NOISE CANCELLING USING ADAPTIVE FILTER ALGORITHMS

OMAR MOHYALDEEN ALMALLAHMED

JULY 2015

NOISE CANCELLING USING ADAPTIVE FILTER ALGORITHMS

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF

ÇANKAYA UNIVERSITY

BY

OMAR MOHYALDEEN ALMALLAHMED

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF

COMPUTER ENGINEERING

JULY 2015

Title of the Thesis: **Noise Cancellation Using Adaptive filter Algorithms.**

Submitted by **Omar Mohyaldeen ALMALLAHMED**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.

Prof. Dr. Taner ALTUNOK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Müslim BOZYİĞİT
Head of Department

This is to 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.

Assist Prof. Dr.Yuriy ALYEKSYEYENKOV
Supervisor

Examination Date: 27.07.2015

Examining Committee Members

Assist. Prof. Dr.Yuriy ALYEKSYEYENKOV	(Çankaya Univ.)
Assist. Prof. Dr. Abdül Kadir GÖRÜR	(Çankaya Univ.)
Assos. Prof. Dr. Fahd JARAD	(UTAA Univ.)

TATEMENT OF NON-PLAGIARISM PAGE

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

Name, Last Name : Omar ALMALLAHMED

Signature :

Date : 27.07.2015

ABSTRACT

NOISE CANCELLING USING ADAPTIVE FILTER ALGORITHMS

ALMALLAHMED, Omar
M.Sc., Department of Computer Engineering
Supervisor: Assist. Prof. Dr. Yuriy ALYEKSYEYENKOV

July 2015, pages 84

The aim of this research is to study the theoretical adaptive filters for noise canceling problem. Firstly, it describes the notion using adaptive filters. Secondly, it presents three more common adaptive filters the RLS, NLMS and LMS algorithms. In addition, it describes the implementation of adaptive filters. Furthermore, the aim of this study is to solve some of the noise problems that affect the performance of the systems. The use of Noise Canceling theories is the most important solution for this problem, and for that research adaptive filters are used to solve these problems. The most important algorithms for these filters are RLS, NLMS and LMS algorithms. The research tests these algorithms with adaptive filters applications (noise cancellation, signal prediction, system identification, echo cancellation), and have chosen the best algorithm based on the value of Mean Square Error (MSE). The research presents two examples of a solution for the noise problem of the voice signal using Simulink MATLAB for different environments. The first example is active noise canceling theory, where, the color noise and then white noise (Gaussian noise) on the audio signal is used, after that three algorithms RLS, LMS, NLMS are applied to the audio signal to detect and cancel both noises. The second example is Channel Equalization, which also applies the three algorithms to solve the very same problem. Finally, the results of this Simulink are discussed and future work is present.

Keywords: Adaptive Filter, Noise Canceling, RLS, NLMS, LMS Algorithms**.**

ÖZ

UYARLAMALI FİLTRE UYGULAMALARININ KULLANILMASI İLE GÜRÜLTÜNÜN YOK EDİLMESİ

ALMALLAHMED, Omar
Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı
Tez Yöneticisi: Yrd. Doç. Dr. Yuriy ALYEKSYEYENKOV

Temmuz 2015, 84 sayfa

Bu araştırmanın amacı gürültünün yok edilmesi probleminde uyarlamalı filtrelerin (adaptif filtre) teorik çalışmasının gerçekleştirilmesidir. Araştırma ilk olarak, uyarlamalı filtrelerin kullanım kavramını açıklamaktadır. Çalışmada ikinci olarak, RLS, NLMS ve LMS gibi bilinen üç uygulamalı filtre sunulmuştur. Tezde, buna ek olarak, uyarlamalı filtrelerin uygulamaları da anlatılmaktadır. Bu çalışmanın diğer bir amacı ise, sistem performansını etkileyen bazı gürültü problemlerini çözüme ulaştırmaktır. Bu problemin en önemli çözümü gürültü yok etme teorisinin kullanımıdır, Bu tür problemlerin çözümünde uyarlamalı filtreler kullanılmaktadır. Bu filtrelerin en önemli algoritmaları RLS, NLMS ve LMS'tir. Araştırmada, uyarlamalı filtre uygulamaları (gürültü yok etme, sinyal tahmini, sistem tanımlaması, yankı iptali) ile bu algoritmalar test edilmiş ve ortalama hata değerlerine göre içlerinden en iyi olan algoritmanın seçimi sağlanmıştır. Araştırmada Simulink

MATLAB kullanılarak farklı ortamlardaki ses sinyallerinin gürültü iptali çözümüne dair örnekler sunulmuştur. Bunlardan ilki beyaz gürültü (Gaus Gürültüsü) kullanımı, ikincisi ise ses sinyali üzerinde renkli gürültü kullanımı ve sonrasında gürültünün bulunup yok edilmesi için RLS, MLS VE NLMS' in kullanılmasıdır. İkinci örnek ise Kanal Denkleştirme konusudur ve burada da aynı sorunu çözmek için yine sözü edilen üç algoritma kullanılmaktadır. Tezde, son olarak, Simulink sonuçları tartışılmış ve gelecekteki çalışmalara dair fikirler ve öneriler sunulmuştur.

Anahtar Kelimeler: Uyarlamalı Filtreler, Gürültünün Yok Edilmesi, RLS,NLMS, LMS Algoritmaları

ACKNOWLEDGEMENTS

First of all, I praise God for providing me with this opportunity and granting me the capability to proceed successfully throughout my study. Also, I would like to express my sincere gratitude to Assist. Prof.Dr.Yuriy ALYEKSYEYENKOV for his supervision, special guidance, suggestions, and encouragement through the development of this thesis.It is a pleasure to express my special thanks to my parents, my wife, my children Alfarooq and Basmala, my brother Qusay Mohyaldeen ALMALLAHMED, my sisters and my friends, especially Taha ALJADIR, Mouath SALIM and Sinan ALKHAYAAT for their valuable support.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

FIGURES

FIGURES

FIGURES

LIST OF TABLES

TABLES

LIST OF ABBREVIATIONS

AF	Adaptive Filters
ANC	Adaptive Noise Canceller
DSP	Digital Signal Processing
FIR	Finite Impulse Response
IIR	Infinite Impulse Response
ISI	Inter Symbol Interference
LMS	Least-Mean-Squared
MSE	Mean-Squared Error
NC	Noise Cancelling
NLMS	Normalized Least-Mean-Squared
RLS	Recursive Least-Squares
WLS	Weighted Least-Squares

CHAPTER1

INTRODUCTION

1.1 Background

In the current world of electronics,the increase in the number of electrical appliances as well as of the industrial equipment has led to the emergence of noise problems, This makes it necessary for us to think about the noise cancellation solutions, particularly,one of which is the use of silencers and blocker vibration along with the use of filters to solve such problems.

The purpose of this research is to study the adaptive filter (AF), as well as algorithms for the filter along with private applications of the adaptive filter theory, that give some examples of these applications which help in the understanding of the technical method for such filters and how to cancel the noise [1].Here, in this part,it is explained the most important algorithms and how they work as well as the characteristics of each algorithm and what are the advantages and disadvantages of these algorithms.

In this chapter. A quick introduction to adaptive filters and some applications is given along with, to the main purpose of this thesis which is explaining about Noise Cancelling (NC)

1.2 Overview

Ever since signal processing has become the requirement of modern era particularly after the advancement in the field of engines apart from other areas where noise is generated may it be sound or vibration or any other type of noise signal processing of the requirements of the modern era, especially after the great advances in the field of engines, as well as in other areas that cause noise to generate may it be the sound or vibration or any other type of noise.

All this led to an interest in the Noise Canceling theories and hard work has been put in to develop all theories that study Noise Canceling or Noise Reduction. The adaptive filters are the most important means to reduce this phenomenon, which is discussed in this paper. In addition to the above, the real-time implementation is calculated in most of the Fixed filters due to the increase in the cost and difficulty of calculating the real-time. While, in adaptive filters the feature of a real-time makes it easy to calculate.

On the other hand, it is not important to know the properties of the signal and transport properties of the channel as is the case in fixed filters where it is necessary to know the properties of the signal or the transport properties of the channel which is an essential condition to cancel noise [2].

One of the main advantages of adaptive filters is that it works in an environment that even though is not known, but it has the ability to adapt to the environment and the ability to detect differences in the time variables. In addition to the low cost of these filters, its implementation is simple and uncomplicated [3]. The main processing of the adaptive filter can be classified into two major processes, the first process is the filter process which filters the signal from any noise in the signal, this process depend to the second process of the adaptive filter. The second process optimizes weights to the parameters to optimize the performance of the adaptive filter to get a better result. The adaptive filter can be divided into two types, this classification depends on the impulse response. The first type is Finite Impulse Response (FIR) adaptive filter, which starts to respond when the value of the parameters decreases to be close to zero after a limited time period. The second type is Infinite Impulse Response (IIR) adaptive filter, this type depend to the inner feedback technique, and has infinity response [4].

Audio noise, which is the subject of research of this thesis is the kind noise that can in accordance to adaptive filter canceled this kind of noise without leading to the corruption of the signal and the impact on the hardware performance.

There are two types of noise: active noise and passive noise. The adaptive filters have the ability to cancel the active noise and to minimize the impact of fires on the hardware as well as the psychological impact. For ease of the design and ease of work, these filters lead to the removal of low frequency noise that has caused worry to the designers of communications devices in the past [5].

1.3 Objectives of the Thesis

The purpose of this study is to discuss the adaptive filter and to study the theories of the adaptive filters (Identification, Inverse Modelling, Prediction and Interference Cancelling), after that compare the most important algorithms (LMS, NLMS and RLS) in the practical part,where the research uses the SIMULINK of MATLAB program to simulate the adaptive filter and the algorithms.Also the research discusses the example proposal in the Active Noise Cancelling (ANC) when the original signal is audio.

In the next chapter the research discusses the adaptive filter theories and the adaptive filter algorithms.

CHAPTER 2

ADAPTIVE FILTER THEORY

In this chapter, the research discusses the applications of adaptive filters and what are the uses of these applications. Moreover, it discusses the theories of adaptive filters and what are the problems faced and then later explain about Wiener filters. Thenexplanation of the algorithms for adaptive filters is given, i.e. algorithms, Recursive Least Squares (RLS) and Least Mean Squares (LMS) algorithms. In addition, to this the research also explains the details of the Least Mean Squares (LMS) algorithm, as well as explain the most significant details of the versions of this algorithm (LMS), a normalized LMS algorithm (NLMS).

2.1 Adaptive Filter

Let's suppose Adaptive filters have the capacity of adaptation with unknown surroundings. These filters are versatile; therefore, they are used differently. (They can be operated in an unknown system) in addition to that, are low in cost (hardware cost of implementation, compared to the non-adaptive filters, acting in the same system).

One of the main issues which makes it the most important applications of digital signal processing (DSP) is that it has the ability to track the input signal and the time of the statistics works and adjustment of parameters [6] [4]. Truly, adaptive filters are utilized in many applications successfully. As a result, these applications are divided into four divisions: prediction, modelling, inverse, identification and interference cancelling and thus discussed in the next chapter accordingly.

Here it is necessary to mention, the applications stated earlier have similar characteristics: the adaptive filter receives a signal and compares it with an eligible response, producing an error which is then utilized to modify the filter's adaptable coefficients, that is known as weight so as to make the error lesser. In an ideal sense, that error is optimized, in some cases to zero, and in a desirable signal in some others [1] [4].

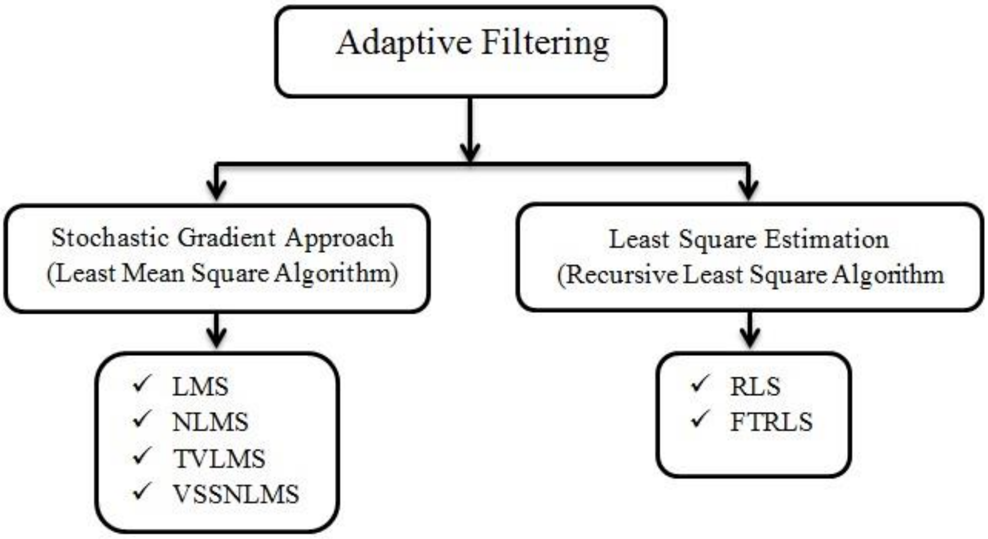

Figure 2.1 Pyramidal of Adaptive filter

Figure 2.1 shows the types of adaptive filters and algorithms for each type

2.2 Active Noise Cancellation

The (ANC) Active Noise Cancelling belongs to the Interference Cancelling type. It can also be referred as an adaptive noise cancelling or active noise canceller. This algorithm has a purpose which is to minimize the noise interference, or cancel that perturbation in case there is an optimal situation is available [6]. This algorithm also adopts an approach that tries to imitate the original signal $s(n)$. Therefore, the goal of this study is to use an ANC algorithm in order to cancel speech noise interference; however, this algorithm can also be employed to deal with any other type of corrupted signal. Here ANC can be viewed in figure 2.2, as given below,

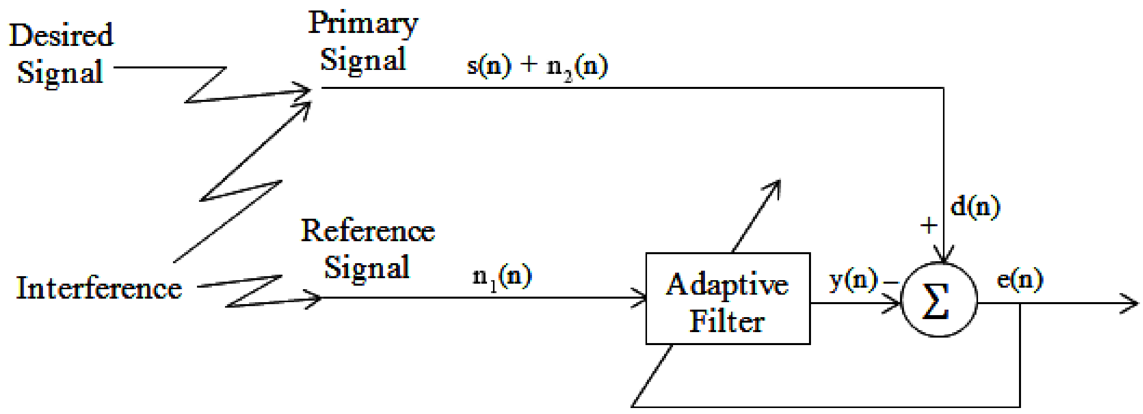

Figure 2.2 Active Noise Cancelling

The goal of the ANC is to reduce the noise interference corrupting the original input signal. One can see in the figure above that the desired signal $d(n)$ is composed of unknown signals, that calling the $s(n)$ corrupted for an additional noise $n_2(n)$, and generated for the interference. The adaptive filter is then installed in a place where the only input is the interference signal $n_1(n)$. Both signals $n_1(n)$ and $n_2(n)$ are correlated. The output of the filter y(n) is compared with the desired signal $d(n)$, generating an error $e(n)$. This error in the system output can be used to adjust the variable weights of the adaptive filter in order to minimize the noise interference. In an optimal situation, the output of the system $e(n)$ is composed of the signal $s(n)$ which is free of the noise interference $n_2(n)$ [7].

2.3 Implementation of Adaptive Filters

2.3.1 Introduction

Before explaining adaptive filters applications, it is important to give a brief explanation of these applications, and how to use them, as well as what is the purpose of the same applications.

2.3.2 General Applications

The adaptive filters are the finest filters currently used because of the low cost as well as ease use in addition to their versatility to adapt. On these basis, they divided into four types of applications, and thus focus in this section is to explain these applications and uses of each one of them. Chapter three explained these applications in detail and their representation in practice and compare their results using the three algorithms(LMS, LMS and RLS) [6] [8].

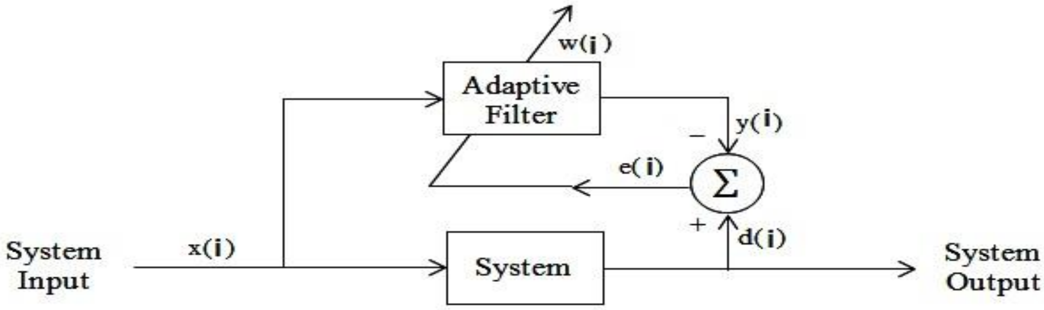

Figure 2.3 System Identification

i. **Identification**: This can be explained by the case illustrated in Figure 2.3, where the case contains which may have any system and signal output of a given system represents the desired signal $d\ (i)$. The reference signal $x(i)$ is to represents the input signal in the adaptive filters as the same input signal of the system. Here, the output of the adaptive filter $y(i)$ in comparison with the desired signal $d\ (i)$ produces error signal $e(i)$, which has many benefits. The most important benefit, is used for adjusting the weights, these weights are utilized to reduce the error.

   The purpose of designing an a AF is this that it supply an approximation for any system [8] [9].

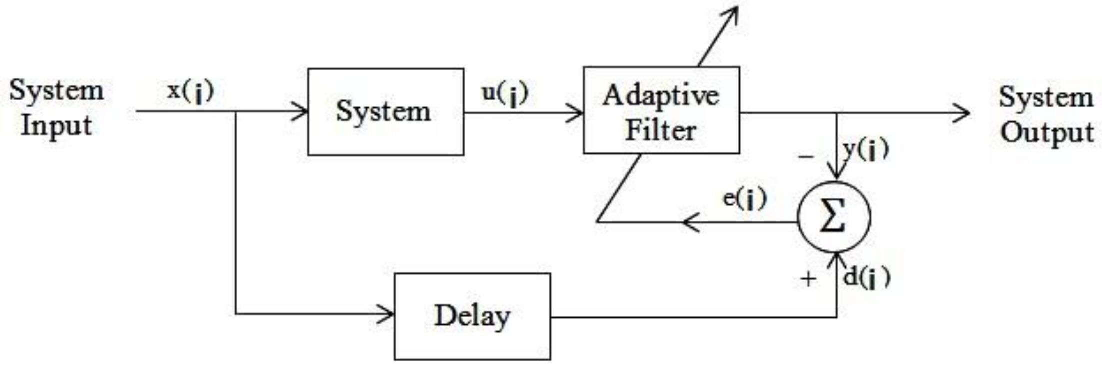

Figure 2.4 Inverse Modelling

ii. **Inverse Modeling**: As illustrated in Figure 2.4 in this case, the input signal to the Adaptive filters $u(i)$ is an output signal of the system, the desired signal $d(i)$ is the same signal input to the system $x(i)$, which however undergone some delay versions. The comparison between the desired signal $d(i)$ and the output of the adaptive filter $y(i)$ will generate an error signal which this signal (error signal) utilizes to reduce the weight and thus reduce the error.

Here, the purpose is of tracking and discovering the feedback transfer job of the any system [9].

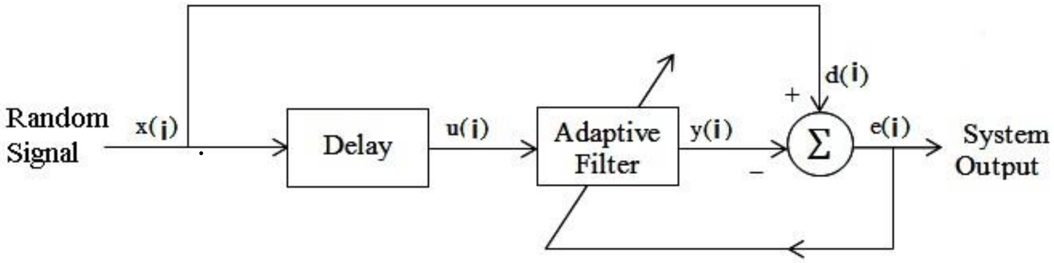

Figure 2.5 Prediction

iii. **Prediction**:In this case, as noted in Figure 2.5, there is only one signal $x(i)$, hence the input signal is the same as the desired signal $d\ (i)$, where as the input signal to the adaptive filter is $x(i)$, but it's a delay version $u(i)$, where the error signal generates in comparison between the output of the adaptive filter $y(i)$ and the desired signal $d\ (i)$. This is"used to provide a prediction of the present value of a random signal"[9].

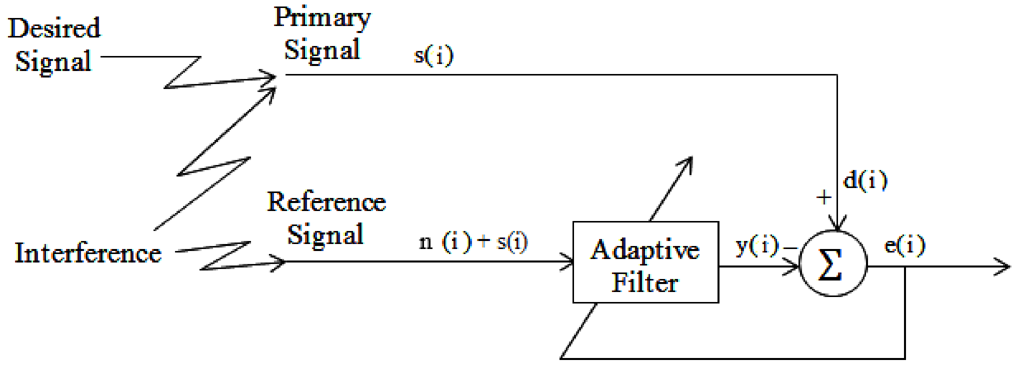

Figure 2.6 Jammer suppress

iv. Jammer Suppress

This application can be seen in Figure 2.6, which is the most important one in the adaptive filter applications. It contains two signals, the original signal without the noise $s(i)$ and the signal with noise $s(i) + n(i)$. In this application, the primary signal which represents the desired signal and the reference signal which represents the corrupted signal. The source signal $s(i)$ is the same as the desired signal $d(i)$, and the corrupted signal is an input of adaptive filters. When compared between the adaptive filter output $y(i)$ and the source signal $s(i)$, the result will represent the error signal $e(i)$ and this signal is used for optimizing the weights $w(i)$, the weight which is used for correcting the compared signal. This application is used for canceling unwanted intervention from an original signal [4] [9].

2.4 Performance Measures in AF

In this portion the research tries to define some of the important terms discussed in this thesis.

2.4.1 Mean Square Error

The Mean Square Error (MSE) is the factor used for measuring the effective solution of the problem, therefore, the MSE value must be small or close to zero, that means that the solution can be more effective then. There are some reasons that affect the value of MSE. These reasons are, firstly, the quality of the noise, secondly, the error in the step size (slow convergence or fast convergence), thirdly, the quantization of the noise and finally the noise signal amplitude [2].

2.4.2 Convergence Rate

This term is used for the convergence between the parameters, that means this factor is used for calculating the ratio which is used for optimizing results. The speed convergence is the best case in the adaptive filter, the convergence rate is not independent on other factors [2].

2.4.3 Filter Length

This factor is very important in the adaptive filter, where the filter length is subject to many parameters that determines the length of the filter. The most important factor is the size of the noise signal and the type of this signal, which increase the number of iterations which mean increasing in the accuracy of the filters, as well as the number of iterations which have an effect on the convergence rate. On the other hand, when there is an increase in the filter length, which leads to increase of the computations operation, thus making an impact to the MSE and its stability [2].

2.5 Wiener Filter Theory

The purpose of using the Wiener filter is to cancel the existing noise in the signal or minimize noise ratio, which is done by estimating the ratio of noise in the signal. This filter uses desired signal and compare it with a reference signal to reduce the noise, and also where it estimates signal required without noise. This filter has the potential to impose noise signal as well as input signal at random.

To derive the expression of a finite impulse response wiener filter, considers a discrete time filter as shown in Figure 2.1 [6].

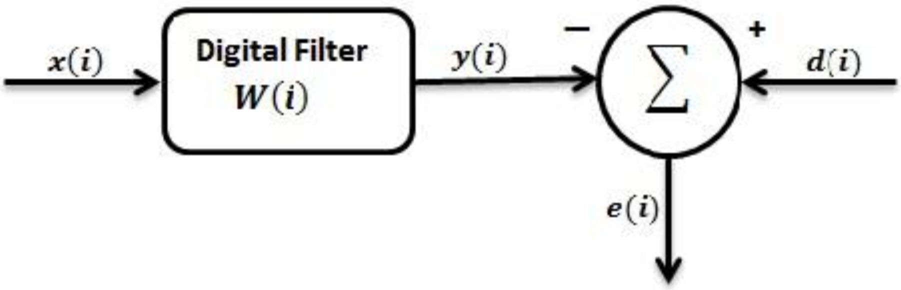

Figure 2.7 Wiener Filter

Where $x(i)$: is the input signal ,$w(i)$: is the coefficients of wiener filter, $y(i)$: is the output signal , $d(i)$ is the desired signal $e(i)$ is the error signal.

Here the output signal $y(i)$ is calculated

$$y(i) = \sum_{n=0}^{q-1} w_n(i)x(i-n) \tag{2.1}$$

Or

$$y(i) = w(i)^t x(i) \tag{2.2}$$

Where $x(i-n)$ is the input signal specimens, the value of the weights of the filter is w_n . In fact, w_n is the same filtering parameter in adaptive filter.

The difference of desired signal $d(i)$ and output signal $y(i)$ is defined as error signal $e(i)$ [10].

$$e(i) = d(i) - y(i) \tag{2.3}$$

The signal of the square error same the signal of the cost function J. [6]

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

The aims of these filters are to lessening the cost function. Filter's optimal parameters can be accessed by creating thresholds to the cost function. Where the equations are:

$$\nabla J = -2E\{x(i-n)e(i)\} \tag{2.5}$$

When $n = 0, 1,2,3, \dots . q$

The ideal situation is when the error value is very close to zero, and very close to the value of the cost function. In this case, it becomes an ideal adaptive filters work and which gives us the signal output close to the original signal. As previously mentioned that one of the best cases in adaptive filters is to have a cost function value close to zero and which can be accessed in this case by the following equation [11]:

$$E\{x(i-n)e_{min}(i)\} = 0 \qquad 2.6$$

When $n = 0, 1,2, \dots . q$, and the e_{min} is less in error value which is an optimal case.

When compensation equation (2.6) using equations (2.1) and (2.3) it will get the $y(i) = \sum_p^q w_{opt}(p)E\{x(i-n)x(i-p)\}$

$$y(i) = E\{x(i-n)d(i)\} \qquad 2.7$$

Where $n = 0, 1,2, \dots . q$

With the imposition of the w_{opt} is the ideal weight for the filters, and the value of (q) is the length of the filter.

$$r(p-n) = E\{x(i-n)x(i-p)\} \qquad 2.8$$

$$s(-n) = E\{x(i-n)d(i)\} \qquad 2.9$$

"Where $r(p-n)$ is the auto correlation function of the input signal and $s(-n)$ is the cross-correlation function of the input signal and also the desired response"[14].

It can simplify the equation (2.7) to become in this format

$$\sum_{p=0}^{q-1} w_{opt}(p)r(p-n) = s(-n) \qquad 2.10$$

The last equation (2.10) represents the Wiener Hopfequation, this equation can be represented as the following [10].

$$Rw_{opt} = s \qquad 2.11$$

As R is the autocorrelation an array of the input signal $x(i)$, S is the cross-correlation function of input signal $x(i)$ and desired signal $d(i)$. $\boldsymbol{w}_{opt}$ is the vector consisting of the optimal filter coefficients which minimizes $E\{e^2(i)\}$ known as mean squared error (MSE) [2].

2.6 Adaptive Filter Algorithms

2.6.1 Least Mean Square Algorithm

The research considers this algorithm as the most important and basic algorithm in the adaptive filters, and this algorithm (LMS) is a linear regression algorithm. The working principle of this algorithm is to find the best parameters, which reduce the squared error. "LMS algorithm is rounded to the most descent algorithms using the expected value of the Squared Error signal. LMS is useful in implementations where the full knowledge of the error signal is not obtainable"[10],[12].

The working principle of the MSN algorithm is the principle of iteration of parameters, calculate the tap weight update:

$$w(i+1) = w(i) + {}^{1}/_{2}\,\mu[-\nabla J] \qquad 2.12$$

Where the value of $(w(i+1))$ represents the updated weight parameters, the value of (μ) represents a constant value which is a step size, with respect to the value of (J), as it is previously stated that it represents the value of the cost function, with a note that the value of the cost function is equal to the instantaneous value of the squared error signal$e(i)$.

$$J = e^2(i) \qquad 2.13$$

When compensation equations (2.2) in the (2.3) will produce an equation of the cost function

$$\nabla e^2(i) = -2e(i)x(i) \qquad 2.14$$

The new equations after updating parameters will be derived as

$$w(i+1) = w(i) + \mu[x(i)e(i)] \qquad 2.15$$

And

$$e(i) = d(i) - y(i) \qquad 2.16$$

And

$$y(i) = w(i)^L x(i) \qquad 2.17$$

The value of the Size step (μ)should be confined between

$$0 < \mu < {}^{2}/_{\lambda_{max}} \qquad 2.18$$

Where λ_{max} is the largest eigenvalue of the autocorrelation matrix of $x(i)$, ***R*** [10]

Where

$$\lambda_{max} = \sum_{n=0}^{q-1} E\{|x(i-n)|^2\} \qquad 2.19$$

$$0 < \mu < 2 \Big/ \textstyle\sum_{n=0}^{q-1} E\{|x(i-n)|^2\} \qquad 2.20$$

Input	*x(i)* = Input signal. *w(i)* =Tap weight. And *d(i)*= desired signal
Output	*w(i+1)*= tap weight update. *y(i)*=Filter output signal
Parameters	Tap input = λ_{max} . Step size= μ $\lambda_{max} = \sum_{n=0}^{q-1} E\{|x(i-n)|^2\}, 0 < \mu < 2/\lambda_{max}$
Initialization	At the beginning of the algorithm, any values undefined offset zero *w*(0) = 0.
Steps	One: - filtering process $y(i) = w^t(i)x(i)$ Two: - Error assessment $e(i) = y(i) - d(i)$ Three:- update the tap weight $w(i+1) = w(i) + 2\mu\, e(i)x(i)$

Table 2.1 Represents a Summary of the LMS Algorithm

2.6.2 Normalized Least-Mean-Square Algorithm

NLMS algorithm is developed from the LMS algorithm, which has been added to some of the parameters on the LMS algorithm to solve some of the problems LMS algorithm. In the LMS algorithms, the value of the $x(i)$ is directly proportional to the $2\mu\, e(i)x(i)$.In this case, the increase in the value of

$x(i)$ leads to an increase in the value of the noise in the signal and amplify it. That goes back to the direct correlation relationship between the $x(i)$ on the one hand and the value of the step size (μ) and the value of the input $x(i)$ and the value of the error $e(i)$on the other hand. As discussed above, it makes it difficult to guess the value of the step size and thus affect the stability of the algorithm [2].

All of these reasons have led to the development of LMS algorithm and in solving all the problems and improving the stability, by adding more of the parameters.The development of the LMS algorithm makes a new algorithm which is called the Normalized Least-Mean-Square (NLMS) algorithm.

The NLMS algorithm calculates the size Step by the following equation:

$$\mu = \frac{\bar{\mu}}{\delta + \|x(i)\|^2} \qquad 2.21$$

Parameter δ is a constant and it is small, the parameter $\bar{\mu}$ is the step size of the NLMS and the value $0 < \bar{\mu} < 2$.[13]

To calculate the tap weights after the update using the following equation:

$$w(i+1) = w(i) + 2\mu\, e(i)x(i)$$

Since the

$$\mu = \frac{\bar{\mu}}{\delta + \|x(i)\|^2}$$

$$w(i+1) = w(i) + 2\,\frac{\bar{\mu}}{\delta + \|x(i)\|^2}\, e(i)x(i) \qquad 2.22$$

Input	*x(i)*= Input vector. *w(i)*=Tap weight. And *d(i)* =desired output
Output	*w(i+1)* = Tap weight update. *y(i)* =Filter output
Parameters	Step size = $\bar{\mu}$. δ = Constant $2 > \bar{\mu} > 0$
Initialization	At the beginning of the algorithm, any value undefined offset zero *w(0)*= 0.
Steps	One: - Filtering process $y(i) = w^t(i)x(i)$ Two:- Error assessment $e(i) = y(i) - d(i)$ Three :-Update the tap weight $w(i+1) = w(i) + 2\,\frac{\bar{\mu}}{\delta + \lVert x(i) \rVert^2}\; e(i)x(i)$

Table 2.2 Represents a Summary of the NLMS Algorithm

2.6.3 Recursive Least-Squares Algorithm

As it noted previously that the LMS algorithm depends on the reduction of MSE, while the recursive least squares algorithm depends on finding MSE. In addition, the slow convergence of the LMS algorithm, has been treated in the RLS algorithm and made fast convergence, and this is the most important difference between these two algorithms.

It is true that RLS algorithm is the most reliable and the best in terms of results, but on the other hand is very complex in terms of calculation of the parameters in this algorithm.

The cost function in the RLS algorithm which represents the Weighted Least Squares (WLS) can be calculated by the following equation:

$$J(i) = \sum_{a=0}^{n} \lambda^{i-1} e^2(i) \qquad 2.23$$

In this algorithm, when the value of lambda is (λ) between zero and one, this condition is called forgetting factor, and the usefulness of this factor is that it gives the value of previous error weights after updating the parameters.

$$e(i) = d(i) - y(i) = d(n) - w^L(i-1)x(i) \quad 2.24$$

The input signal $x(i)$ can be described mathematically by the following equation

$$x(i) = [x(i), x(i-1), \dots x(i+c-1)]^L \quad 2.25$$

Tap weight parameters $w(i)$can be described by the following equations

$$w(i) = [w_0(i), w_1(i), \dots w_{c-1}(i)]^L \quad 2.26$$

The purpose of the Add phi (Φ) parameter get less value cost function, and can get a less cost function values when the value of the weights of the parameters is in perfect condition, and are written in the form of a matrix.

$$\Phi(i)\hat{w}(i) = z(i) \quad 2.27$$

The c by c Dimensions of the matrix $\Phi\ (i)$

$$\Phi(i) = \sum_{a=0}^{n} \lambda^{i-1} x(i) x^L(i) \quad 2.28$$

"The *c-by-1* cross-correlation vector is ***z****(i)* between the tap inputs of the transversal filters and the desired response is defined by" [14]:

$$z(i) = \sum_{a=0}^{n} \lambda^{i-1} x(i) d^*(i) \quad 2.29$$

As * represents the complex coupling.

Now here one must calculate the value of the matrix inverse Φ^{-1}, as well as the value of lambda inverse λ^{-1}so that can be calculated the value of the gain vector$g(i)$ [14][12].

$$\Phi_A^{-1}(i) = \lambda^{-1}\Phi_A^{-1}(i-1) - \lambda^{-1} g(i) x^L(i) \Phi_A^{-1}(i-1) \quad 2.30$$

Then

$$g(i) = \frac{\lambda^{-1}\Phi_A^{-1}(i-1)x(i)}{1 + \lambda^{-1} x^L(i) \Phi_A^{-1}(i-1) x(i)} \quad 2.31$$

Accordingly, the calculation of the tap weight $w(i)$ the equation is

$$w(i) = w(i+1)g(i)e^*(i) \quad 2.32$$

As * represents the complex coupling

Input	*x(i)*= Input vector. $\hat{w}(i-1)$ =Tap weight. Φ_A^{-1} $(i-1)$ = correlation matrix. And *d(i)* =desired output
Output	*w(i)* =Tap weight update. Correlation matrix updates = $\Phi_A^{-1}(i)$. And $y_{i-1}(i)$=Filter output
Parameters	Number of tap = c. Constant =δ . Forgetting factor = λ.
Initialization	At the beginning of the algorithm, any values undefined offset zero *w*(0) =0 , $\Phi_A^{-1}(0) = 0$.
Steps	One: - calculating the gain vector *g(i)* $g(i) = \dfrac{\lambda^{-1}\Phi_A^{-1}(i-1)x(i)}{1+\lambda^{-1}x^L(i)\Phi_A^{-1}(i-1)x(i)}$ Two:- Filtering process $y(i) = w^L(i)x(i)$ Three :-error assessment $e(i) = d(i) - y(i)$ Four:-Update the tap weight $w(i) = w(i+1)g(i)e^*(i)$ Five :- update correlation matrix $\Phi_A^{-1}(i) = \lambda^{-1}\Phi_A^{-1}(i-1) - \lambda^{-1}g(i)x^L(i)\Phi_A^{-1}(i-1)$.

Table 2.3 Represents a Summary of the RLSAlgorithm

CHAPTER3

IMPLEMENTATION AND RESULTS

In this chapter, the research discusses the applications (Identification, Inverse Modelling, Prediction and Interference Cancelling) to the adaptive filters, and the research applies the three algorithms (LMS, NLMS and RLS) to all applications of adaptive filters, and compare the results based on the Mean-squared error (MSE).

3.1 System Identification

System identification is the one of the applications of the adaptive filters, this application is used to determine the unknown systems, whether these systems are Analog or Digital. This application is used in control systems, communication and in a DSP also it can be used to track the time in the unknown systems [15]. In chapter 2 paragraph 2.3.2 was discussed the system identification, and in this section, it solves this problem (system identification) by using the three algorithms(LMS,NLMS,RLS) [7], [20].

3.1.1 The Application

In this section can be discussed the problem of the system identification and how can this problem be solved. In figure 3.1 it shows the system identification. It can be noticed in this figure that here is one signal, this signal is an input to the adaptive filter and the unknown system, the output of the unknown system is the desired signal, then compared between the desired signal and the output of the adaptive filter that produces the error.

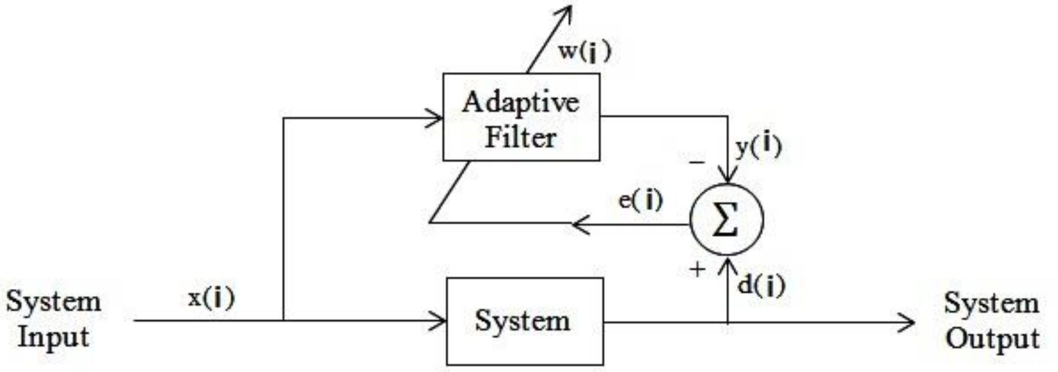

Figure 3.1 System Identification

3.1.2 Solution Using the LMS Algorithm

As explained in Chapter two, the LMS algorithm is based on the reduction of the value of square error as well as relying on finding the value of the cost function. The LMS algorithm can be applied to the identification problem in order to compare the results of this algorithm with other algorithms to the same application.

It is important to calculate the value of the cost function in the LMS algorithm, and this is calculated by the following equation [10] [11]:

$$J(i) = E|e^2(i)| \qquad 3.1$$

The problem with this application (Identification) is whether how to calculate the value of the desired signal that being the output signal of an unknown system. Using the equations that have been inferred in chapter two in paragraph 2.3.2, the following equation which used to calculate the desired signal can be obtained which represents the output signal value of the unknown system,

$$d(i) = w_0^L x(i) + i(i) \qquad 3.2$$

Figure3.2 shows problem (identification), and how to solve this problem by using the LMS algorithm as well as how to represent it use of MATLAB program.

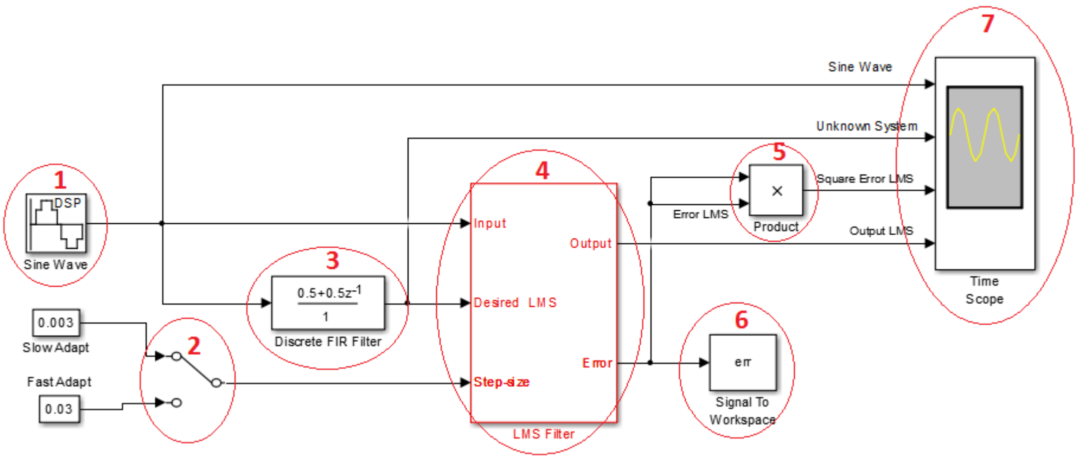

Figure 3.2 Solution using the LMS algorithm

Where

1- Is the input signal (sine wave) the amplitude of this signal =1 and the frequency =100 HZ.

2- Manual switch to select the value of the step size (μ) in this case the value of μ =0. 003.

3- Any system.

4- The adaptive filter (LMS or NLMS algorithm) blocks and the value of the filter length (q) =32.

5- Squared error, when compared between the desired signal and the output signal of the adaptive filter that produces an error signal.

6- Is a DSP tool called signal to work space block used for finding the Mean-squared error.

7- Time scope, used to display the result.

After running the Simulink MATLAB program, the results gathered areas shown in Figure 3.3

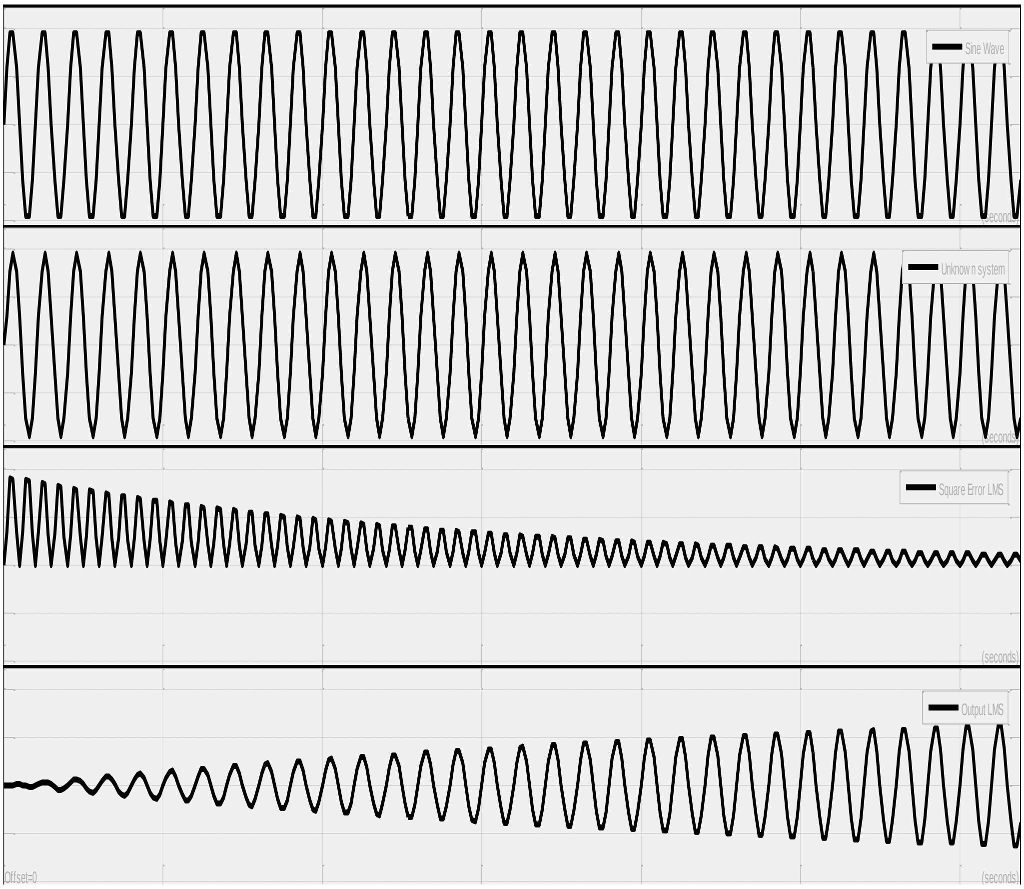

Figure 3.3 Pertinent signals when using the LMS algorithm to solve the System Identification problem

Figure 3.3 shows all of the input signal $x(i)$ and the desired signal $d(i)$, which represent the system output signal, as well as squared error $e^2(i)$in addition to the output signal of the filter $y(i)$, respectively, during the time of the implementation of 0.3 seconds. While taking into consideration the value of the length of the filter (the number of iterations) which is equal to 5.

In the figure 3.3 the squared error, the error value is very big, where the value of wave amplitude equal 1 at the time zero. Then that begins to decreasing to become zero when the time equal 0.077.

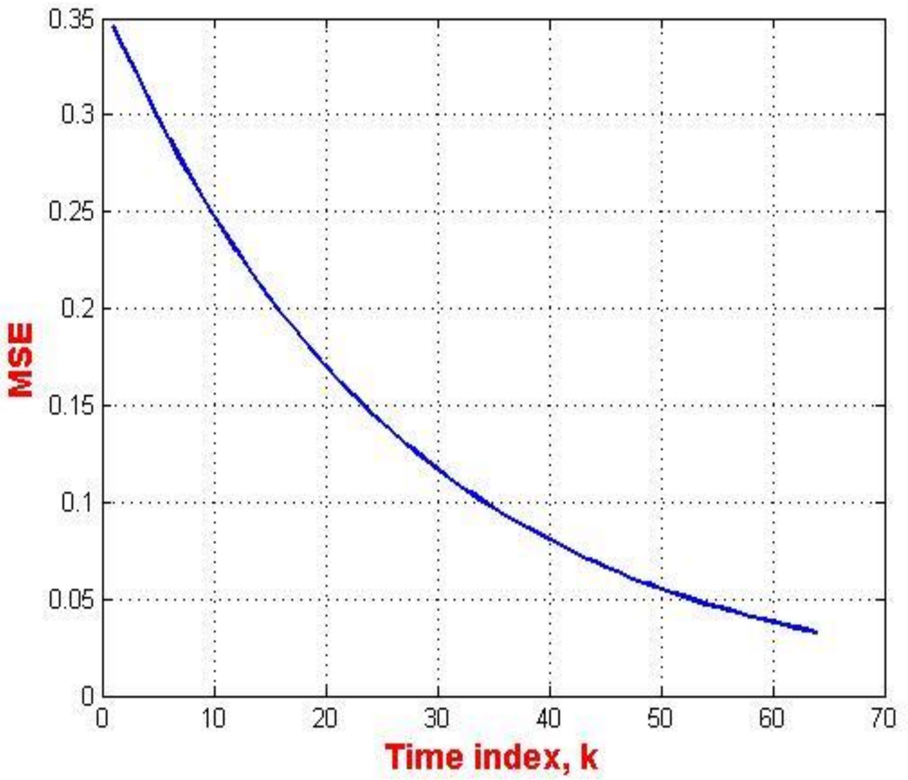

Figure 3.4 MSE of the LMS algorithm

In Figure 3.4 as it noted that the value of MSE is 0.35, here the purpose of finding a cost function, the mean-squared error is to see how the algorithm responses to reach the optimum limit. When analyzing Figure 3.4, as noted, that the convergence rate is the average rate, which is very slow and this algorithm needs about 150 samples at least during execution time, knowing that the execution time is 0.3 seconds.

3.1.3 Solution Using the NLMS Algorithm

Previously discussed in Chapter two, paragraph (2.3.2), that the biggest problems in the LMS algorithm is the difficulty in their application if the input signal $x(i)$ is very large, the second problem is the slow convergence, which leads to the development of an LMS algorithm and the work of the NLMS algorithm.

The application of this algorithm (NLMS) is to test the identification problem by the fast convergence after it has been tested on the slow convergence.

Figure 3.2 shows how to represent the LMS algorithm in the Simulink MATLAB program, the difference in the representation of NLMS algorithm is a step size (μ), where the size of the step in the NLMS algorithm is equal to 0.03 and the value of the constant is δ =0.9

The results that got after running the Simulink MATLAB program shown in Figure 3.6

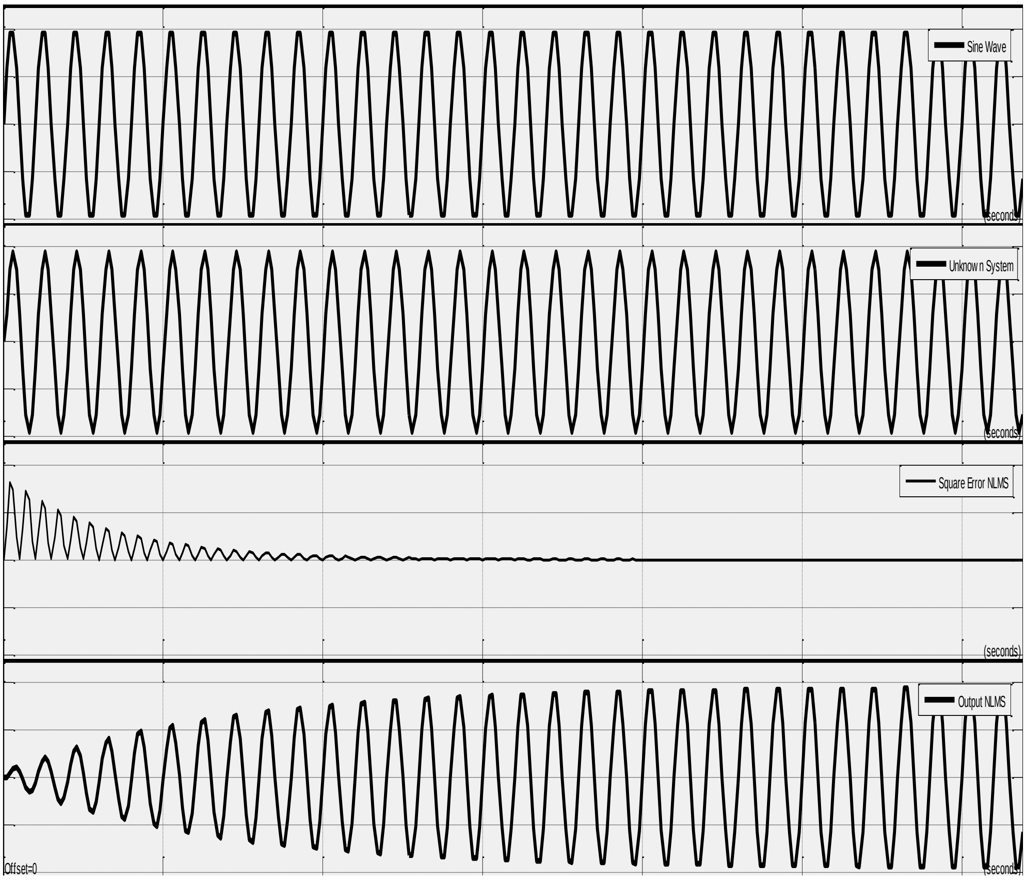

Figure 3.5 Pertinent signals when using the NLMS algorithm to solve the System Identification problem.

Figure 3.5 shows all of the input signal $x(i)$ and the desired signal $d(i)$, which represent the system output signal, as well as squared error $e^2(i)$ in addition to the output signal of the filter $y(i)$, respectively, during the time of the implementation of 0.3 seconds. While taking into consideration the value of the length of the filter (the number of iterations) which is equal to 5.

In the Figure 3.6 the squared error, where the value of wave amplitude equals 0.82 at the zero Sec. Then it begins decreasing to become very close to zero when the time equals 0.154.

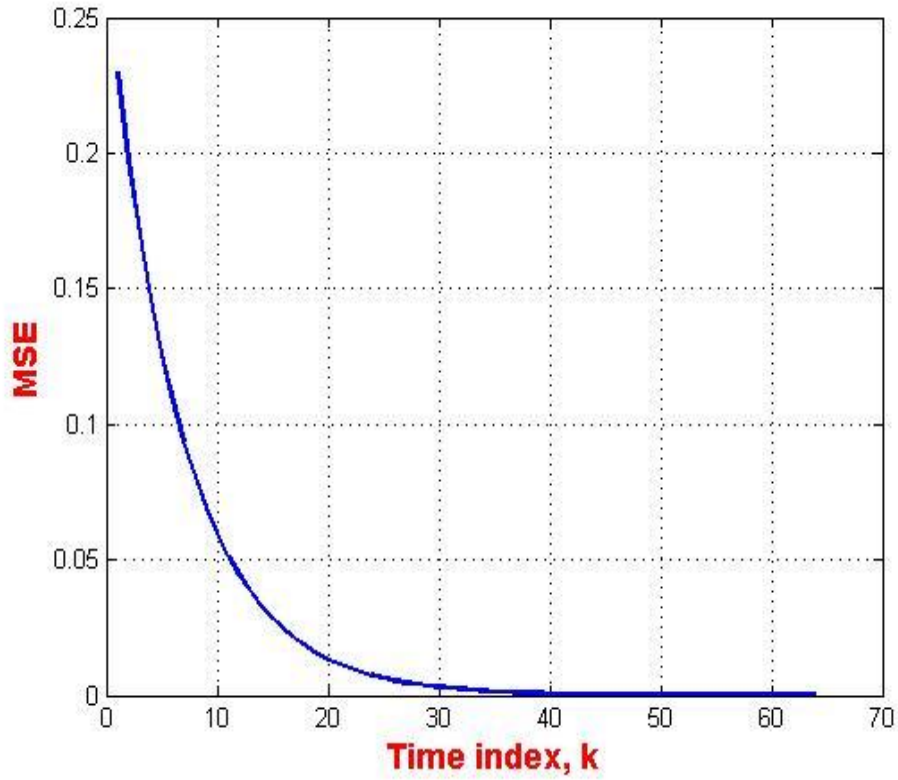

Figure 3.6 MSE of the NLMS algorithm

In Figure 3.6, it can be noticed that the value of the MSE starts at 0.023 in the first time and the MSE value in the NLMS is less than the MSE in the LMS algorithm, and the convergence rate in the NLMS algorithm is fast and has the value of the convergence rate is 35 samples. It means the convergence rate in the NLMS algorithm is faster about three times more of the LMS algorithm.

3.1.4 Solution Using the RLS Algorithm

The RLS algorithm is the most usable algorithms used for solving the problem of identification as being more effective. The value of the cost function which has the same value MSE is very less and that increases the proportion of stability, in addition to this, using fast convergence is also a good stability. For the purpose of calculating the value of the cost function [10] [11], as used the following equation to calculate the *J(i)*

$$J(i) = \sum_{a=0}^{n} \lambda^{i-a} e^2(i) \qquad 3.3$$

Where the lambda value between zero and one, called the forgetting factor, a small constant value used to give small weight to error signal while *e(i)* is the error value [15].

Using the equations in Chapter two, paragraph 2.3.4, and then applying this equation to the system identification problem, it can get the following equation:

$$d(i) = w_0^L x(i) + i(i) \qquad 3.4$$

Figure 3.7 shows the problem (identification), and how to solve this problem by using the RLS algorithm as well as how to represent it using MATLAB program,

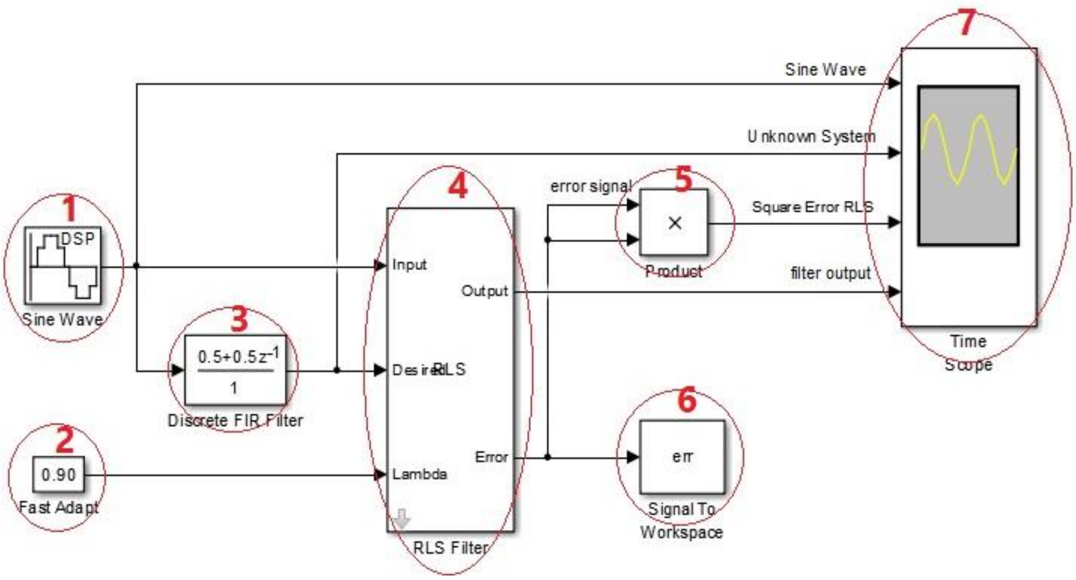

Figure 3.7 Solution using the RLS algorithm

Where

1- Is the input signal (sine wave) the amplitude of this signal =1 and the frequency =100 HZ.
2- Lambda value $\lambda = 0{,}9$.
3- Any system.

4- The adaptive filter (RLS algorithm) block, the value of the filter length (q) =32.

5- Squared error, when comparing between the desired signal and the output signal of the adaptive filter produces an error signal.

6- Is a DSP tool called signal to work space block used for finding the Mean-squared error.

7- Time scope, used to display the result.

The results that got after running the Simulink MATLAB program shown in Figure 3.8

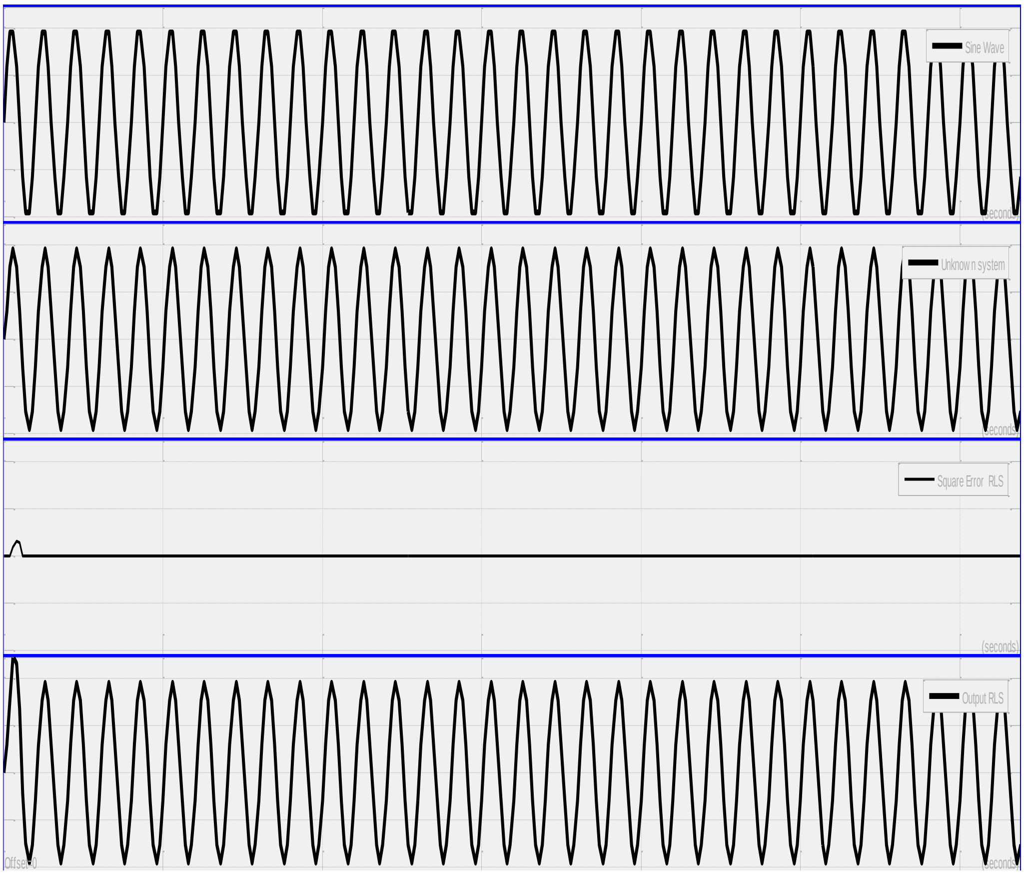

Figure 3.8 Pertinent signals when using the RLS algorithm to solve the System Identification problem.

Figure 3.9presents the zooming in the Figure 3.8, here can see the signal output between 0 to the second 0.03.

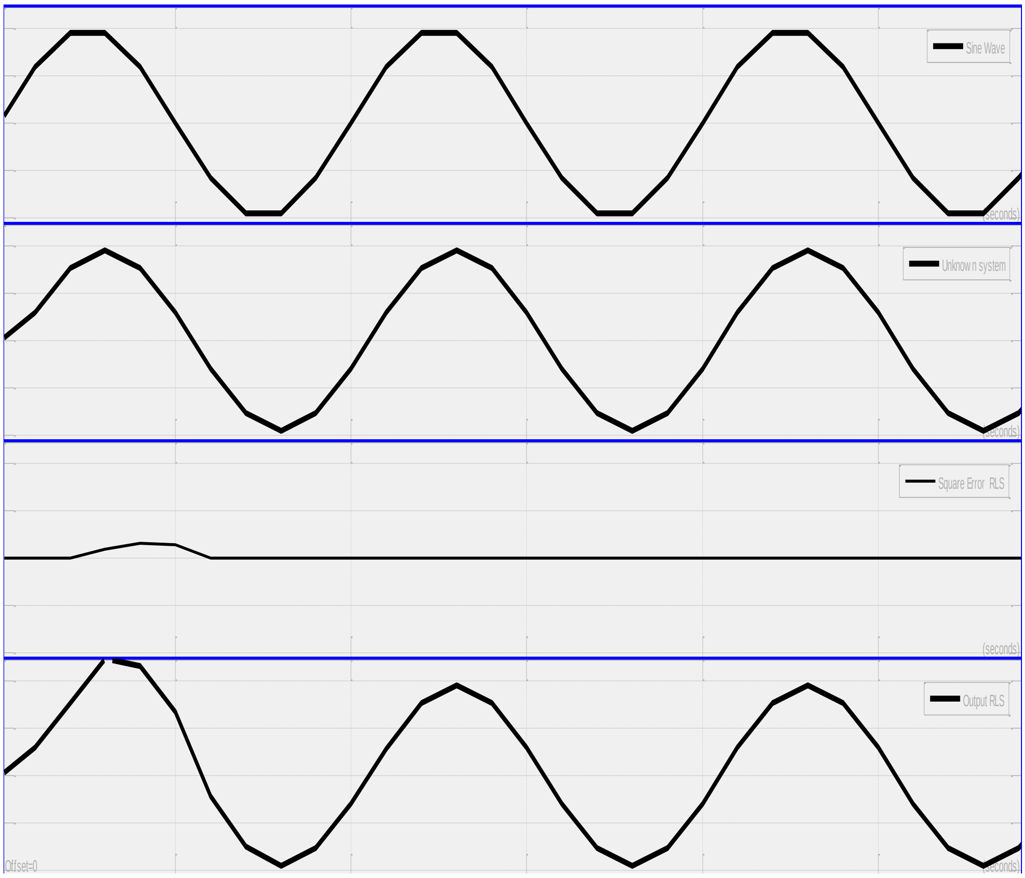

Figure 3.9 Image for figure 3.8 Scaled by 20.

Where as Figure 3.8 shows all of the input signal $x(i)$ and the desired signal $d(i)$, which represents the system output signal, as well as squared error $e^2(i)$ in addition to the output signal of the filter $y(i)$, respectively, during the time of the implementation of 0.3 seconds. While taking into consideration the value of the length of the filter (the number of iterations) which is equal to 5 the initialized value of the $\Phi_A^{-1}(0) = \delta I$. The best value of the δI when the constant equal is 0.3. The forgetting factor λ equals 0.9.

In the Figure 3.8 the squared error is starting close to zero and then increasing to large amplitude in the second 0.0015 where the value of wave amplitude equals 0,1.Then begins the decrease to become zero when the time equal 0.0028.

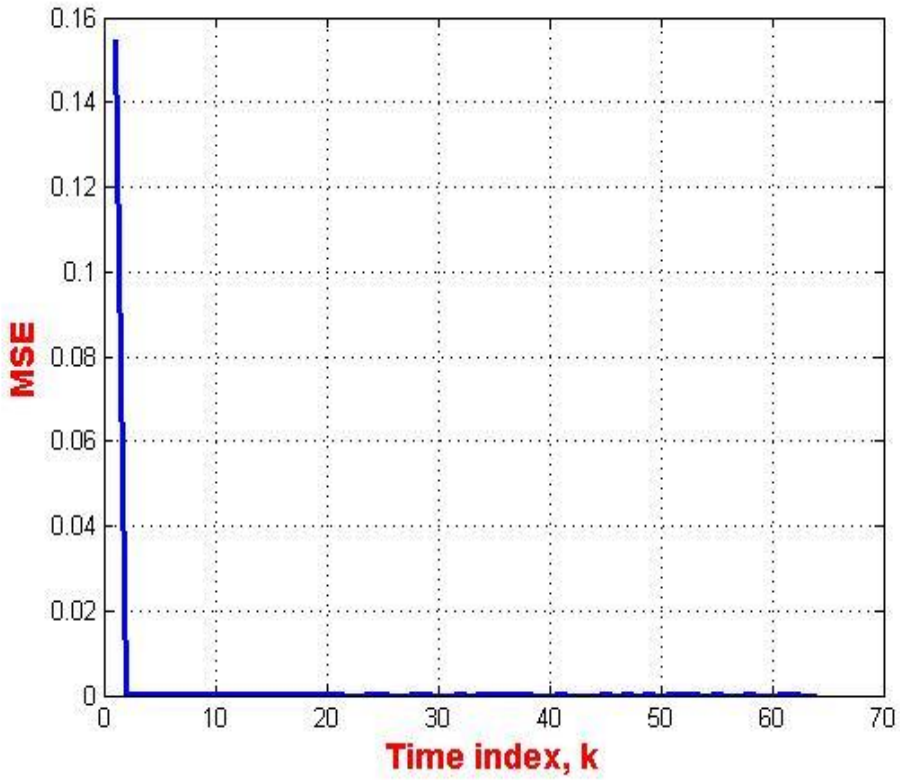

Figure 3.10 MSE of the RLS algorithm

In Figure 3.10 as noted that the value of the MSE is 0.16, this value is smaller as compared to the LMS algorithm and NLMS algorithm. The convergence rate too is very small, and has value is 2 samples during implementation time.

3.1.5 Comparison of Results

Reading all the details and then analyzing them are the most important requirements of real-time applications which is done by selecting the appropriate algorithm.The choice of non-correct algorithm in real-time applications leads to the high cost of implementation or poor implementation, where the poor implementation will lead to instability then leads to difficult repair of these mistakes or adoption of these results.The term to get the best results and choose the best algorithm in adaptive filters should adopt the following factors to solve the system identification problem.

- Convergence rate:The number of iterations is the most important requirements that must be taken into consideration, and the number of these iterations utilized to reach the nearest required value,"if the algorithm has a fast rate of convergence, it means that the algorithm adapts rapidly to a stationary unknown environment"[15].
- Cost is one of the most important thing that should be noted during the algorithm selection of appropriate adaptive filter. The costs include both the execution time as well as the equipment which is used in the process of filtration. In addition to using computers specifications must also take into account the mathematical operations used and the size of the memory suitable for these mathematical operations[15].
- Trace: the ability of the algorithm to track statistical differences in the unknown environment [15].

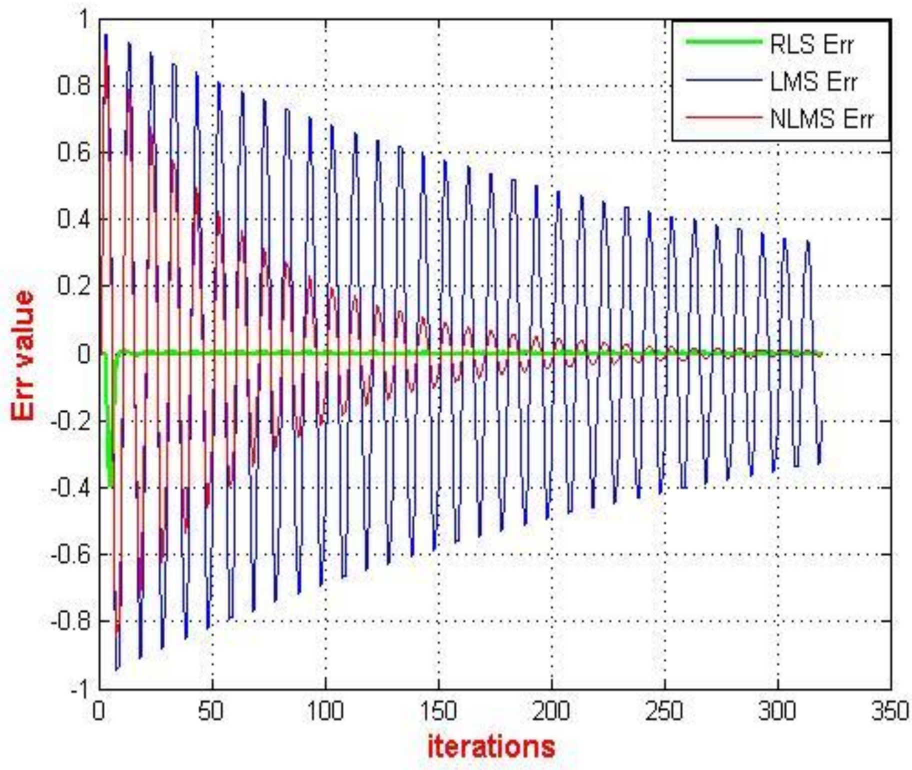

Figure 3.11 Comparison between error signal in the three algorithms

In Figure 3.11 a comparison is shown between the three algorithms (LMS, NLMS, RLS) through the error signal $e(i)$, which can be understood by this figure that the value has less error when using the algorithm (RLS) as the highest value of the error signal is equal to -0.4 this value is very small as compared with an LMS algorithm and NLMS algorithm, which becomes the error value zero after two training cycles. The error in the LMS algorithm is very big and spends a lot of time.

The dependence on the value of the error signal is not the only way to choose the best algorithm in adaptive filters, and the dependence on the value of MSE is the best way to choose the best algorithm. Figure 3.12 represents MSE for the three algorithms and the comparison between these algorithms.

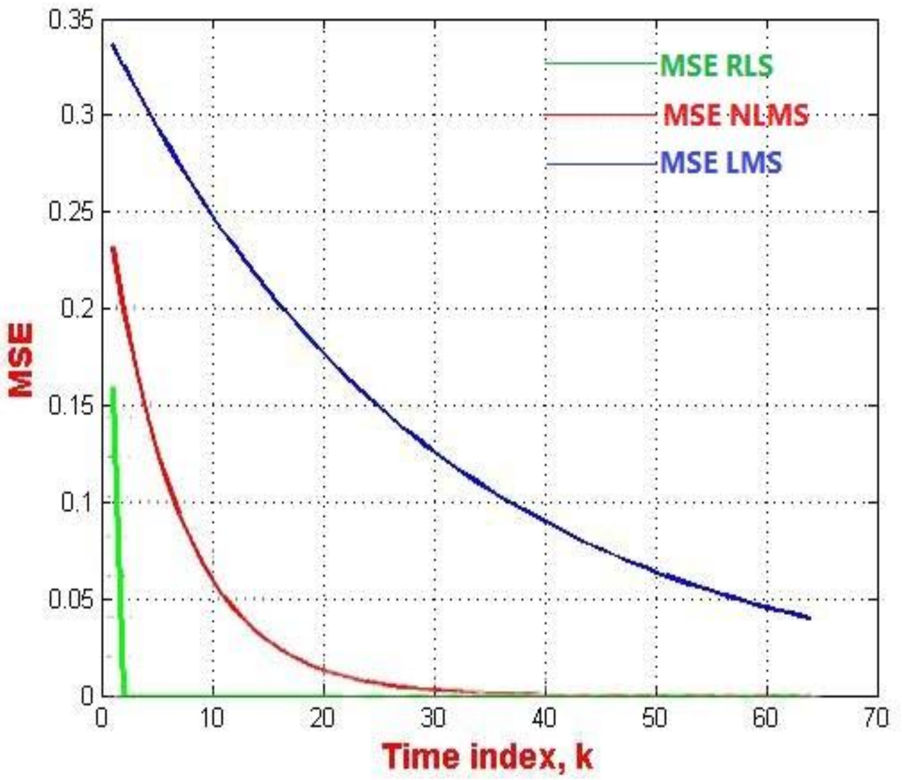

Figure 3.12 Comparison between MSE in the three algorithms

As seen in the Figure 3.12, that the lower value in MSE in the RLS algorithm, and the highest value in the LMS algorithm with regards to the convergence rate, the RLS algorithm has the fastest convergence rate and the LMS algorithm have the slowest algorithm.

From these results, as can be said, that the best algorithm is used in the application system identification is RLS algorithm.

3.2 Prediction

The purpose of using this application is to predict the value of the any signal (unknown characteristics, as well as an unknown environment). This application can be used to find the error of the measurement in the path of objects in the image also, video and all applications in this field [2]. In this part of the research have given a solution to the problem of prediction through the use of the three algorithms (LMS, NLMS and RLS).

3.2.1 The application

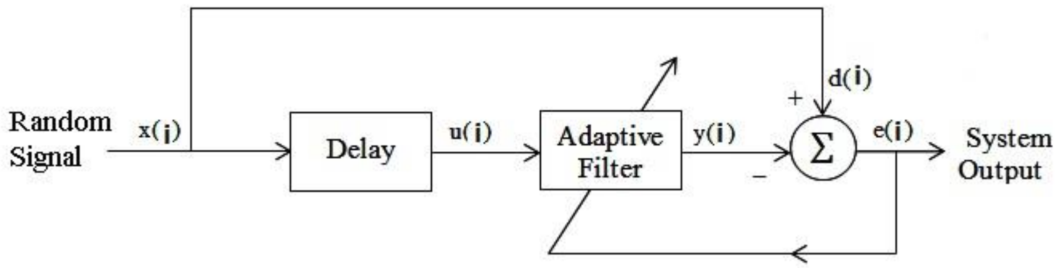

Figure 3.13 liner Prediction

Figure 3.13 illustrates the predictproblem.The purpose of this application is filtering a random signal and extract noise from this signal, from the Figure shown above, as noted that there is only one input signal.The input signal to the adaptive filter $u(i)$ is a signal output version delayed of the original signal (random signal) $x(i)$,the desired signal $d(i)$ is the same random signal $x(i)$, error signal produced when comparing the output signal of the adaptive filter with the desired signal.

3.2.2 Solution Using the LMS Algorithm

In paragraph 2.3.2 of Chapter two, as discussed the details of this algorithm (LMS), and now in this part the research discussing how to apply this algorithm to the problem of prediction, and the response of this algorithm to solve this problem. By notice the Figure 3.13 it can be observed that the input signal adaptive filter is $u(i)$, and the desired signal $d(i)$ is the same random signal $x(i)$, and thus get the following equations:

$$x(i) = d(i) \qquad 3.6$$

$$y(i) = w^L(i)u(i) \qquad 3.7$$

$$e(i) = x(i) - y(i) \qquad 3.8$$

$$w(i+1) = w(i) + 2\mu e(i)u(i) \qquad 3.9$$

Here Figure 3.14 shows problem (prediction), and how to solve this problem by using LMS algorithm and how to represent it using MATLAB program.

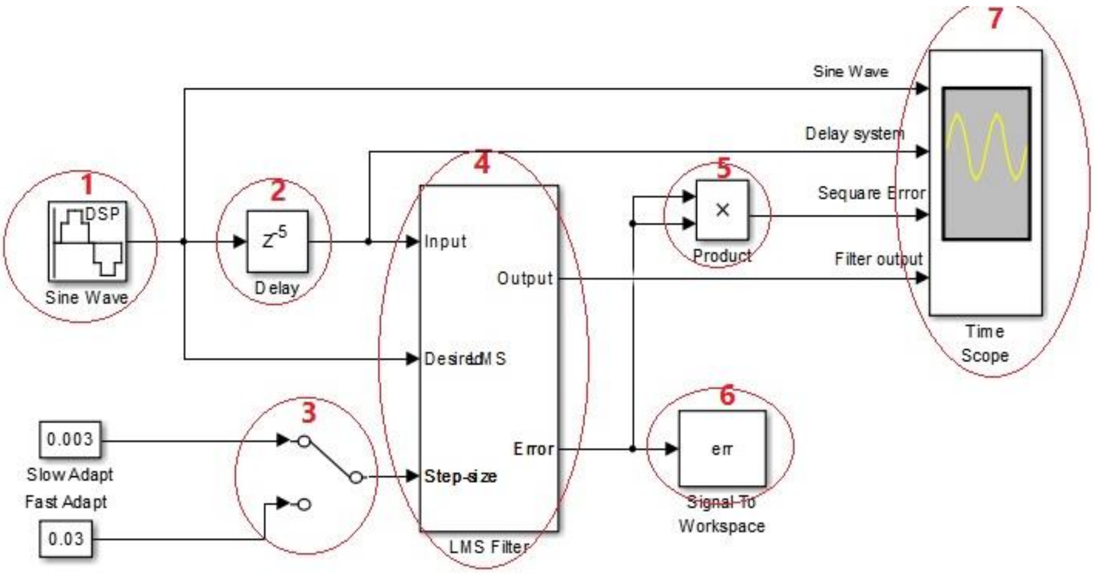

Figure 3.14 Solution using the LMS algorithm

Where

1- Is the input signal (sine wave) the amplitude of this signal =1 and the frequency =100 HZ.
2- Delay system and delay time = 0.005 second.
3- Manual switch to select the value of the step size (μ) in this case the value of μ =0. 003.
4- The adaptive filter (LMS,NLMS algorithm) blocks and the value of the filter length (q) =5.
5- Squared error, when comparing between the desired signal and the output signal of the adaptive filter produces an error signal.

6- Is a DSP tool called signal to work space block used for finding the Mean-squared error.

7- Time scope, used to display the result.

The results that got after running the Simulink MATLAB program shown in Figure 3.14

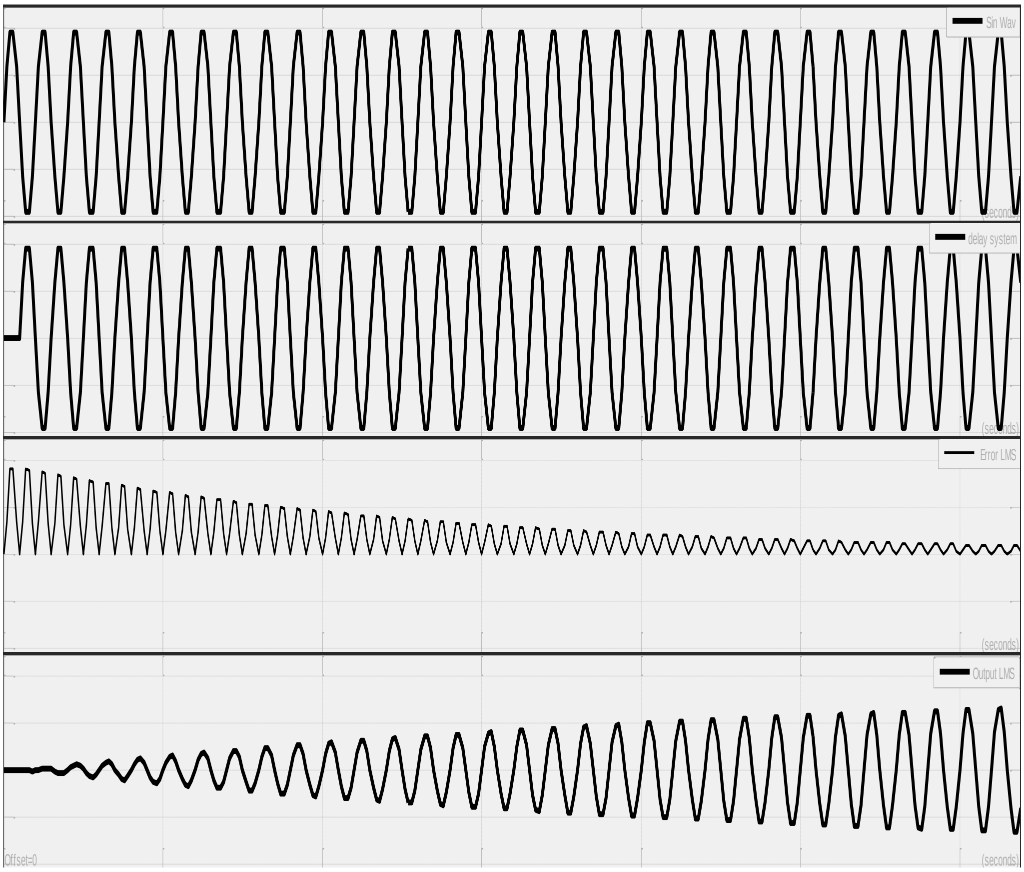

Figure 3.15 Pertinent signals when using the LMS algorithm to solve the Prediction problem.

Figure 3.15 shows the input signal $x(i)$ which represents the desired signal $d(i)$, and the output of the system delay which delays to 0.005 second, as well as squared error signal $e^2(i)$ in addition to the output signal of the filter $y(i)$, respectively, which is during the time of the implementation of 0.3 seconds. This is while taking into consideration the value of the length of the filter (the number of iterations) which is equal to 5.

In the figure 3.15 the square error, where the value of wave amplitude equal 1 at the time 0.002, then begins decreasing to become zero when the time equal 0.65.

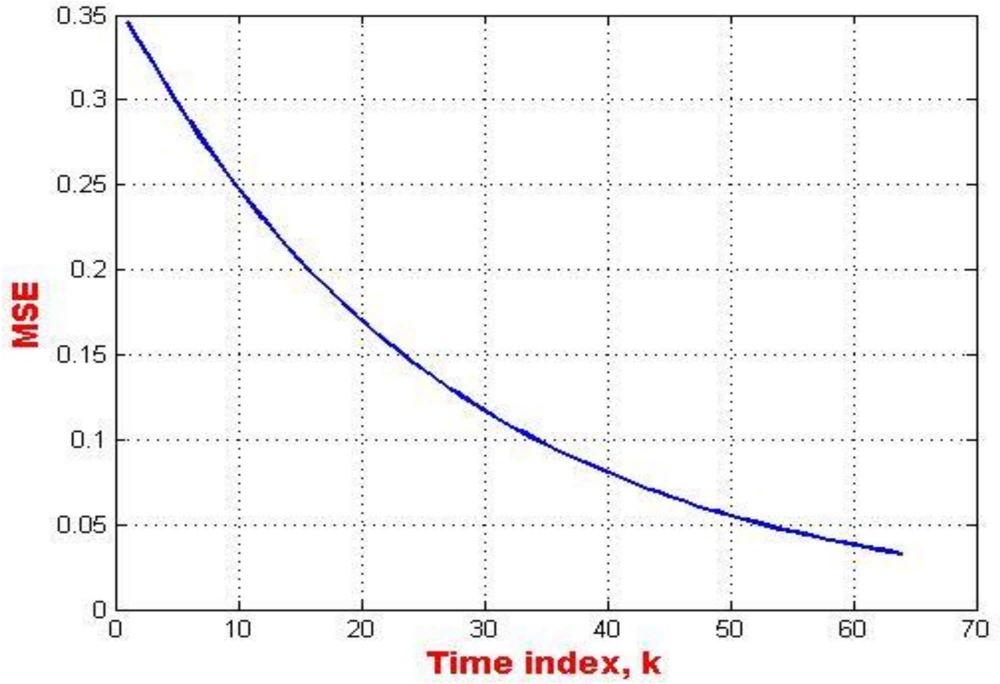

Figure 3.16 MSE of the LMS algorithm

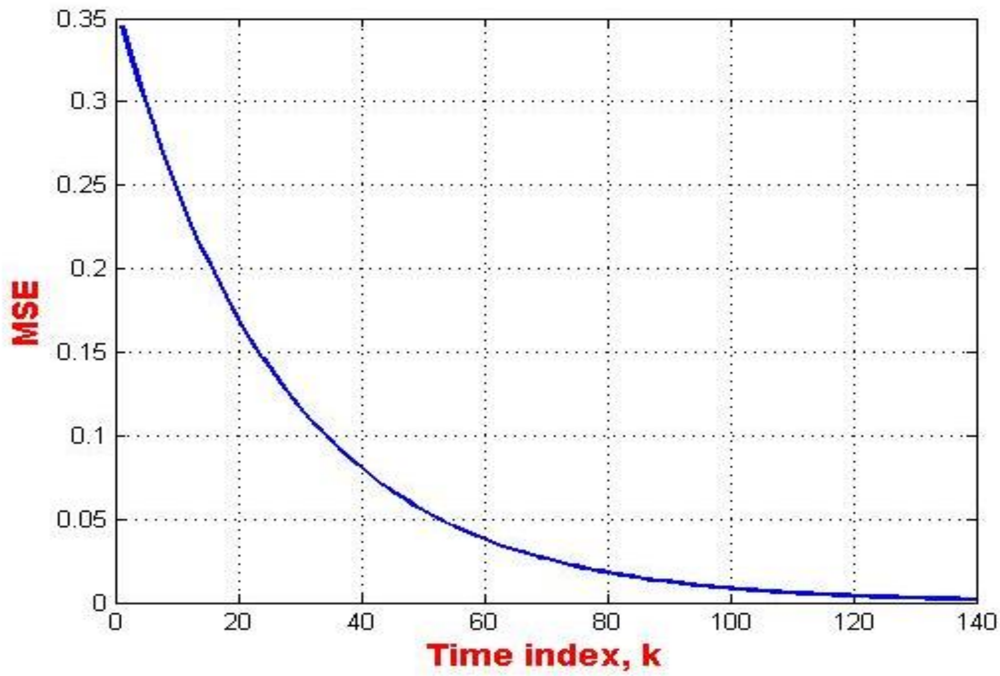

Figure 3.17 MSE of the LMS algorithm when implementation time is 0.8 second.

The Figure 3.17 represents the value of the MSE when implementation time is 0.8 second.

In Figure 3.16, were noticed that the value of the MSE starts at0.35 ,it is the highest value at the second 1. Notably, the value of the MSE is large as well as the convergence rate is very slow and this algorithm needs about 120 samples during execution time, bearing in mind that the execution time is 0.8 seconds, and this is one of the problems in this algorithm.

3.2.3 Solution Using the NLMS Algorithm

It is explained in Chapter two, paragraph 2.6.2, how the NLMS algorithm works and what teams between the LMS algorithm and the NLMS algorithm, when applying the NLMS algorithm to the problem of predict, must take into account some of the changes that have taken place in the equations calculating of the parameters. After conducting these updates on the equations in order to adapt to address the problem of prediction, getting to these equations:

$$x(i) = d(i) \quad 3.10$$

$$y(i) = w^{L}(i)u(i) \quad 3.11$$

$$e(i) = x(i) - y(i) \quad 3.12$$

$$\mu = \frac{\bar{\mu}}{\delta + \|x(i)\|^2} \quad 3.13$$

$$w(i+1) = w(i) + 2\mu e(i)u(i) = w(i) + 2\frac{\bar{\mu}}{\delta + \|x(i)\|^2}e(i)u(i) \quad 3.14$$

Figure 3.14 shows how to represent the LMS algorithm in the Simulink MATLAB program, the difference in the representation of NLMS algorithm is the step size (μ), where the size of the step in the NLMS algorithm is equal to 0.03, and value of the constant is δ =1.0.

After running the Simulink MATLAB program, could see the results that obtained represented in Figure 3.19

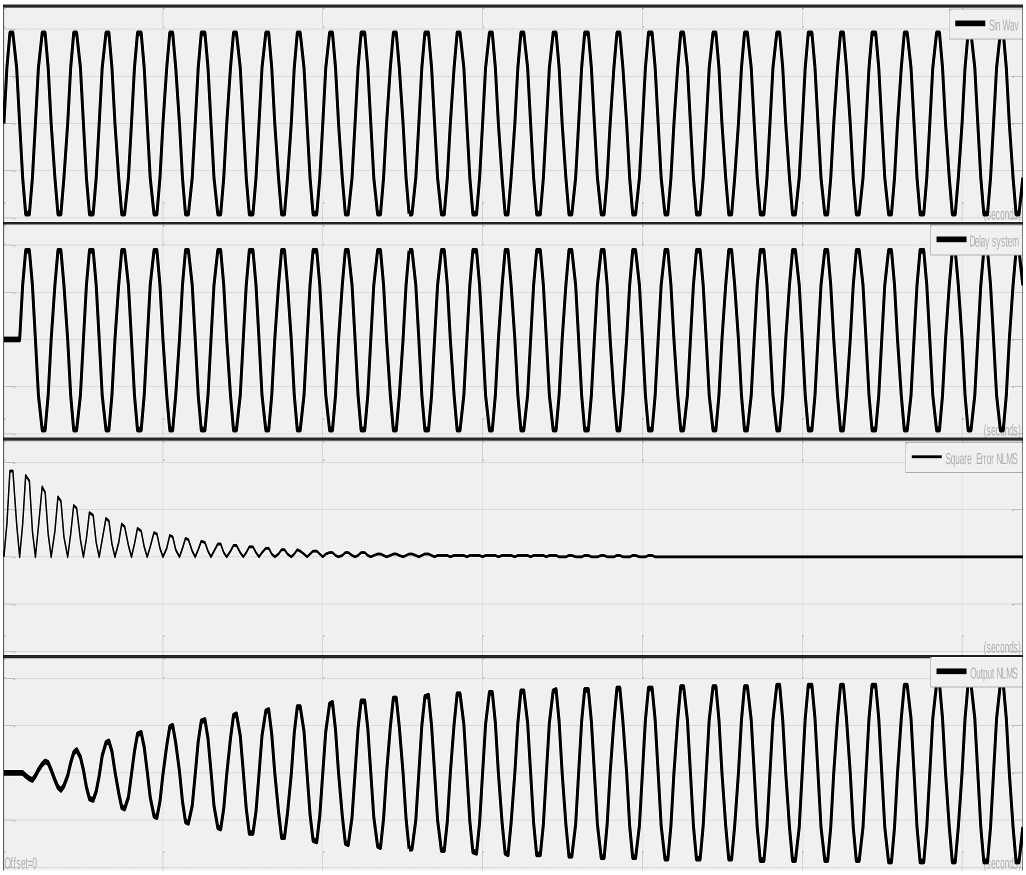

Figure 3.18 Pertinent signals when using the NLMS algorithm to solve the Prediction problem

This Figure 3.18 shows the input signal $x(i)$ which represents the desired signal $d(i)$, and the output of the system delay which delays to 0.005 second, as well as squared error signal $e^2(i)$ in addition to the output signal of the filter $y(i)$, respectively, particularly during the time of the implementation of 0.3 seconds. This is while taking into consideration the value of the length of the filter (the number of iterations) which is equal to 5.

In the Figure 3.18 the square error, where the value of wave amplitude equals almost 1 at the time of 0.002 second, then begins to decrease to become zero when the time equals to 0.15.

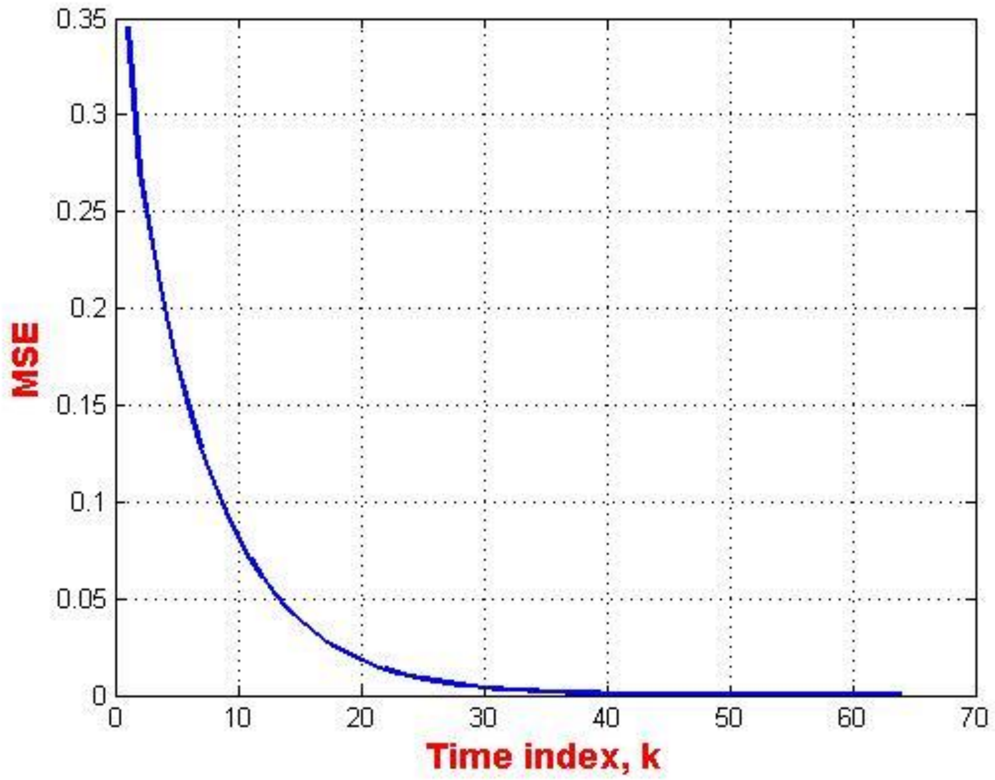

Figure 3.19 MSE of the NLMS algorithm

In Figure 3.19 as noted that the value of MSE is 0.35, here the purpose of finding a cost function, the mean-squared error is to see how the algorithm responses to reach the optimum limit. When analyzing Figure 3.19, could easily discover that the convergence rate is of the average speed and this algorithm needs about 25 samples during execution time, knowing that the execution time is 0.3 seconds.

3.2.4 Solution Using the RLS Algorithm

In this paragraph, the RLS algorithm has been applied to solve the problem of Predict, after it has applied algorithms (LMS, NLMS), and that is done for the application of this algorithm and to extract the results correctly. Actually, it must adapt equations until get the correct results [15], and the equations for these are

$$x(i) = d(i) \qquad 3.15$$

$$g(i) = \frac{\lambda^{-1}\Phi_A^{-1}(i-1)u(i)}{1+\lambda^{-1}u^L(i)\Phi_A^{-1}(i-1)u(i)} \quad 3.16$$

$$y(i) = w^L(i)u(i) \quad 3.17$$

$$e(i) = x(i) - y(i) \quad 3.18$$

$$w(i+1) = w(i) + g(i)e^*(i) \quad 3.19$$

$$\Phi_A^{-1}(i) = \lambda^{-1}\Phi_A^{-1}(i-1) - \lambda^{-1}g(i)u^L(i)\Phi_A^{-1}(i-1) \quad 3.20$$

Where described all variables in paragraph 2.3.4 of Chapter two

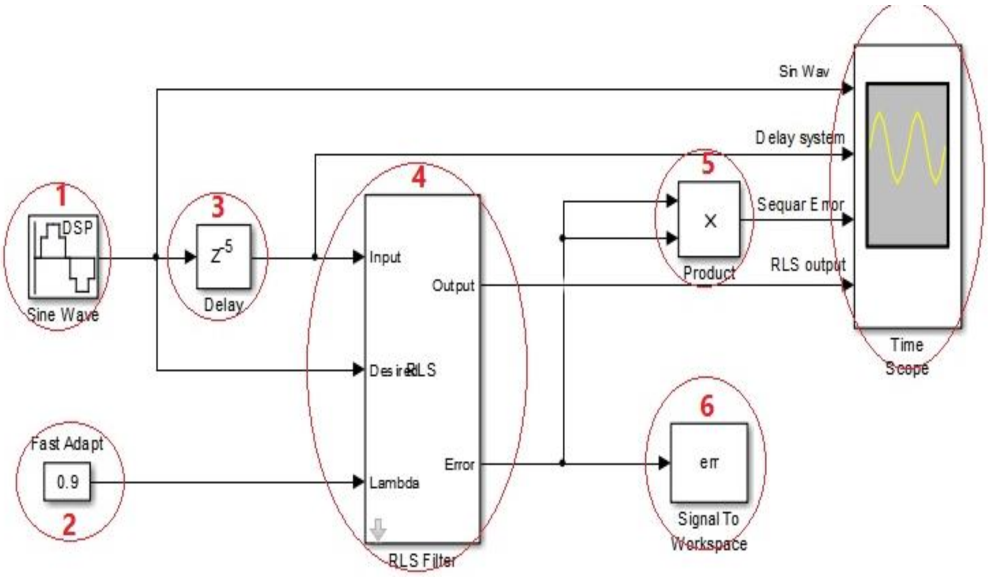

Figure 3.20 Solution using the RLS algorithm

Where

1- Is the input signal (sine wave) the amplitude of this signal =1 and the frequency =100 HZ.
2- Lambda value $\lambda = 0.9$.
3- Delay system and delay time is =0. 005 second.
4- The adaptive filter (RLS algorithm) blocks and the value of the filter length is (q) =5.
5- Squared error, when comparing between the desired signal and the output signal of the adaptive filter produces an error signal.
6- Is a DSP tool called signal to work space block used for finding the Mean-squared error.
7- Time scope, used to display the result.

The results that got after running the Simulink MATLAB program shown in Figure 3.21 given below.

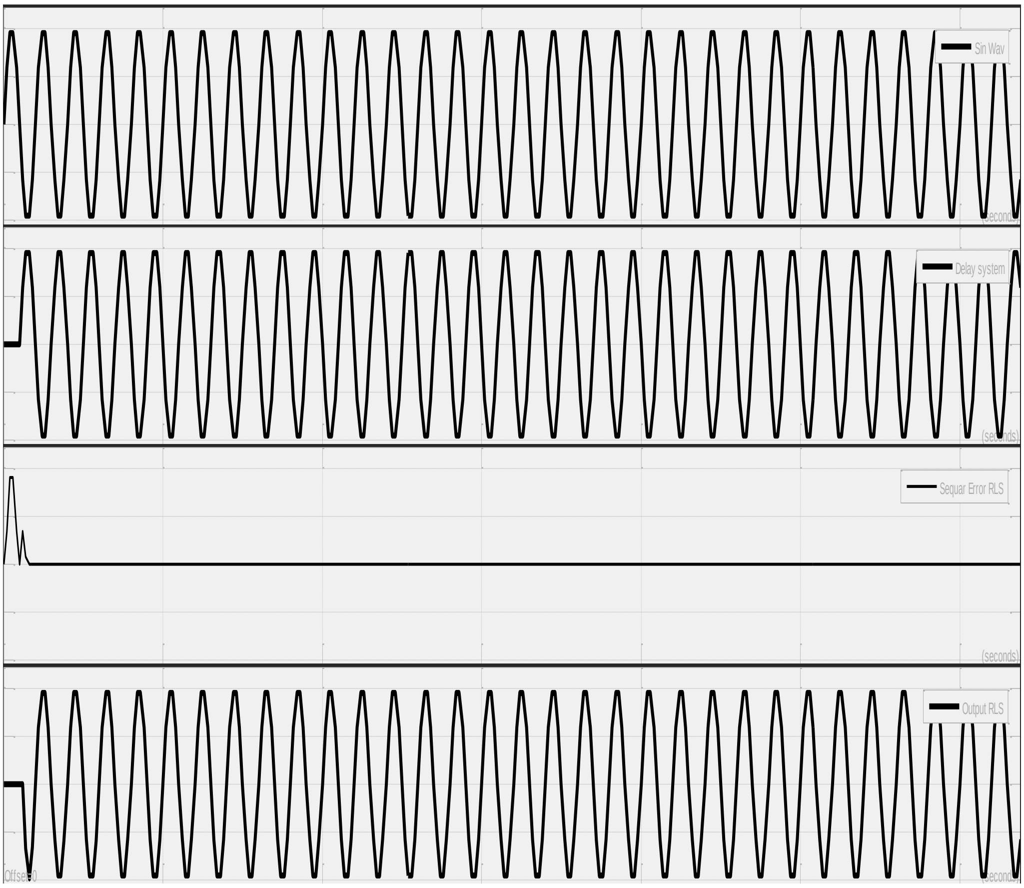

Figure 3.21 Pertinent signals when using the RLS algorithm to solve the Prediction problem

Figure 3.21 shows the input signal $x(i)$ which represents the desired signal $d(i)$, and the output of the system which is delayed and the delay value is 0.005 second, as well as the squared error signal $e^2(i)$ in addition to the output signal of the filter $y(i)$, respectively, particularly during the time of the implementation of 0.3 seconds. While taking into consideration the value of the length of the filter (the number of iterations) which is equal to 5. In the Figure 3.21 the squared error, is starting from zero and increasing to larger amplitude in the 0.002second where the value of wave amplitude is closer to 1, then it begins decreasing to zero when the time equals 0.038, after that, increasing amplitude until it becomes 0.4 at the 0.006 second and then begins to decline until reaching zero in the 0.007 second.

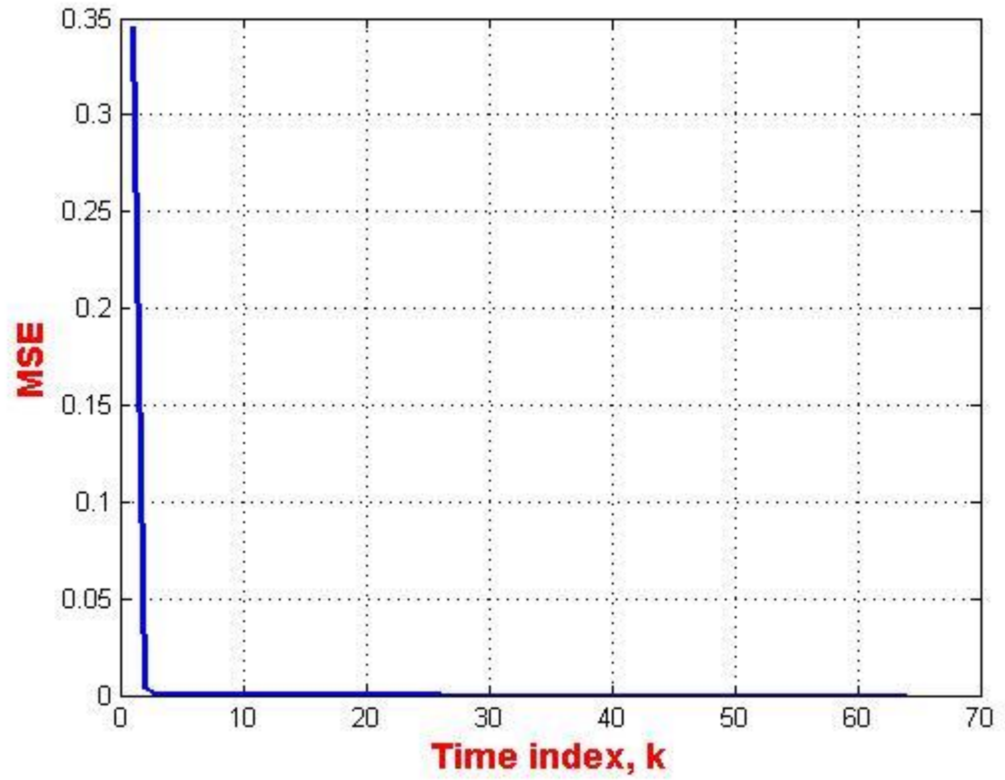

Figure 3.22 MSE of the RLS algorithm.

In the Figure 3.22, in this algorithm, as noted, that the rate of convergence is very fast, where the value of the rate of convergence is at this almost 2.5 algorithm, meaning that it needs 2.5 samples during execution time, knowing that the execution time is 0.3 seconds.

3.2.5 Comparison of Results

The analysis of the results and choice ofthe best algorithm to solve the problem of predict requires taking into account the factors mentioned in paragraph 3.1.5, one of these factors is the rate of convergence, cost and tracking.Figure 3.23 shows the comparison between MSE andthe three algorithms (LMS, NLMS, RLS).as noted that it is very easy to find any of these algorithms having a faster convergence rate and the algorithms having the slow convergence rate.as noted that the rate of convergence of LMS algorithm is very slowas compared toother algorithms.At the same execution time, the rate of congruence in the LMS algorithm takes four times more than it in the NLMS algorithm, and takes 100 times in the RLS algorithm. The second factor is the cost, the RLS algorithm is more expensive because it has many mathematical operations and this is one of the most important disadvantages of algorithm RLS.

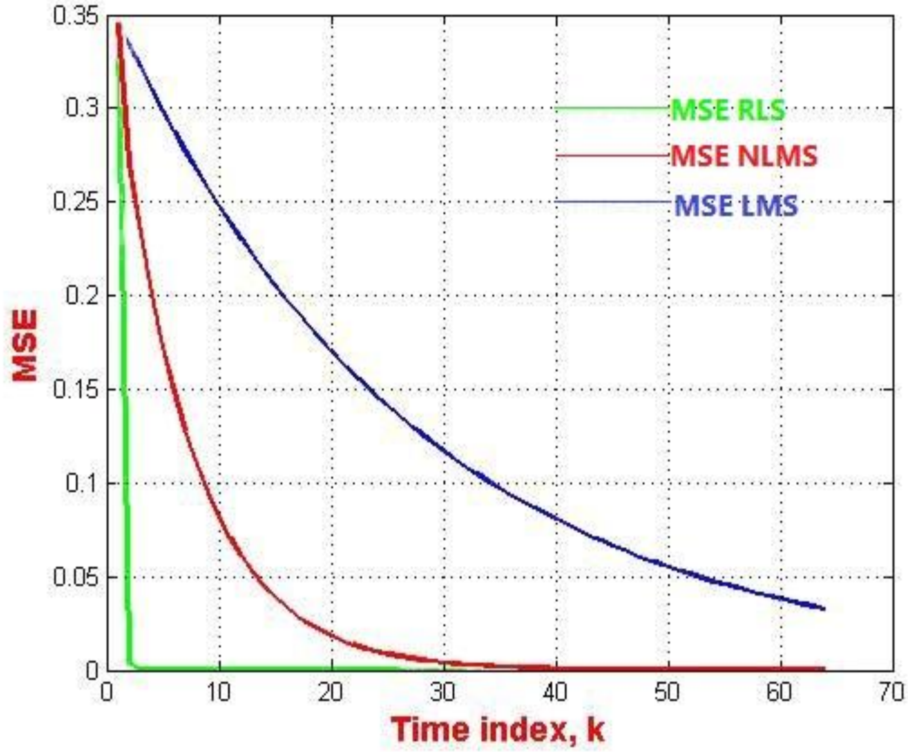

Figure 3.23 Comparison between MSE and algorithms LMS, NLMS and RLS

Figure 3.24 shows the comparison between the error of the (LMS, NLMS and RLS)algorithms, and this is the third factor which is tracked when it noted that there is less error in the RLS algorithm, which maintains on the properties of the signal.

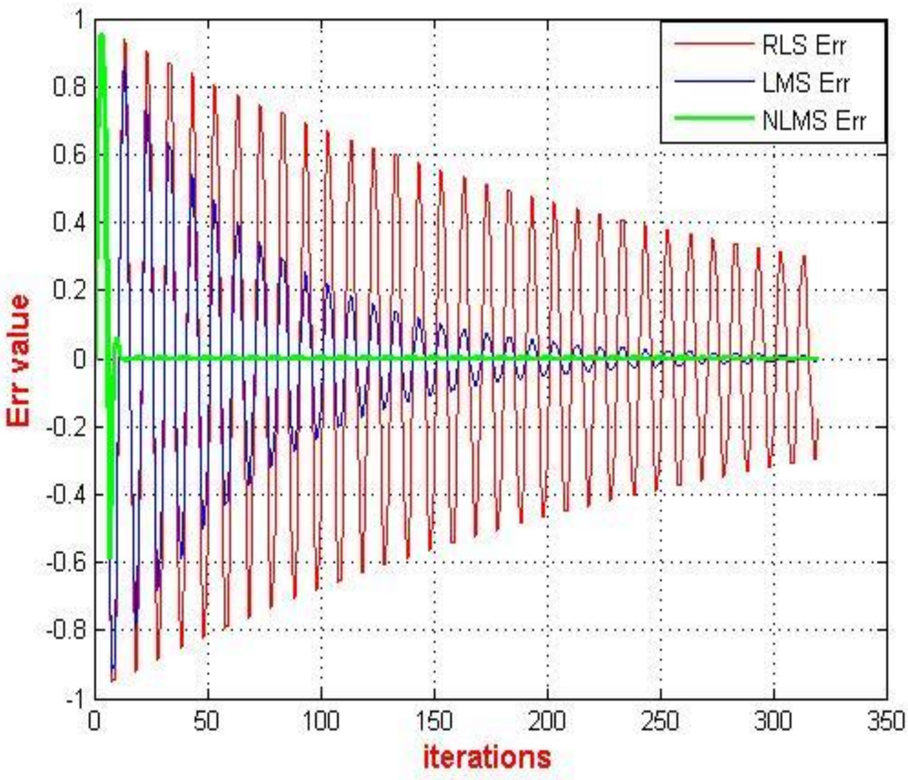

Figure 3.24 Comparison between the error signal in the three algorithms

3.3 Inverse Modelling

In Chapter two, applications of adaptive filters are discussed and described. Along with the uses of this application discussed. The purpose of using this application is tracking and discovering the feedback transfer task of the system and cancelling out noise from the signals to improve Signal to Noise Ratio (SNR) [16].

3.3.1 The Application

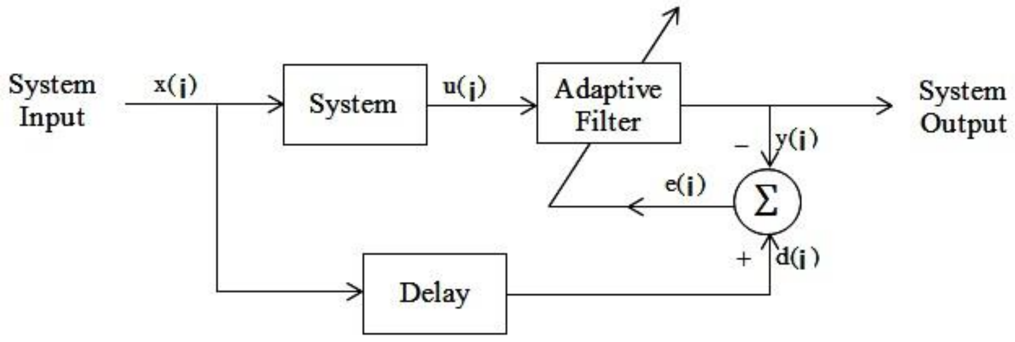

Figure 3.25 Inverse Modeling

In this case the input of adaptive filter u(i) is the output of the unknown system, the desired signal (i) is the same of the original signal and the input of the unknown system x(i),(but that consists of delayed version). When comparing between the d(i) and the output of the adaptive filter, then generation is $e(i)$. Three algorithms (LMS, NLMS and RLS) are used in this application.

3.3.2 Solution Using the LMS Algorithm

In paragraph 2.6.1 of the chapter two, has been discussing the details of this algorithm (LMS), and this part will discuss how to apply this algorithm to the problem of prediction, and the response of this algorithm to solve this same problem.

In this case the equations which are used to calculate the parameter of adaptive filter rely on the type of unkown system and the value of delay systems. In this case the value of the delay system is 0.005sec, that means the value of the delay time is 0.005sec.

Figure 3.26 shows the problem (Inverse Modeling), and how to solve this problem by using LMS algorithm and how to represent it by using MATLAB program.

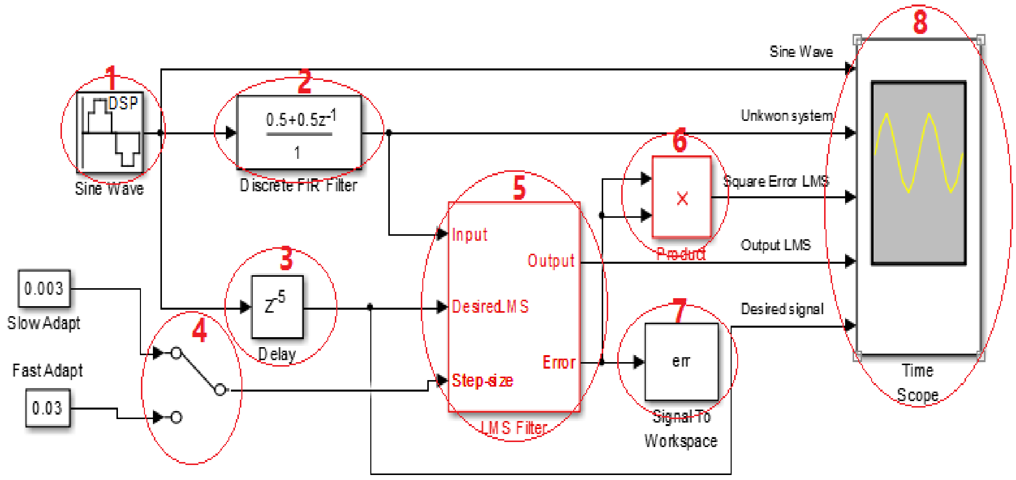

Figure 3.26 Solution using the LMS algorithm

Where

1- Is the input signal (sine wave) the amplitude of this signal =1 and the frequency =100 HZ.
2- Unknown system
3- Delay system and delay time = 0.005 second.
4- Manual switch to select the value of the step size (μ) in this case the value of μ =0. 003.
5- The adaptive filter (LMS, NLMS algorithm) blocks and the value of the filter length is (q) =5.
6- Squared error, when comparing between the desired signal and the output signal of the adaptive filter that produces an error signal.
7- Is a DSP tool called signal to work space block used for finding the Mean-squared error.
8- Time scope, used to display the result.

The results that got after running the Simulink MATLAB program shown in Figure 3.27.

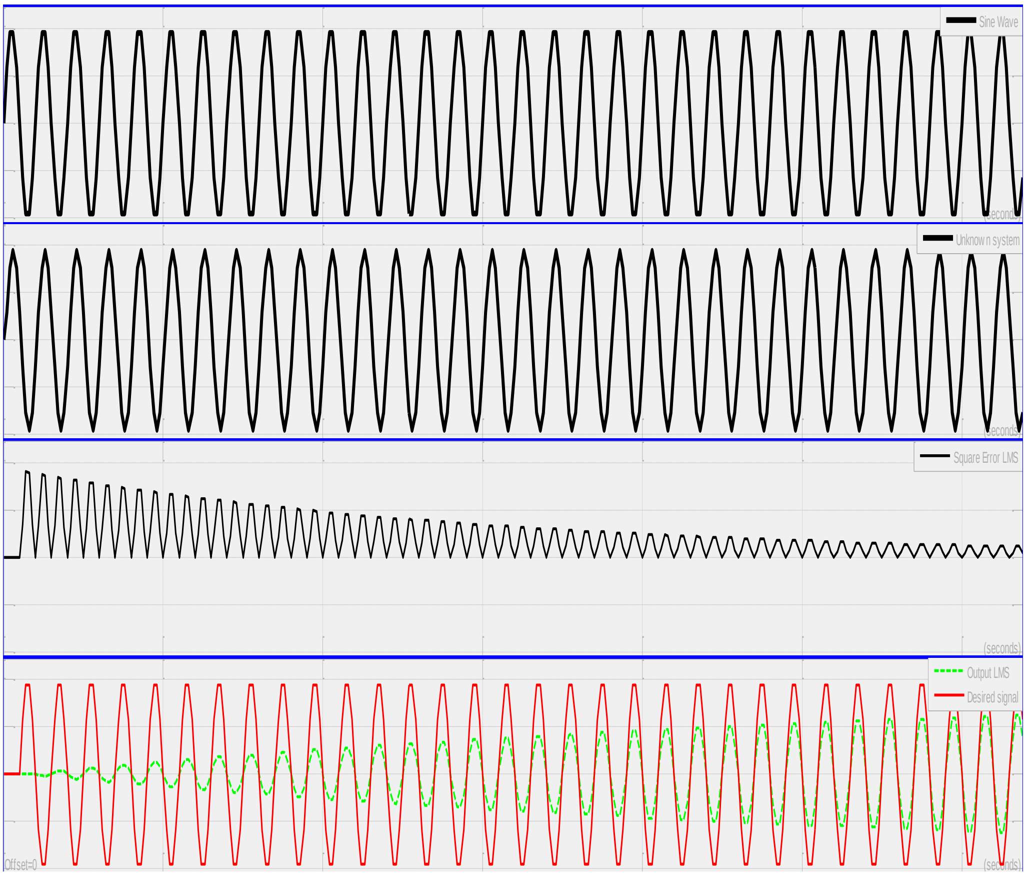

Figure 3.27 Pertinent signals when using the LMS algorithm to solve the Inverse Modeling problem

Figure 3.27 shows all of the input signal $x(i)$ and the input of the adaptive filter which represents the system output signal, as well as squared error signal $e^2(i)$ in addition to the desired signal $d(i)$ which represents the delay output signal. Finally the output signal of the filter is $y(i)$, during the time of the implementation of 0.3 seconds respectively,. While taking into consideration the value of the length of the filter (the number of iterations) which is equal to 5. In the Figure 3.27, in the start of the error squared signal as noted the presence of 0.005 second delay, after that the value of wave amplitude equals 1. Then starts decreasing until it becomes zero when the time equals 0.7.

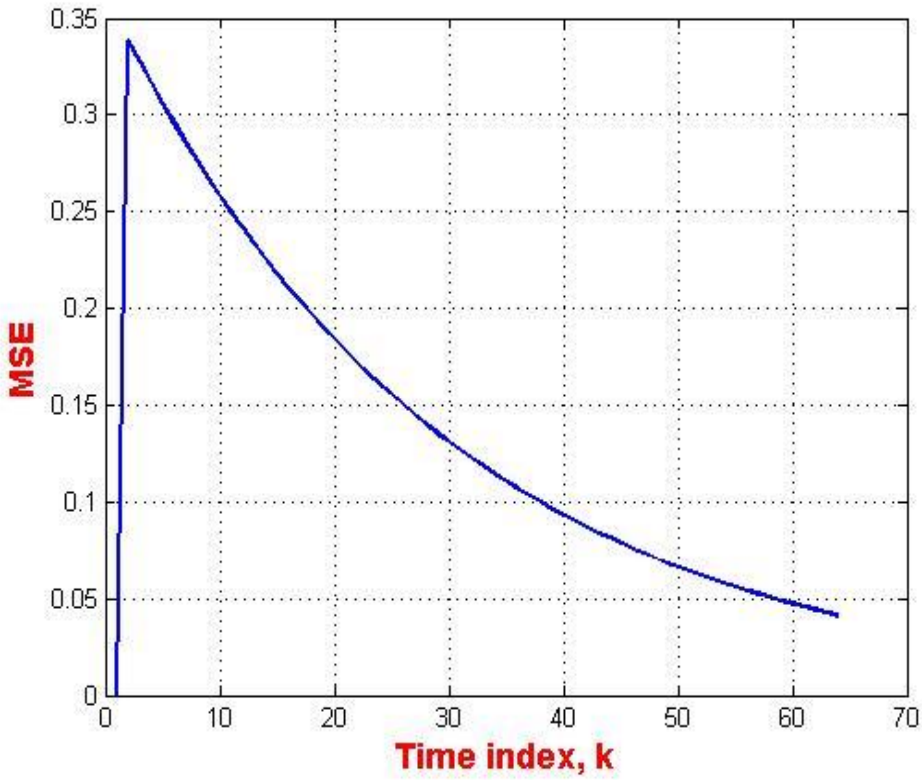

Figure 3.28 MSE of the LMS algorithm

In Figure 3.28, as noted the value of MSE started at 0.005 second because of the delay of the desired signal. The highest value of MSE is 0.34, which as noted in this figure where the convergence rate is very slow and gets congruence between the parameters after almost 120 model during execution time.

3.3.3 Solution Using the NLMS Algorithm

To solve the problem of the slowness of the convergence rate in the LMS algorithm and developing the NLMS algorithm, the paragraph 2.6.2 discusses how to avoid this problem. In the inverse modeling application, the parameters in the adaptive filter depend on the value of the system and the value of system delay.

Figure 3.26 shows how to represent the LMS algorithm in the Simulink MATLAB program and the difference in the representation of NLMS algorithm is a step size (μ), where the size of the step in the NLMS algorithm is equal to 0.03, and value of the constant δ =1.0.

After running the Simulink MATLAB program, get the results as shown in Figure 3.29 below.

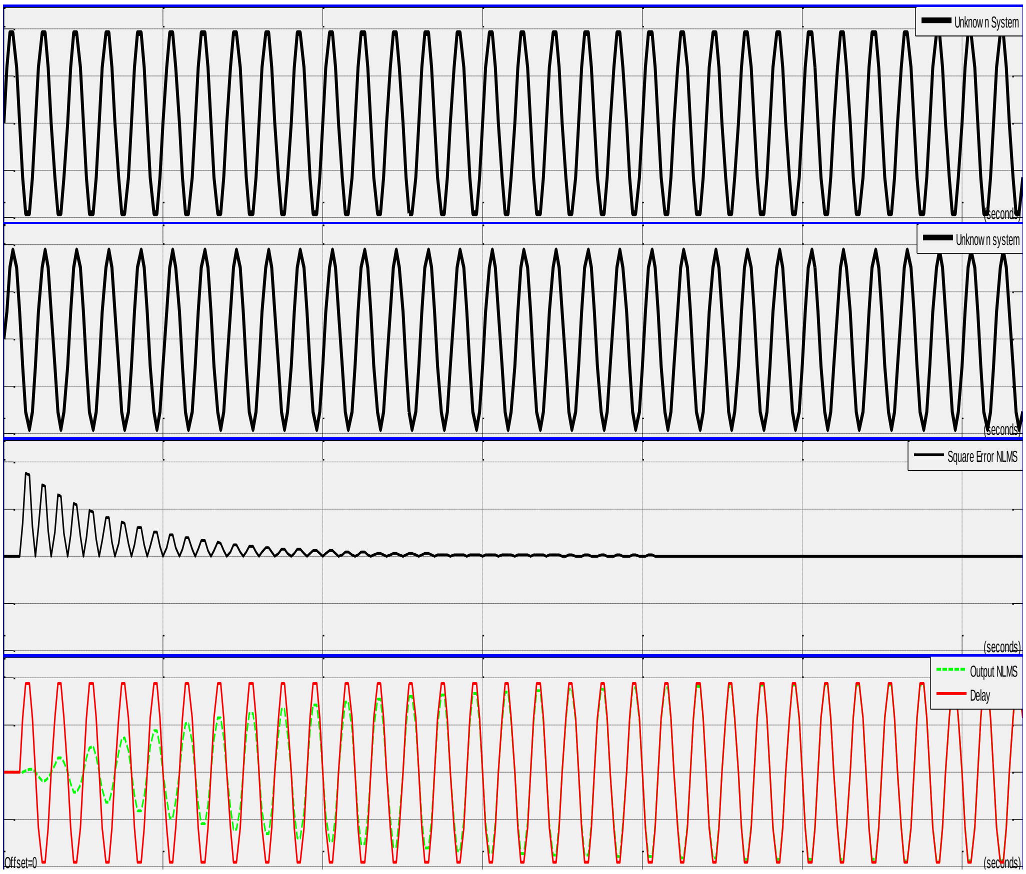

Figure 3.29 Pertinent signals when using the NLMS algorithm to solve the Inverse Modelling problem.

Figure 3.29 shows the input signal $x(i)$ and the input of the adaptive filter which represents the system output signal, as well as squared error signal $e^2(i)$ in addition to the desired signal $d(i)$which represents the delay output signal. And finally, the output signal of the filter $\mathbf{y}(\mathrm{i})$, respectively, during the time of the implementation of 0.3 seconds. While taking into consideration the value of the length of the filter (the number of iterations) which is equal to 5 and implementation time is 0.3Sec.

In the figure 3.29, in the start of the error squared signal, as noted the presence of 0.005 second delay, after that the value of wave amplitude equal 1. Then it decreasing to become zero when the time equal 0.145.

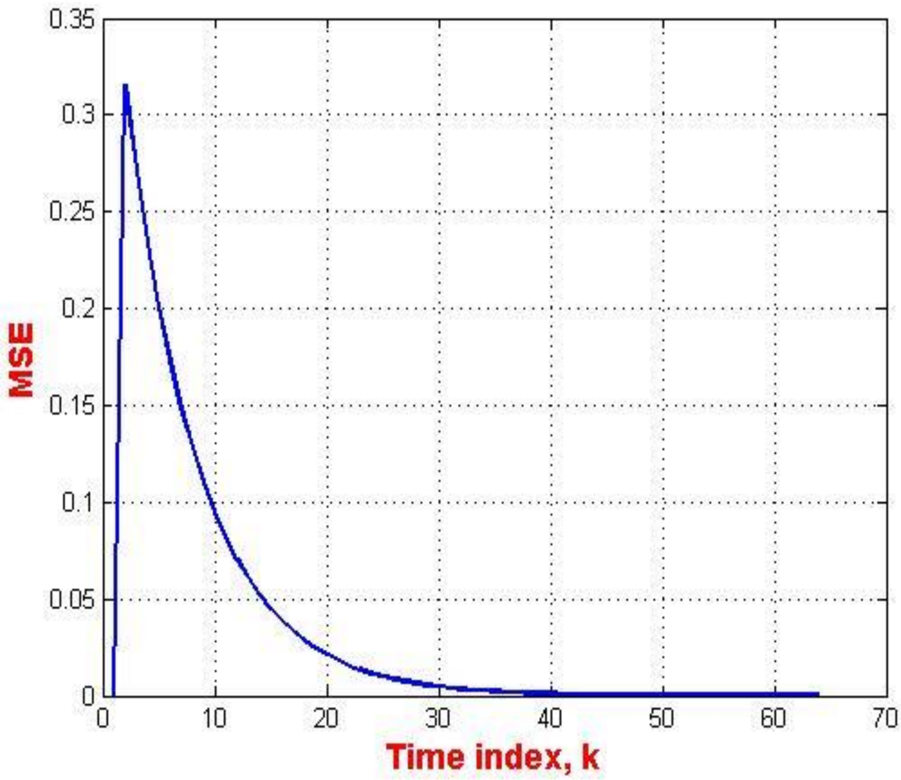

Figure 3.30 MSE of the NLMS algorithm

In Figure 3.30, as noted the value of the MSE started in the 0.005 second because the delay of the desired signal, and the value of MSE is 0.32, that the purpose of finding a cost function. The mean-squared error is to see how the algorithm gives response to reach the optimum limit. When analyzing Figure 3.29, as found that the convergence rate is the average speed and this algorithm needs about 30samples during execution time, knowing that the execution time is 0.3 seconds.

3.3.4 Solution Using the RLS Algorithm

The RLS algorithm is a very important algorithm among the adaptive filter algorithms. This algorithm is very complex, because it includes a lot of equations, which is one of the disadvantages of the RLS algorithm, but the advantage of this algorithm is that it is very fast in the convergence rate. The figure 3.31 shows how it can represent this algorithm in the Simulink MATLAB program.

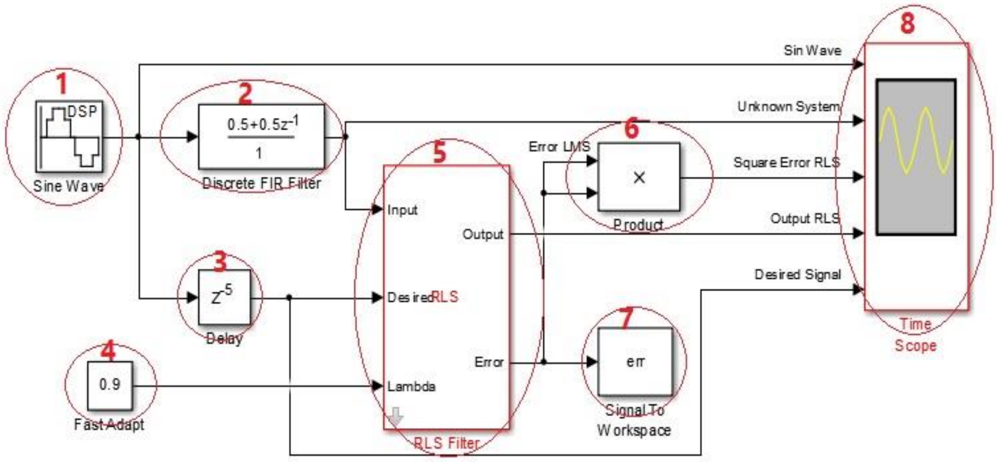

Figure 3.31 Solution using the RLS algorithm

Where

1- Is the input signal (sine wave).The amplitude of this signal is 1 and the frequency is 100 HZ.
2- Unknown system
3- Delay system and delay time = 0.005 second.
4- Lambda value $\lambda = 0{,}9$.
5- The adaptive filter (LMS, NLMS algorithm) blocks and the value of the filter length (q) =5.
6- Squared error, when comparing between the desired signal and the output signal of the adaptive filter produces an error signal.
7- Is a DSP tool called signal to work space block used for finding the Mean-squared error.
8- Time scope, used to display the result.

The results that got after running the Simulink MATLAB program shown in Figure 3.32.

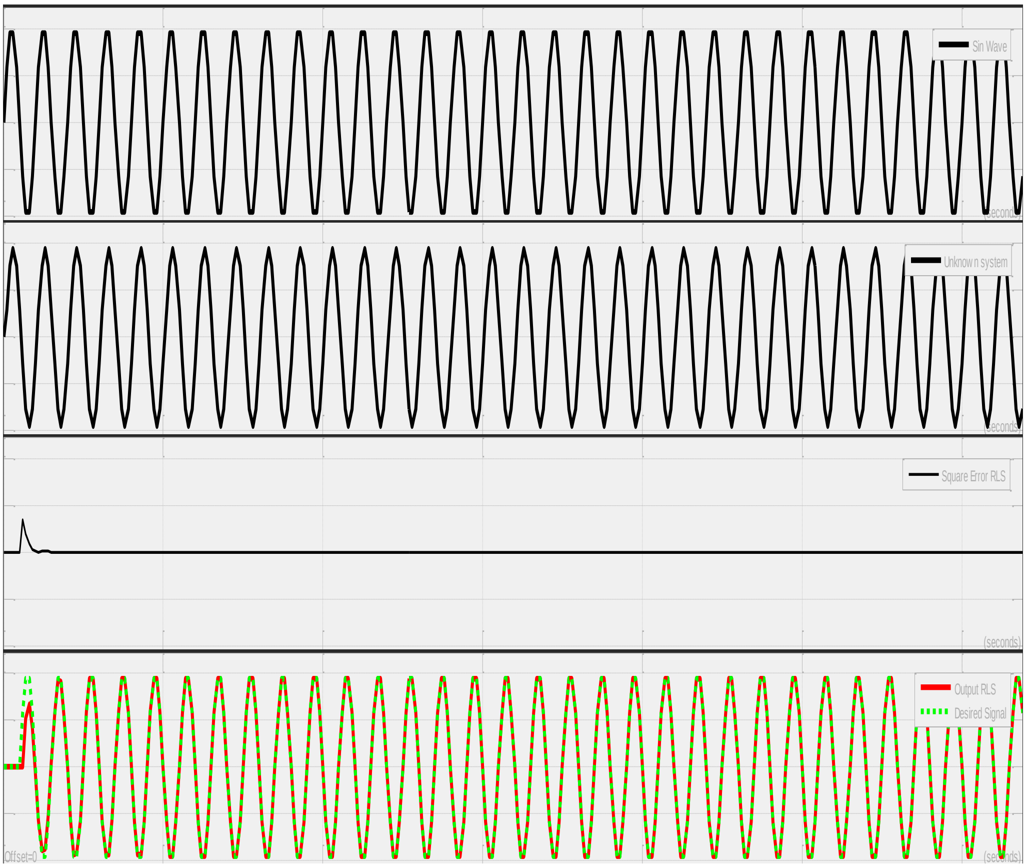

Figure 3.32 Pertinent signals when using the RLS algorithm to solve the Inverse Modelling problem.

In the Figure 3.32, the first signal is the original signal (sine wave), it represents the input signal; $x(i)$.The second signal is the unknown system's output; $\mathrm{u(i)}$, which representsthe input of adaptive filters, and the third signal is the squared error, $e^2(i)$. Finally,the two signals- the yellow signal is the output of the adaptive filter; $y(i)$, and the red signal is the output of the system delay that is delayed by 5msecand represents the desired signal; $d(i)$. It must take in consideration the value of the filter's length (the number of iterations) which is here equal to 5.

The squared error is delayed by 5 msec due to the affect of the delay system. The squared error value and the time is very small, the amplitude of the squared error is 0.45 and the time of the squared error is 0.04 Sec.

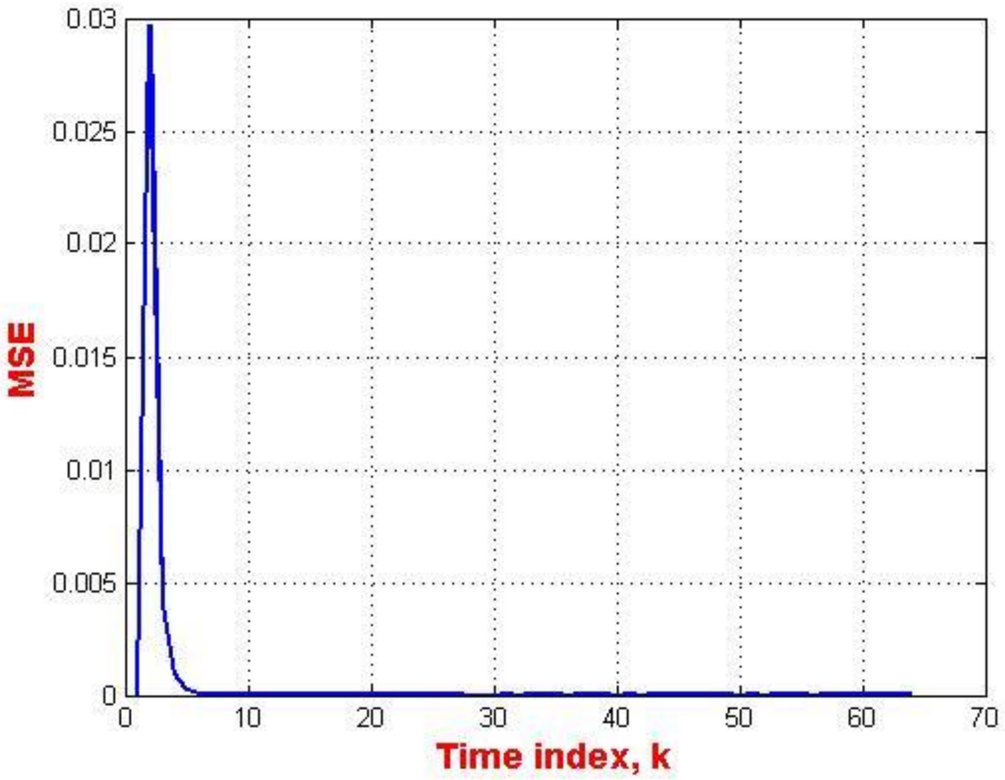

Figure 3.33 MSE of the RLS algorithm

Figure 3.33 shows the MSE of the RLS algorithm. In this figure as noted the value of MSE which started at time 0.005 Sec because of the delay system, and that the value of the MSE is very small, 0.03. Besides the value of convergence rate is 6 samples in the implementation time.

3.3.5 Comparison of Results

So far, it has been discussed in the paragraph 3.1.5 and 3.2.5 the factors which can be analyzed in order to know which algorithm is the best among the three algorithms. The factors are the convergence of rate, the cost, and the error in these algorithms.

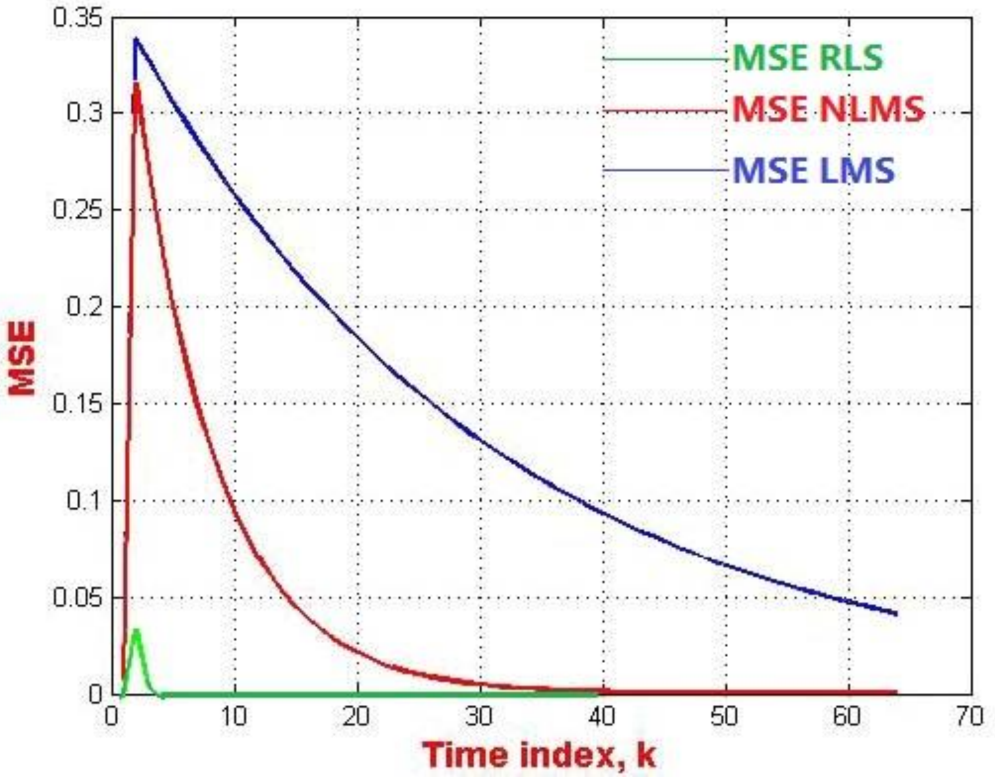

Figure 3.34 Comparison between MSE for the algorithms LMS, NLMS and RLS.

In Figure 3.34, as noted the value of MSE, the smallest value of the MSE in RLS algorithm is 0.03,and it is notable that the rate of convergence of LMS algorithm is very slow compared with other algorithms. At the same execution time, the rate of congruence in the LMS algorithm takes twice the time of the NLMS algorithm, andtakes200 times the in the RLS algorithm’s time.

Additionally, the Figure 3.35 shows the comparison of the error value in the three algorithms, the lowest error value in the RLSis 0.6, at time=0.05 Sec.

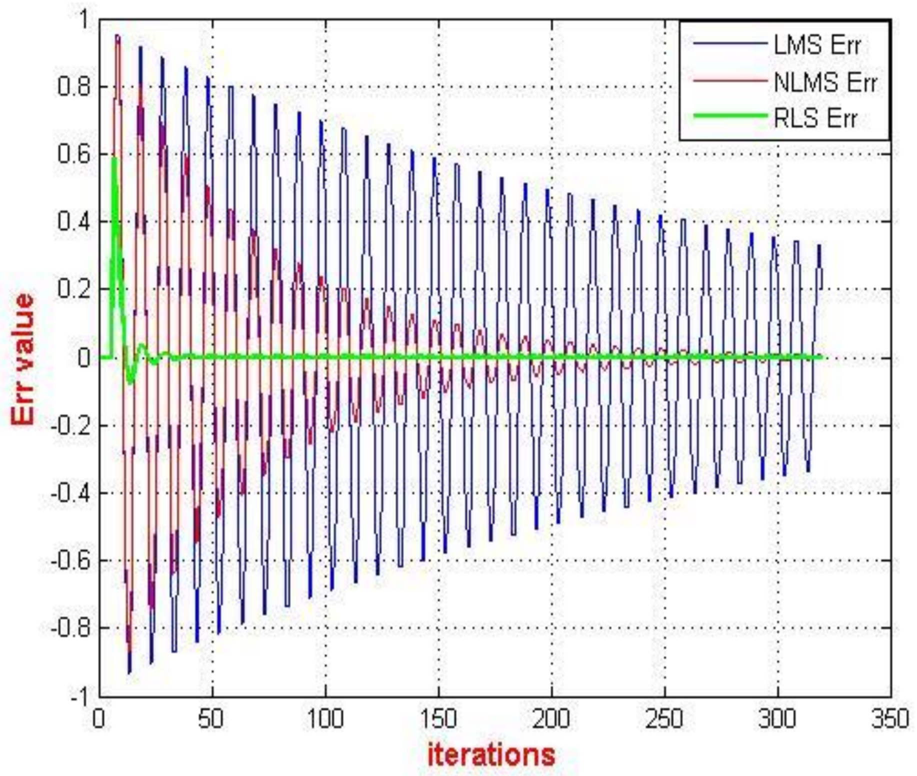

Figure 3.35 Comparison between the error signal in the three algorithms

3.4 Jammer Suppress

In this section, the research discusses the most important application in the adaptive filter, which is the Interference cancelling. This application is used in noise cancelling. In fact, the noise cancelling becomes very important at this time because of the evolution of communication devices. Figure 3.36 shows the interference cancelling.

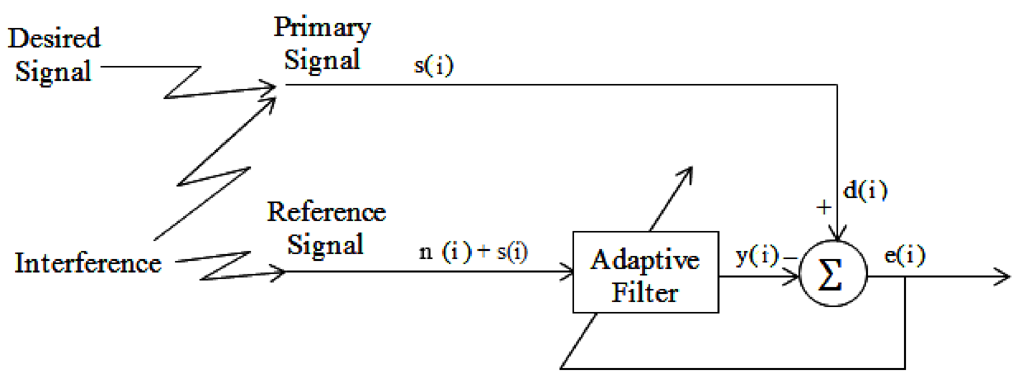

Figure 3.36 Jammer suppress

This figure shows the Interference Cancelling. The primary signal contains the source signal $x(i)$, which represents the desired signal $d(i)$. The reference signal contains the source signal $x(i)$, and the noise signal $n(i)$ is added. The $y(i)$ represents the output of the adaptive filter, after comparing the output of adaptive filter $y(i)$ and the desired signal $d(i)$, which produces the error $e(i)$.

3.4.1 Solution Using the LMS Algorithm

In this case the research uses the LMS algorithm to solve the noise cancelling problem. It is easy to calculate the LMS algorithm's parameters. This equation is to calculate the parameters [7] [20]:

$$x(i) = d(i) \qquad 3.21$$

$$y(n) = w^{L}(i)x(i) \qquad 3.22$$

$$e(i) = y(i) - x(i) \qquad 3.23$$

$$w(n+1) = w(n) + 2\mu e(i)x(i) \qquad 3.24$$

Figure 3.37 shows problem (Interference Cancelling), how to solve this problem by using LMS algorithm, and how to represent it using MATLAB program.

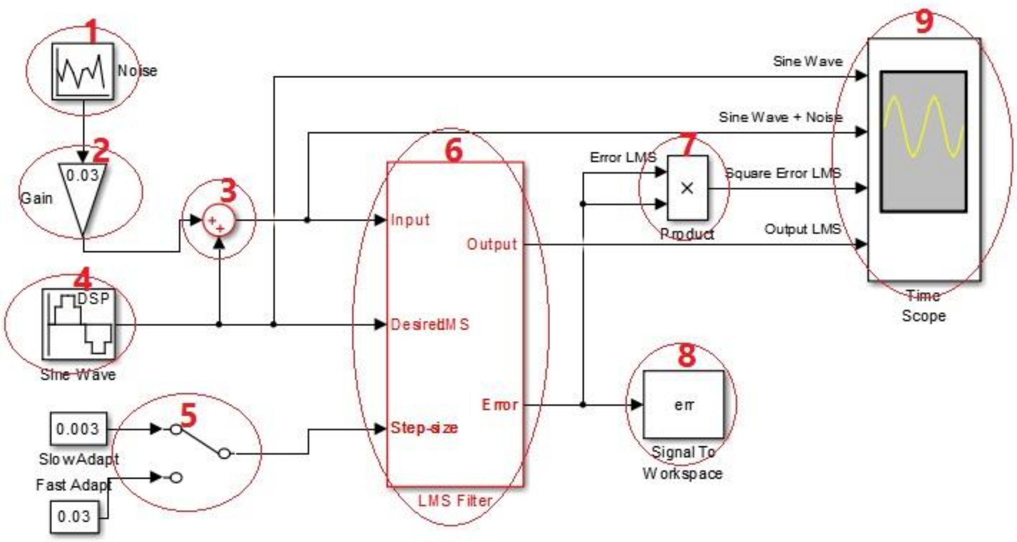

Figure 3.37 Solution using the LMS algorithm

Where:

1- Is the noise source, which is here Gaussian noise (white noise).

2- The Gain is used to normalize the noise in this Simulink by multiplying the Gaussian noise by the factor of 0.03.
3- Sum, is used to sum up two signals or more, in this case it is used to add the noise signal to the sine wave.
4- The input signal (sine wave) the amplitude of this signal is 1 and the frequency is 100 HZ.
5- Manual switch to select the value of the step size (μ) in this case the value of μ is 0. 003.
6- The adaptive filter (LMS,NLMS algorithm) blocks and the value of the filter length (q) =5.
7- Squared error, when comparing between the desired signal and the output signal of the adaptive filter produces an error signal.
8- Is a DSP tool called signal to work space block used for finding the Mean-squared error.
9- Time scope- used to display the result.

The results that got after running the Simulink MATLAB program shown in Figure 3.38.

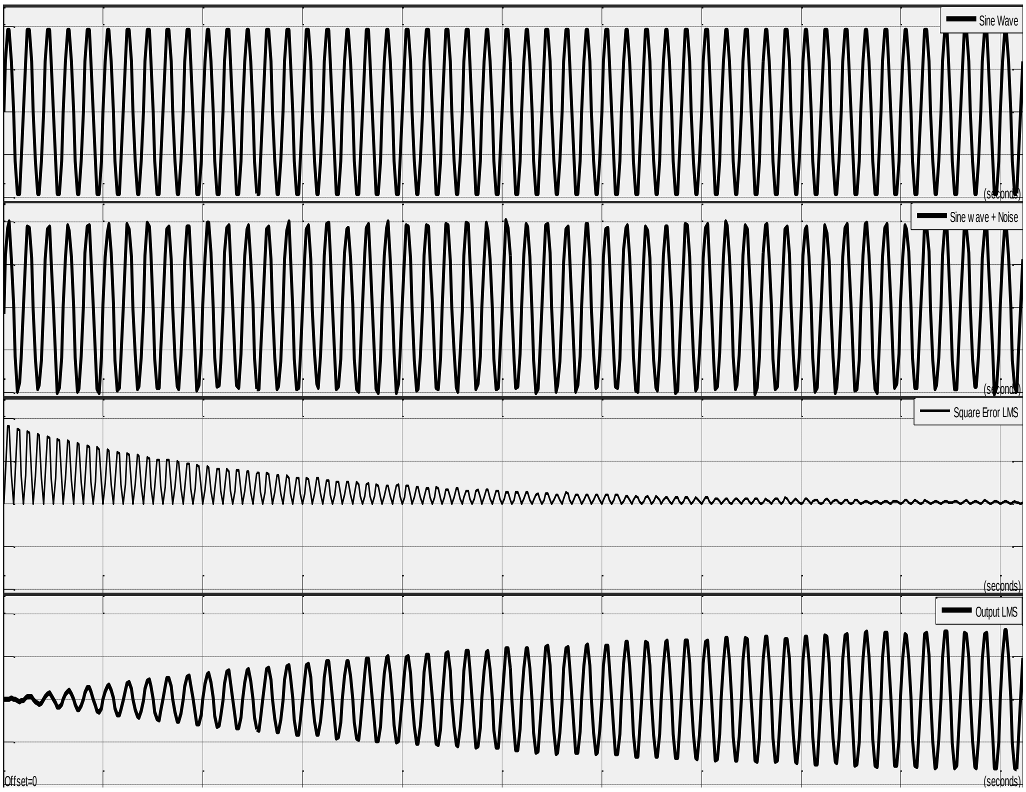

Figure 3.38 Pertinent signals when using the LMS algorithm to solve the Gamer suppresses problem.

In figure 3.38, as noted the input signal $x(i)$ (sine wave) which represents the original signal and also represents the desired signal $d(i)$. Additionally, the sine wave had been added to the Gaussian noise, then introduced to the Gain component. Then, as noted the squared error's $e^2(i)$, value is very big in the beginning because of the slow convergence of the LMS algorithm, and the value of iteration (length of the filter), which's purpose is training the adaptive filter. Lastly, let's take a look at the output of the LMS filter $y(i)$. Where, the value of iteration (length of the filter) is 5 and the implementation time is 0.5 Sec. In the part squared error, it is notable that the value of squared errors $e^2(i)$ is very large. The amplitude of the squared error in the beginning is 1 after which it decreases till it becomes zero. As noted the LMS filter as not effective as it spends relatively more time to filter noise.

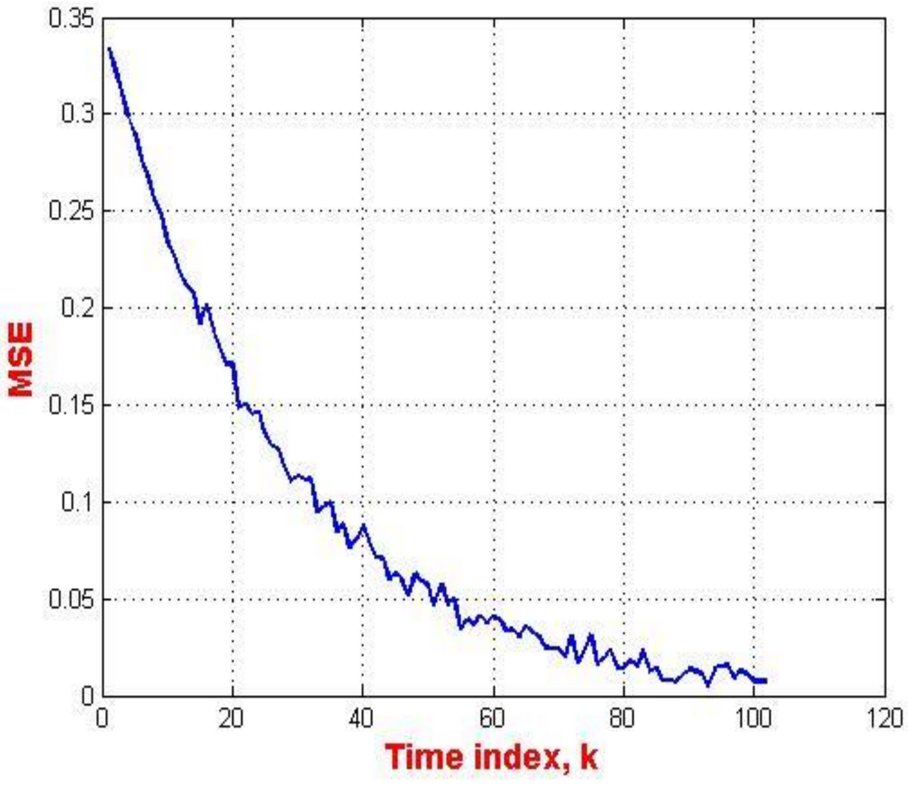

Figure 3.39 MSE of the LMS algorithm

In the figure 3.39, as seen the value of the MSE is initially 0.34, the convergence rate is for almost 100 samples during the implementation time. Note that the value of iteration is very small, and at least 10 iterations in the same implementation time (0.5 Sec) must use.

3.4.2 Solution Using the NLMS Algorithm

As it has been discussed in paragraph 3.4.1, the LMS algorithm is not suitable for noise cancelling. Therefore, it is better to test the NLMS algorithm in order to obtain a value for the fastest convergence rate. In chapter two, the NLMS algorithm is discussed as well as the equations which is used to filter the noise in this algorithm. If those equations are used, then they must be changed to be eligible so as to solve the problem.

$$d(i) = x(i) \quad 3.25$$

$$y(i) = w^{L}(i)x(i) \quad 3.26$$

$$e(i) = x(i) - y(i) \quad 3.27$$

$$w(i+1) = w(i) + 2\mu e(i)x(i) \quad 3.28$$

$$\mu = \frac{\bar{\mu}}{\delta + \|x(i)\|^2} \quad 3.29$$

Figure 3.37 shows how to represent the LMS algorithm in the Simulink MATLAB program. The difference in the representation of NLMS algorithm is a step size (μ), where the size of the step in the NLMS algorithm is equal to 0.03 (fast convergence), and the value of the constant δ =1.0.

The results obtained after running the Simulink MATLAB program are as shown in figure 3.40

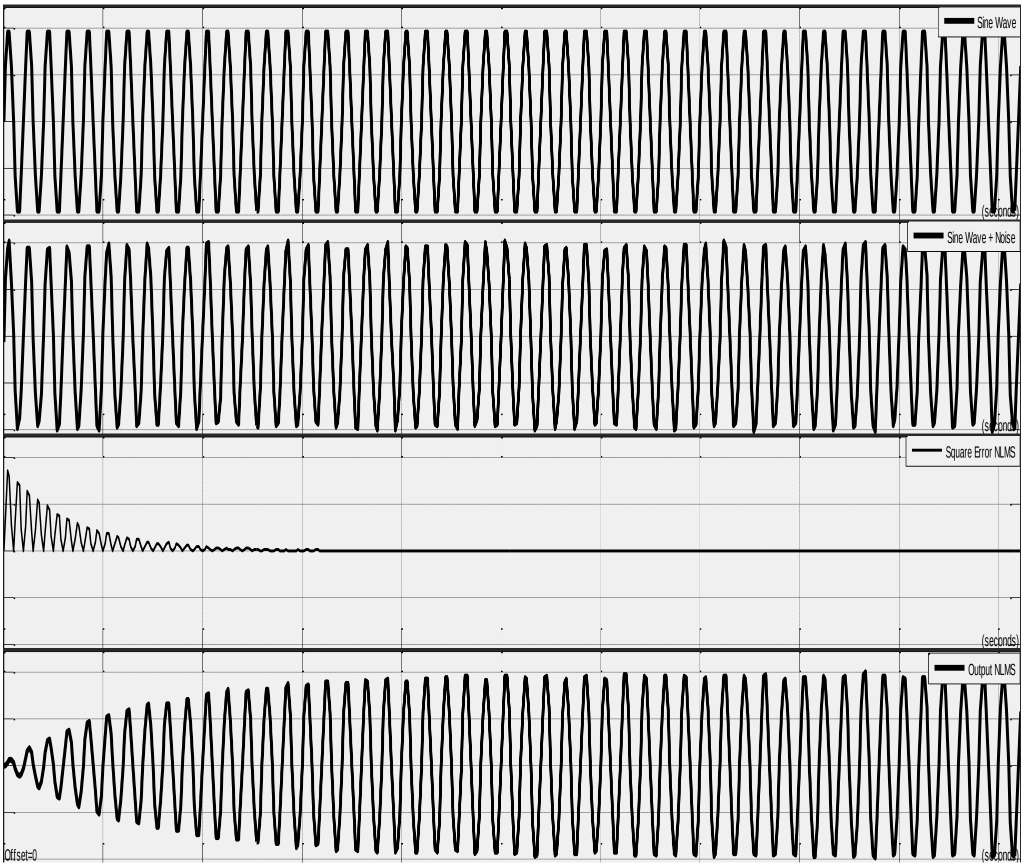

Figure 3.40 Pertinent signals when using the NLMS algorithm to solve the Gamer suppresses problem.

In the Figure 3.40, the original signal $x(i)$which that is the same of the desired signal $d(i)$, which be represented by a sine wave signal. The amplitude of this signal is 1, and the frequency is 100 HZ. The input signal in the adaptive filter includes the original signal (sine wave) $x(i)$ and add a noise, where the noise signal is the Gaussian noise multiplied by the factor 0.03 to normalize the noise. In addition, there is the squared error $e^2(i)$. Finally, there is the output of the NLMS filter $y(i)$.

The squared error $e^2(i)$, as noted the amplitude of the squared in the beginning is close to 1, then decrease the amplitude until it becomes zero at 0.144 Sec during implementation time (0.5 Sec). As noted from the time spent that this algorithm is a good algorithm for noise cancelling. But, as noted the squared error $e^2(i)$ has the value which begins to oscillate at the (0.4 Sec).

The figure 3.41 represents the MSE of the NLMS algorithm.

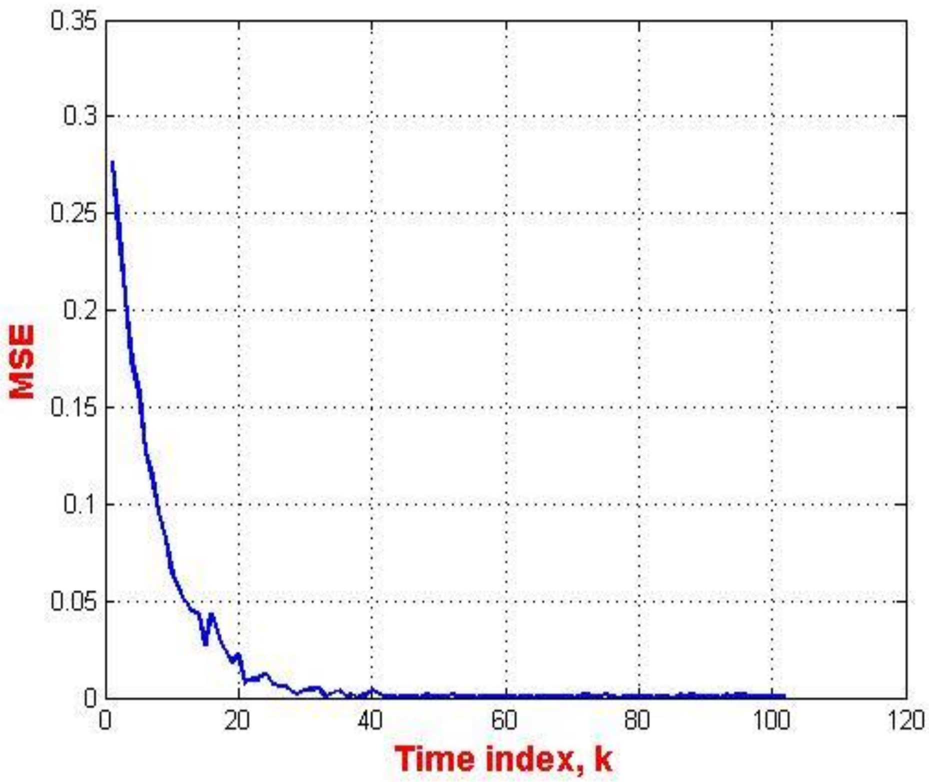

Figure 3.41 MSE of the NLMS algorithm

In figure 3.41, the value of MSE is 0.28, and the value of convergence rate is 37 samples at implementation time (0.5 Sec). This is relatively good score and the rate of convergence is fairly لاll.

3.4.3 Solution Using the RLS Algorithm

Active noise cancelling (ANC) uses many technologies to cancelling the noise. One of these techniques is the RLS algorithm, this algorithm is the most important algorithm in the adaptive filter. The research discusses in chapter two the RLS algorithm in addition to the advantage and disadvantage as well as the equations used to filter the noise in this algorithm, now, to solve the problem git need to update these equations to solve this problem.

$$d(i) = x(i) \qquad 3.30$$

$$y(i) = w^L(i)\, x(i) \qquad 3.32$$

$$k(i) = \frac{\lambda^{-1}\Phi_A^{-1}(i-1)x(i)}{1+\lambda^{-1}x^L(i)\Phi_A^{-1}(i-1)x(i)} \qquad 3.31$$

$$e(i) = x(i) - y(i) \qquad 3.33$$

$$w(i+1) = w(i) + k(i)e^*(i) \qquad 3.34$$

$$\Phi_A^{-1}(i) = \lambda^{-1}\Phi_A^{-1}(i-1)\lambda^{-1}k(i)x^L(i)\Phi_A^{-1}(i-1) \qquad 3.35$$

The figure 3.42 shows how an algorithm can be represented in the Simulink MATLAB program.

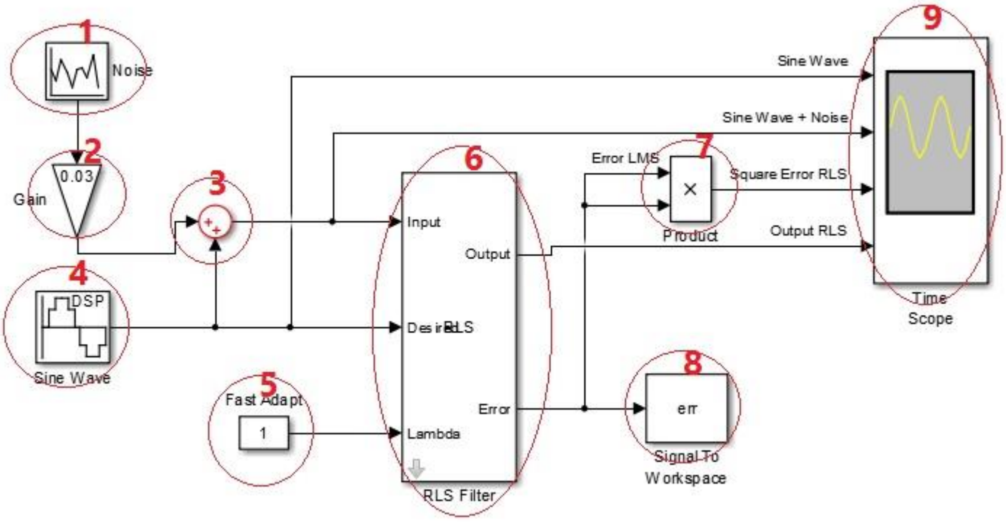

Figure 3.42 Solution using the RLS algorithm

Where

1- It is the noise source, it is the Gaussian noise (white noise).
2- The Gain, uses to normalize the noise in this Simulink by multiplying the Gaussian noise by 0.03.
3- Sum, uses to summation two signals or more, in this case it is used to add the noise signal with the sine wave.
4- Is the input signal (sine wave) the amplitude of this signal =1 and the frequency =100 HZ.
5- Lambda value $\lambda = 0{,}9$.
6- The adaptive filter (RLS algorithm) blocks and the value of the filter length is(q) =5.

7- Squared error, when compared between the desired signal and the output signal of the adaptive filter produces an error signal.

8- Is a DSP tool called signal to work space block used for finding the Mean-squared error.

9- Time scope, used to display the result.

The results that got after running the Simulink MATLAB program shown in Figure 3.43.

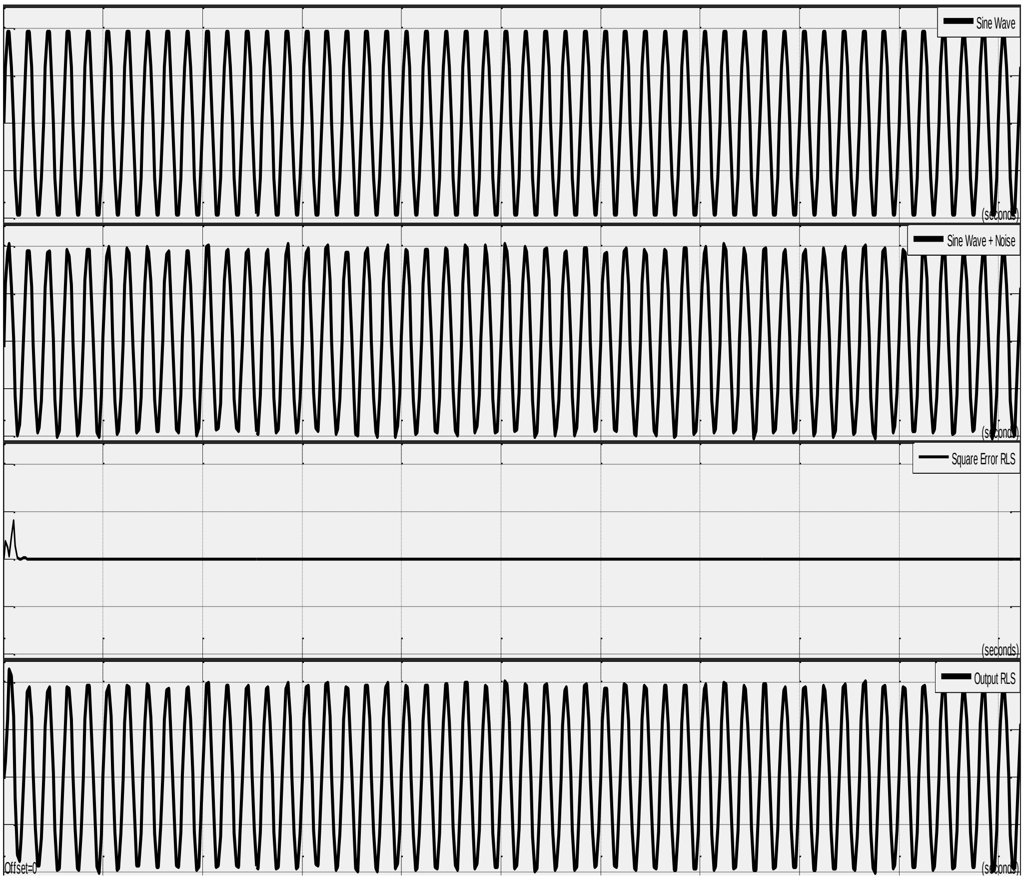

Figure 3.43 Pertinent signals when using the RLS algorithm to solve the Gamer suppresses problem.

In the Figure 3.43 the MATLAB result of the RLS algorithm, which shows in the first part can be seen the desired signal $d(i)$ or the original signal $x(i)$ it is of the same value, in the second part can see the noise signal $n(i) + x(i)$, when adding the original signal to the white noise (Gaussian noise) after multiplying by the factor 0.003 to normalize noise, the signal result is the input of the RLS filter.The third part is the squared error, the error is produced by comparison between the desired signal $d(i)$ and the output signal of the adaptive filter $x(i)$. Finally the output of RLS filter $y(i)$. The implementation time is 0.5 Sec, the value of iteration (length of the filter) is 5.

The value of the squared error is very small infact equal to 0.40 at a time of 0.008 Sec. That means the result of the RLS algorithm is good and effective for noise cancelling.

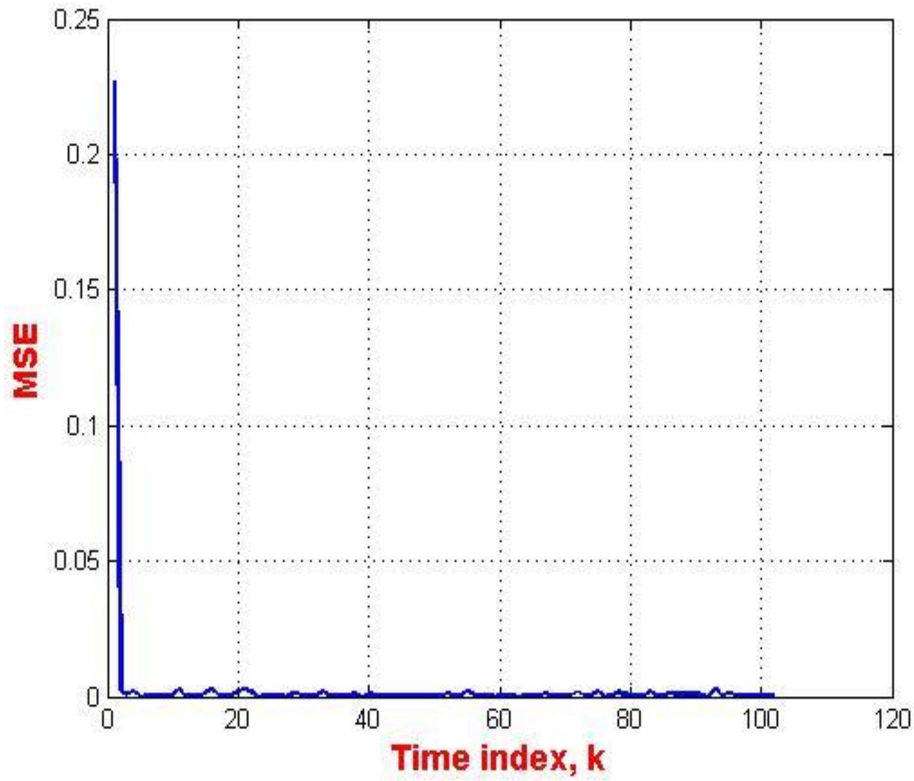

Figure 3.44 MSE of the RLS algorithm

The Figure 3.44 shows the MSE of the RLS filter.In this figure the value of MSE is 0.23, and the value convergence is 2.5 samples in the implementation time (0.5 Sec), this result is considered ideal because of less implementation time, and the number of iterations few as well.

3.4.4 Comparison of Results

As discussed in section 3.1.5 the factors which can depend on selecting the best algorithm in order to solve any problem when using the adaptive filter or using ANC. The first factor is the cost, because of the complex equations in the RLS filter, so needing a hardware with special specifications, thus, RLS algorithm has the highest cost. The second factor is the MSE, the Figure 3.45 shows the comparison among the three algorithms.

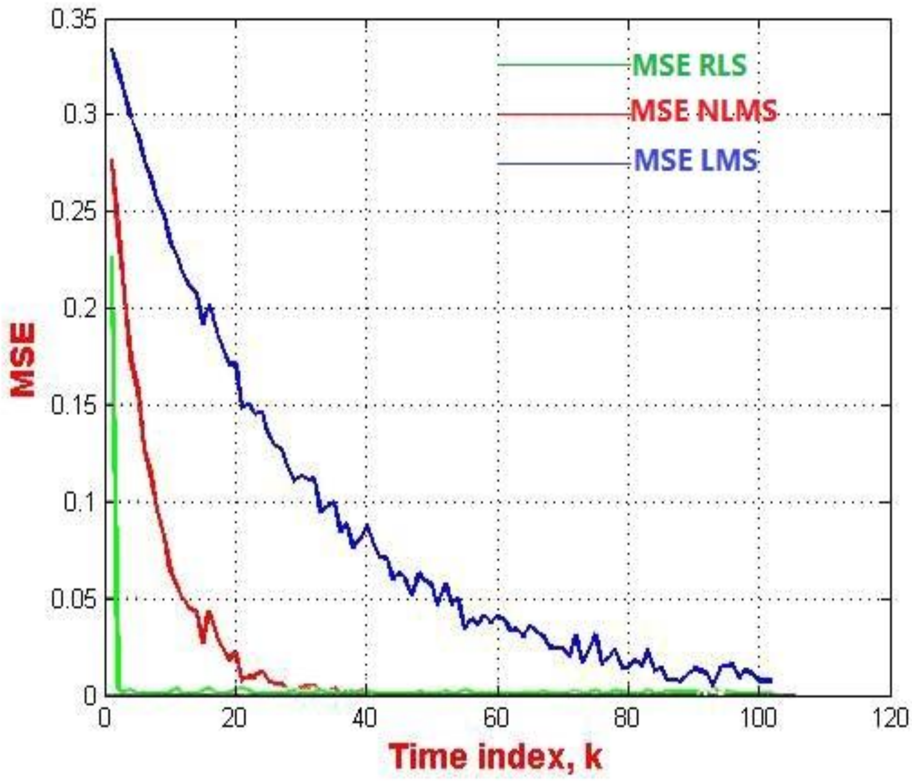

Figure 3.45 Comparison between MSE for the algorithms LMS, NLMS and RLS

In Figure 3.45,the least value in the RLS algorithm is 0.23 and in the NLMS algorithm is 0.28, finally in the LMS algorithm is 0.34 that is the highest among the three algorithms. The convergence rate, in LMS algorithm is 100 samples, in the NLMS algorithm is 37 samples, and in the RLS algorithm is 2.5 samples. That means the RLS algorithm has faster convergence rate. The last factor is Tracing, the Figure 3.47 shows the comparison among errors in the three algorithms.

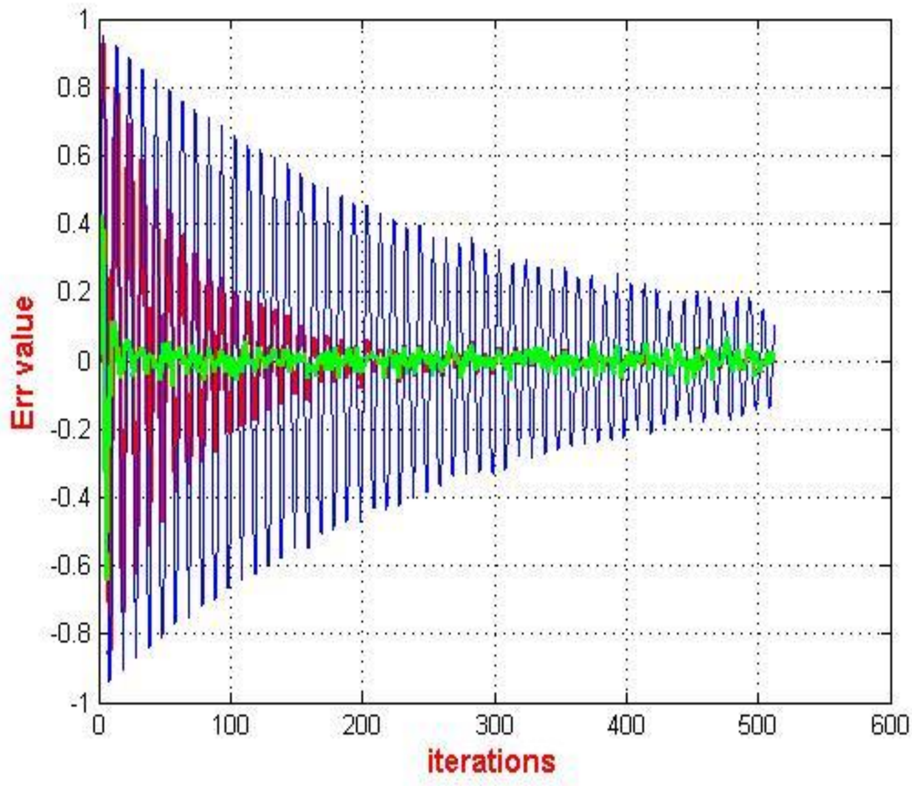

Figure 3.46 Comparison between the error signal in the three algorithms

Figure 3.46 shows, the comparison among the error signals, the error signal in the LMS algorithm is represented by the blue signal, the error signal in the NLMS algorithm is represented by the red signal, and the error signal in the RLS algorithm represented by the green signal. As seen in Figure 3.46 shows the error in the LMS algorithm which is very big and spends a lot of time to cancel the noise. On the other hand, the RLS algorithm has less error and spends a short time to cancel the noise.

CHAPTER 4

PRACTICAL APPLICATION

4.1 Introduction

In this chapter, firstly the research discusses Active Noise Cancelling, where the research presents the examples of the ANC, and uses of the three algorithms (LMS, NLMS and LMS) of analyzing these examples and comparing the results. The research uses two types of noise, the Gaussian noise (white noise) and the color noise to test these algorithms.Secondly the researcher discusses the Channel Equalization, and uses the three algorithms (LMS, NLMS and LMS) to analyze these examples and compare the results.

4.2 Active Noise Cancelling

In this part the research represents this case ‘Active Noise Cancelling’ in the proposed example, that shown in the figure 4.1, and compare the results of the three algorithms.

4.2.1 Using Gaussian Noise

In this section of the thesis, the research presents an example of the adaptive filter and the practical environment for this filter, where the Gaussian noise (white noise) is used to test the three algorithms.The Figure 4.1 shows an example of the adaptive filter, which can test the algorithms on this filter, where all the algorithms (LMS, NLMS, RLS) has been applied and discuss the results.

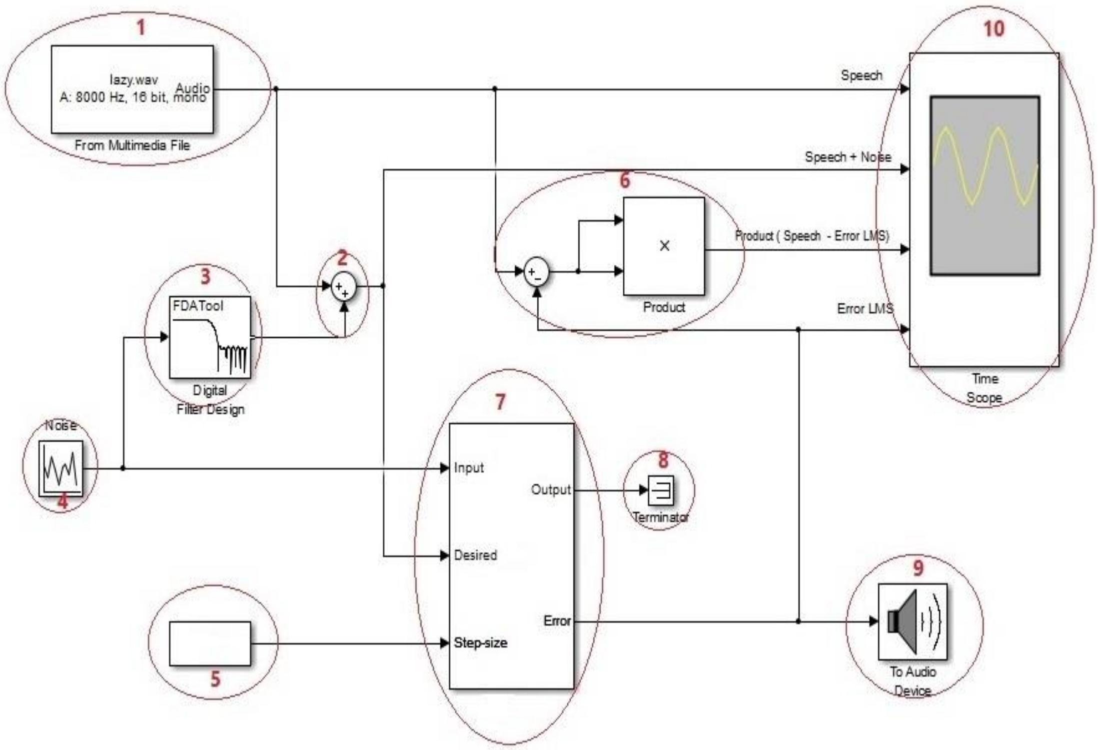

Figure 4.1 Proposed example for the ANC using Gaussian noise

Where:

1. The source signal $x(i)$ which is represented by the song.wav. The sampling frequency8000 HZ.
2. Sum, used for the summation of two signals or more, in this case it is used for adding the noise signal to the source signal.
3. Lowpass filter, in this example is used to change the properties of the noise signal, thus the noise signal becomes different from the noise source.
4. It is the noise source, it is the Gaussian noise (white noise).
5. The step size, which is in the algorithms (LMS and NLMS) is μ, and in the RLS algorithm is λ.
6. Squared error, when comparing the desired signal and the output signal of the adaptive filter produces an error signal.
7. The adaptive filter block
8. Terminator.
9. Speaker to listen to the output sound.
10. Time scope, used to display the result.

4.2.1.1 Solution Using the LMS Algorithm

This paragraph describes the simulation and the actual results of LMS algorithm when applying Gaussian noise corruption input signal.

The ANC program, using the LMS algorithm to solve the problem the signal corrupted by a Gaussian noise, shown in Figure 4.1, and how can it be solved by using a simulation MATLAB program.

In this example the value of step size μ is 0.0003 (slow convergence), and the value of the filter length (q) is 32, the initial value of filter weights is 0.1, and the implementation time is 6 Sec. The Figure 4.2 represents the LMS filter output.

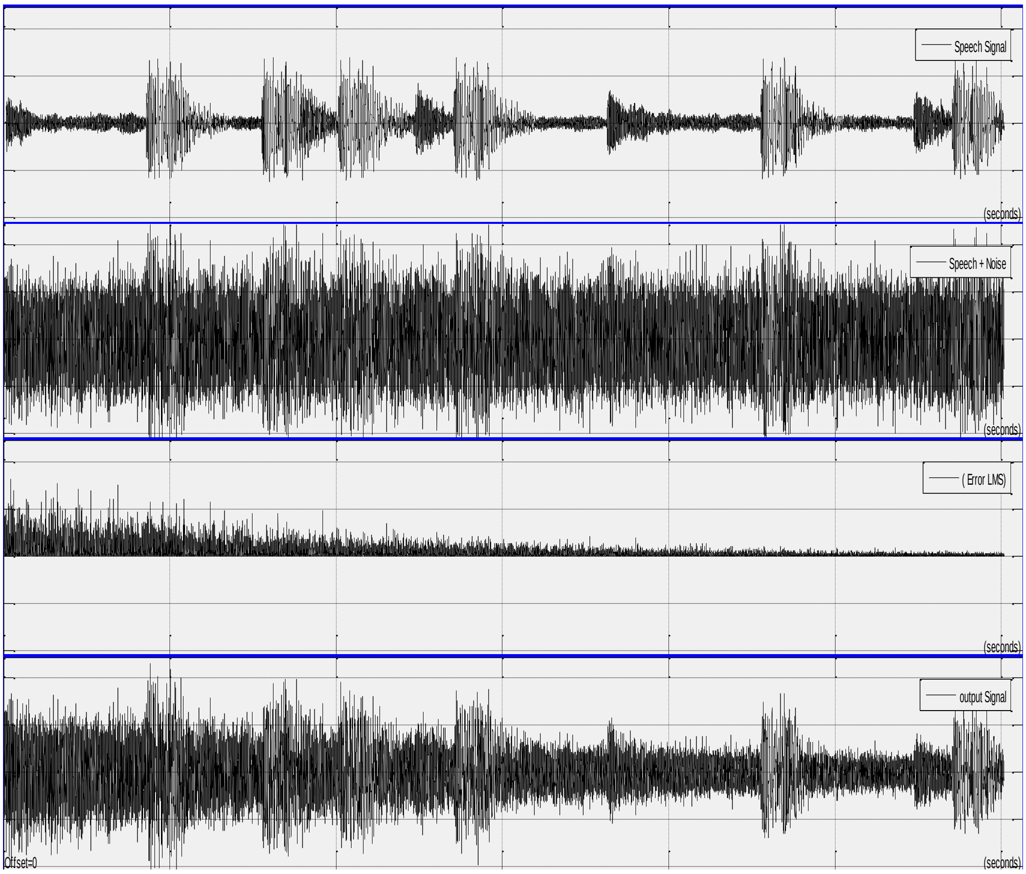

Figure 4.2 Output of the LMS algorithm when using Gaussian noise.

In the Figure 4.2, as seen the source wave $x(i)$ in the first part. Whereas in the second part as seen the noise signal $n(i)$ which represents the desired signal $d(i)$, the third part is the squared error $e^2(i)$, is the result of comparing the source signal $x(i)$ to the error signal $e(i)$.Eventually, gives the output signal.

In this algorithm, at the beginning of implementation, the value of squared error $e^2(i)$ is very big, the error takes 10.25 Sec to cancel the noise, until the value of the squared error becomes equal to zero.

4.2.1.2 Solution Using the NLMS Algorithm

In this section the fast convergence rate has been applied after that the slow convergence rate has been used, and also the Gaussian noise has used on the original signal $x(i)$. Figure 4.1 shows the proposed example, and how it can be applied to the NLMS algorithm, in this case the step size (μ) is a 0.03 (fast convergence), and the initial value of filter weights is 0.1, and the value of the filter length (q) is 32, where the implementation time is 6 Sec. The Figure 4.3 represents the NLMS filter output.

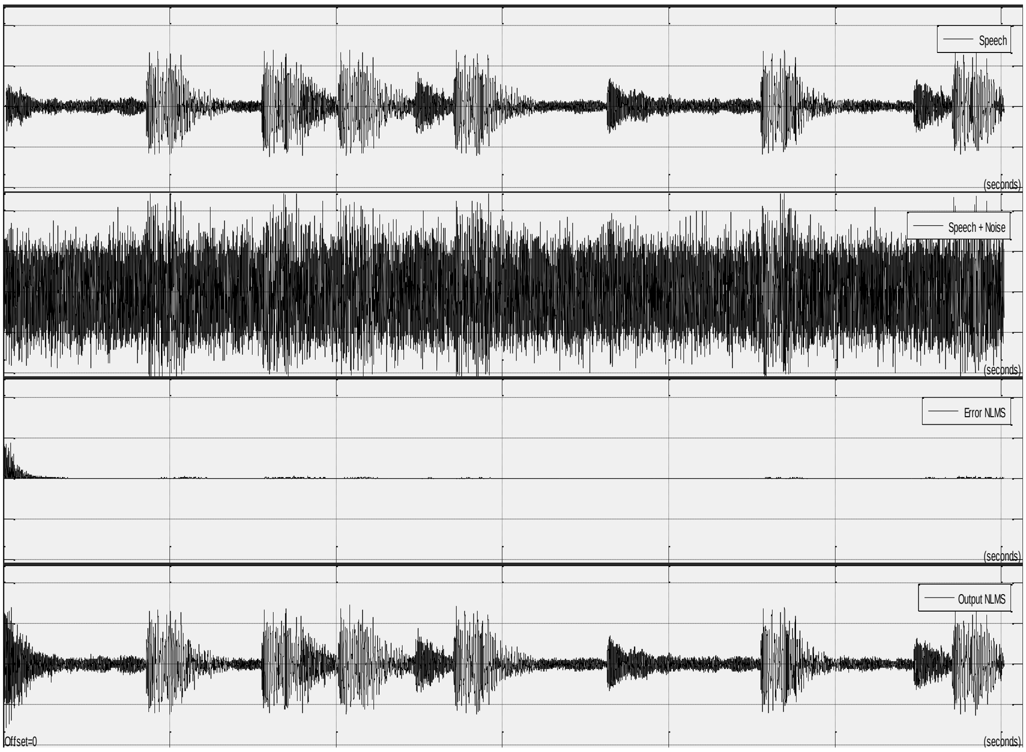

Figure 4.3 Output of the NLMS algorithm when using Gaussian noise.

The Figure 4.3 shows, the input signal $x(i)$, this signal is the sound wave and extension (.wav). The Gaussian noise produces a random signal with uniformity or Gaussian (normal) distribution. Consequently, the squared error is used to measure the ratio of the error. Finally the output signal of the NLMS filter, and that is supposed to be an equal input signal.

In the Figure 4.3, the value of squared errors, as noted, the algorithm spends almost 0.449 seconds to become an acceptable conversion property, to become the squared error closer to zero. From the result in Figure 4.3, can notice that the NLMS algorithm is good for canceling the noise because they spent a little time to do it. But as seen in the Figure 4.3 in the squared error and specifically between the 1.568 Sec and the 1.849 Sec, presents some of the noise, which means that the NLMS algorithm is not the most stable.

4.2.1.3 Solution Using the RLS Algorithm

Figure 4.1 contains the active noise cancellation software built to work with the wave being corrupted by Gaussian noise. This filter has a length of 32, and initial value of filter weights was 0.1. The lambda λ was is chosen to be equal to 1, after the implementation of the test results which are as in the following form.

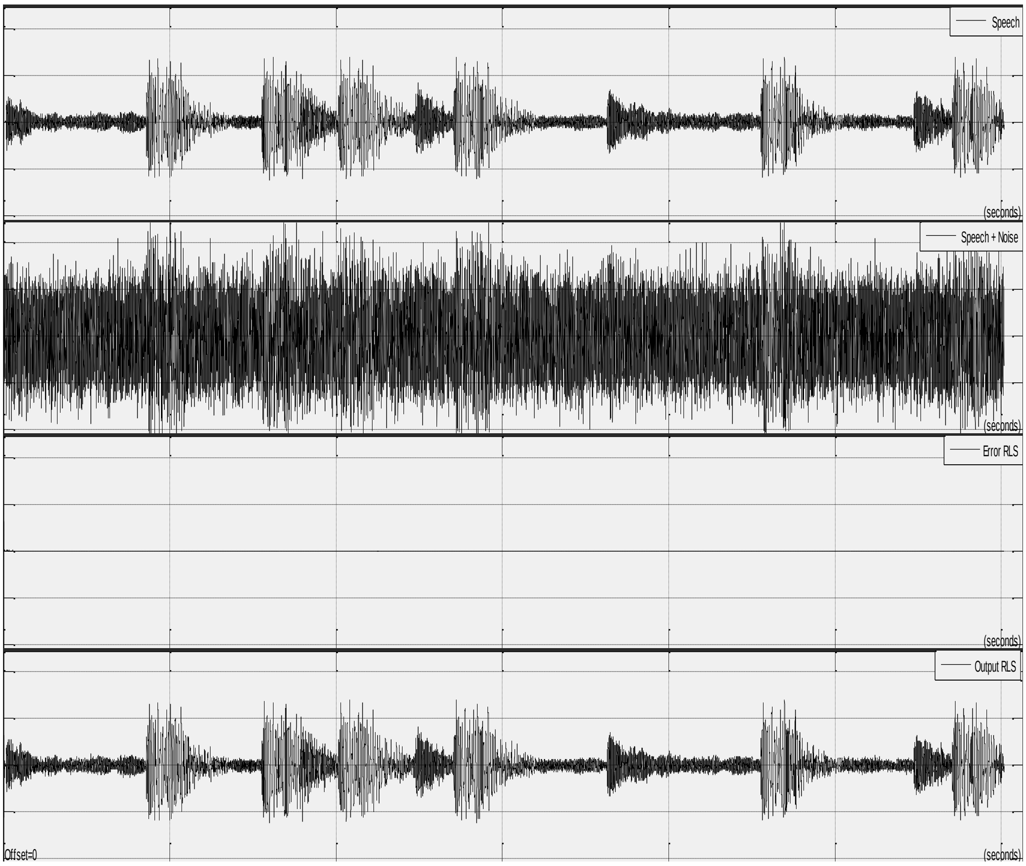

Figure 4.4 Output of the RLS algorithm when using Gaussian noise

The Figure 4.4 represents the four signal results, these signals are, the input signal $x(i)$,The noise signal $n(i)$ which represents the desired signal $d(i)$, the noise signal is the Gaussian noise (white noise) processed by low pass filter so as to change the characteristics of the noise signal and producean other noise signal.The squared error, when comparing between the error signal $e(i)$of the ANC and the input signal $x(i)$ will produce the squared error. Finally the output of the RLS algorithm.

Figure 4.5 represents the zooming in the Figure 4.4, as seen the signal output between 0 to the second 0.025.

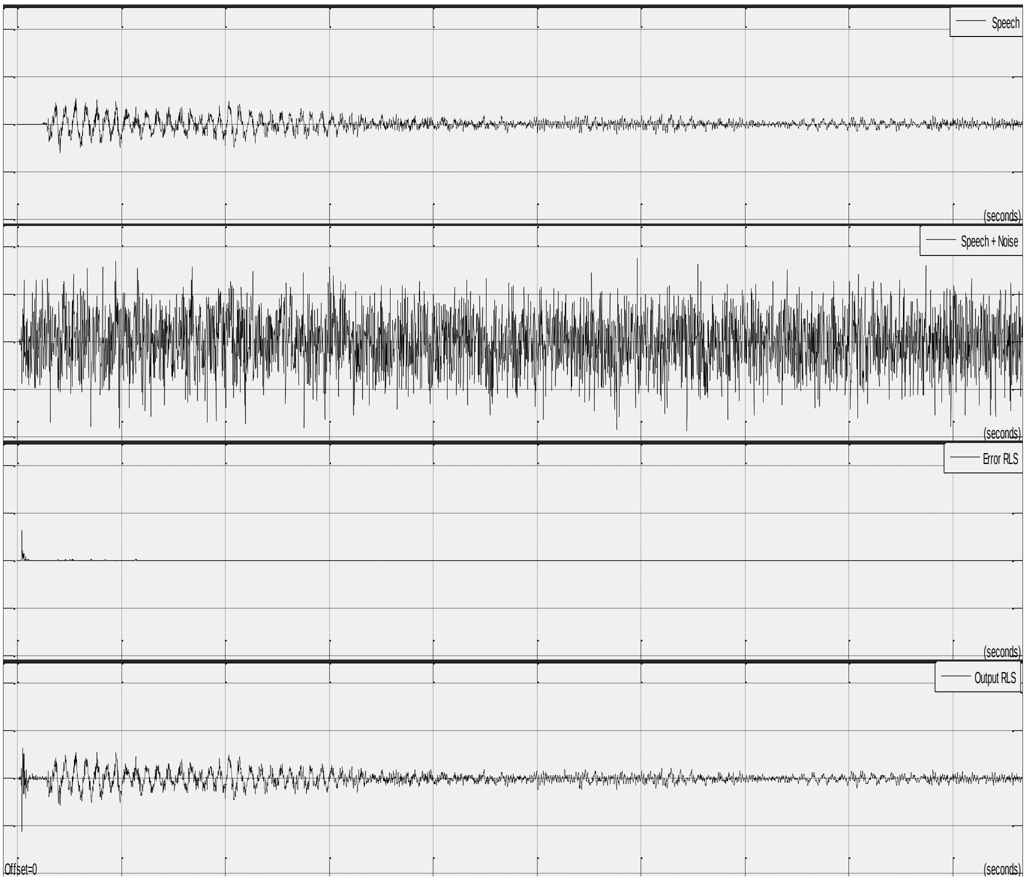

Figure 4.5 The output of RLS algorithm

Figure 4.5 represents the zooming of figure 4.4, where signal output can seen between 0 and 0.025 second.

The Figure 4.5 shows the squared error as very little, and the convergence rate, spends very short time, less than 0.002 Sec, that means the convergence rate is very fast.

4.2.2 Using Color Noise

In this section the three algorithms have tested on the proposed example, applying the color noise and comparing the results. Figure 4.6 shows the design of the active noise cancel, and the difference between the results when a Gaussian noise is used along with a color noise.

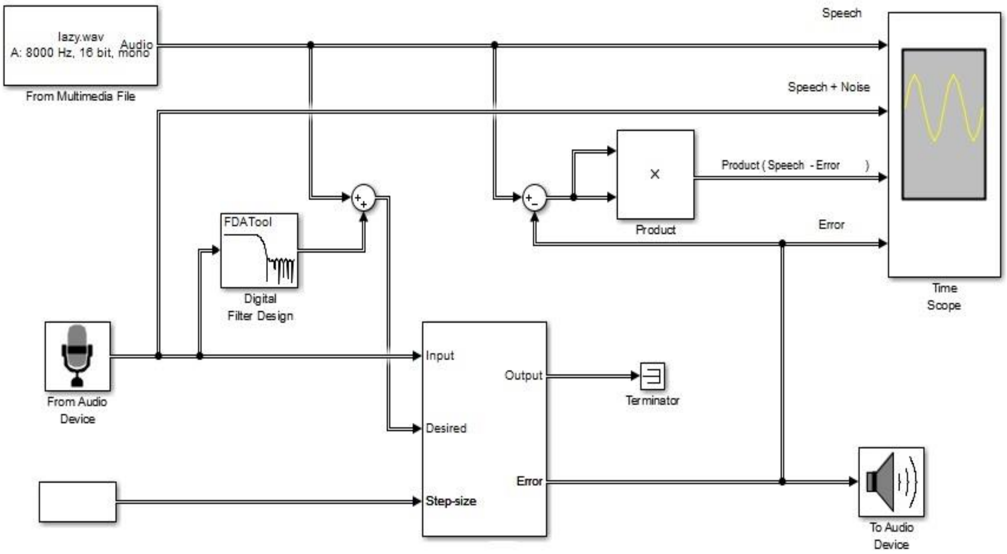

Figure 4.6 Proposed example for the ANC using color noise

The difference between the example that uses the Gaussian noise and the example that uses the color noise is: in the color noise the microphone has used as a source of the noise, that means a noise source is random, but in the Gaussian noise a steady source is used

4.2.2.1 Solution Using the LMS Algorithm

The length of the filter has been chosen to be 32, and the step size μ is 0.0003. The input signal now malformed with colored noise. The noise source is a random signal, the microphone is used to detect to this noise. In this section the LMS algorithm used to test the proposed example. Figure 4.7 shows the results of this algorithm.

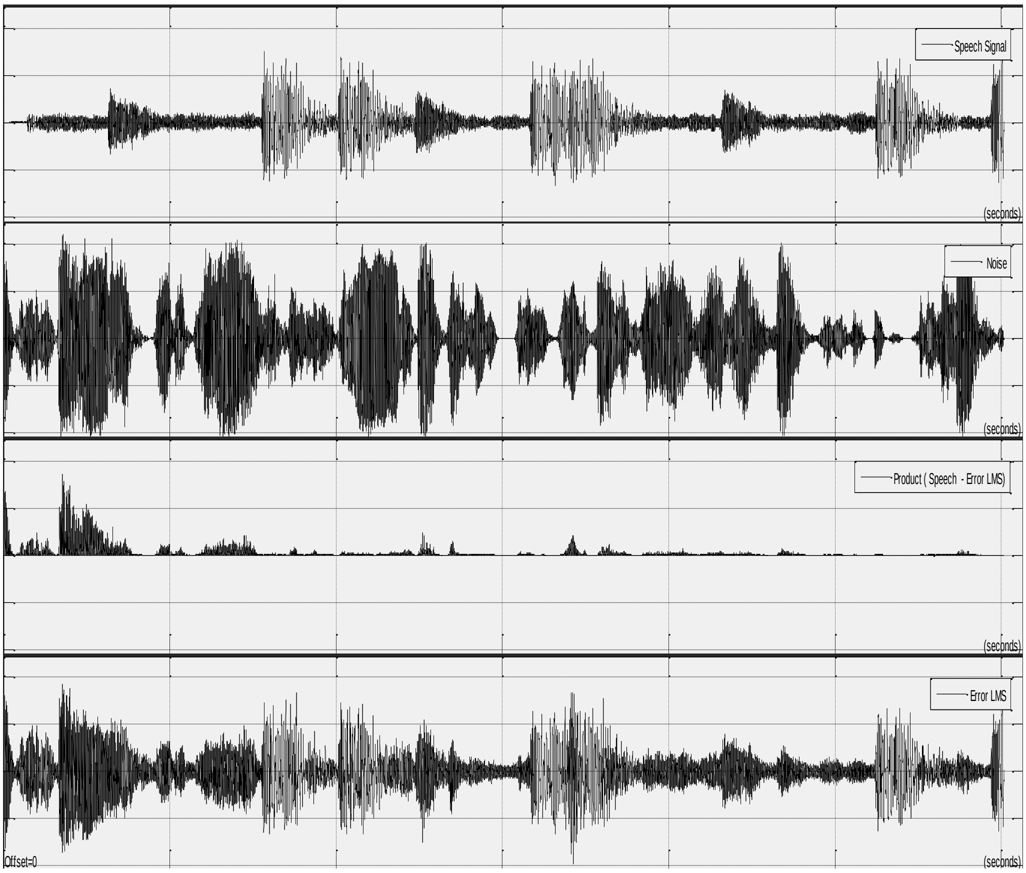

Figure 4.7 The Simulink output when applying the LMS algorithm and using color noise

In the Figure 4.7, the first signal is the input signal $x(i)$ which is an audio wave. The frequency of this wave is 8000 HZ. The second signal is noise signal $n(i)$ this noise is called the color noise, the difference between the white noise (Gaussian noise) and the color noise is that the color noise is random. The third signal is the squared error. Finally, the output of the LMS filter.

In Figure 4.7 as noted that the effect of the color noise on the original signal is very small, the squared error is small, depends upon the value of the convergence rate in the color noise on the power of the noise, namely, the convergence rate in a color noise is not fixed. The result in an LMS algorithm is good in this case that the noise power is little. While taking into consideration the value of the step size (μ) which is equal to 0.0003.

4.2.2.2 Solution Using the NLMS Algorithm

As discussed earlier that the NLMS algorithm is an update of the LMS algorithm, where an NLMS algorithm is to address some of the mistakes in the LMS algorithm. Figure 4.6 shows the Simulink cancelling the color noise. The length of the filter is 32, the step size μ is 0.003 (fast convergence) and the δ constant is 1.0. The color noise is detected by the microphone. The Figure 4.8 shows the Simulink result.

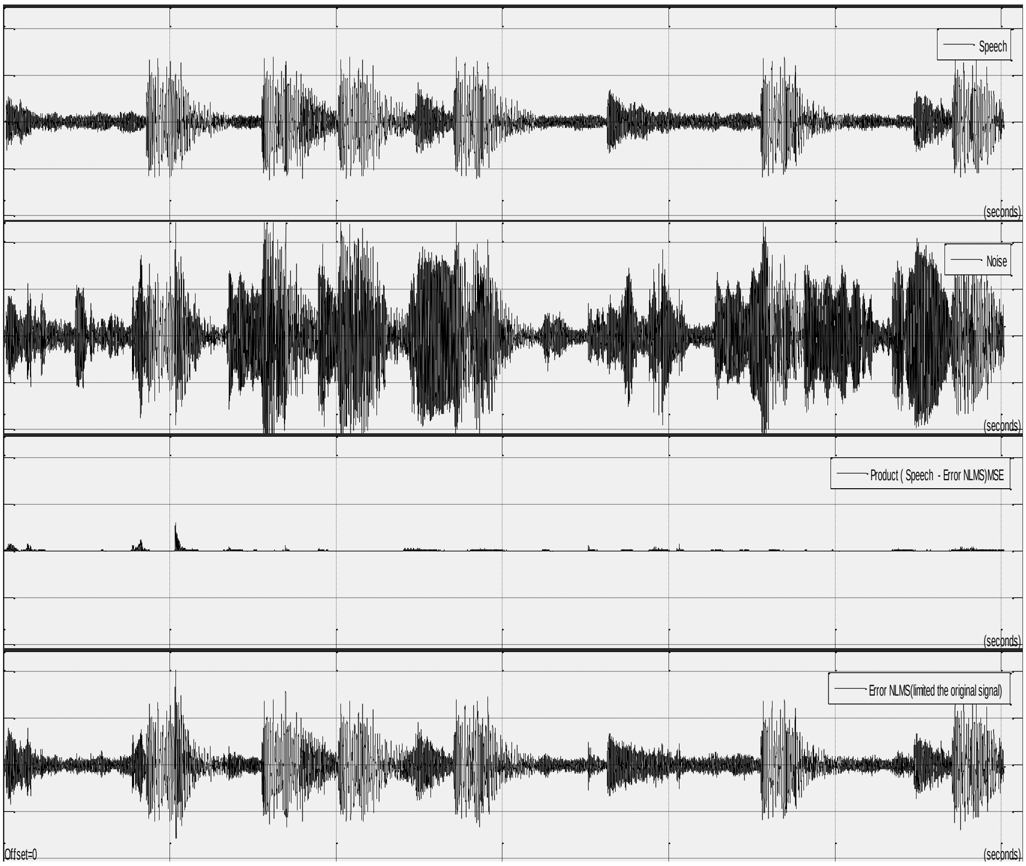

Figure 4.8 The Simulink output when applying the NLMS algorithm and using color noise

Figure 4.8, has four sections, the first section represent's the original signal or the input signal $x(i)$ The frequency of the sound signal of the original signal is 8000 HZ. The second section is the noise signal $n(i)$, which contains the original signal added to the color noise. The third section represents the squared error, and has a value close to zero. The final section is the filter's output $y(i)$, when comparing the original signal with the output signal the results must be equal to or close to zero.

In the Figure 4.8, especially in the third section in the squared error, as noted, that the value of the squared error is small, but spends a long time training the adaptive filter and cancelling the noise.

4.2.2.3 Solution Using the RLS Algorithm

In this portion the RLS algorithm has been applied to the proposed example and test this algorithm on the ANC to know the effect of this algorithm on the ANC. The Figure 4.6 shows the Simulink cancelling the color noise. The length filter is 32, the value of λ is 1, the initial value for the other parameters in the RLS filter is zero. The color noise was detected by the microphone. The Figure 4.9 shows the Simulink result.

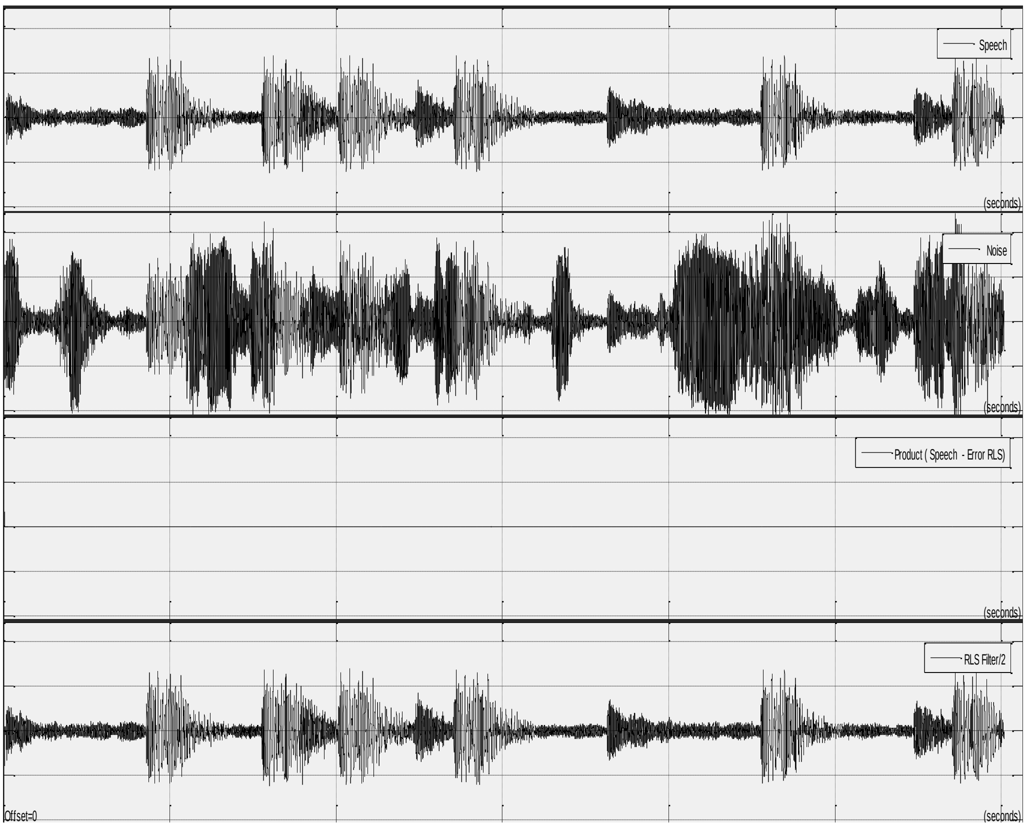

Figure 4.9 The Simulink output when applying the RLS algorithm and using color noise

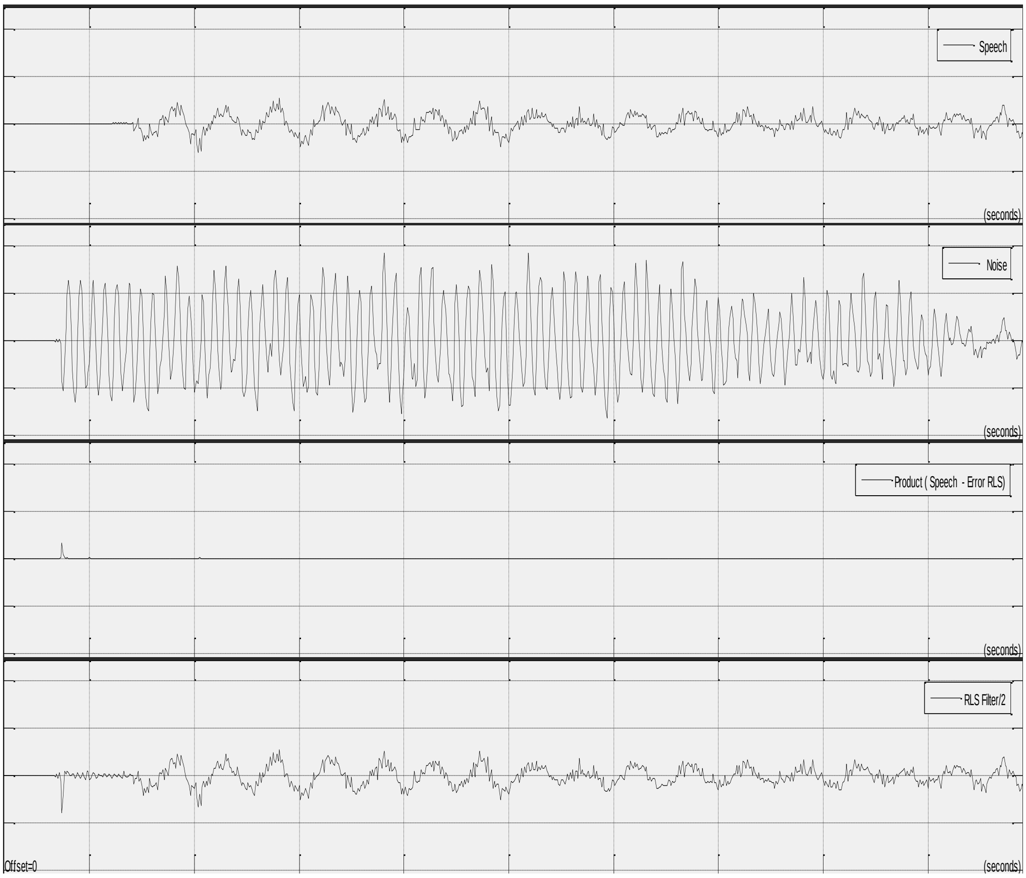

Figure 4.10 Represents the zoom view from the figure 4.9

In the Figures 4.9 and 4.10, notably, the first signal is the input signal $x(i)$.It is audio wave. The frequency of this wave is 8000 HZ. The second signal is noise signal $n(i)$ this noise is called the color noise, the difference between the white noise (Gaussian noise) and the color noise is that the color noise is random. The third signal is the squared error. Last but not the least, the output of the RLS filter.

In the figure 4.10, the value of the convergence rate is a very fast in comparison to the LMS algorithm and NLMS algorithm.

4.3 Channel Equalization

Telephone, wireless and optical channels, in other words telecommunications channels, are vulnerable to Inter Symbol Interface (ISI). A frequency-selective and/or dispersive communication link between sender and receiver causes detrimental effects. The simple method to downplay from this detrimental effect is channel equalization. Thus higher rates of data are empowering. ISI is introduced by an incorporated equalizer within the receiver and channel. The ISI, in telecommunication, is a modality of signal distortion, in which one interface with a subsequent symbol is defined as an Inter Symbol Interface (ISI). This phenomenon is undesirable to have as it has the same effect as a noise in the previous symbol. Consequently, the communication will be less reliable. The multipath propagation, or inherent non-linear frequency response, of a channel causes the successive symbols to "blur" together, and at worst, it may cause the ISI [17] [18].

In this section as discussed the Channel Equalization problem, where all the algorithms (LMS, NLMS, RLS) have been applied and analyze the results. The Figure 4.11 shows how these can solve this problem.

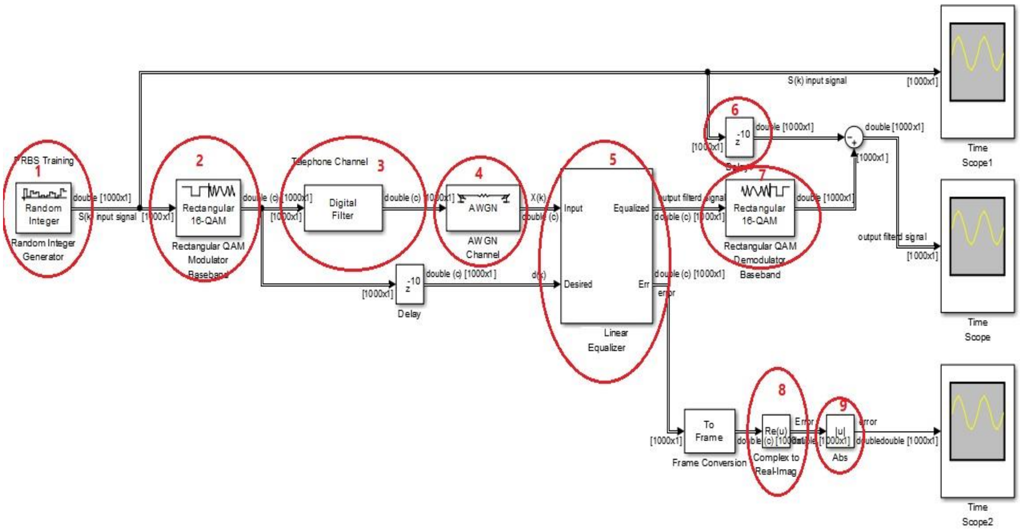

Figure 4.11 Channel Equalization

Where:

1. Random integer generation .
2. To modulate the signal
3. Telephone channel.
4. Gaussian noise.
5. Adaptive filter.
6. Delay system and delay time = 0.010 second.
7. To extract the modulation.
8. To Separate the imaginary signal from the real signal, and cancel the imaginary signal.

9. The absolute value.

4.3.1 Solution Using the LMS Algorithm

This section shows the results of the application of the LMS algorithm to test how effective it is of at solving the Channel Equalization problem. The figure 4.11shows how to represent this problem in the Simulink MATLAB. The Figure 4.12 shows the result of the Simulink.

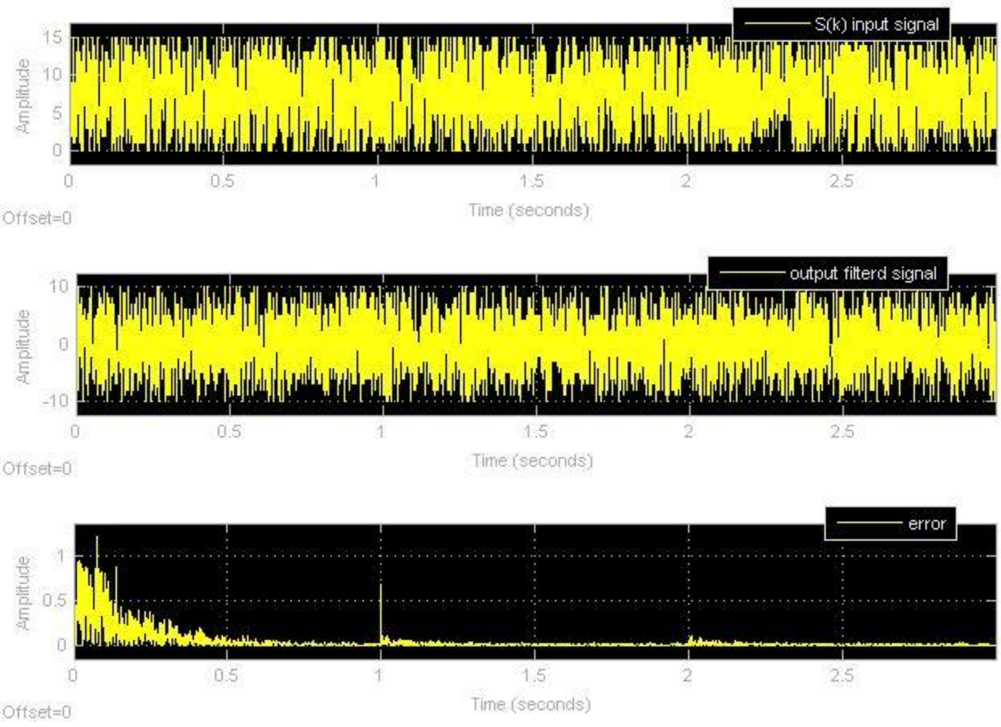

Figure 4.12 Pertinent signals when using the LMS algorithm to solve the Channel Equalization

In the Figure 4.12 as be seen three signals. The input signal is the random signal $s(k)$ the whose amplitude varies between 0 and 15.The output signal $y(k)$has an amplitude that varies between (-10 to 10). Finally, the Error signal $e(k)$.This signal's amplitude varies between (0 to 1). The Error signal is quite noticeable because it takes a long period of time, moreover the value of the Error in the 2nd second increases, after which the value of the Error signal becomes close to zero. The convergence rate is very slow, i.e. takes 0.5 second.

The implementation time is 3 Sec and the value of the filter's length(the number of iterations) is equal to 64.

4.3.2 Solution Using the NLMS Algorithm

After applying the LMS algorithm on the figure 4.11, now the NLMS algorithm is applied to the same problem that was shown in figure 4.11. The figure 4.13 shows us the result that is obtained from the NLMS algorithm

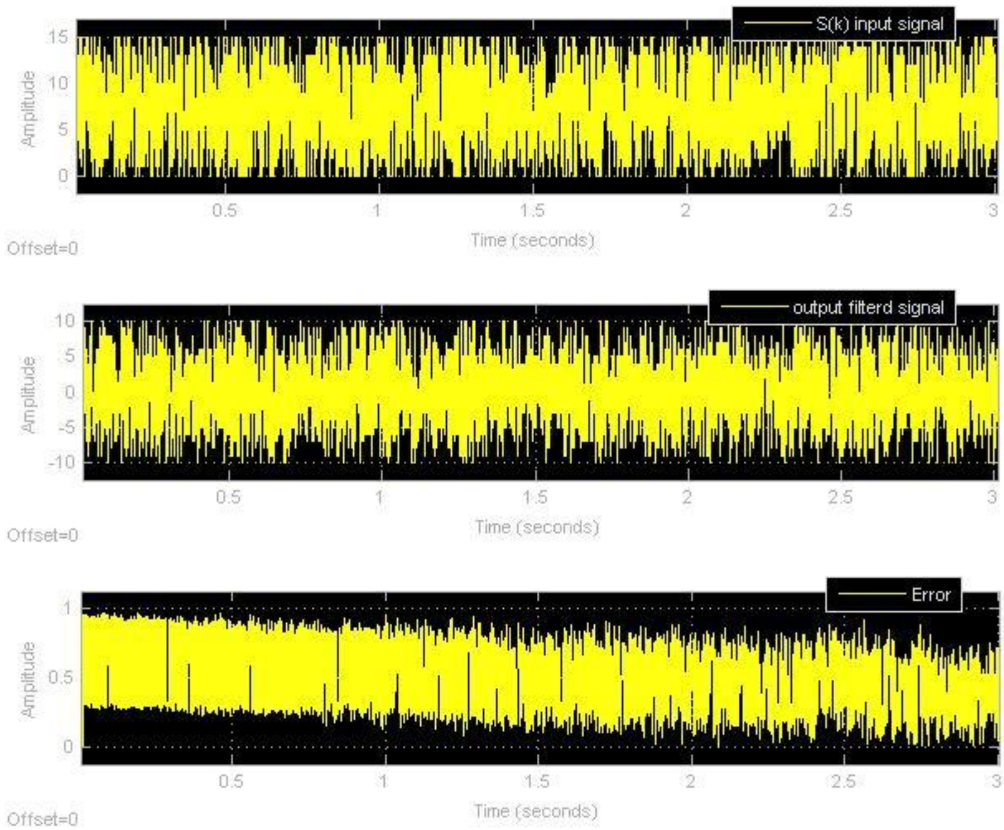

Figure 4.13 Pertinent signals when using the NLMS algorithm to solve the Channel Equalization

In figure 4.13, the input signal $s(k)$ is seen, the output signal $y(k)$, and the Error signal $e(k)$.The amplitude of the input signal varies between (0 to 15), the amplitude of the output signal between (-10 to 10), the Error value is very big and, hence, takes a very long time. Therefore, it can be said that the LMS algorithm has failed to solve this problem (Channel Equalization). The implementation time is 3 Sec and the value of the filter length (the number of iterations) is equal to 64.

4.3.3 Solution Using the RLS Algorithm

In the chapter 3 all the theories of the adaptive filter were applied and discussed the modus operandi of these theories,then explained the derivation of the equations defining the theories. The figure 4.13 shows the result of the application of the RLS algorithm to figure 4.11

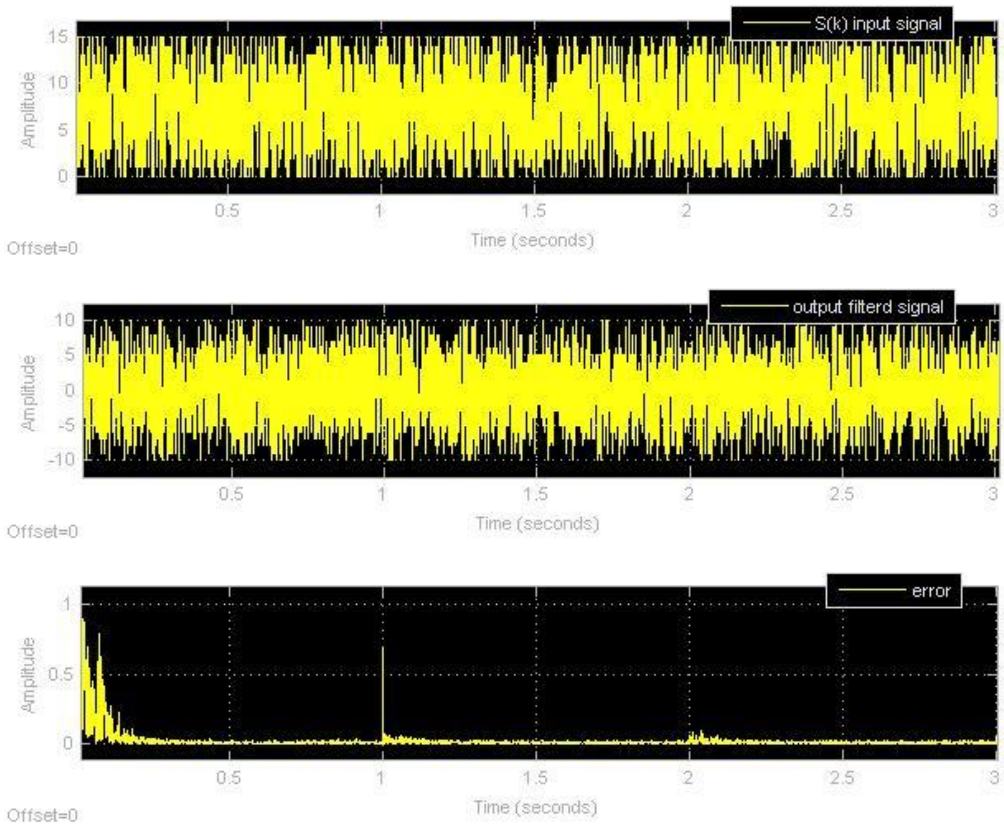

Figure 4.14 Pertinent signals when using the RLS algorithm to solve the Channel Equalization

The figure 4.14 shows three signals.The first signal is the input signal $x(k)$, the second signal is the output signal $y(k)$, and the last signal is the Error signal $e(k)$. The amplitude of the input signal varies between (0 to 15), and the amplitude of the output signal varies between (-10 to 10). The Error signal in the RLS algorithm is spending the normal amount of time. The Convergence rate is the fastest, and spends 0.19 seconds. Knowing that the implementation time is 3sec.Again, the value of the length of the filter (the number of iterations) is equal to 64.

CHAPTER5

CONCLUSIONS

5.1 Conclusions

This study discussed the adaptive filter algorithms (LMS, NLMS, RLS) and compared these three algorithms with one another. The results obtained from applying the adaptive filter theories, were presented, and noise canceling from the audio and sine wave, to determine the best algorithm. It observed that the RLS algorithm gave the best, and the most stable results. This algorithm required a comparatively smaller number of iterations to result with a convergence rate that was very fast. Furthermore, despite the value of the MSE and Error rate being small, the algorithm is very complex and is expensive. From the results of the NLMS algorithm, can observe instability especially in the Channel Equalization application, and that it had a greater iteration number than the RLS algorithm. Moreover, although it is cheap, the value of MSE and Error rate are large and the convergence rate is fast. The LMS algorithm is a simple, non-complex algorithm, but needs a very large number of iterations, and has a low convergence rate. Its MSE and Error rate is also very big.

Algorithm	Stability	Complex	Number of Iterations	Convergence of Rate	MSE
LMS	Stabile	Simple	Very high	Slow	Very big
NLMS	Non Stabile	Simple	High	Fast	Big
RLS	Stabile	Complex	Low	Very fast	Small

Table 5.1 Comparison Between the Three Algorithms

From the table above, it can be seen that the RLS algorithm is better, more robust and more stable than the MLNS algorithm and the LMS algorithm.

Additionally, RLS algorithm has a very low error ratio compared with the other algorithms.

The adaptive filters are more effective in the color noise than in the Gaussian noise. Since the MSE is more stable, it spends comparatively little time to filter the color noise. The NLMS algorithm failed in the Channel Equalization. The RLS algorithm shows the best performance among the three algorithms.

5.2 Future Work

The future work in the field of the adaptive filter is to further develop the RLS algorithm in order tominimiseit'sdisadvantages most importantly it's complex equations and high cost**.**

REFERENCES

1. **Dhiman, J., Ahmad, S., & Gulia, K. (2013),** "*Comparison Between Adaptive Filter Algorithms (LMS, NLMS and RLS)*". International Journal of Science, Engineering and Technology Research, 2(5), pp-1100

2. **Hameed, A. A. (2012),** "*Real-Time Noise Cancellation Using Adaptive Algorithms (Doctoral dissertation*", Eastern Mediterranean University (EMU)).

3. **Tokhi, M. O., & Leitch, R. R. (1991, September), "***Design and Implementation of Self-Tuning Active Noise Control Systems. In Control Theory and Applications*", IEE Proceedings D (Vol. 138, No. 5, pp. 421-430). IET.

4. **Douglas, S. C. (1999),** *"Introduction to Adaptive Filters"*, The Journal of the Acoustical Society of America, 25(6), 1130-1136

5. **Tienan, L., Liguo, W., Baochang, X., Aihua, X., & Hang, Z. (2002).** "*Adaptive Noise Canceler and its Applications For Systems with Time-Variant Correlative Noise. In Intelligent Control and Automation*", 2002. Proceedings of the 4th World Congress on (Vol. 2, pp. 1412-1415). IEEE.

6. **Simon, H. , (2002), "***Adaptive Filter Theory*". Prentice Hall, 2, 478-481**.**

7. **Gale, Z. (2011),**"*GMC Terrain Crossover Uses Noise Cancelling Tech to Quiet Cabin, Help Efficiency*". Truck Trend Magazine**.**

8. **Bellanger, M. (2001),** "*Adaptive Digital Filters*". Signal Processing, IEEE Transactions on, 44(7), 1323-1327**..**

9. **Dhamija, S., & Tayal, A. (2011),** "*Adaptive Filtering Algorithms for Channel Equalization and Echo Cancellation*". International Journal of Computer Technology and Applications, 2, 1577-1581.

10. **Falcão, R. M. A. (2012),** "*Adaptive Filtering to Noise Cancellation*" (Doctoral dissertation, Universidade do Porto).

11. **Farhang-Boroujeny, B. (2013),** "*Adaptive Filters*": Theory and Applications. John Wiley & Sons. ". International Journal of Computer Technology and Applications, 2, 1653-1558

12. **Manikandan, S., & Ebenezer, D. (2008, January***),"A New Prediction Based Adaptive Median Filter for Restoration of Degraded Audio Signals. In Signal Processing, Communications and*

Networking", 2008. ICSCN'08. International Conference on (pp. 203-207). IEEE.

13. **Tummarello, G., Nardini, F., & Piazza, F. (2003, May),** "*Step Size Control in NLMS Acoustic Echo Cancellation Using a Neural Network. in Circuits and Systems*", 2003. ISCAS'03. Proceedings of the 2003 International Symposium on (Vol. 5, pp. V-705). IEEE.

14. **Widrow, B., Glover Jr, J. R., McCool, J. M., Kaunitz, J., Williams, C. S., Hearn, R. H., ... & Goodlin, R. C. (1975),** *"Adaptive Noise Cancelling"*: Principles and Applications. Proceedings of the IEEE, 63(12), 1692-1716.

15. **Dinniz, P. S. (2008**),"*Adaptive Filtering: Algorithms and practical Implementations*". Rio de Janeiro: Springer Sciences+ Business Media, LLC. P1, 77-78.

16. **Park, J. H., Kim, J., Lee, J. K., Kim, S. W., & Lee, C. W. (2007, March),** *"Performance Analysis of Adaptive Noise Canceller in Real-Time Automobile Environments. In System Theory*", 2007. SSST'07. Thirty-Ninth Southeastern Symposium on (pp. 169-172). IEEE.

17. **Widrow, B., & Lehr, M. A. (1995)**, *"Noise Canceling and Channel Equalization"*. In *Conf.* Rec. IEEE Int. Conf. Communications (pp. 648-650).

18. **Avalos, J. G., Velazquez, J., & Sanchez, J. C. (2011),** "*Applications of Adaptive Filtering*". INTECH Open Access Publisher.

19. **Nagumo, J. I., & Noda, A. (1967),** "*A Learning Method For System Identification. Automatic Control*", IEEE Transactions on, 12(3), 282-287.

20. **Sinn, C. V., & Gotze, J. (2003, April),** "*Comparative Study of Techniques to Compute FIR Filter Weights in Adaptive Channel Equalization. In Acoustics, Speech, and Signal Processing*", 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on (Vol. 6, pp. VI-217). IEEE.

21. **Bitmead, R. R., & Anderson, B. D. (1980),** "*Performance of Adaptive Estimation Algorithms Independent Random Environments*". Automatic Control, IEEE Transactions on, 25(4), 788-794**.**

22. **Panuska, V. (1969, November),** "*An adaptive Recursive-Least-Squares Identification Algorithm. In Adaptive Processes (8th) Decision and Control*", 1969 IEEE Symposium on (pp. 65-65). IEEE.

23. **Diniz, P. S. (2013**),"*Fundamentals of Adaptive Filtering*". In Adaptive Filtering (pp. 13-78). Springer US.

24. **Górriz, J. M., Ramírez, J., Cruces-Alvarez, S., Puntonet, C. G., Lang, E. W., & Erdogmus, D. (2009),** *"A Novel LMS Algorithm Applied to Adaptive Noise Cancellation. Signal Processing Letters*", IEEE, 16(1), 34-37.

25. **Tate, C. N., & Goodyear, C. C. (1983),** "*Note on the Convergence of Linear Predictive Filters, Adapted Using the LMS Algorithm*". Electronic Circuits and Systems, IEE Proceedings G, 130(2), 61-64.

26. **Kong, V. S. X. (1995),** *"Adaptive Signal Processing Algorithms".* 1995. SSST'07. Thirty-Ninth Southeastern Symposium on (pp.170-173). IEEE

27. **Singh, A. (2001),** *"Adaptive Noise Cancellation".*2001 IEEE Symposium on (pp. 67-69). IEEE.

28. **Starer, D., & Nehorai, A. (1992),** *"Newton Algorithms for Conditional and Unconditional Maximum Likelihood Estimation of the parameters of Exponential Signals in Noise".* Signal Processing, IEEE Transactions on, 40(6), 1528-1534.

29. **Caraiscos, C., & Liu, B. (1983, April),** *"A round-off error analysis of the LMS adaptive algorithm. in Acoustics, Speech, and Signal Processing",* IEEE International Conference on ICASSP'83. (Vol. 8, pp. 29-32). IEEE.

30. **Wheeler, P. D., Rawlinson, R. D., Pelc, S. F., & Dorey, A. P. (1978, May),** *"The Development and Testing of an Active Noise Reduction System For Use in Ear Defenders",* in Inter-Noise and Noise-Con Congress and Conference Proceedings (Vol. 1978, No. 1, pp. 977-982). Institute of Noise Control Engineering.

APPENDICES A

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: ALMALLAHMED, Omar

Date and Place of Birth: 25 September 1979, Mosul/Iraq

Marital Status: Married

Phone: +90 5346115766

Email: omar_mohyaldeen@yahoo.com, c1271551@student.cankaya.edu.tr

EDUCATION

Degree	**Institution**	**Year of Graduation**
M.Sc.	Çankaya Univ. , Computer Engineering	2015
B.Sc.	Technical College / Mosul	2003
High School	Al Risalah Secondary School	1997

WORK EXPERIENCE

Year	**Place**	**Enrollment**
2005- Present	The Ministry of Education / General – Directorate of Education in Nineveh	Specialist

FOREIN LANGUAGES

Arabic, Advanced English, Beginner Turkish

PUBLICATIONS

1. Omar Mohyaldeen., "*Comparison Between Embedding on Edges in Spatial and Frequency Domains*", The Second International Conference on Education Technologies and Computers (ICETC2015) IEEE.

HOBBIES

Reading, Traveling, Meeting New People, Swimming, Football, Watching Movies and Listening to Music.