$, $\kappa = 100$, $N = 32$, SNR$= 5$dB, $\omega_s = \pi/6$. L-LMS: $\mu = 0.001$, $\gamma = 0.001$. FRS-LMS: $\mu = 0.017$, $\omega_b = \pi/9$, $w_1 = 20$, $w_2 = 0.002$.

much faster than the L-LMS algorithm (FRS-LMS converges after 220 time samples, whereas the L-LMS converges after 1600 time samples).

4.2.1.2. Constrained Correlated Impulsive Noise. In this part, a constrained correlated impulsive noise process is generated with $\epsilon = 0.2$, $\kappa = 100$ and added to the received signal $s(k)$. The correlated noise was generated by filtering a constrained white impulsive noise process with an FIR filter having the magnitude response characteristic shown in Fig. 4.5. $s(k)$ is a sinusoidal signal with frequency $\omega_s = \pi/3$ and unit amplitude. Simulations were done with the parameters; $N = 32$ taps, SNR$= 5$dB. For FRS-LMS the parameters were; $\mu = 0.0015$, $\omega_b = \pi/6$, $w_1 = 10$, $w_2 = 0.001$. For L-LMS algorithm the parameters were; $\mu = 0.00016$, $\gamma = 0.0001$. In Fig. 4.8 we observe that they both

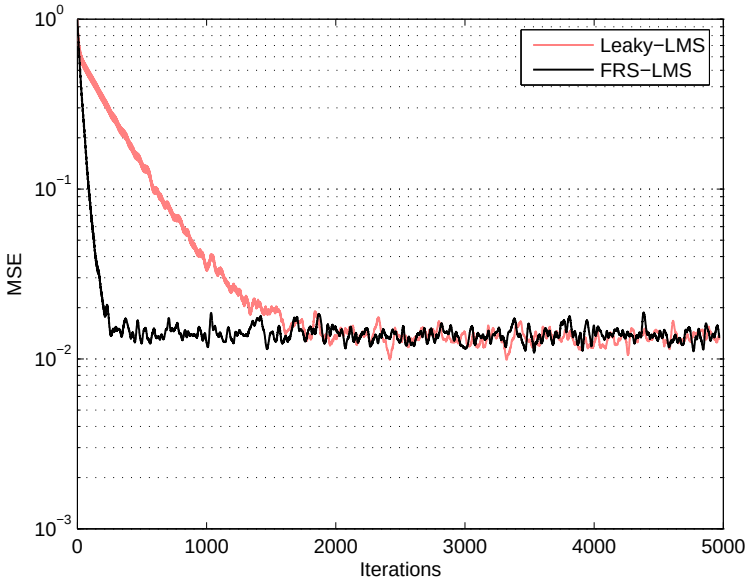

Figure 4.8. The ensemble MSE for FRS-LMS and L-LMS in constrained correlated impulsive noise, $\epsilon = 0.2$, $\kappa = 100$, $N = 32$, SNR= 5dB, $\omega_s = \pi/3$. L-LMS: $\mu = 0.00016$, $\gamma = 0.0001$. FRS-LMS: $\mu = 0.0015$, $\omega_b = \pi/6$, $w_1 = 10$, $w_2 = 0.001$.

have similar performances in terms of MSE (MSE=0.015), but the FRS-LMS converges much faster than the Leaky-LMS (FRS-LMS converges after 300 time samples, whereas the L-LMS converges after 1900 time samples).

4.2.2. Input Signal with Impulsive Noise

In this section, for the simulations it is assumed that a sinusoidal signal $s(k)$ with frequency $\omega_s = \pi/3$ is corrupted by additive noise. The filter length, $N = 32$ taps. The cases where the noise contains impulsive components and when it becomes correlated are investigated in detail.

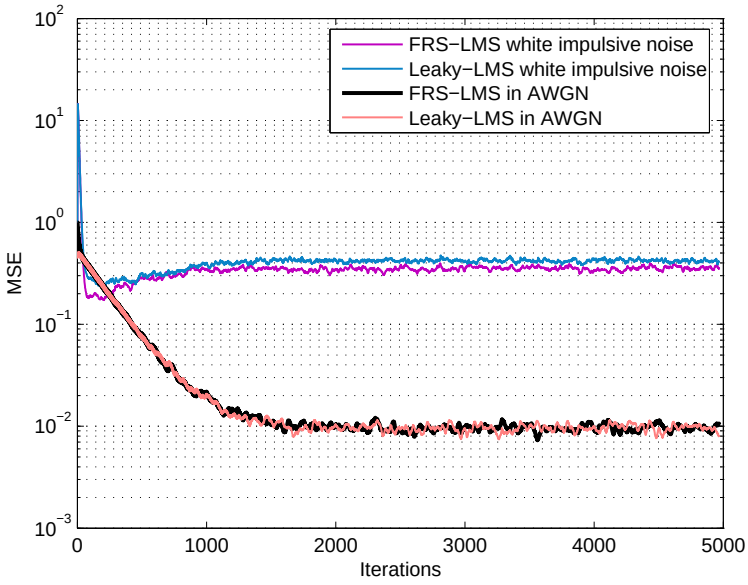

Figure 4.9. The ensemble MSE for FRS-LMS and L-LMS in AWGN and white impulsive noise environments, $\epsilon = 0.2$, $\kappa = 100$, $N = 32$, SNR= 5dB, $\omega_s = \pi/3$, $\mu = 0.00025$. L-LMS: $\gamma = 0.001$. FRS-LMS: $\omega_b = \pi/6$, $w_1 = 20$, $w_2 = 0.001$.

4.2.2.1. White Impulsive Noise. **A.** In order to study the effects of the impulsive components (outliers) of the noise process. Under the same conditions, an AWGN process has been added to the signal $s(k)$ and then a white impulsive noise process is added to $s(k)$.

Firstly, the signal $s(k)$ is assumed to be corrupted by AWGN. Simulations of the FRS-LMS algorithm were done using the parameters; $\mu = 0.00025$, $\omega_b = \pi/3$, $w_1 = 20$, $w_2 = 0.002$ and SNR= 5dB. For the L-LMS algorithm, the parameters used were; $\mu = 0.00025$, $\gamma = 0.001$. Then, using the same parameters a white impulsive noise process is generated with $\epsilon = 0.2$, $\kappa = 100$ and added to $s(k)$. These parameters are used to model severely impulsive noise [20]. From Fig. 4.9In the AWGN case, the L-LMS and the FRS-LMS algorithms have similar MSE (MSE=0.01). This does not contradict with Fig. 4.1 because decreasing μ will decrease the MSE to a certain minimum point, decreasing

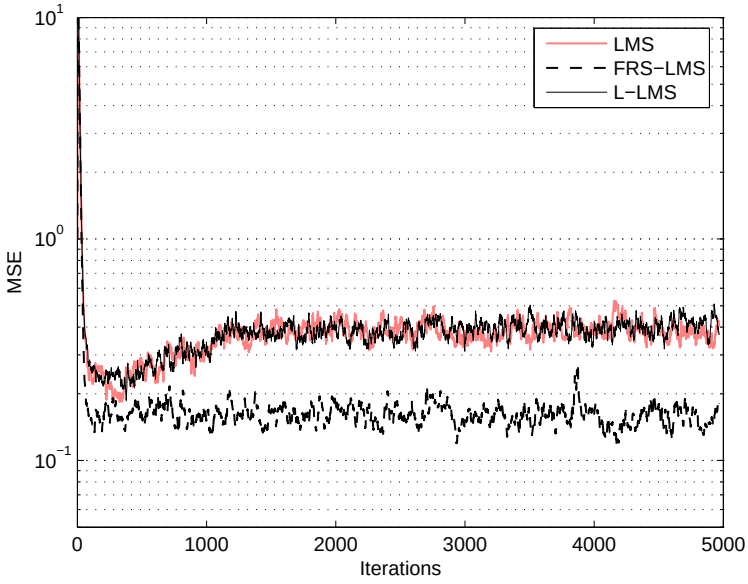

Figure 4.10. The ensemble MSE for LMS, L-LMS, ML-LMS, and FRS-LMS in additive white impulsive noise, $N = 33$, SNR$= 5$dB, $\mu = 0.00025$. L-LMS: $\gamma = 0.001$. FRS-LMS: $\omega_b = \pi/4$, $w_1 = 100$, $w_2 = 0.001$.

μ further will not decrease the MSE. Whereas in the impulsive noise case, the FRS-LMS converges to a relatively lower MSE (MSE=0.19) than the L-LMS (MSE=0.42). By comparing the AWGN and the impulsive noise cases, it is important to note that the outliers in the noise process has decreased the performance of both algorithms severely (approximately 20 and 40 times more MSE for FRS-LMS and L-LMS respectively). However, the FRS-LMS is relatively more robust to impulsive noise since the magnitudes of the outliers are reduced by the weight function.

B. In this part, it is assumed that a BPSK modulated signal $x(k)$ corrupted by additive noise. All the simulations of this part were done with a filter length $N = 33$ taps and in the case of MSE performance comparison, the SNR$= 5$dB. The weight function was selected under the assumption that the signal is within the noise band.

A white impulsive noise process is generated with $\epsilon = 0.2$, $\kappa = 100$. The signal

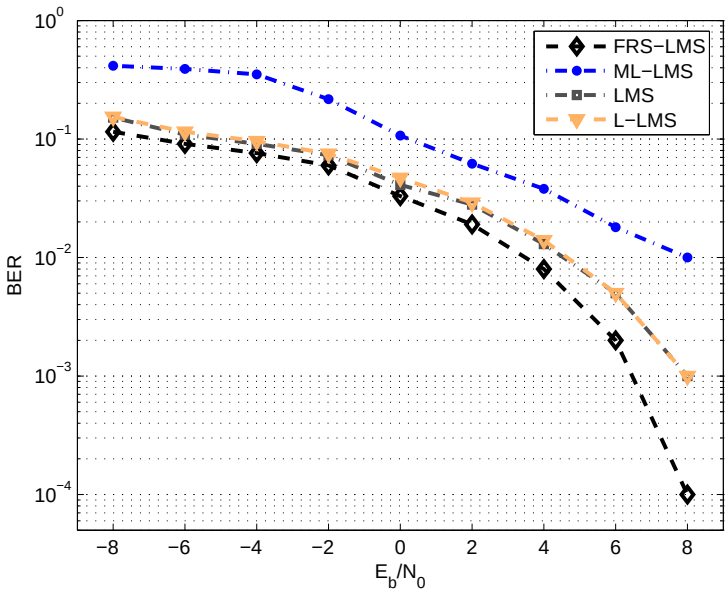

Figure 4.11. BER Performance for LMS, L-LMS, ML-LMS, and FRS-LMS in additive white impulsive noise, $N = 33$, $\mu = 0.00025$. L-LMS: $\gamma = 0.001$. FRS-LMS: $\omega_b = \pi/4$, $w_1 = 100$, $w_2 = 0.001$.

$x(k)$ is assumed to be corrupted by additive white impulsive noise. Simulations were done for LMS, L-LMS, ML-LMS and FRS-LMS algorithms with step-size $\mu = 0.00025$. For the L-LMS algorithm $\gamma = 0.001$. For the FRS-LMS algorithm, $\omega_b = \pi/4$, $w_1 = 100$, $w_2 = 0.002$. Fig. 4.10 shows that the FRS-LMS converges to a smaller MSE ($\mathrm{MSE}_{\mathrm{FRS-LMS}} = 0.15$), the LMS and L-LMS algorithms performs almost the same, because for a very small μ and γ the L-LMS performance becomes very similar to the LMS in white noise environments ($\mathrm{MSE} = 0.4$), and the ML-LMS fails to converge. Divergence of the ML-LMS algorithm may be due to the large value of noise variance; any small change in the first element in the weigh vector gives a relatively large change in the estimated leakage matrix ($\hat{\mathbf{A}}(k)$). In BER rate performance, Fig. 4.11 shows that the FRS-LMS is approximately 1.5dB better than the standard LMS and L-LMS algorithms, and 4.8dB better than the ML-LMS algorithm when BER=10^{-2}.

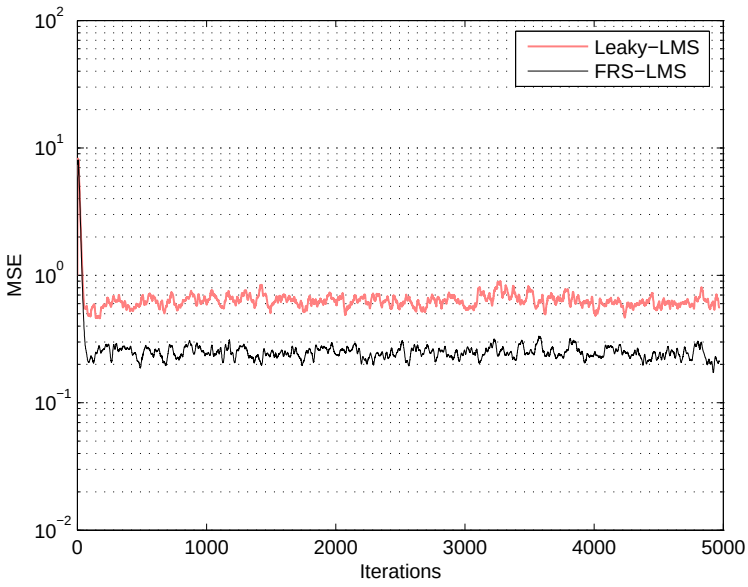

Figure 4.12. The ensemble MSE for FRS-LMS and L-LMS in correlated impulsive noise, $\epsilon = 0.2$, $\kappa = 100$, $N = 32$, SNR= 5dB, $\omega_s = \pi/3$, $\mu = 0.00025$. L-LMS: $\gamma = 0.001$. FRS-LMS: $\omega_b = \pi/6$, $w_1 = 20$, $w_2 = 0.001$.

4.2.2.2. Correlated Impulsive Noise. **A.** In order to study the effects of correlated impulsive noise on convergence, a correlated impulsive noise process is generated with $\epsilon = 0.2$, $\kappa = 100$. The FRS-LMS algorithm has the parameters $\mu = 0.00025$, $\omega_b = \pi/6$ and $w_1 = 100$, $w_2 = 0.001$, SNR= 5dB and the L-LMS algorithm has $\mu = 0.00025$, $\gamma = 0.001$. Fig. 4.12 shows that for the same convergence speed, the FRS-LMS converges to a lower MSE (MSE=0.24) than the L-LMS (MSE=0.62). The FRS-LMS is able to reduce the effects of the outliers and the correlated noise by shaping the frequency response. When we compare Fig. 4.9 (the white impulsive noise MSE curves) and Fig. 4.12, it can be observed that due to the correlated noise (in the presence of outliers) the MSE increases approximately 1.25 and 1.5 times for FRS-LMS and L-LMS respectively.

In the Gaussian mixture noise model for impulsive noise, the SNR is assumed to be constant, and increasing the impulsive component strength κ or the outlier frequency ϵ

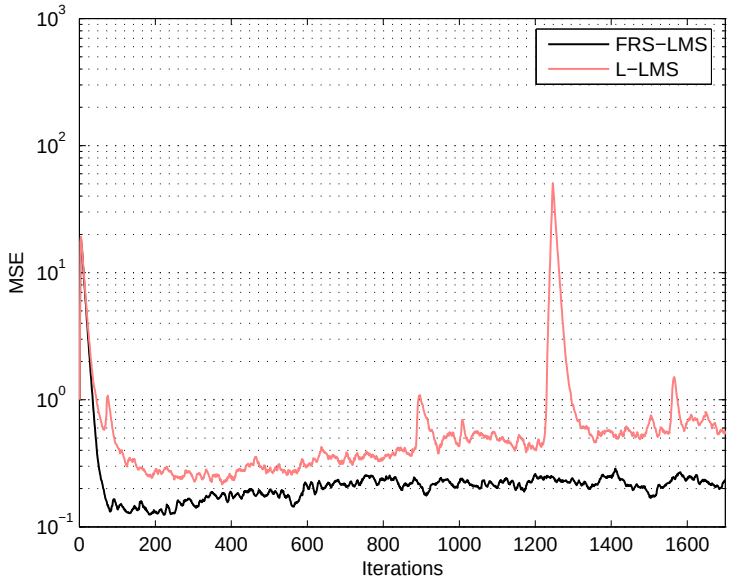

Figure 4.13. The ensemble MSE for FRS-LMS, L-LMS in additive correlated impulsive noise, $N = 33$, SNR$= 5$dB, $\mu = 0.00025$. L-LMS: $\gamma = 0.001$. FRS-LMS: $\omega_b = \pi/6$, $w_1 = 100$, $w_2 = 0.001$.

leads to an increase in the total noise power. An alternative approach [20], is to keep the total noise variance constant in order to study the effects of the variation in the shape of the noise distribution. In other words, when κ is increased for fixed ϵ, the nominal noise variance has to be decreased and vice versa. When Fig. 4.8 and Fig. 4.6 are compared, it is noticed that in the same correlated noise, the effect of constrained impulsive noise is very small. This is due to the fact that in constrained impulsive noise case, increasing the probability (ϵ) of the impulsive components will lead to a corresponding decrease in the background noise.

B. For this part, it is assumed that a BPSK modulated signal $x(k)$ corrupted by additive correlated impulsive noise. A correlated impulsive noise process is generated by using the first-order autoregressive model (AR(1)) $v(k+1) = \rho v(k) + v_0(k)$, where $v_0(k)$ is a white impulsive noise process with $\epsilon = 0.2$, $\kappa = 100$, and ρ is the correlation parameter ($\rho =$

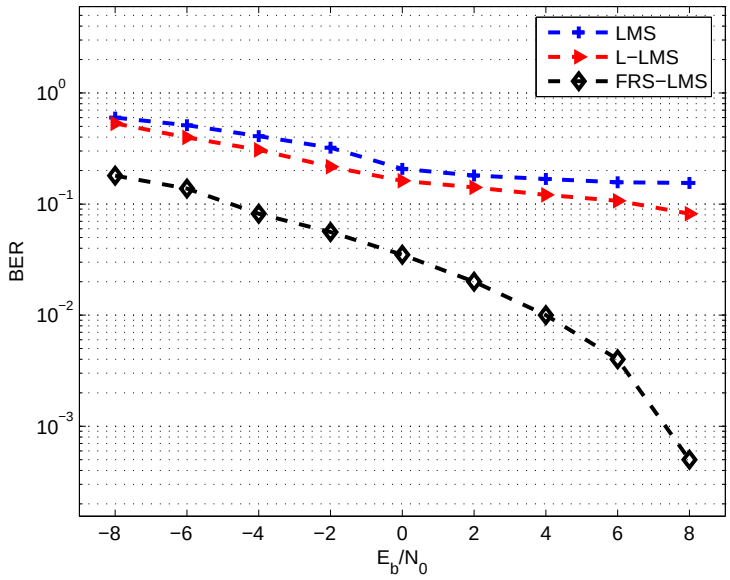

Figure 4.14. BER Performance for LMS, L-LMS, and FRS-LMS in additive correlated impulsive noise, $N = 33$, $\mu = 0.00025$. L-LMS: $\gamma = 0.001$. FRS-LMS: $\omega_b = \pi/6$, $w_1 = 100$, $w_2 = 0.001$.

0.6). Simulations were done for the LMS, L-LMS, ML-LMS, and FRS-LMS algorithms with step-size $\mu = 0.00025$. For the L-LMS algorithm $\gamma = 0.001$. For the FRS-LMS algorithm, $\omega_b = \pi/6$, $w_1 = 100$, $w_2 = 0.002$. In terms of MSE, the FRS-LMS converges where the other algorithms completely fail to converge as shown in Fig. 4.13. This failure of convergence is due to the impulses (outliers) that cause large adaptation errors at each iteration which in turn lead to a larger step-size; this large step-size contributes to larger MSEs in estimating the tap weights. Fig. 4.14 shows the superior BER performance of the FRS-LMS algorithm (FRS-LMS algorithm is approximately 11dB better than the L-LMS algorithm when BER=10^{-1}). Here, it should be noted that LMS and L-LMS algorithms do not converge in this channel with these parameters.

4.3. Mobile Communication Channel Model

In mobile communications, channel estimation is a major area of research. Due to different obstructions in the channel, the received signal will be faded as a result of vectorially summed shifted versions of the transmitted signal as mentioned in the previous chapter. The general model of the received signal can be expressed as (4.1), where $s(k)$ is a wide-sense stationary channel parameter and $n(k)$ is a zero-mean additive noise [33]. Estimating $s(k)$ from $x(k)$ is an important problem, with diverse applications. Channel estimation using adaptive filtering is one of the available methods, but has some convergence and complexity problems [1]. Fixed low pass filtering [29] has deteriorations when the cutoff frequency of the low pass filter is different from the maximum Doppler frequency. Wiener and Kalman filtering [25] [26] techniques require high implementation complexity. The ML-LMS algorithm proposed in [8], is simple and nearly blind (it requires only the knowledge of noise variance) with performance close to that of optimal Wiener filtering. However, the algorithm does not converge in fading channels with correlated noise. The proposed FRS-LMS algorithm does not require training sequences or the knowledge of the noise variance but requires an estimate of the signal frequency which can be recovered from its periodogram [12], as mentioned in the previous section. The algorithm has been shown to have fast convergence, even in fading channels with correlated noise.

In this thesis, an adaptive channel estimation method using the FRS-LMS algorithm is proposed. The performances of FRS-LMS, the ML-LMS and the standard LMS algorithms in estimating channel parameters are compared. Finally, the superior BER performance of the FRS-LMS algorithm in fading channels with additive noise is shown for a DS-CDMA system.

4.3.1. The Communication Channel

The following are assumed in the channel model used in the simulations [30]: The channel has an LOS path, and two Non-Line-of-Sight (NLOS) paths, there is no effect on the instantaneous phase of the transmitted signal, and the magnitudes of attenuation parts are assumed to be constant over a symbol interval. Based on the above assumptions, the received signal can be expressed as:

$$s(k) = \sqrt{P_0}x(k) + \sqrt{P_1}r_1x(k) + \sqrt{P_2}r_2x(k-\tau), \tag{4.2}$$

where P_0, P_1 and P_2 represent the relative power levels of the three multipath components. r_1 and r_2 represent the independent Rayleigh random variables, which in turn represent the attenuation of the NLOS paths. τ is the relative delay between the two NLOS paths. By taking the Fourier Transform of (4.2) we obtain,

$$S(f) = \sqrt{P_0}X(f) + \sqrt{P_1}r_1X(f) + \sqrt{P_2}r_2X(f)e^{-j2\pi f\tau} \tag{4.3}$$

From (4.2) and (4.3)

$$H(f) = \sqrt{P_0} + \sqrt{P_1}r_1 + \sqrt{P_2}r_2\exp(-j2\pi f\tau), \tag{4.4}$$

where $H(f)$ is the transfer function of the channel.

If the product $f\tau$ is relatively large over the frequency range of the signal, then the signal is frequency selective; otherwise, it is flat. τ is expressed in terms of the sampling period.

The parameters for WCDMA are listed in Table 4.1 [34] and these were used in the simulations.

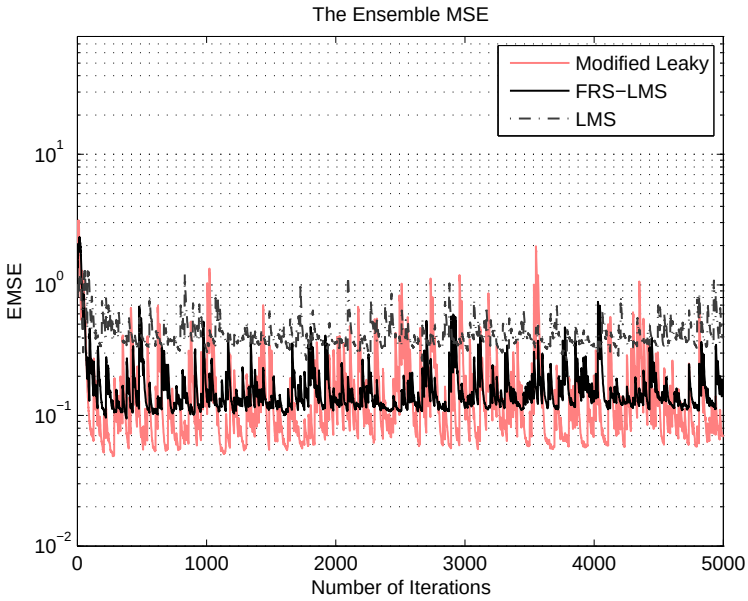

Figure 4.15. The ensemble MSE for LMS, ML-LMS, and FRS-LMS in AWGN channels, $N = 5$, $f_d = 185$Hz, SNR$= 5$dB, $\mu = 0.0025$. For FRS-LMS: $\omega_b = \pi/4$, $w_1 = 20$, $w_2 = 0.002$.

Table 4.1. WCDMA Parameters

Modulation	QPSK
Carrier Frequency	2.0 GHz
Channel Chip Rate	3.84 Mcps
Channel Symbol Rate	15 ksps
Mobile Speed	100 Km/hr

4.3.2. Input Signal in Fading Channel with AWGN

In this experiment, a complex additive white gaussian noise process is generated and added to the received signal $s(k)$. $s(k)$ was created by using QPSK modulation over a fading mobile channel with a Doppler frequency $f_d = 185$Hz, which corresponds to a mobile speed of 100 km/hr. Simulations were done using the parameters, $\mu = 0.0025$, $N = 5$ taps, and SNR=5dB. For FRS-LMS: $\omega_b = \pi/4$ and $w_1 = 20$, $w_2 = 0.002$. Fig. 4.15 shows the performances of the ML-LMS and the FRS-LMS and the standard LMS algorithms under the same conditions. It can be observed from the figure that both FRS-LMS and ML-LMS approximately have a similar MSE performance (0.1) whereas, the standard LMS algorithm has a higher MSE (0.4). Fig. 4.16 shows that the FRS-LMS is approximately 5dB better than the standard LMS and 1.8dB better than the ML-LMS when BER=10^{-1}. It should be noted that even though the FRS-LMS and the ML-LMS algorithms seem to converge approximately to the same MSE, the FRS-LMS has a higher BER performance. This is because of the small difference in the measured steady-state MSE values ($\mathrm{MSE}_{\mathrm{FRS-LMS}} = 0.1021$ and $\mathrm{MSE}_{\mathrm{ML-LMS}} = 0.1342$ for FRS-LMS and ML-LMS respectively).

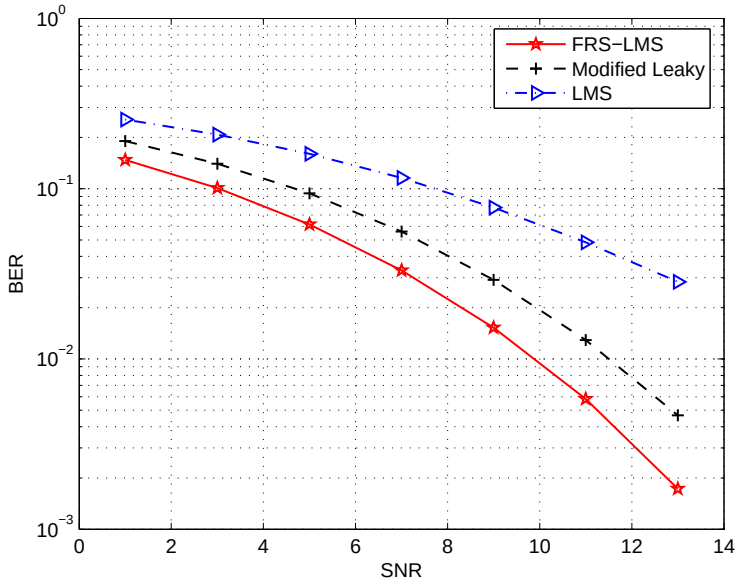

Figure 4.16. BER Performance Comparison for LMS, ML-LMS, and FRS-LMS in AWGN channels. $N = 5$, $f_d = 185$Hz, SNR$= 5$dB, $\mu = 0.0025$. For FRS-LMS: $\omega_b = \pi/4$, $w_1 = 20$, $w_2 = 0.002$.

4.3.3. Input Signal in Fading Channel with Correlated Gaussian Noise

In this experiment, a correlated additive Gaussian noise process is added to the received signal, $s(k)$. The correlated noise was generated by filtering a complex additive white Gaussian noise process with an FIR filter having the magnitude response characteristic shown in Fig. 4.5. $s(k)$ was created in the same way as in the previous part. Simulations were done using the parameters, $\mu = 0.0025$, $N = 5$ taps, and SNR=5dB. The FRS-LMS algorithm is simulated using the same parameters described in the previous section with window size $\omega_b = \pi/2$ and $w_1 = 20$, $w_2 = 0.002$. The ML-LMS algorithm is also simulated under the same conditions. The results in Fig. 4.17, show that FRS-LMS converges to an MSE=0.3 whereas the ML-LMS fails to converge having a small steady state drift. In terms of BER performance, Fig. 4.18 shows that the FRS-LMS is approximately 4dB better than the ML-LMS when BER=10^{-1}.

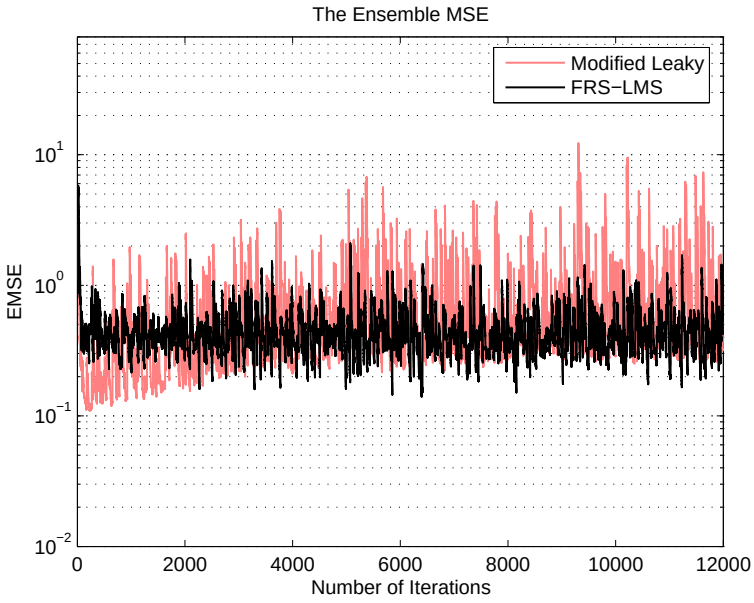

Figure 4.17. The ensemble MSE for FRS-LMS and ML-LMS in correlated Gaussian noise channels, $N = 5$, $f_d = 185$Hz, SNR$= 5$dB, $\mu = 0.0025$. For FRS-LMS: $\omega_b = \pi/2$, $w_1 = 20$, $w_2 = 0.002$.

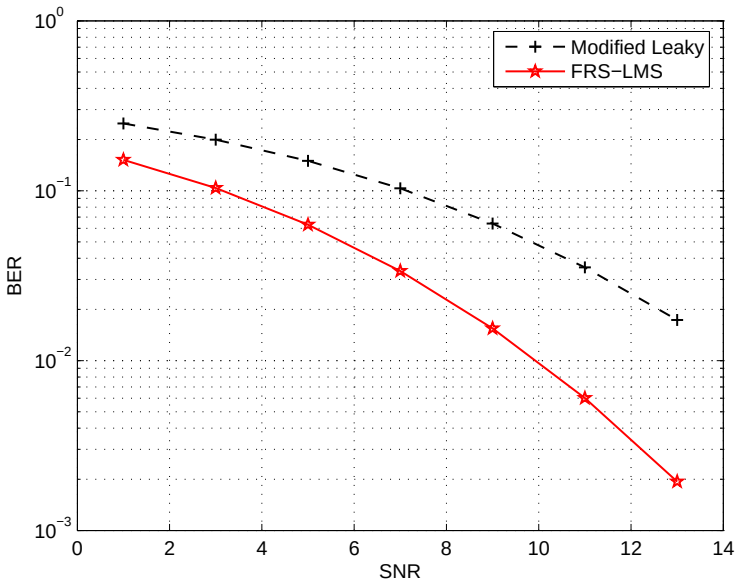

Figure 4.18. BER Performance Comparison for FRS-LMS and ML-LMS in correlated Gaussian noise channels. $N = 5$, $f_d = 185$Hz, SNR$= 5$dB, $\mu = 0.0025$. For FRS-LMS: $\omega_b = \pi/2$, $w_1 = 20$, $w_2 = 0.002$.

CHAPTER 5

CONCLUSION AND FUTURE WORK

In this thesis, a comparison, in terms of Mean Square Error (MSE) performance, among the standard Least Mean Square (LMS), Normalized LMS (NLMS), Leaky LMS, Modified Leaky LMS (ML-LMS) and Frequency Response Shaped Least Mean Square (FRS-LMS) has been investigated in Additive White Gaussian Noise (AWGN) and correlated Gaussian noise environments. Simulation results show that the FRS-LMS outperforms the other algorithms in terms of MSE or speed of convergence.

The performance of the FRS-LMS adaptive algorithm in estimating a sinusoidal signal in white and correlated impulsive noise channels is investigated. The FRS-LMS can be considered as a generalized version of the Leaky-LMS algorithm and has a similar computational complexity. By shaping the frequency response and reducing the effects of outliers the FRS-LMS algorithm shows robust performance in impulsive and correlated noise. The results indicate that the FRS-LMS provides superior MSE and BER performance in white and correlated impulsive noise environments.

An adaptive channel estimation algorithm using FRS-LMS is proposed and its performance was compared to the Modified Leaky LMS and the standard LMS algorithms. The proposed algorithm does not require training sequences or the knowledge of the noise variance. In fading channels with AWGN, it is shown that FRS-LMS and Modified Leaky LMS algorithms have similar MSE performance which in turn is better than

that of the standard LMS algorithm. In fading channels with correlated additive Gaussian noise, the FRS-LMS converges while the Modified Leaky LMS fails to converge. In terms of BER performance, for a fading channel with AWGN, the FRS-LMS had a small performance gain over the Modified Leaky LMS. On the other hand, in fading channels with correlated additive Gaussian noise, the FRS-LMS algorithm performed significantly better than the Modified Leaky LMS algorithm.

As a future work, a statistical analysis (based on the probability distribution function of the noise) of the FRS-LMS algorithm could be done. In its current form, the FRS-LMS algorithm's computational complexity is comparable with the Leaky LMS algorithm. Finding a way to decrease its computational complexity further in order to make it comparable with the standard LMS algorithm would lead to significant application areas.

REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, Upper Saddle River, NJ, 2002.

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

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

[4] G. Tummarello, F. Nardini and F. Piazza, "Stepsize Control In NLMS Acoustic Echo Cancellation Using A Neural Network Approach", *International Symposium on Circuits and Systems*, Vol. 5, pp. 705-708, May 2003.

[5] M. Kamenetsky and B. Widrow, "A Variable Leaky LMS Adaptive Algorithm", *IEEE Conference Signals, Systems and Computers*, Vol. 1, pp. 125-128, November 2004.

[6] J. M. Cioffi. "Limited-precision Effects In Adaptive Filtering", *IEEE Transactions on Circuits and Systems*, Vol. 7, pp. 821-833, 1987.

[7] P. M. Clarkson, *Optimal and Adaptive Signal Processing*, CRC Press, Inc, 1993.

[8] J. Sungkwon, C. Jihoon and H.L. Yong, "Modified leaky LMS algorithm for channel estimation in DS-CDMA systems", *IEEE Communications Letters*, Vol. 6, No. 5, pp. 202-204, May 2002.

[9] D. R. Morgan, M. M. Sondhi, J. Benesty, T. Gansler, and S.L. Gay, *Advances in Network and Acoustic Echo Cancelation*, Springer, 2001.

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

[11] F. Sang and H.G. Yeh, "The Use of Transform Domain LMS Algorithm to Adaptive Equalization", *IECON on Industrial Electronics, Control, and Instrumentation*, Vol.3, pp. 2061-2064, November 1993.

[12] O. Kukrer and A. Hocanin, "Frequency-response-shaped LMS adaptive filter", *Elsevier Digital Signal Processing*, Vol. 16, pp. 855-869, August 2006.

[13] S. Haykin, *Communication Systems*, John Wiley and Sons, Singapore, 1994.

[14] S. S. Pillai and M. Harisankar, "Simulated performance of a DS spread spectrum system in impulsive atmospheric noise", *IEEE Transactions Electromagnetic Compat.*, Vol. 29, pp. 8082, 1987.

[15] M. Bouvet and S. C. Schwartz, "Comparison of adaptive and robust receivers for signal detection in ambient underwater noise", *IEEE Transactions Acoustic, Speech, Signal Processing*, Vol. 37, pp. 621626, 1989.

[16] M. Shao and C. L. Nikias, "Signal processing with fractional lower order moments: Stable processes and their applications", *Proc. IEEE*, Vol. 81, pp. 9861009, July 1993.

[17] J. Haring and A. J. Vinck, "Coding and Signal Space Diversity for a Class of Fading and Impulsive Noise Channels", *IEEE Transactions on Information Theory*, Vol. 50, pp. 887-895, May 2004.

[18] B. Seyfe and S. Valaee, "A New Choice of Penalty Function for Robust Multiuser Detection Based on M-Estimation," *IEEE Transactions on Communications Theory*, vol. 53, pp. 224- 227 , February 2005.

[19] X. Wang and H. V. Poor, “Robust multi-user detection in non-Gaussian channels”, *IEEE Transactions on Signal Processing*, Vol. 47, No. 2, pp. 289-305, February 1999.

[20] H. Deliç and A. Hocanın, “Robust detection in DS/CDMA”, *IEEE Transactions on Vehicular Technology*, Vol. 51, No. 1, January 2002.

[21] T. S. Rappaport, *Wireless Communications*, Upper Saddle River, Prentice Hall, 1996.

[22] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, New York, Cambridge University Press, 2005.

[23] B. Sklar, “Rayleigh fading channels in mobile digital communication systems part II: Mitigation”, *IEEE Communications Magazine*, Vol. 35, No. 7, pp. 102-109, July 1997.

[24] A. F. Molisch, *Wireless Communications*, Chichester, John Wiley and Sons, 2006.

[25] A. Aghamohammadi, H. Meyr and G. Ascheid, “Adaptive synchronization and channel parameter estimation using an extended Kalman filter”, *IEEE Transactions on Communications*, Vol. 37, pp. 1212-1218, November 1989.

[26] J. K. Cavers, “An analysis of pilot symbol assisted modulation for Rayleigh fading channels”, *IEEE Transactions on Vehicular Technology*, Vol. 40, No. 5, pp. 686-693, November 1991.

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

[28] K. Mayyas and T. Aboulnasr, “Leaky LMS algorithm: MSE analysis for Gaussian data”, *IEEE Transactions on Signal Processing*, Vol. 45, No. 4, pp. 927-934, April 1997.

[29] M. L. Moher and J. Lodge, "TCMP-A modulation and coding strategy for Rician fading channels", *IEEE Journal on Selected Areas in Communications*, Vol. 7, pp. 1347-1355, December 1989.

[30] W. H. Tranter, K. S. Shanmugan, T. S. Rappaport and K. L. Kosbar, *Principles of Communication Systems Simulation with Wireless Applications*, Prentice Hall, 2004.

[31] D. Starer and A. Nehorai, "Newton algorithms for conditional and unconditional maximum likelihood estimation of the parameters of exponential signals in noise", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 40, No. 6, pp. 15281534, 1992.

[32] M. S. Ahmad, A. Hocanin and O. Kukrer, "Performance of the frequency-response-shaped LMS algorithm in impulsive noise", *IEEE International Conference on Signal Processing and Communications*, 2007, (under review).

[33] M. S. Ahmad, A. Hocanin and O. Kukrer, "Adaptive channel estimation using frequency-response-shaped LMS algorithm in DS-CDMA systems", *IEEE Global Communications Conference*, 2007, (under review).

[34] *Physical channels and mapping of transport channels auto physical channels (FDD)*, 2000. 3G TS-25.211 version 3.20.