Recursive Inverse Adaptive Filtering for Fading Communication Channels

Faris Yousef Shokri Jaar

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
July 2019
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ali Hakan Ulusoy
Acting Director

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

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

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

Prof. Dr. Aykut Hocanin
Supervisor

Examining Committee

1. Prof. Dr. Aykut Hocanin
2. Prof. Dr. Osman Kükrer
3. Assoc. Prof. Dr. Mehmet Toycan

ABSTRACT

Adaptive filtering is an important and rapidly developing discipline in the signal processing and communications field. Many different algorithms were developed throughout the years. LMS and RLS being the most well-known and studied of the algorithms. Like in many disciplines in engineering there exists many disadvantages in these algorithms and so variations were proposed. The main parameters sought after in a new algorithm were convergence speed, stability, and computational complexity. LMS suffers in the former as it has generally low convergence speed. Meanwhile, RLS main disadvantage is the computational complexity it introduces, in addition to being unstable in varying channel settings. RI adaptive filtering algorithm proposed in 2009 provides some solutions to the aforementioned issues.

In this thesis, the use of the RI algorithm in common communication channel setting is simulated. The performance of the algorithm is studied and compared to some other adaptive filtering algorithms in terms of BER. Both channel estimation and channel equalization performances are investigated and commented upon. The conclusions reached will be that RI outperforms RLS in terms of convergence speed and computational complexity in all cases studied while giving similar results to NLMS in frequency selective channels but outperforming it in flat fading channels.

Keywords: Adaptive filter, recursive inverse, RLS, communication channel

ÖZ

Uyumlu filtreleme sinyal işleme ve iletişim alanının önemli ve gelişmekte olan bir disiplinidir. Alanın gelişme sürecinde farklı uyumlu filtre algoritmaları geliştirilmiştir. Least Mean Squares (LMS) ve Recursive Least Squares (RLS) algoritmaları en fazla bilinen ve başarımı araştırılan algoritmaların başında gelmektedir. Geliştirilen algoritmaların olumlu ve olumsuz yönleri bulunmakta ve araştırmacılar uygulamaya özge farklı geliştirmeler önermektedirler. Bu kapsamda en önemli başarım parametreleri yakınsama hızı, kararlılık ve hesaplama karmaşıklığıdır. LMS genel olarak düşük yakınsama hızına sahiptir. RLS ise, yüksek hesap karmaşıklığına ve kararlılık sorunlarına sahip olmakla birlikte, hızlı ve ilintili gürültü ortamlarında bile çalışabilen bir algoritmadır. 2009 yılında önerilen Recursive Inverse (RI) uyarlanır algoritması, yukarıda bahsedilen olumsuz yanların gelişmesine katkı koymaktadır.

Tezde, RI algoritmasının iletişim kanallarında başarımı benzetimler yoluyla incelenmiştir ve farklı güncel algoritmalarla Bit Hata Oranı (BER) açısından karşılaştırılmıştır. Kanal kestirimi ve kanal dengeleme başarımları incelenerek ulaşılan BER değişimleri gözlemlenmiştir. RI algoritmasının, RLS, LMS ve Normalized Least Mean Squares (NLMS) algoritmalarından yakınsama hızı ve hesaplama karmaşıklığı açısından avantajları farklı ortamlarda belirlenmiş ve sunulmuştur.

Anahtar Kelimeler: Uyumlu filtre, sönümlemeli haberleşme kanalı, kanal dengeleme

ACKNOWLEDGMENT

Taking this opportunity, I would like to thank my supervisor for his guidance to make sure this work turns out in the best shape possible.

Of course, I cannot forget to mention my mother Samia, my father Yousef, my brother Fadi and Tipsy for their continuous support and the confidence they instill in me.

I would also like to extend my gratitude to the Department of Electrical and Electronic Engineering at Eastern Mediterranean University represented by the chair of the department Prof. Dr. Hasan Demirel.

Finally, my sincere thanks goes to the jury members of my thesis defense for their constructive feedback of my work on this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

IIR	Infinite impulse response
FIR	Finite impulse response
RI	Recursive inverse
RLS	Recursive-least-squares
NLMS	Normalized least-mean-square
BER	Bit error rate
LMS	Least-mean-square
VSSLMS	Variable step size LMS
DCTLMS	Discrete cosine transform LMS
TDVSS	Transform domain LMS with variable step-size
RRLS	Robust RLS
SFTRLS	Stabilized fast transversal RLS
MSE	mean-square-error
$\mathbf{w}$	Tap weights vector
$\mathbf{R}$	Autocorrelation matrix
$\mathbf{p}$	Cross-correlation vector
k	Step size
β	Forgetting factor
$\mathbf{x}$	Tap input vector
$\mathbf{I}$	Identity matrix
AWGN	Additive white Gaussian noise
SNR	Signal-to-noise ratio

Chapter 1

INTRODUCTION

1.1 Introduction

One of the well-established topics in digital signal processing is adaptive filtering. A field that had many researchers and innovation in over the years [1]. Adaptive filters are more suited for uses when the statistics of the data is not known in advance as they provide the ability to adapt to changes in the impulse response, hence the prospect of employability in many applications in digital signal processing and communication systems.

Adaptive filtering comprises of two basic processes; generating an out signal after filtering some input data, and secondly the process of adaptation. The adaptation process aims to minimize a specific cost function by adjusting the filter coefficients. The way these processes are done may vary as the algorithms change.

Adaptive filtering can and has been implemented in both infinite impulse response (IIR) [2]; filters that are characterized by the fact that the impulse response never truly settles to zero but may decay, and finite impulse response (FIR) [2]; filters that have a finite impulse response defining it. The focus of this thesis will be solely on FIR filters.

1.2 Problem Statement

The fact that the common communication channel model is that of an ever-changing and varying one requires the adaptive filter to be robust enough to be able to adapt fast and with as little errors as possible. RI algorithm [3] which will be discussed later, provides a better alternative than other adaptive filters in such an environment.

1.3 Thesis Objective and Contribution

The main objective of this thesis is to apply the RI algorithm to such a channel, with the aim of testing and gauging the performance of RI compared to RLS [1] and NLMS in terms of channel estimation and equalization when it comes to the calculated BER. Some example cases will be laid out in which RI is the better option compared to the other algorithms due to its ability to better adapt to variations in the channel.

1.4 Thesis Overview

The thesis is organized in the following way, after the brief introduction in this chapter, Chapter 2 provides some overview of the various adaptive filtering algorithms and their shortcomings. Chapter 3 lays out the Recursive Inverse algorithms that will be under the subject of assessment, and the various components implemented to obtain the simulation results. Chapter 4 presents the simulation results and invites discussion upon them. Finally, Chapter 5 concludes the thesis with final observations and the possibility of future work.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

The two algorithms that had the biggest effect and popularity are least mean square (LMS) [1] and recursive least squares (RLS) [1] algorithms. LMS algorithm is one of the earliest and most well-known adaptive algorithms and was used in many applications [4-6]. To outperform the different LMS algorithms, a different algorithm was developed which was the recursive least squares (RLS) algorithm. RLS algorithm employs an estimated autocorrelation matrix (and its inverse) and relies on recursion in updating the estimates. Many algorithms have been proposed that try to increase efficiency and enhance the performance of adaptive filtering.

2.2 LMS

LMS algorithm has the advantage of its simplicity both software and hardware-wise with fast convergence if the step size is tuned right [1]. Normalized Least Square Algorithm (NLMS) was proposed [7-10] to offer lower gradient noise amplification in the case of large input values. This fixed step size utilized in the previous algorithms turns into a disadvantage and to circumvent it, variable step size LMS variants were proposed like VSSLMS [11], still these algorithms fair badly with highly correlated input signals and in additive impulsive noise. Transform domain variants of LMS like the discrete cosine transform LMS (DCTLMS) [12] and the transform domain LMS with variable step-size (TDVSS) [13] and provide good performance in environments with correlated noise, but because of the need by these algorithms for some transformation processes and even information regarding the variance of the noise, they suffer from high computational complexity.

2.3 RLS

To obtain an improved efficiency in comparison to the different LMS variants. Various people developed the recursive-least-squares (RLS) algorithm [1], [14]-[15]. Although the RLS algorithm does succeed in that, it has the disadvantage of suffering from high computational complexity and the possibility of numerical instability due to the use of the inverse autocorrelation matrix, this issue is also shared by the different variants of RLS like the ones discussed below.

Another shortcoming of the algorithm is the tuning required for the forgetting factor and the need for it to be a relatively high value approaching unity so that stability and convergence are guaranteed, this issue leads to a limiting scope of use in applications that involve non-stationary environments [16]-[17].

Some variants of RLS proposed was the robust RLS (RRLS) [18]. Instead of a fixed step size, this algorithm utilized a variable one which leads to better performance in varying environments. One disadvantage of RRLS lies in the fact that it has a very high computational complexity even when compared to RLS.

Another improvement to the algorithm proposed was the stabilized fast traversal RLS (SFTRLS) [19]-[20] were proposed in the efforts to reduce the complexity of the RLS algorithm but the tradeoff was the lower performance that this algorithm provided.

2.4 RI

Recursive Inverse (RI) algorithm [21] was developed in 2009. The RI algorithm was created to remedy some of the challenges and problems mentioned earlier concerning these algorithms. The RI algorithm has the advantage of not updating the inverse autocorrelation matrix unlike the RLS algorithm, this technicality leads to a substantial decrease in computational complexity [3]. Regarding how the RI algorithm performs, the results were of a comparable level compared to RLS in terms of MSE convergence albeit as previously mentioned with a lower complexity especially when using a large number of filter taps. RI also has a much better performance than that of LMS and with significantly higher convergence speed also [3].

Chapter 3

METHODOLOGY

3.1 Introduction

The previously developed Recursive Inverse (RI) algorithm offers a faster convergence compared to LMS and better performance than RLS in a non-stationary environment which is our concern as communication channels are usually modeled to be of varying nature.

This chapter will start by showing the RI algorithm derivation. Following will be the discussion for the test environment and channels to be used in the simulations along with the way the performance comparisons will be carried out to obtain the attached results.

3.2 Derivation of the Recursive Inverse Algorithm

The derivation of the update equation of the RI algorithm and its convergence analysis is detailed in [21], and the derivation will be summarized below.

The FIR filter coefficients optimum solution can be derived from the Wiener-Hopf equation [1], the filter weights can be calculated as

$$\mathbf{w}(k) = \mathbf{R}^{-1}(k)\mathbf{p}(k) \tag{1}$$

The time parameter is represented by k (k = 1,2,3,…), the filter weight vector is represented by $\mathbf{w}(k)$, $\mathbf{w}(k)$ is the vector evaluated at time k, an estimate of the autocorrelation matrix of the input vector is represented by $\mathbf{R}(k)$, whereas a cross-correlation vector between the desired output signal value and the input of the filter taps is estimated in the parameter $\mathbf{p}(k)$. Solving (1) is needed at each time-step as the filter coefficients are updated. Additionally, another restriction imposed by the inverse of the autocorrelation matrix is that inevitability of said matrix should be guaranteed at each time-step.

The correlations in (1) are updated in a recursive manner as follows

$$\mathbf{R}(k) = \beta\mathbf{R}(k-1) + \mathbf{x}(k)\,\mathbf{x}^{\boldsymbol{T}}(k) \tag{2}$$

$$\mathbf{p}(k) = \beta\mathbf{p}(k-1) + d(k)\mathbf{x}(k) \tag{3}$$

a value close to 1 is mostly chosen for the forgetting factor, here given the symbol of β. Substituting the above two equations into the Wiener-Hopf equation produces

$$\mathbf{w}(k) = \{\beta\mathbf{R}(k-1) + \mathbf{x}(k)\,\mathbf{x}^{\boldsymbol{T}}(k)\}^{-1}[\beta\mathbf{p}(k-1) + d(k)\mathbf{x}(k)] \tag{4}$$

rearranging the resultant equation after taking the matrix inversion lemma [22] into use

$$\begin{aligned}\mathbf{w}(k) &= \left[\boldsymbol{I} - \frac{\mathbf{R}^{-1}(k-1)\mathbf{x}(k)\boldsymbol{x}^{T}(k)}{\beta + \mathbf{x}^{T}(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)}\right]\mathbf{w}(\mathrm{k}-1) \\ &\quad + \left[\frac{\mathbf{R}^{-1}(k-1)\mathbf{x}(k)}{\beta + \mathbf{x}^{T}(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)}\right]d(k) \\ &= \mathbf{w}(k-1) + \mu(k)\mathbf{R}^{-1}(k-1)\mathbf{x}(k)e(k)\end{aligned} \tag{5}$$

Newton-LMS algorithm [23]-[27] has the update equation as in equation (5) with the error $e(k)$ equaling:

$$e(k) = d(k) - \mathbf{x}^{T}(k)\mathbf{w}(k-1) \tag{6}$$

Equation below shows the step-size calculation

$$\mu(k) = \frac{1}{\beta + \mathbf{x}^{\boldsymbol{T}}(k)\mathbf{R}^{-\mathbf{1}}(k-1)\mathbf{x}(k)} \tag{7}$$

Taking the inverse of the autocorrelation matrix is required for Newton-LMS. But for RI that is not necessary. The RRLS shares some similarity with Newton-LMS but with the major difference being that the inverse autocorrelation matrix is updated instead of updating the estimated correlation matrices.

The iterative solution of the wiener equation will yield the following equation that will eventually converge to a solution.

$$\mathbf{w}_{n+1}(k) = [\mathbf{I} - \mu\mathbf{R}(k)]\boldsymbol{w}_{\boldsymbol{n}}(k) + \mu\mathbf{p}(k), \qquad n = 0, 1, 2, \ldots \tag{8}$$

if μ is chosen to satisfy the criterion of convergence, then

$$\mu < \frac{2}{\lambda_{\max(\mathbf{R}(k))}} \tag{9}$$

Considering the equation used to update the correlations in (2), and then evaluating the expected value of $\mathbf{R}(k)$ we obtain,

$$\overline{\mathbf{R}}(k+1) = \beta\overline{\mathbf{R}}(k) + \mathbf{R}_{xx}\,, \tag{10}$$

where $\mathbf{R}_{xx} = E\{\mathbf{x}(k)\mathbf{x}^T(k)\}$ and $\overline{\mathbf{R}}(k) = E\{\mathbf{R}(k)\}$ solving equation (10) gives out

$$\overline{\mathbf{R}}(k) = \frac{1-\beta^k}{1-\beta}\mathbf{R}_{xx}\,, \tag{11}$$

As $k \rightarrow \infty$

$$\overline{\mathbf{R}}(\infty) = \frac{1}{1-\beta}\mathbf{R}_{xx}\,. \tag{12}$$

Equation (11) allows for the understanding of the behavior of the eigenvalues of the autocorrelation matrix and Eq. (12) gives the values at the limit. Knowing criterion in (9) must be satisfied at the limit, we get

$$\mu < \frac{2(1-\beta)}{\lambda_{\max}(\mathbf{R}_{xx})} \tag{13}$$

Eq. (13) puts a restriction on μ such that it can only take values much smaller than those permitted by (7) when using $\mathbf{R}_{xx}$ instead of $\mathbf{R}(k)$, thus it is better to use a variable step-size μ as in

$$\mu(k) < \frac{2}{\lambda_{max}(\mathbf{R}(k))} = \left(\frac{1-\beta}{1-\beta^k}\right)\left(\frac{2}{\lambda_{max}(\mathbf{R}_{xx})}\right) = \frac{\mu_{max}}{1-\beta^k} \ , \tag{14}$$

Or

$$\mu(k) = \frac{\mu_0}{1-\beta^k} \quad \text{where } \mu_0 < \ \mu_{max}. \tag{15}$$

The iteration in (8) has the disadvantage of being computationally complex. Using a step-size that is variable reduces the iterations needed at each time step down to 1. Thus, the RI algorithm update equation of the weight values is therefore:

$$\mathbf{w}(k) = [\mathbf{I} - \mu(k)\mathbf{R}(k)]\mathbf{w}(k-1) + \mu(k)\mathbf{p}(k) \tag{16}$$

The RI algorithm has the edge in terms of complexity as it has lower computational complexity compared to RLS type algorithms that require updating the inverse autocorrelation matrix unlike RI as can be seen from Eq. (16).

3.3 Channel and Test Environment

In the following, some key points and characteristics of the test environment that will be used in the simulations along with some channel models to obtain the performance metrics will be discussed.

3.3.1 Rician Multipath Fading Channel

The Rician multipath channel is a communication channel model widely used as a test environment to verify and study the performance of different Algorithms and modulation techniques. This channel will be used in some simulations as will be noted where necessary.

The channel is characterized by having a Rician distributed coefficient for the line of sight component and a Rayleigh distributed coefficients for the remaining delayed paths, these coefficients act as the fading coefficients. Also, the channel introduces uniform phase jitter to the input signal.

3.3.2 Doppler Shift

The Doppler phenomenon is a physical phenomenon caused by the sender or receiver being in motion, moving either in the direction of or opposite of the other. This phenomenon leads to a change in the frequency of the signal transmitted or received leading to the possibility of introducing errors to the signal. The model built to test the performance of the different algorithms takes Doppler shift into consideration because of the nature of the common commination channel of containing movement in some of its components. Ignoring the Doppler phenomenon will lead to a static channel whereas including it leads to a non-stationary impulse response of the channel as it contributes to different cases of multipath interference due to the presence of the frequency shift and thus the varying impulse response.

3.3.3 Stationary Channel

For the sake of testing the convergence performance, some simulations will be made over a stationary channel model. Such a channel [28] is laid out in Eq. 17 below

$$h(n) = \left\{ \begin{array}{ll} \left(\frac{1}{2}\right)\left[1 + \cos\left(\left(\frac{2\pi}{3.3}\right)(n-2)\right)\right], & n = 1,2,3 \\ 0\,, & otherwise \end{array} \right\} \quad (17)$$

3.3.4 Channel Equalization Block Diagram

The general block diagram of the channel equalization system which will be implemented can be seen below. The input signal goes through the channel, the channel used will be stated where necessary in the simulation results chapter later on. After the signal passes the channel, it is corrupted by noise mainly AWGN. The result of this operation then passes the adaptive traversal equalizer and the difference between the output of said filter and a delayed version of the original input is used as error feedback to the many adaptive filtering algorithms used.

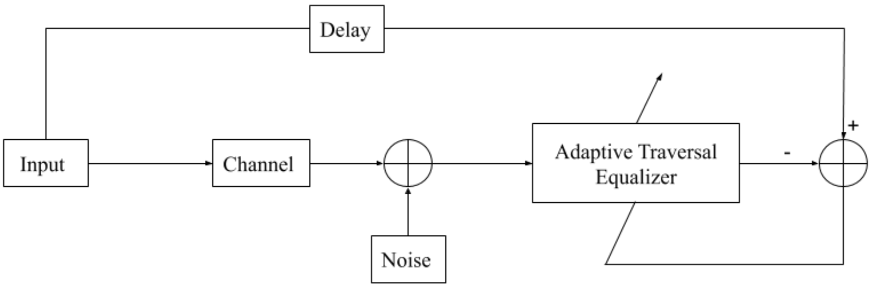

Figure 1: Channel equalization block diagram

3.3.5 AWGN

After the signal passes through the implemented channel, it is corrupted by AWGN which stands for Additive White Gaussian Noise. Additive because the values that are being introduced are added to the input of such channel. White refers to the fact that it is frequency independent having constant power density for all frequencies. Gaussian means that these values follow a Gaussian distribution. Finally, noise refers to an unwanted component corrupting the desired signal.

Gaussian is the most commonly used noise model in simulation because it is the one that most resembles the real world noise effects on signals. According to the central limit theorem, the effect of many different probability density functions can be approximated by the Gaussian distribution.

Noise introduced to the desired signal usually comes from other signals present in the medium of transmission, like air in wireless communications as well as thermal noise caused by electronic components used in the transmitter and receiver circuits.

3.3.6 Modulation Technique

The modulation technique used throughout the simulations is Binary Phase Shift Keying (BPSK). Some cases use BPSK with 0 initial phase shift while others use BPSK modulation with an initial phase shift of π/4. The use of either case will be noted wherever necessary.

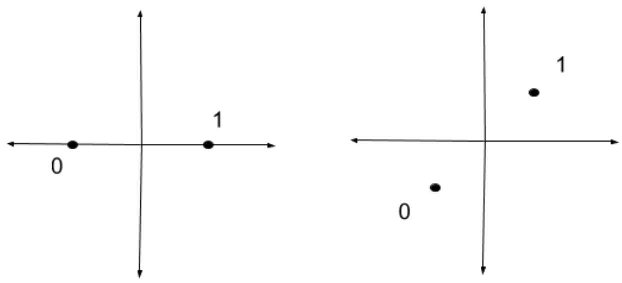

Figure 2: Constellation diagram of BPSK with no initial phase shift (left), and with an initial phase shift of π/4 (right)

3.3.7 Monte Carlo Simulations

Monte Carlo simulations are a form of a statistical analysis tool widely used in many non-engineering fields and also engineering fields, to simulate predictions for systems that contain probabilistic parameters.

Throughout the simulations to follow, it should be noted that the results are obtained using the Monte Carlo simulation by having many runs of the same program averaged out to provide a more accurate representation of the results by limiting the effect of uncertainty.

3.4 Multipath Fading Channels

Given the fact that the varying channel implemented in the simulations is a multipath fading channel, some brief explanation of such channels is to be presented in this section.

There are mainly two ways these channels can be categorized. The two distinctive properties that determine what each channel may be classified are called the coherence bandwidth and the coherence time. Coherence bandwidth may be simply defined as the frequency band within which the different frequency components experience similar levels of attenuation. This coherence bandwidth is inversely proportional to the arrival times of the delayed signals. Coherence time means the time period during which the channel approximately retains similar structure without much change.

In the following, the main classification of channels depending on coherence bandwidth and coherence time will be presented.

3.4.1 Flat Fading Channels

Flat fading channels are characterized by the fact that nearly all of the transmission bandwidth lays within the coherence bandwidth thus leading to different frequency components being affected by nearly equal gain values as was mentioned before.

Such a channel can usually be represented by an impulse response that has one or very few significant values.

Fading channels will be flat if the line of sight component of the channel environment has a significantly higher power that other delayed signals arriving reflected off different elements in the environment. In other words, a flat fading channel is one in which all significant delayed versions of the signal arrive before the arrival of the next bit or symbol. This model can be seen in some cases of Rician multipath fading as this kind of channel includes a line of sight component. Whereas for Rayleigh multipath fading channel, this fading model is less prevalent as Rayleigh channels usually lack the line of sight component.

3.4.2 Frequency Selective Channels

This type of channel has the characteristic of having a transmission bandwidth that is larger than the coherence bandwidth of the channel. This leads to some frequency components of the signal to be affected by different levels of attenuation.

This type of channels may be represented by channel impulse response that has many significant values unlike flat fading.

Frequency selective channels are those in which the delayed signals could arrive after subsequent symbols are received at the receiver leading to distortions to that symbol. Such a configuration may result because of the existence of many reflections caused by the different environmental obstacles.

3.4.3 Fast Fading Channels

The categorization of how fast a channel is depends as previously mentioned on the coherence time of the channel. The channel is generally considered fast if its coherence time is less than the symbol period. Thus, channel variations are faster than baseband signal variations.

The change in the channel characteristics can be observed in the change of the impulse response representing such a channel. The impulse response generally retains similar values during the coherence time.

Fast fading happens when the maximum Doppler shift affecting the channel is off a considerable amount in comparison to the transmission rate of the channel. This happens in real world when there is high amount of mobility in the environment, for example moving cars.

3.4.4 Slow Fading Channels

Flowing the explanation for fast fading channels, a channel is generally said to be slow fading if its coherence time is larger than symbol period, which means the channel retains similar characteristics for a longer period of time than fast fading. Thus, channel variations are slower than baseband signal variations. These slow changes are reflected by the slow changing impulse response representing the channel.

This case of fading usually happens when the maximum Doppler shift affecting the transmission is low valued in comparison to the transmission bandwidth. This can be reflected in real situations when there is not much movement in the transmission environment due to nearly stationary components making it up.

It should be noted that the first two classifications are independent than the latter two, that means that it is possible to obtain a fading channel that is a fast flat fading channel or one that is a slow frequency selective fading channel, and so on.

3.5 Proposed Simulation System

The main concern and aim is to develop a system that tests the performance of different adaptive filtering algorithms in terms of channel equalization in a non-stationary environment setting. These results will be discussed in section 4.2. the following figure shows the block diagram of the implemented system.

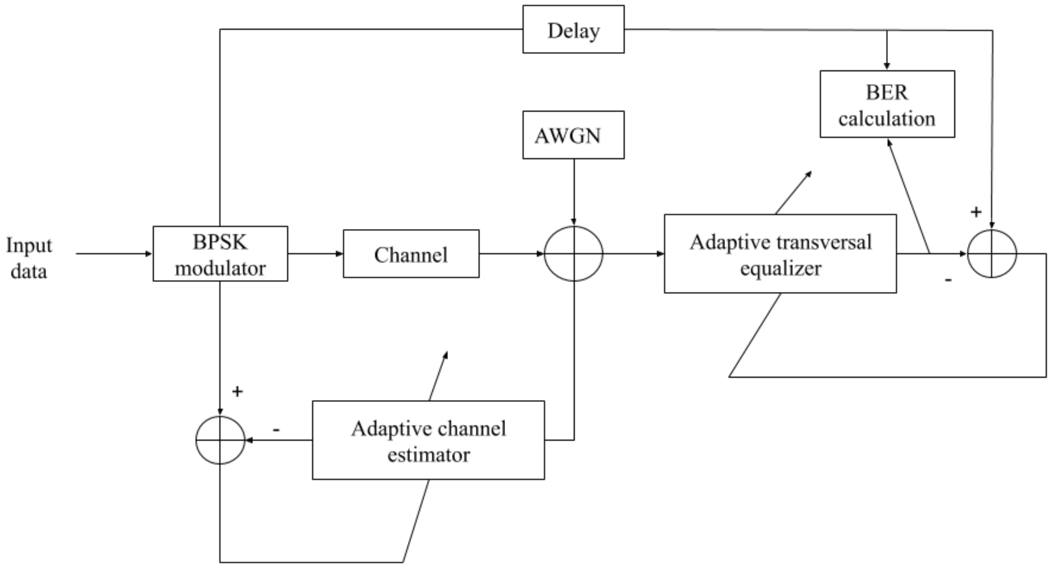

Figure 3: Block diagram of the implemented system

The simulation starts with the generation of input bits represented as having a value of either 0 or 1 randomly. These input bits are then given to a BPSK modulator; the modulator has an initial phase shift of $\pi/4$ for ease of further analysis.

The modulated signal is then processed by the MATLAB object (comm.RicianChannel). To simulate the effect of the signal passing through a Rician multipath fading channel, exact parameters used to prime the object are laid out and discussed later in the next chapter of simulation results.

After that, the resultant signal is simulated as having gone through a AWGN by adding Gaussian generated noise to the signal with the noise having different variance values according to the SNR level given.

The signal then passes to the adaptive equalizer part of the system which changes the weights of the filter such that to try and make the effect of passing through both the multipath fading channel and the equalizer as close to the effect of not passing through either as possible. In addition, the output of both the multipath fading channel and the AWGN also goes through a channel estimator to help give a visualization of the corruptive channel impulse response.

The method in which the adaptive equalizer in adapting is chosen to be using the phase of the original bit signal as the modulation method used is BPSK and the original information of the bit is encompassed in its phase. Furthermore, unlike when doing system identification there is a delay inserted in the case of channel equalization.

Now for the important part of figuring out whether a bit was received after going through all the previous processes was in error or not. Again, since an important part of the signal is its phase, the method used in determining the existence of an error or not is through the phase.

The bit is accepted as being of correct value if the phase of said bit is within a region of 0.5π radians of the original phase. Because of the nature of signal phases, a phase difference of more than 0.5π radians in either direction (positive or negative) is considered to be a difference that will lead to an error.

In the end, the number of errors is added up to find the BER rate of the system for a specific SNR level and then the SNR is changed to different values for the final BER vs SNR plot. Also, it is worth noting that this whole process is done many times to minimize the effect of uncertainties through Monte Carlo simulations.

Chapter 4

SIMULATION RESULTS

This chapter will layout the results of the simulations showcasing the RI algorithm performance in comparison to different adaptive filtering algorithms. Some simulations will be done on a stationary channel to specifically show the minimization of error as time steps increase. Then, a varying channel is tested to show the performance of adaptation of the RI, RLS and NLMS algorithms, by using the system described just briefly before.

4.1 Stationary Channel

For this simulation, X_n is a Bernoulli input sequence of alternating 1 and -1 representing a simplified Binary Phase Shift Keying (BPSK). The corrupting channel is modeled as defined by the impulse response in Eq.16 and represented in figure below. disturbance noise is a zero mean normal random variable with its variance depending on the SNR level defined.

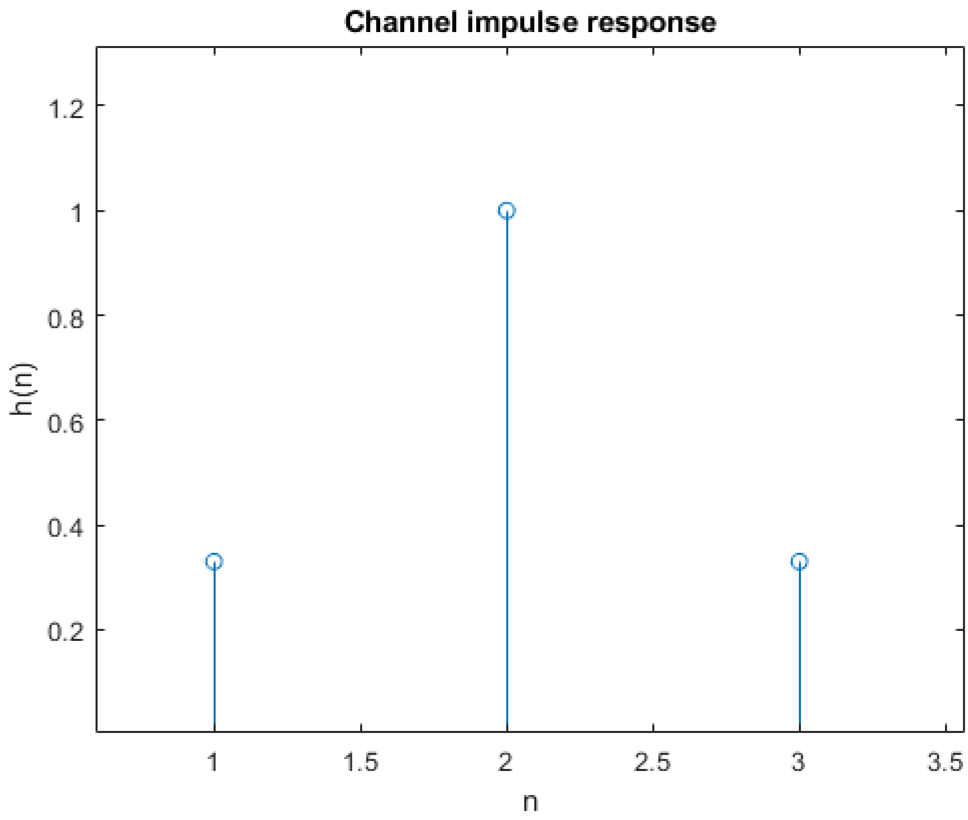

Figure 4: Impulse response for the stationary channel

The simulation was done as follows. First, the input sequence of [1, -1] was convolved with the channel impulse response defined earlier. Then the additive white Gaussian noise was generated according to the desired SNR value and after that, an attempt at reconstructing the input signal is made. The error between the actual value and the calculated one is used in the adaptive algorithm to equalize the channel.

The first thing to look at is the performance of the filter algorithm in reducing the bit errors after it has converged. Figures below show the histogram of signal values in 3 different cases all under AWGN with SNR of 4. Without passing through the corrupting channel (figure 2). After passing through the channel but without being equalized and finally after passing through both the channel and the traversal filter (figure 3).

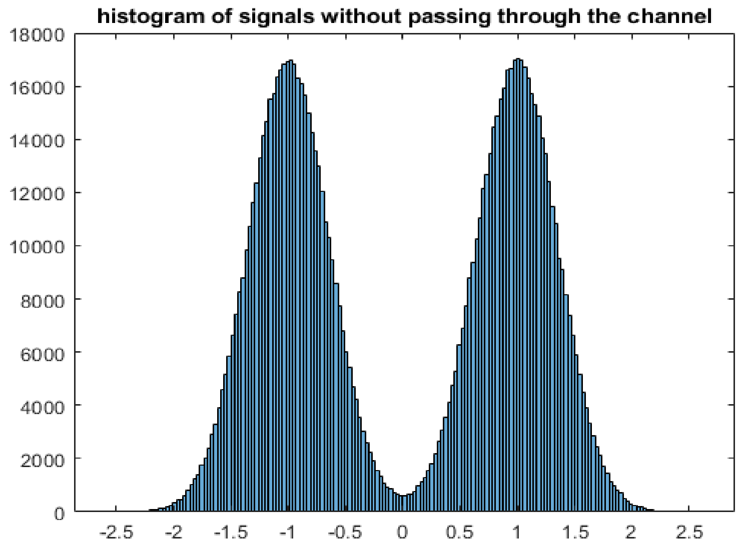

Figure 5: Histogram of the signals without passing through the corruptive channel.

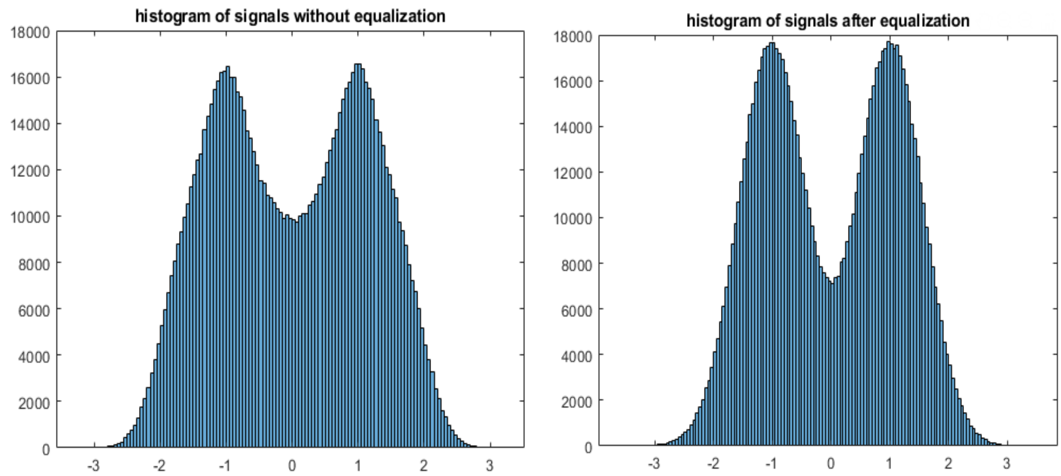

Figure 6: Histogram of signals without channel equalization(left), and with equalization(right)

Previous figures serve the purpose of providing insight into the effect channel equalization would have on the communication channel. As can be seen, the probability of the decision being made in error decreases after the signal is equalized as the inter symbol interference introduced by the corruptive channel is reduced leading to more accurate decisions being made. The difference of interest here is the one between the cases where the corruptive channel is used that shows the decrease in ISI after channel equalization.

Because of the heavy use and importance of channel equalization in the simulations done, the following figure shows the performance of EMSE convergence of different adaptive filtering algorithms for channel equalization, the simulation is done in a stationary channel as described at the begging of section 4.1. Parameters used in the simulation were as follows, delay for all algorithms was set to 7, for RI, β=0.97 and μ_0 = 0.004. Whereas RLS had β=0.991. As for LMS a step size of value 0.0024 was used, a step size equal to 0.03 was used for NLMS in addition to ϵ being equal to 0.01. The additive white Gaussian noise level used was 15dB.

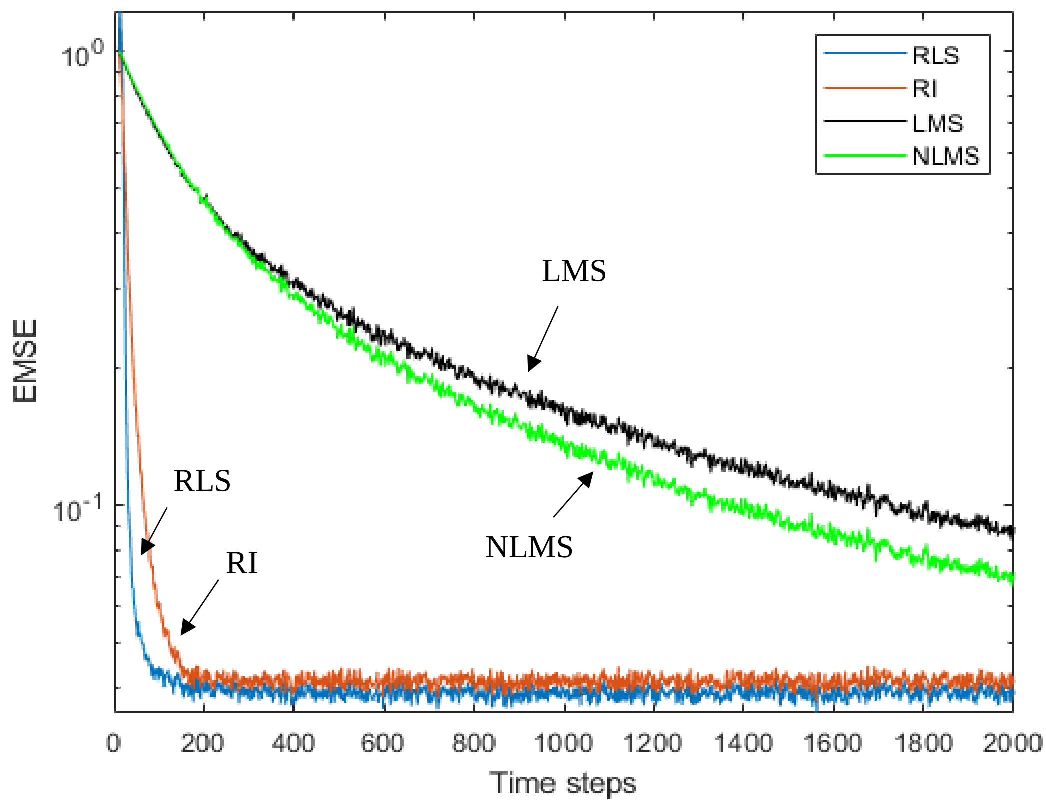

Figure 7: EMSE convergence for RI, RLS, LMS, and NLMS in a stationary channel

The previous figure demonstrates the performance of different algorithms in converging to the minimum mean square error in a channel equalization setting. It can be seen that both RI and RLS converge to a close value of minimum mean square error with a minuscule difference. As for LMS and NLMS, it is obvious that they both would take more time in converging to a steady value of MSE due to their feature of lacking memory of preceding time steps.

The following simulation was done to observe the BER performance of the different algorithms in a stationary environment. The method makes the bit value decision after each time step and sums up the number of errors after the end of the test message. This means that each run only used 1 test sequence with 1 bit checked after each time step. Parameters used in the simulation were as follows, for RI, β=0.97 and μ_0 = 0.004. Whereas RLS had β=0.991. for LMS the step size used was 0.0024 and as for NLMS the step size was set to 0.03 with ϵ=0.01.

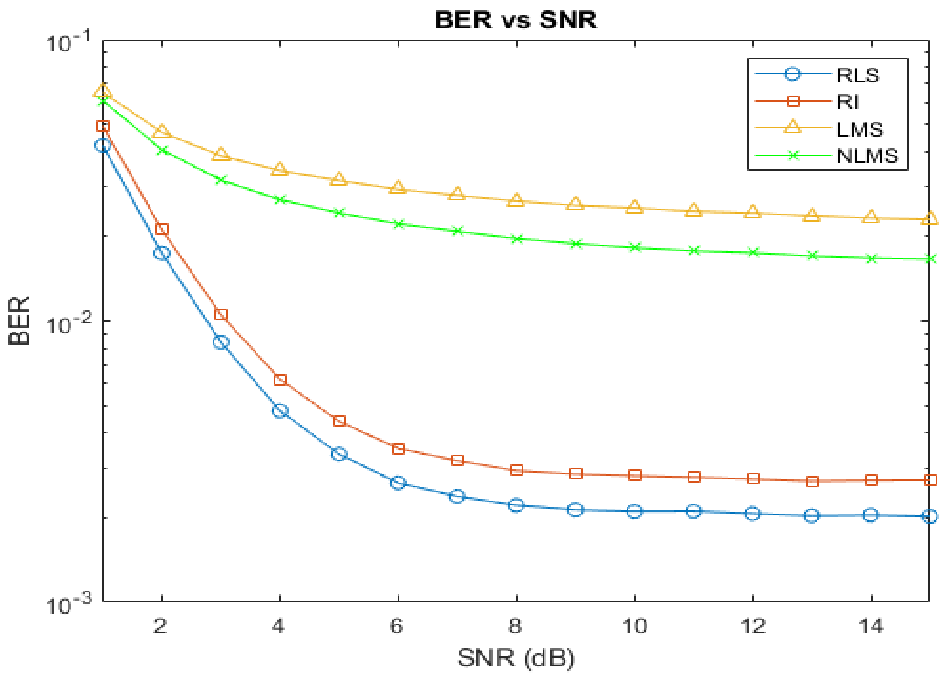

Figure 8: BER vs SNR for RI and RLS in a stationary channel

The figure above shows that RLS is performing better than RI by around 1dB, the reason for this is that as shown previously the RLS algorithm converged faster than RI thus creating this gap in the comparisons. LMS and NLMS, on the other hand, show bad performance which can also be backed up by the figure detailing EMSE convergence previously. The apparent but unexpected convergence of values especially those for RLS and RI is due to the limited number of samples used in the simulation.

This section laid out some performance comparisons between these different adaptive filtering algorithms, but these results are not conclusive as the channel was a stationary one and the samples were limited, such a set up might not be optimal to compare the different algorithms, as it might not truly show the advantages and disadvantages of these algorithms.

4.2 Rician Channel

The following simulations were done to test the performance of RI and RLS and NLMS in a continuously changing channel model represented by the Rician multipath fading channel as was described in section 3.5. Different cases of the calibration of the channel are presented.

The channel equalization model is the one given in figure 1. The parameters for the simulations were β=0.8 and μ_0 = 0.004 for RI. Whereas for RLS, β=0.9999 to guarantee stability and initial autocorrelation matrix diagonals being equal to 100. For NLMS a step size of 0.03 was chosen with ϵ being equal to 0.01. The input signal was a BPSK modulated signal represented by random [1, -1]. The different parameters used for the calibration of the channel were mostly fixed with a variation being applied only in the input data sample rate to produce the various cases to be discussed. For reproducibility, the random stream used in the MATLAB object was "*mt19937ar with seed 10*".

A good way to illustrate and understand the performance differences would be to look at the ability of the algorithms in terms of estimating the channel presented so those figures will also be made available.

The first case done was with an input data sample rate of 500KHz, delay vector of [0 , 2 , 4]x 10^{-6} s, multipath gain vector of [0 , -6 , -9] dB for the paths considered, and maximum Doppler shift of 3Hz that is necessary to produce the varying nature of the channel and a Rician distribution k factor of 10. This configuration produces a slow fading frequency selective channel. The following figures show a few snapshots of the frequency repose of this channel, BER vs SNR for RLS, RI and LMS, and a look at the estimate produced by each algorithm to help understand the difference in performance.

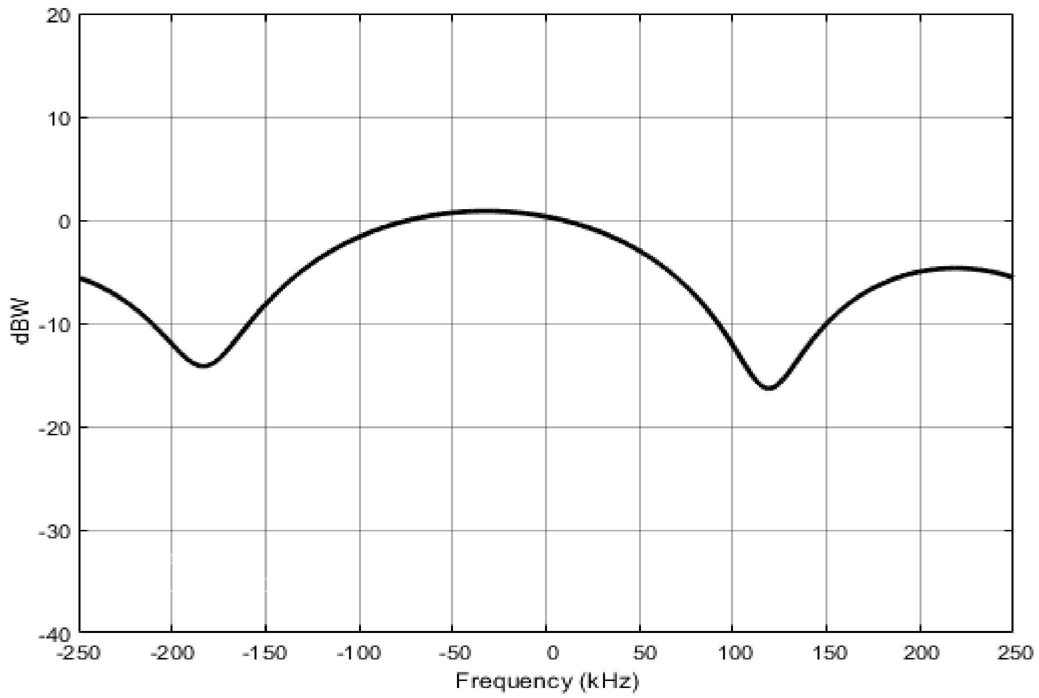

Figure 9: Frequency response of the channel with a sampling rate of 500KHz at sample time 500

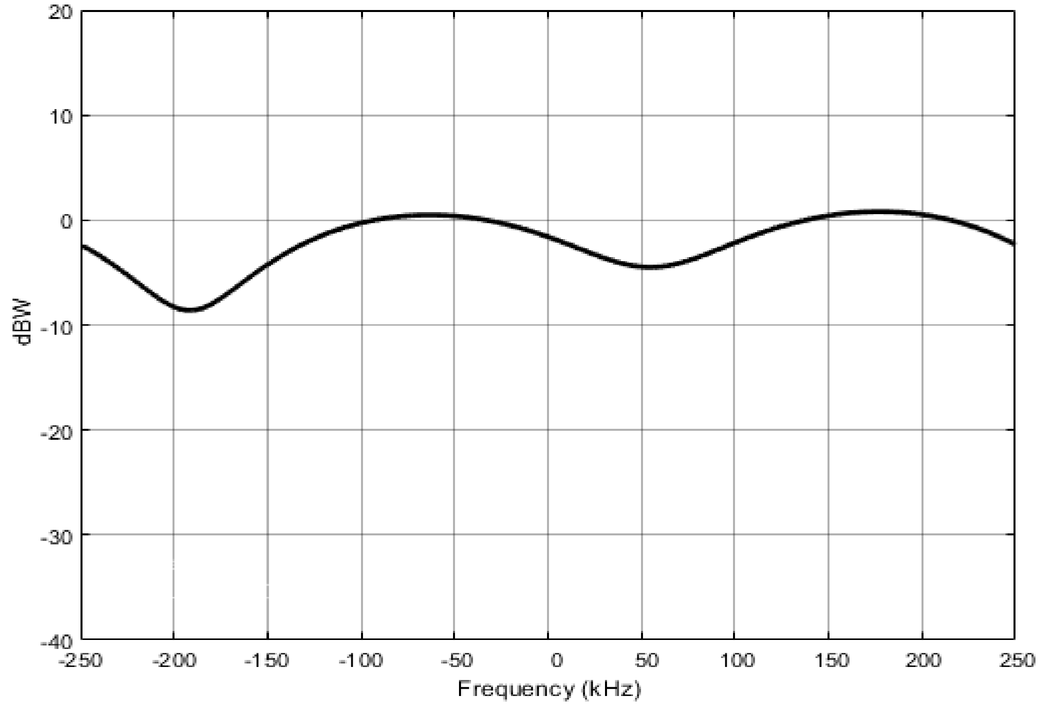

Figure 10: Frequency response of the channel with a sampling rate of 500KHz at sample time 6000

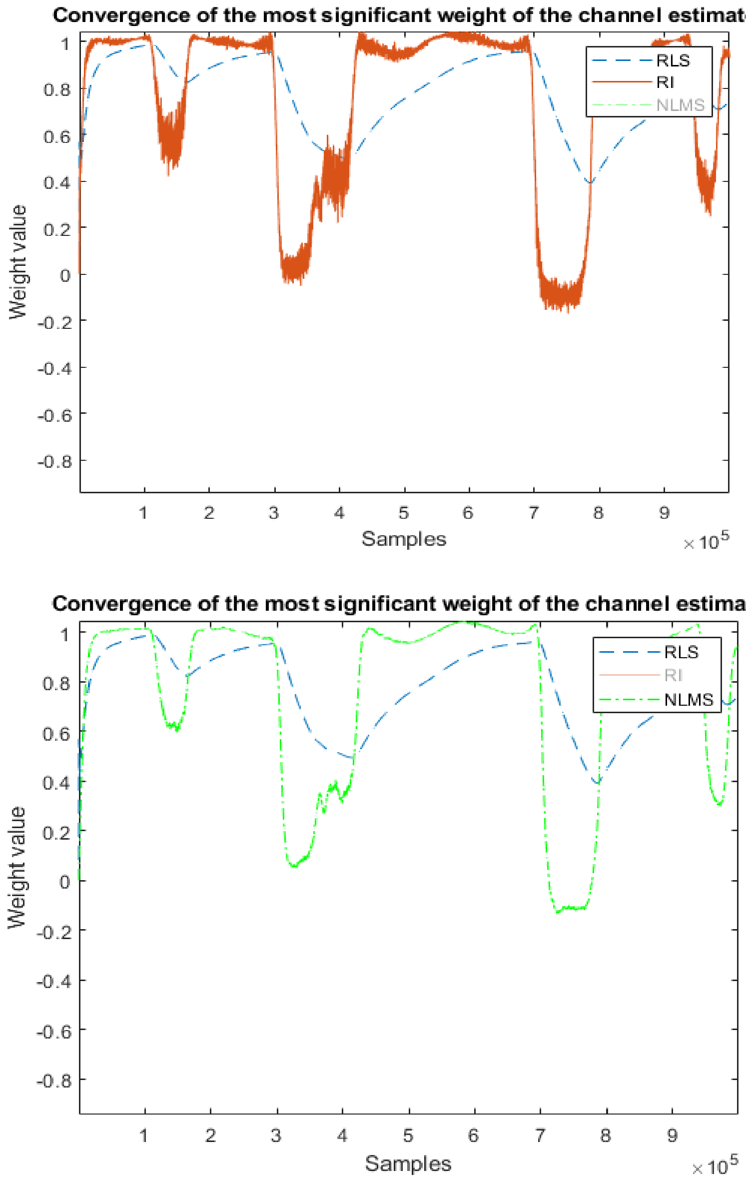

Figure 11: Convergence of most significant filter weight for a sampling rate of 500KHz, RLS and RI on top, RLS and NLMS on bottom

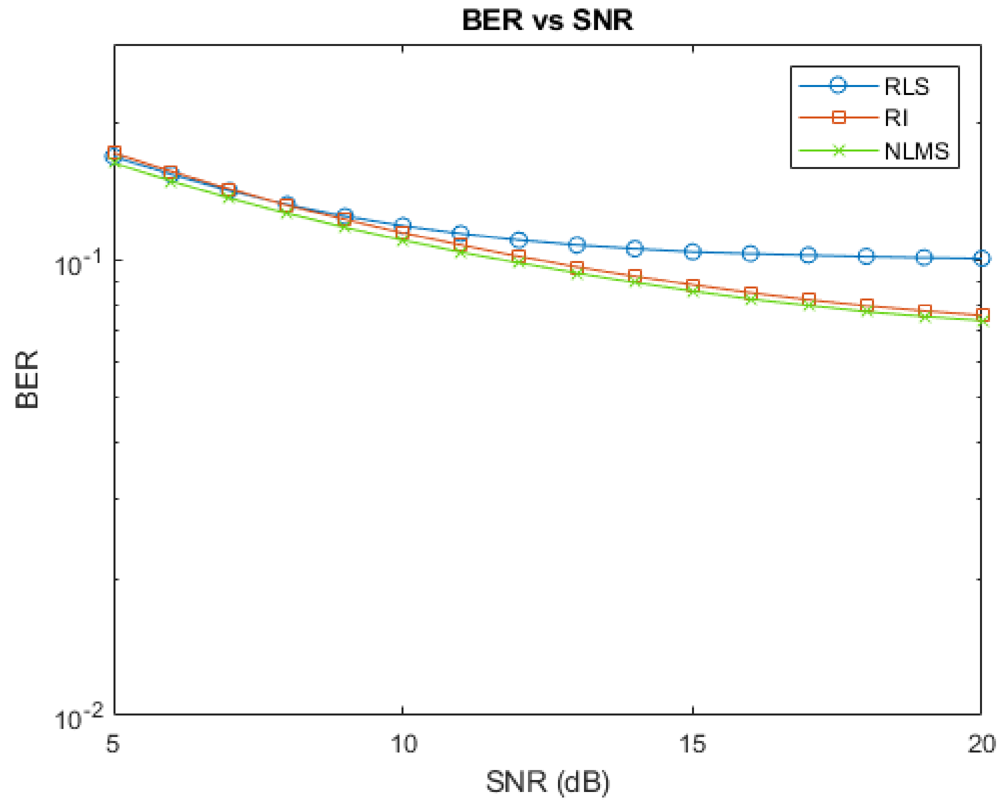

Figure 12: BER vs SNR for RI, RLS, and NLMS for a sampling rate of 500KHz

Previous two figures show the faster convergence of RI in this frequency selective environment and the better performance of RI over RLS into terms of BER especially at higher SNR levels while at the same time being less complex. Whereas RI shows a very similar performance to NLMS at nearly all SNR levels shown.

The similarity in results between all 3 algorithms can be attributed to the fact that the impulse response of the channel is a slow fading one but is producing frequency selective characteristics so that the speed of convergence of RI does not play a big of a role so it does not hold a big advantage.

A second case that had frequency selective characteristics but with the weights changing faster than in the first case was done with an input data sample rate of 50KHz, delay vector of [0, 2, 4]x 10^{-6} s, multipath gain vector of [0, -6, -9] dB, and maximum Doppler shift of 3Hz to produce the variation in the channel, and a Rician distribution k factor of 10. This model represents a less frequency selective channel than in the first case but with faster fading. The following figures show BER vs SNR for RLS, RI and LMS, and a look at the estimate produced by each algorithm to help understand the difference in performance.

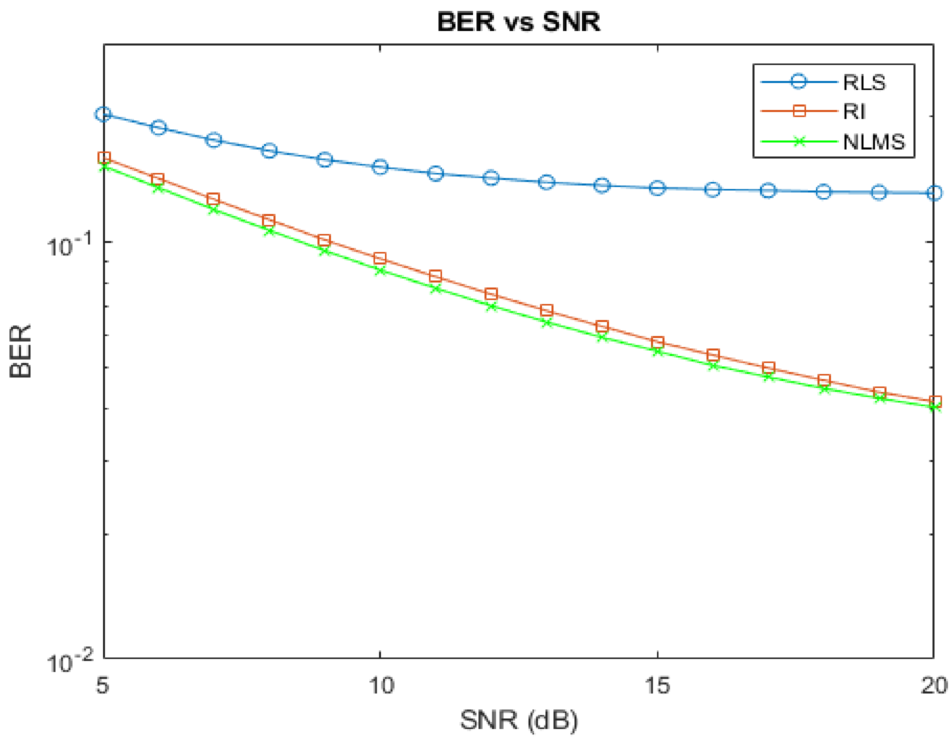

Figure 13: BER vs SNR for RI and RLS for a sampling rate of 50KH

It can be noticed from previous figure that now the distinction in performance between RI algorithm on one hand, and RLS on the other is now more prevalent due to the fact that the impulse response of the channel is now more rapidly changing and the faster and better adaptability of the RI algorithm plays a more important role in setting the algorithm up to produce fewer bit errors, whereas compared to NLMS, RI is still producing similar results at nearly all SNR levels shown mainly due to the fact that it lacks memory of previous values. The difference reaches around 7dB between RI and RLS at the highest SNR level tested here of 20dB.

The third case was conducted with an input data sample rate of 20KHz, delay vector of [0, 2, 4]x 10^{-6} s, a multipath gain vector of [0, -6, -9] dB, and maximum Doppler shift of 3Hz, and a Rician distribution k factor of 10. These settings produce a flat fading channel that has fast variations as will be seen. Below figure shows a representation of this channel at a random instance of time.

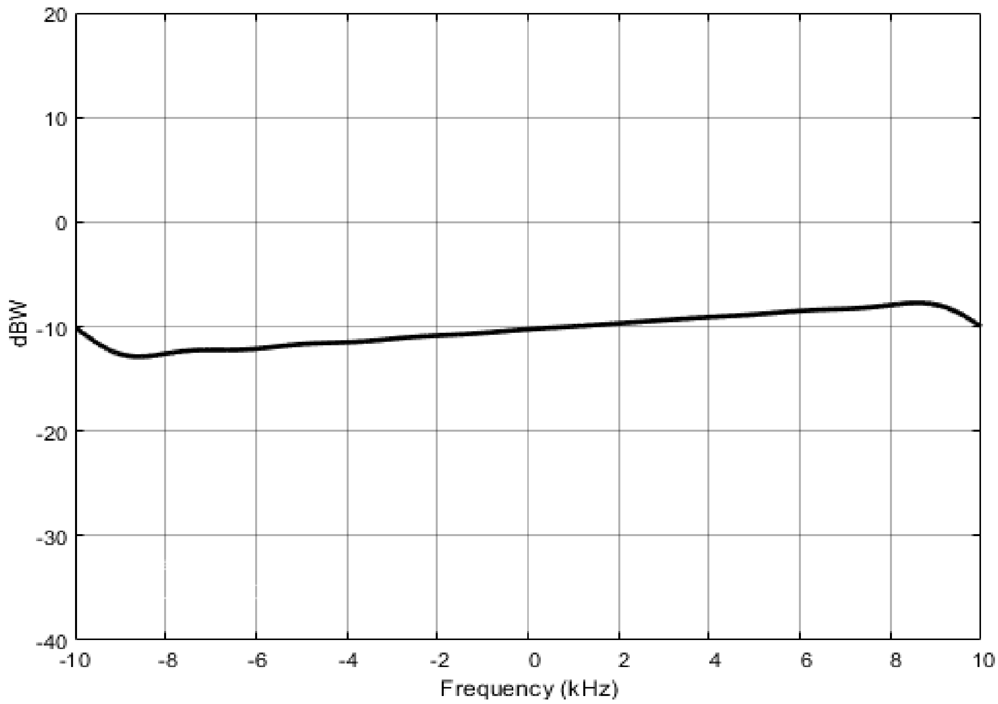

Figure 14: Frequency response of the channel with a sampling rate of 20KHz at sampling time 100000

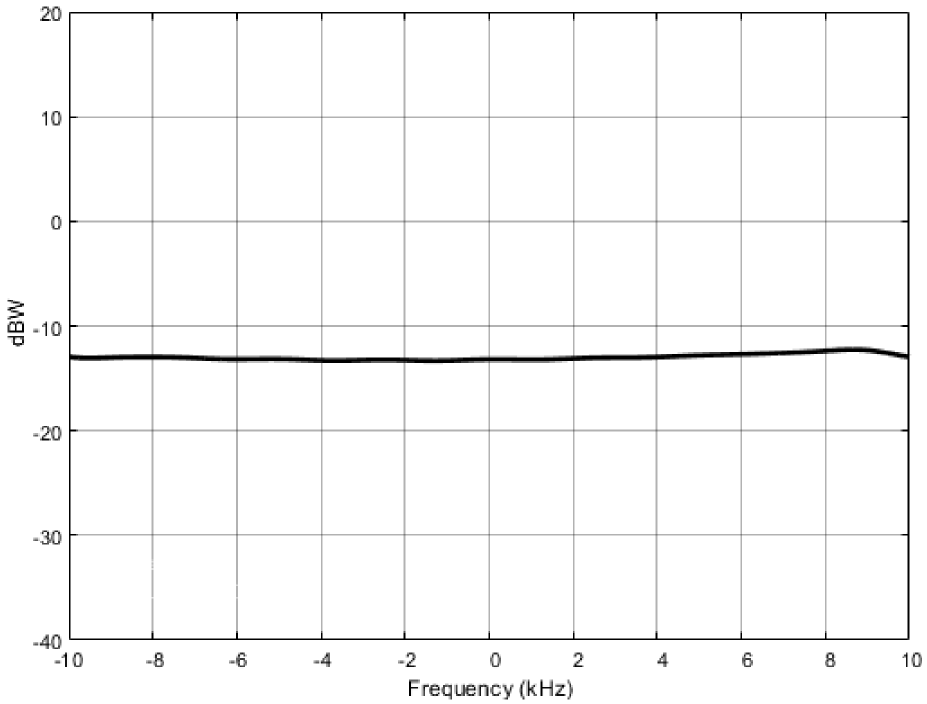

Figure 15: Frequency response of the channel with a sampling rate of 20KHz at sampling time 6300

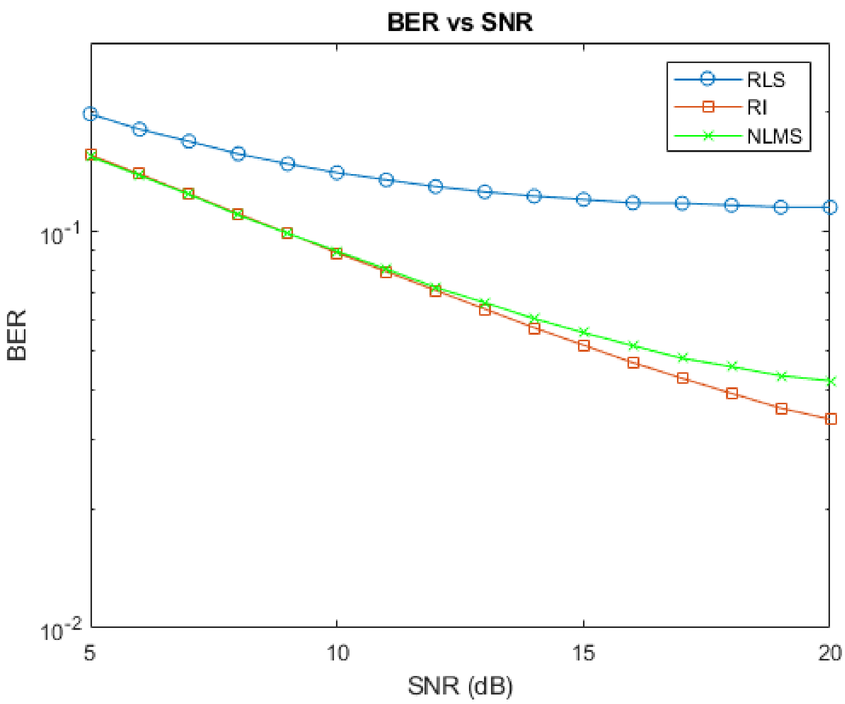

Figure 16: Convergence of most significant filter weight for a sampling rate of 20KHz

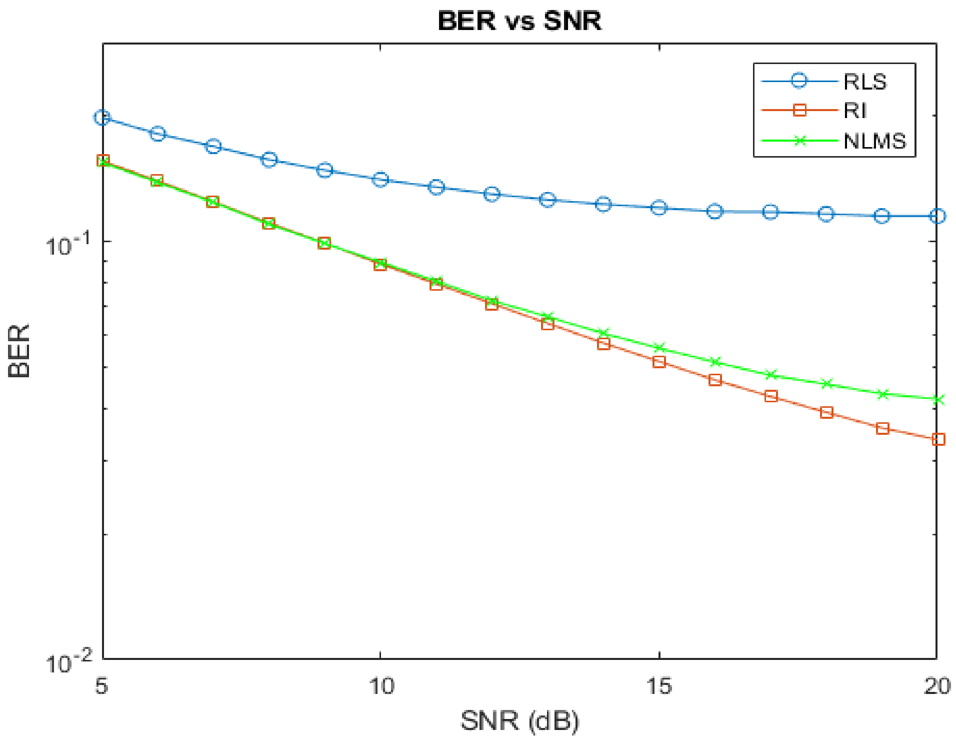

Figure 17: BER vs SNR for RI. RLS and NLMS for a sampling rate of 20KHz

The figures for the case parented above shows how RI outperforms RLS because of its ability to adapt to changes faster as the channel presented is one of fast variations but no frequency selectiveness, whereas RLS shows bad performance in that regard as the channel changes before the algorithm even has the chance to adapt due to the nature of the algorithm of requiring very high forgetting factor which leads to worse adaptation to variation. NLMS, on the other hand, is still showing good performance compared to RI with the RI having a slight advantage at higher levels of SNR.

The improvement in BER performance of RI over RLS and NLMS increases with the incrimination of SNR levels to reach approximately 7dB and 1dB respectively at the highest SNR level tested.

To further understand the performance of RI in the system constructed over other adaptive filtering algorithms like RLS and NLMS, a case is presented where the data sampling frequency is chosen to be 10KHz with the other parameters being same as in preceding cases. Following are figures of the convergence of the most significant weight value and BER vs SNR.

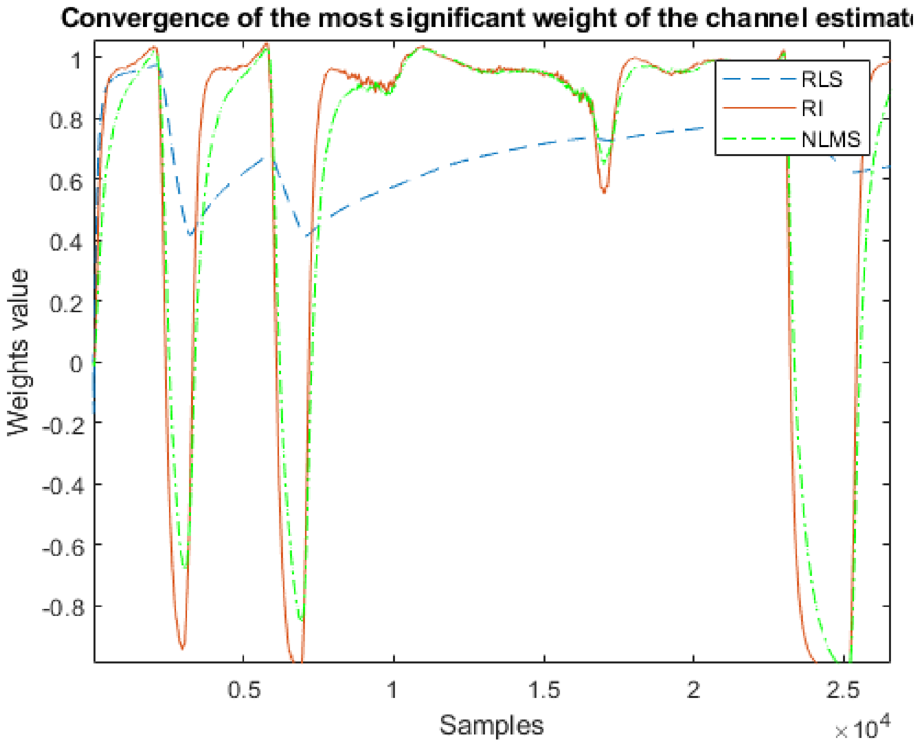

Figure 18: Convergence of most significant filter weight for a sampling rate of 10KHz

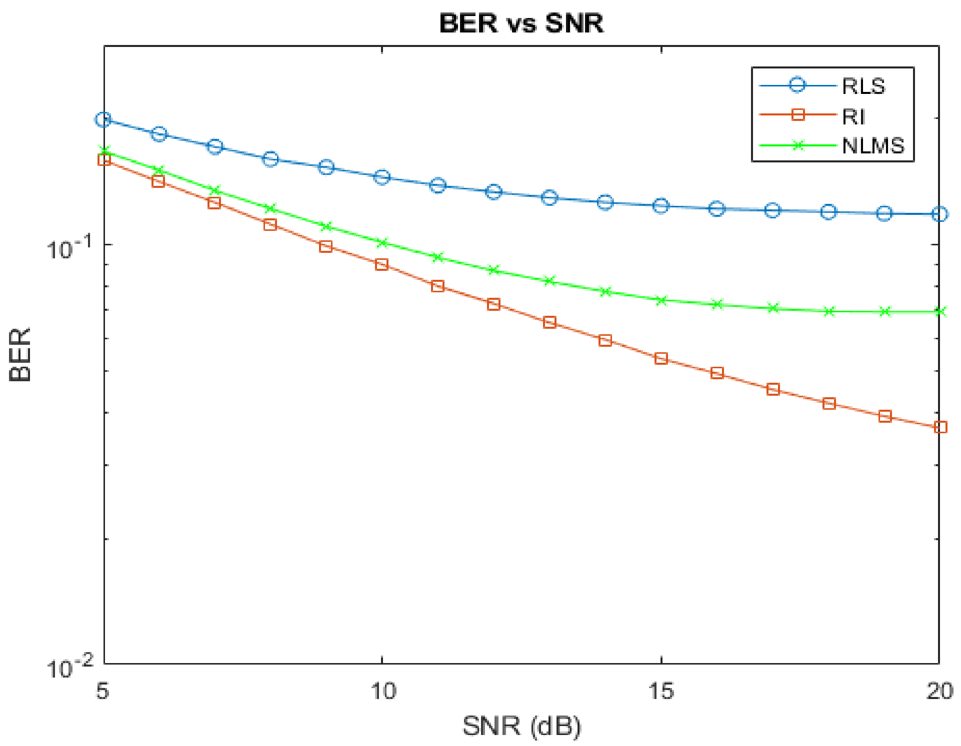

Figure 19: BER vs SNR for RI. RLS and NLMS for a sampling rate of 10KHz

As can be seen in previous two figures, this case that presents a fast-changing flat fading channel shows a better performance of RI over both RLS and NLMS, especially at higher SNR levels.

The improvement in BER performance of RI over RLS and NLMS increases with the incrimination of SNR levels to reach approximately 6dB and 3dB respectively at the highest SNR level tested.

As can be inferred from the figures the RI algorithm adapts to the change in the channel impulse response with greater speed and accuracy compared to the RLS algorithm which shows an abysmal performance in this regard, often not converging before another change occurs in the channel and environment.

In the case of NLMS, the results show us that both it and RI deliver an approximately equal performance when it comes down to the issue of bit errors in the cases where the impulse response of the channel changes at a slow rate (slow fading). The RI algorithm shines in the regard of bit errors compared to NLMS whenever the channel has a fast variation in the impulse response (fast fading) as can be seen in the last two cases due to its superior ability in adapting to these changes.

Chapter 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

In this thesis, a simulation system for testing the SNR vs BER performance of different adaptive filtering algorithms was presented.

Results presented compared the performance of RI against other algorithms mainly RLS and NLMS in a Rician multipath fading channel. Simulations were done in various channel settings including fast flat fading channel and slow fading frequency selective channel model.

It was noticed that in the slow fading frequency selective cases presented RI performed better than RLS and had similar results to NLMS. The real advantage that RI had over the other algorithms was in the fast flat fading cases, as the results illustrated that RI was clearly the better algorithm in adapting to the channel variations.

5.2 Future Work

Future work to be implemented to better understand and appreciate the performance of the RI algorithm would be to simulate the same system in other noise models such as ACGN, AWIN, and ACIN.

In addition, the system could be further improved by implementing a different way of channel equalization, namely decision feedback equalizer (DFE) which would in principle improve the results of the equalization process.

REFERENCES

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

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

[3] M. S. Ahmad, O. Kukrer and A. Hocanin, "Recursive inverse adaptive filtering algorithm," *Elsevier Digital Signal Processing*, vol. 21, no. 4, 2011, pp. 491-496.

[4] C. V. Sinn and J. Gotze, "Comparative study of techniques to compute FIR filter weights in adaptive channel equalization," *IEEE International Conference on Acoustics, Speech and Signal Processing*(ICASSP03), vol. 6, April 2003, pp.217-220.

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

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

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

[8] A. E. Albert and L. S. Gardner, *Stochastic Approximation and Nonlinear Regression*, MIT Press, Cambridge, MA, 1967.

[9] R. R. Bitmead and B. D. O. Anderson, "Lyapunov techniques for the exponential stability of linear difference equations with random coefficients," *IEEE Transactions on Automation and Control*, vol. AC-25, 1980, pp. 782-787.

[10] R. R. Bitmead and B. D. O. Anderson, "Performance of adaptive estimation algorithms in independent random environments," *IEEE Transactions on Automation and Control*, vol. AC-25, 1980, pp. 788-794.

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

[12] D. I. Kim and P. De Wilde, "Performance analysis of the DCT-LMS adaptive filtering algorithm," Signal Processing, vol. 80, no. 8, August 2000, pp. 1629-1654.

[13] R. C. Bilcu, P. Kuosmanen and K. Egiazarian, "A Transform domain LMS adaptive filter with variable step-size," *IEEE Signal Processing Letters*, vol. 9, no. 2, February 2002, pp. 51-53.

[14] R. Hastings-James, R. and M. W. Sage, "Recursive generalised-least-squares procedure for online identification of process parameters," *Proceedings of the Institution of Electrical Engineers*, vol. 116, no. 12, 1969, pp. 2057-2062.

[15] V. Panuska, "An adaptive recursive-least-squares identification algorithm,"*8th IEEE Symposium on Adaptive Processes and Decision and Control*, vol. 8, part1, 1969, pp. 65-69.

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

[17] S. Ciochina, C. Paleologu, J. Benesty and A. A. Enescu, "On the influence of the forgetting factor of the RLS adaptive filter in system identification," *International Symposium on Signals, Circuits and Systems* (ISSCS 2009), 2009, pp.1-4.

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

[19] G. O. Glentis, K. Berberidis and S. Theodoridis, "Efficient least squares adaptive algorithms for FIR transversal filtering," *IEEE Signal Processing Magazine*, July 1999, pp. 13-41.

[20] D. T. M. Slock and T. Kailath, "Numerically stable fast transversal filters for recursive least squares adaptive filtering," *IEEE Transactions on Signal Processing*, vol. 39, January 1991, pp. 92-113.

[21] M. S. Ahmad, O. Kukrer and A. Hocanin, "Recursive inverse adaptive filtering algorithm," *Fifth International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control* (ICSCCW2009), 2009, pp. 1-3.

[22] D. J. Tylavsky and G. R. L. Sohie, “Generalization of the matrix inversion lemma”, *Proceedings of the IEEE,* vol. 74, no. 7, pp. 1050-1052, July. 1986.

[23] P. S. R. Diniz, M. L. R. de Campos and A. Antoniou, “Analysis of LMS-Newton adaptive filtering algorithms with variable convergence factor,” *IEEE Transactions on Signal Processing,* vol. 43, no. 3, 1995, pp. 617 - 627.

[24] Y. Zhou, S. C. Chan and K. L. Ho, “A New block-exact fast LMS/Newton adaptive filtering algorithm,” *IEEE Transactions on Signal Processing,* vol.54, no. 1, 2006, pp. 374 - 380.

[25] P. P. Mavridis and G. V. Moustakides, “Simplified Newton-type adaptive estimation algorithms,” *IEEE Transactions on Signal Processing,* vol. 44, no. 8,1996, pp. 1932 - 1940.

[26] M. L. R. de Campos and A. Antoniou, “A New quasi-Newton adaptive filtering algorithm,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* vol. 44, no. 11, 1997, pp. 924 - 934.

[27] S. Haykin, *Kalman Filtering and Neural Networks*, John Wiley & Sons, NY,2001.

[28] M. S. Ahmad, O. Kukrer and A. Hocanin, “An efficient recursive inverse adaptive filtering algorithm for channel equalization,” *European Wireless Conference* (EWC 2010), Lucca, Italy, April 2010, pp. 88-92.