Real Time Implementation of Adaptive Line Enhancer

Ahmed Darghaoth

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

Master of Science
in
Computer Engineering

Eastern Mediterranean University
June 2013
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

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

Assoc. Prof. Dr. Muhammed Salamah
Chair, Department of Computer Engineering

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

Prof. Dr. Hasan Kömürcügil
Supervisor

Examining Committee

1. Prof. Dr. Hakan Altınçay

2. Prof. Dr. Hasan Kömürcügil

3. Prof. Dr. Hüseyin Özkaramanlı

ABSTRACT

The contamination of a desired signal by noise (undesired signal) is a main problem encountered in many applications. The digital filters with fixed coefficients exhibit satisfactory performance in cancelling the noise when the desired signal has stationary characteristics. However, when the desired signal has changing characteristics, there may be a performance degradation in eliminating the noise by a digital filter with fixed coefficients. As a remedy to this, adaptive noise cancellation method is used. In the adaptive noise cancellation method, the coefficients of the digital filter are updated by using an adaptive algorithm, until a minimum error is obtained.

In this thesis, the principles of adaptive noise cancellation have been studied by using an adaptive line enhancer (ALE) configuration. The ALE configuration is simulated by using Simulink of Matlab and is implemented by using a digital signal processor (TI TMS320C6416T) in real time.

The performance of ALE under different signal to noise ratio, number of filter coefficients, and step size of the adaptive algorithm has been studied extensively. When step size is small it leads to a good estimation of the sinusoidal signal, but the convergence speed slows down. The ALE output becomes more accurate when more coefficients are used in the filter. When SNR value is increased the estimation becomes more successful resulting in a small error at the ALE output.

Keywords: Adaptive Filters, Adaptive Noise Cancellation, Least Mean Square Algorithm, Adaptive Line Enhancer, Signal to Noise Ratio

ÖZ

Birçok uygulamada, istenen bir sinyalin gürültü tarafından kirlenmesi karşı karşıya kalınan bir problemdir. İstenen sinyalin durağan bir yapısı olduğu durumda, sabit katsayılı sayısal süzgeçler gürültüyü yok etmek için yeterli bir performans sergilerler. Bununla birlikte, istenen sinyalin durağan bir yapısı olmadıgı durumda, sabit katsayılı sayısal süzgeçler gürültüyü yok etme işleminde bir performans azalması ortaya çıkmaktadır.

Bu probleme çare olarak, uyarlanabilir gürültü yoketme metodu kullanılır. Uyarlanabilir gürültü yoketme metodunda, sayısal süzgeçin katsayıları uyarlanabilir bir algoritma vasıtasıyla hata sinyalinin değeri minimum olana kadar güncellenmektedir.

Bu tezde, uyarlanabilir gürültü yoketme metodunun prensipleri uyarlanabilir kanal yükselticisi kullanarak çalışılmıştır. Uyarlanabilir kanal yükselticisinin Matlab'ın Simulink ortamında benzetimi yapılmış ve buradaki Simulink modeli sayısal sinyal işlemcisine (TI TMS320C6416T) yüklenerek, gerçek zamanda Matlab'dan bağımsız olarak kendi başına çalışması sağlanmıştır.

Uyarlanabilir kanal yükselticisinin farklı sinyal gürültü oranı, süzgeç katsayı sayısı, ve uyarlanabilir algoritmanın adım büyüklüğü altındaki performansı ayrıntılı olarak çalışılmıştır. Adım büyüklüğü küçük olduğu zaman, istenen sinüs sinyalinin kestirimi iyi olur, fakat yakınsama hızı yavaşlar. Süzgeçteki katsayıların sayısı artırıldıkça, ALE çıkışı (istenen sinüs sinyali) daha doğru olar. Sinyal gürültü oranı artırıldıkça, sinyal kestirimi daha başarılı olur ve ALE çıkışındaki hata sinyalinin küçülmesine neden olur.

Anahtar Kelimeler: Uyarlanabilir Süzgeçler, Uyarlanabilir Gürültü Yoketme, Enaz Ortalama Kare Algoritması, Uyarlanabilir Kanal Yükselticisi, Sinyal Gürültü Oranı.

ACKNOWLEDGMENT

Praise be to God, prayer and peace be upon the Messenger of God.

While I finish this thesis I have to thank each of gave me a favor.

I would like to thank Prof. Dr. Hasan Kömürcügil for his follow-up research from beginning to evaluate the results in the end, pray for God to reward him with good.

I also extend my sincere thanks and great gratitude to the Eastern Mediterranean University and the chairmanship of the department of the computer engineering for their help during the search and what they gave me in order to facilitate the difficulties I faced.

I cannot fail at this point to thank my esteemed teachers and employees of the department of the computer engineering why they gave me help in completing this research in the form in which it is.

It is fulfilling to extend my thanks to my colleagues who have supported me during the research period with my wishes to them for success.

Finally, I find the words of thanks powerless the expression of what I have gratitude for my family because accorded the care and support was good given after God in the completion of this research.

TABLE OF CONTENTS

LIST OF TABLE

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Introduction

In recent years of the last century there has been increased interest in adaptive filters for various signal processing applications. Adaptive filters have various configurations. The adaptive filter (ADF) configuration, known as adaptive line enhancer (ALE) has been previously proposed by Widrow *et al.* (Widrow, et al., 1975) for separating a correlated component from uncorrelated components. The ALE structure has been successfully applied to spectral estimation by Griffiths (Griffiths, 1975) and to interference refusal by Symons (Symons, 1978). The steady-state response of the ALE under multiple sinusoids has been studied by Zeidler *et al.* (Zeidler, J., Satorius, Chabries, & Wexler, 1978). The response of the ALE to a chirped and Doppler shifted sinusoid has been analyzed by Treichler (Treichler, 1980). The signal to noise ratio (SNR) performance of ALE using the least mean square (LMS) algorithm has been studied by Nehorai and Malah (Nehorai, 1980). The optimized lattice-form of ALE for a sinusoidal signal has been derived by Reddy *et al* (Reddy, Egardt, & Kailath, 1987).

Multiplicative noise models are also considered in some signal processing applications (Swami, 1994). Mounir Ghogho studied the analytic behavior of the LMS-ALE for sinusoids corrupted by multiplicative and additive noise (Ghogho, 1998). Fenglin and Chris studied the frequency properties of an ALE (Fenglin &

Chris, 2004). Asaad used white and color noise cancellation using adaptive feedback cross-coupled line enhancer filter. Saed *et al* studied new ALE based on singular spectrum analysis (Saeid, Tracey, & Abolghasemi, 2012).

This thesis talks about the overall problem of separating a signal that has been corrupted by the noise. The separation process is performed by using an ALE with LMS algorithm. The performance of ALE for one and two sinusoidal signal case is investigated under different parameters such as signal to noise ratio, filter length and step size of the LMS algorithm. Simulation results obtained from Simulink and experimental results obtained from a DSP in real time are presented.

1.2 Adaptive Filtering

The purpose of an adaptive filter (ADF) is to remove the noise from a signal, which is corrupted by a noise. The ADF has variable characteristics whose coefficients cannot be specified in advance. The coefficients are altered mechanically and intelligently by an adaptive algorithm. This algorithm has an important role of supporting ADF to be applied in fields where the exact filtering operation required is unidentified or is non-stationary (M.Kuo & Lee, 2001).

ADFs are usually classified under two categories: finite-impulse-response (FIR) adaptive filter (M.Kuo & Lee, 2001); this filter’s impulse response has a finite duration since it goes to zero after a finite time, and infinite impulse-response (IIR) adaptive filter (M.Kuo & Lee, 2001); it has an internal feedback mechanism and continues to respond indefinitely. IIR adaptive filters have diverse applications and they are beyond the scope of this thesis.

The ADF have various configurations as shown in the Figure 1.1 (Ifeachor & Jervis, 2002) :

(i) Adaptive noise canceller (ANC).

(ii) System-modelling (or system identification).

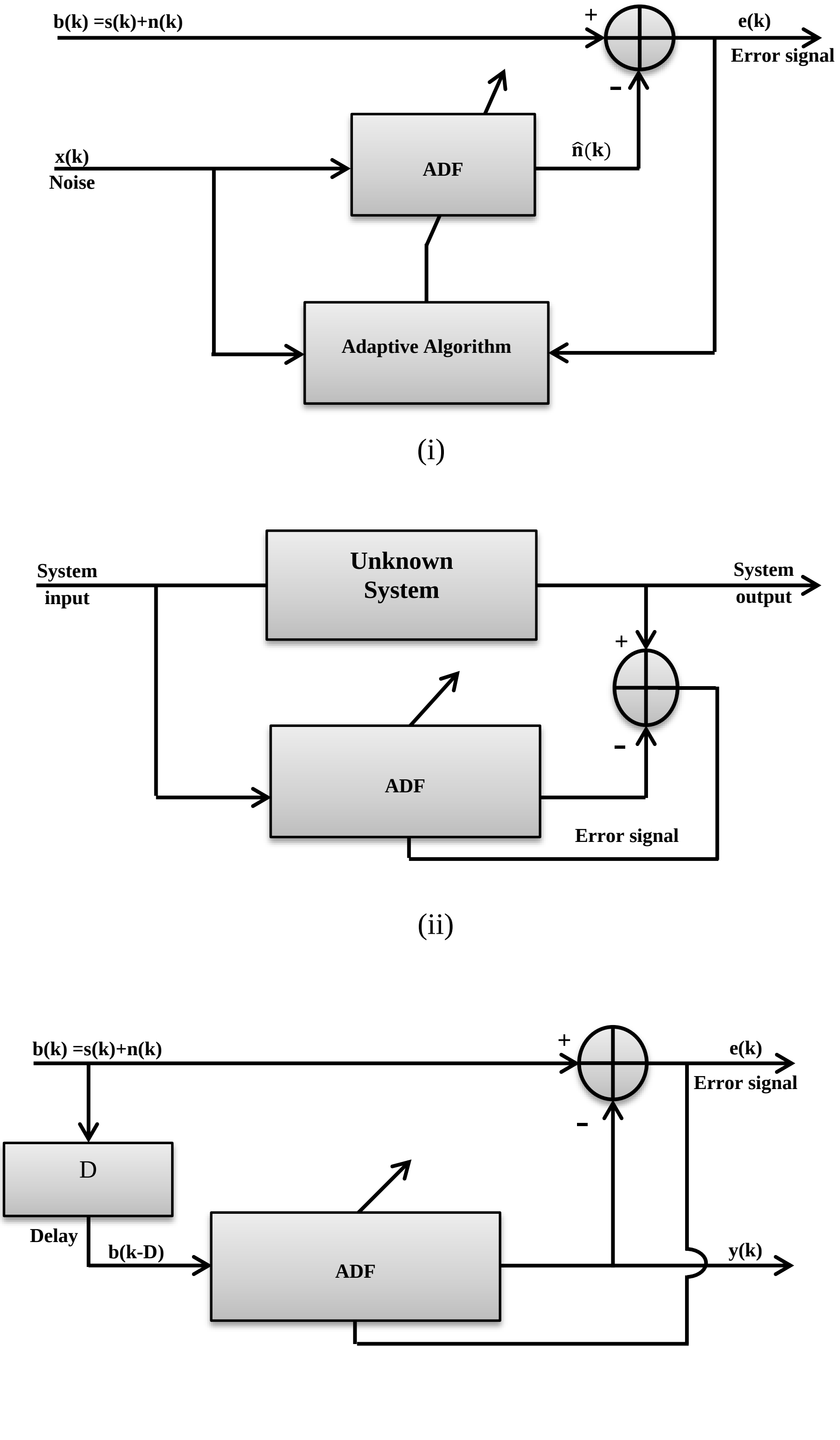
(iii) Adaptive-line-enhancer (ALE).

The ANC configuration which is shown in Figure 1.1(i) has two input signals b(k) and x(k). The signal b(k) is assumed to be b(k) = s(k)+n(k). The s(k) is uncorrelated with x(k) and n(k), and n(k) is correlated with x(k). The signal x(k) is processed by ADF to produce an estimate $\hat{n}(k)$. The estimate of signal s(k) can be obtained by subtracting $\hat{n}(k)$ from b(k).

The system modelling configuration which is shown in Figure 1.1(ii) requires three modules to perform its task.

i. System input (noise signal).
ii. An unknown system module that is an IIR filter.
iii. An adaptive FIR filter module where the FIR filter has *L* coefficients that are adjusted by means of the LMS algorithm.

The three modules are conFigured as shown in Figure 1.1 (iii). From this configuration we can determine how closely the impulse response of the FIR model approximates the impulse response of the unknown system after the LMS algorithm has converged. The discussion about the ALE is given in section 1.2.2.



(i)

(ii)

(iii)

Figure 1.1: Some Configurations of the ADF

1.2.1 Adaptive Algorithms

Adaptive algorithms are used to update the weights, W(k) (coefficients) of the ADF and reduce the error signal, e(k), of the ADF according to some performance measures. These measures are convergence rate, minimum mean square error (MSE), computational complexity, stability, robustness, and filter length (number of weights).

There are several categories of adaptive-algorithms (Dhull, Arya, & Sahu, 2011) :

i. Least-Mean-Square (LMS) Algorithm.

ii. Recursive-Least-Squares (RLS) Algorithm .

iii. Kalman-Filter(KF) Algorithm.

The LMS algorithm is the most efficient algorithm in terms of memory storage and calculations. In addition, the LMS algorithm has been the least suffering in terms of the numerical stability problem when compared with the problem inherent in RLS, and KF algorithms. This makes LMS algorithm is one of the first choices in several applications. However, the RLS algorithm is the best in terms of convergence properties. The LMS algorithm will be explained in chapter 3.

The RLS algorithm was proposed in order to provide superior performance compared to those of the LMS algorithm and its variants, with few parameters to be predefined, especially in highly correlated environments. In the RLS algorithm, an estimate of the autocorrelation matrix is used to decorrelate the current input data. Also, the quality of the steady-state solution keeps on improving over time, eventually leading to an optimal solution (Haykin, 2002).

Even though the RLS algorithm has very good performance in such environments, it actually suffers from its high computational complexity. Also, In RLS algorithm, the forgetting factor has to be chosen carefully such that its value should be very close to one in order to ensure stability and convergence of the RLS algorithm. However, this in turn poses a limitation for the use of the algorithm because small values of forgetting factor may be required for signal tracking if the environment is non-stationary (Sayed, 2008).

The KF is a mathematical technique widely used in the digital computing of control systems, navigation systems, and outer-space vehicles. The KF extracts a signal from a long sequence of noisy or incomplete technical measurements. The KF, and its later extensions to nonlinear problems, represents one of the most widely applied products of modern control theory. The KF uses elements of estimation theory to obtain the best unbiased estimator of a state of a dynamic system using previous measurement knowledge. Previous algorithms suffer from the computational limitation of using all previous information to estimate the state of the system at the next time step. In contrast, the Kalman algorithm makes use of the previous time step information and makes a-priori and a-posteriori predictions which are corrected using the new measurement (Kalman, 1960). The RLS and KF algorithms are beyond the scope of this thesis.

1.2.2 Concept of Adaptive Filters as Adaptive Line Enhancer

In this section it will be described how it is possible to get a narrowband signal (s(k)) which is contaminated by a wideband noise (n(k)). Usually, the n(k) has larger power than the power of s(k) (M.Kuo & Lee, 2001) which results in negative decibels (dB) of signal to noise ratio (SNR). The configuration of an ALE is shown in the Figure 1.2.

The input signal of ALE is assumed to be b(k) = s(k)+n(k). The input signal of the ADF is the delayed version of b(k). The amount of delay must be chosen such that n(k) is de-correlated and s(k) is correlated. The ADF produces an estimate of s(k) which is denoted by $\hat{s}(k)$. The ADF is called linear prediction filter because it predicts the recent sample of s(k) from its previous samples and at the same time weakens the wideband signal. The error signal at the output of ALE can be written as

$$e(k) = b(k)-y(k) = s(k) + n(k) - \hat{s}(k) \approx n(k) \tag{1.1}$$

It is clear that error signal is a wideband noise and the output of the ADF is narrowband signal. Because of its ability to enhance sinusoidal signal in the presence of noise, this scheme is therefore called adaptive line enhancer (Vinay & Jone, 1996), (Bose, 2003).

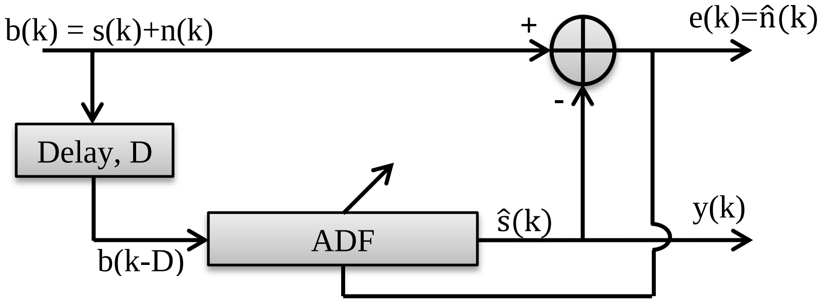


Figure 1.2: The Structure of the ALE

1.3 Objective of the Dissertation

The main objective of this dissertation is to implement a real-time noise cancellation application by using ALE configuration. The ALE configuration is simulated using SIMULINK of the MATLAB, and is implemented by using a DSP card (Texas Instruments (TI) TMS320C TMS320C6416T Digital Signal Processor) in real-time.

The performance of ALE under different signal to noise ratio, number of coefficient, and step size, has been studied extensively. Simulation and experimental results are presented and discussed.

Chapter 2

ADAPTIVE NOISE CANCELLATION

2.1 Basic Principles of Adaptive Noise Cancellation

Adaptive noise cancelation (ANC) is an alternative technique of estimating signals corrupted by additive noise. The ANC is greatly useful in many applications which are listed below (Widrow, et al., 1975) .

i. Cancelling 60Hz Interference in Electrocardiography.

ii. Cancelling the Donor ECG in Heart Transplant Electrocardiography.

iii. Cancelling the Maternal ECG in Fetal Electrocardiograph.

iv. Cancelling Noise in Speech Signals.

v. Cancelling Antenna Sidelobe Interference.

vi. Cancelling Periodic-Interference without an External Reference Source.

vii. Adaptive Self Tuning Filter.

The ANC configuration and its two input signals are $b(k)$ and $x(k)$ are shown in Figure 2.1. The signal $b(k)$ is assumed to $b(k) = s(k) + n(k)$. The signal $s(k)$ is uncorrelated with $x(k)$ and $n(k)$, and $n(k)$ is correlated with $x(k)$. The signal $x(k)$, which is a measure of the contaminating signal, is processed by the ADF to produce an estimate $\hat{n}(k)$. The estimate of signal $\hat{s}(k)$ can be obtained by subtracting $\hat{n}(k)$ from $b(k)$ as follows.

$$e(k) = b(k) - \hat{n}(k) = s(k) + n(k) - \hat{n}(k) \approx \hat{s}(k) \tag{2.1}$$

Hence, it is obvious from equation (2.1) that the ANC scheme removes noise from b(k) and produces the estimate of desired signal s(k).

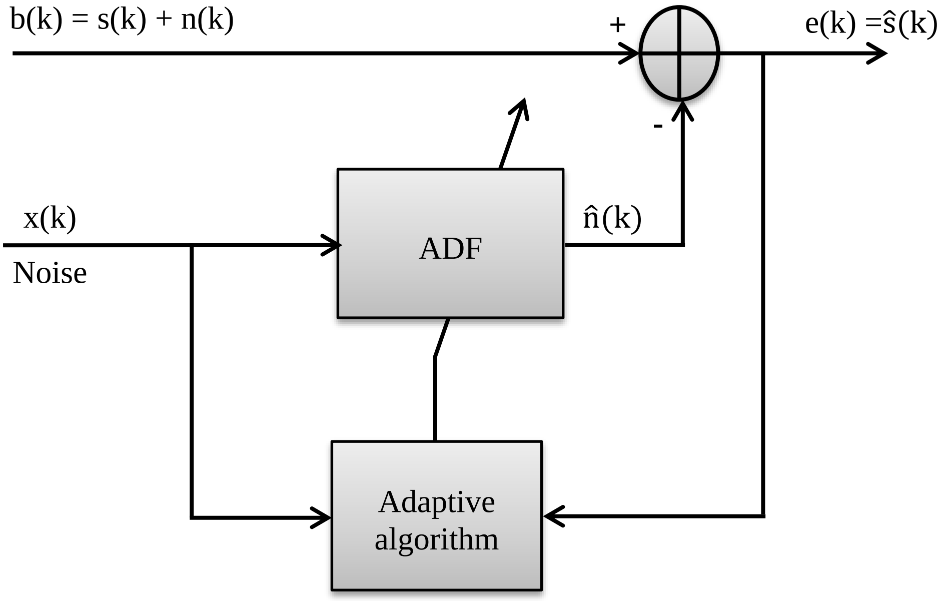


Figure 2.1: Adaptive Noise Canceller Scheme

2.2 Wiener Filter Theory in Adaptive Noise Cancellation

The adaptive filtering algorithms depend on Wiener filter (WF) principle shown in Figure 2.2. The WF can be considered as the optimal filter which achieves the required objective. This filter has two input signals x(k) (noise) and b(k) (signal plus noise). It should be noted that b(k) is correlated with x(k). The signal x(k) is processed by WF so that the output signal is the optimal estimate of signal b(k). The error signal e(k) can be produced by subtracting y(k) from b(k) (Ifeachor & Jervis, 2002), (Bose, 2003), (Kaur, 2011) as follows.

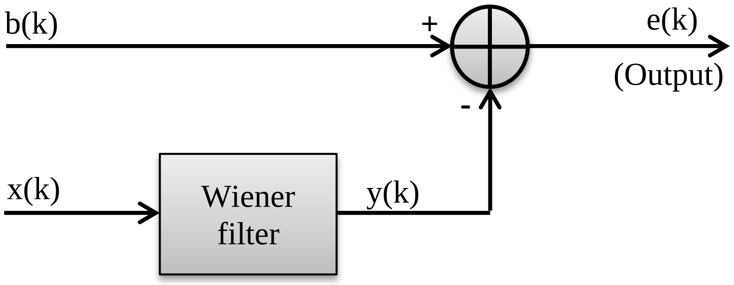


Figure 2.2: The Wiener Filter Configuration

$$\mathrm{e(k)} = \mathrm{b(k)} - \mathrm{y(k)} = \mathrm{b(k)} - \boldsymbol{W}^T(\mathrm{k})\boldsymbol{X}(\mathrm{k})$$

$$= \mathrm{b(k)} - \sum_{i=0}^{L-1} W_{\mathrm{k}}(i)\,\mathrm{x}(\mathrm{k}-i) \tag{2.2}$$

Where L represents the number of coefficients in the WF (FIR filter) and $\boldsymbol{X}$(k) and weight vector $\boldsymbol{W}$ are given as

$$\boldsymbol{X}(\mathrm{k}) = \begin{bmatrix} \mathrm{x(k)} \\ \mathrm{x(k-1)} \\ \vdots \\ \mathrm{x}(\mathrm{k}-(L-1)) \end{bmatrix} \quad \boldsymbol{W}(i) = \begin{bmatrix} W(0) \\ W(1) \\ \vdots \\ W(L-1) \end{bmatrix}$$

The estimated value of the squared error is given as

$$\mathrm{e}^2(\mathrm{k}) = \mathrm{b}^2(\mathrm{k}) - 2\mathrm{b(k)}\boldsymbol{X}^T(\mathrm{k})\boldsymbol{W}(\mathrm{k}) + \boldsymbol{W}^T(\mathrm{k})\boldsymbol{X}(\mathrm{k})\boldsymbol{X}^T(\mathrm{k})\boldsymbol{W}(\mathrm{k}) \tag{2.3}$$

The mean square error (MSE) (J) is obtained by taking the expectations E[.] of both sides of equation (2.3).

$$J = E[\mathrm{e}^2(\mathrm{k})] = E[\mathrm{b}^2(\mathrm{k})] - 2E[\mathrm{b(k)}\boldsymbol{X}^T(\mathrm{k})\boldsymbol{W}(\mathrm{k})] + E[\boldsymbol{W}^T(\mathrm{k})\,\boldsymbol{X}(\mathrm{k})\boldsymbol{X}^T(\mathrm{k})\,\boldsymbol{W}(\mathrm{k})]$$

After replacing $E[\mathrm{b}^2(\mathrm{k})]$ by σ^2, $E[\mathrm{b}(\mathrm{k})\boldsymbol{X}(\mathrm{k})]$ by $\boldsymbol{P}$, and $E[\,\boldsymbol{X}(\mathrm{k})\boldsymbol{X}^T(\mathrm{k})]$ by $\boldsymbol{R}$, the above equation reduces to

$$\mathrm{MSE} = J = \sigma^2 + 2\boldsymbol{P}^T\boldsymbol{W}(\mathrm{k}) + \boldsymbol{W}^T(\mathrm{k})\boldsymbol{R}\boldsymbol{W}(\mathrm{k}) \tag{2.4}$$

Note that the $\boldsymbol{P}$ is cross-correlation vector (with length L) and $\boldsymbol{R}$ is auto-correlation matrix (with size $L{\times}L$). Figure 2.3 shows the plot of the MSE against the filter coefficients. It is clear that it looks like a bowl with a unique minimum point.

Starting with the equation for the mean square error (equation 2.4), derive the Wiener-Hopf equation.

The gradient, $\nabla(\mathrm{k})$, of the MSE is obtained by differentiating the MSE with respect to the weight vector $\boldsymbol{W}$, and setting the result to zero (Haykin, 2002):

$$\nabla(\mathrm{k}) = \frac{dJ}{d\boldsymbol{W}(\mathrm{k})} = \frac{d\sigma^2}{d\boldsymbol{W}(\mathrm{k})} + \frac{d(2\boldsymbol{P}^T\boldsymbol{W}(\mathrm{k}))}{d\boldsymbol{W}(\mathrm{k})} + \frac{d(\boldsymbol{W}^T\boldsymbol{R}\boldsymbol{W}(\mathrm{k}))}{d\boldsymbol{W}(\mathrm{k})} \tag{2.5}$$

Now,

$$\frac{d\sigma^2}{d\boldsymbol{W}(\mathrm{k})} = 0 \tag{2.6}$$

$$\frac{d(2\boldsymbol{P}^T\boldsymbol{W}(\mathrm{k}))}{d\boldsymbol{W}(\mathrm{k})} = \; -2\boldsymbol{P} \tag{2.7}$$

$$\frac{d(\boldsymbol{W}^T(\mathrm{k})\boldsymbol{R}\boldsymbol{W}(\mathrm{k}))}{d\boldsymbol{W}(\mathrm{k})} = 2\boldsymbol{R}\boldsymbol{W}(\mathrm{k}) \tag{2.8}$$

Using these results, and equation 2.5 becomes

$$\nabla(\mathrm{k}) = -2\boldsymbol{P} + 2\boldsymbol{RW}(\mathrm{k}) \tag{2.9}$$

Setting $\nabla(\mathrm{k}) = 0$, and equation 2.5 becomes

$$\nabla(\mathrm{k}) = \frac{dJ}{d\boldsymbol{W}(\mathrm{k})} = -2\boldsymbol{P} + 2\boldsymbol{RW}(\mathrm{k}) = 0 \tag{2.10}$$

The optimum coefficient vector is then given by

$$\boldsymbol{W}_{\mathrm{opt}} = \boldsymbol{R}^{-1}\boldsymbol{P} \tag{2.11}$$

Equation (2.11) is called the Wiener-Hopf equation or solution. The adjustment of the filter weights is the task of the ADF algorithm.

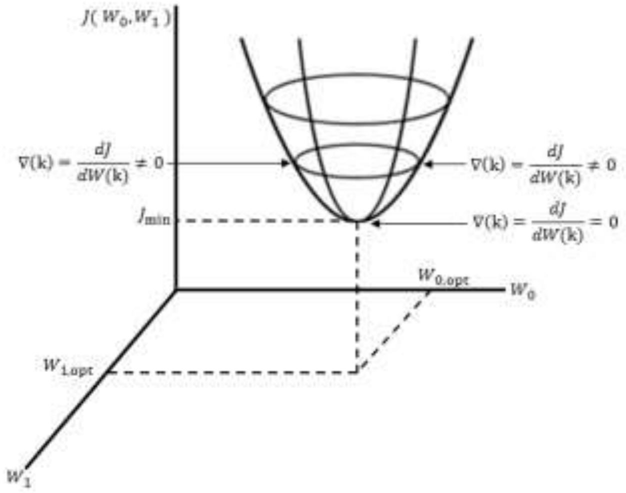

$J(W_0, W_1)$
$\nabla(\mathrm{k}) = \frac{dJ}{dW(\mathrm{k})} \neq 0$
$\nabla(\mathrm{k}) = \frac{dJ}{dW(\mathrm{k})} \neq 0$
$J_{\min}$
$\nabla(\mathrm{k}) = \frac{dJ}{dW(\mathrm{k})} = 0$
$W_{0.\mathrm{opt}}$
W_0
$W_{1.\mathrm{opt}}$
W_1

Figure 2.3: The Error Performance Surface

2.3 Comparison of Wiener and Adaptive Filter Approaches

Table 1: Comparison of Wiener and Adaptive Filter Approaches.

Wiener filter (WF)	Adaptive filter (ADF)
i. When designing a WF a priori knowledge about the actual statistical properties of the data to be processed is required. Only when the properties match the a priori information on which the design of the filter is based, the filter is optimum. It may impossible to design the Wiener filter because this information is not known completely and an appropriate design may no longer be optimum. A possible solution to this problem is to first estimate the statistical parameters of the relevant signals and then compute the filter parameters (Geboren, 1992). ii. The WF solves the signal estimation problem for stationary signals (Geboren, 1992). iii. The WF is optimal in the sense of the minimum MSE (Haykin, 2002).	i. The ADF relies for its operation on a recursive algorithm, which makes it possible for the filter to perform satisfactorily in an environment where complete knowledge of the relevant signal statistics is not available, or when the statistics slowly vary in time (Geboren, 1992), (Haykin, 2002). ii. The ADF is used for estimation of nonstationary signals and systems, or in applications where a sample-by sample adaptation of a process and/or a low processing delay is required (Geboren, 1992). iii. The ADF isn’t optimal filter, it adapts its weights, by using adaptive algorithm (LMS, RLS,and etc), until a minimum MSE is obtained between its output and the desired input signal. (Geboren, 1992), (Ifeachor & Jervis, 2002).

Chapter 3

THE LEAST MEAN SQUARE ALGORITHM

3.1 Introduction

The least mean square (LMS) algorithm is one of the simple, most successful and famous adaptive algorithms which can be applied easily in real-time [Said, 2008]. The LMS algorithm which was developed by Widrow et al. (Widrow, et al., 1975), trains its input correlation matrix and minimizes the MSE. The ratio of maximum to minimum eigenvalues has an important influence on the speed of convergence. If this ratio is small, it would speed up the rate of convergence, and if it is large, it would reduce the speed rate of convergence.

The LMS algorithm makes use of the steepest descent algorithm in which the weights are updated by using the following equation (Ifeachor & Jervis, 2002)

$$W(\mathrm{k}+1) = W(\mathrm{k}) - \mu\nabla(\mathrm{k}) \tag{3.1}$$

Where $W(\mathrm{k})$ is the weight, $\nabla(\mathrm{k})$ is the true gradient vector defined by (2.9), k is the kth sampling instant, and μ is the step size. It should be noted that the stability and the convergence rate depends on the value of step size.

It is clear that in order to be able to compute $W(\mathrm{k}+1)$, $\nabla(\mathrm{k})$ needs information about $\boldsymbol{R}$ and $\boldsymbol{P}$ as shown in equation (2.9).

However, the LMS-algorithm can estimate the weights of the filter without using the matrix $\boldsymbol{R}^{-1}$ in the Wiener-Hopf equation (2.11) by using the direct auto-correlation and cross-correlation computation in real-time. Then the update of weights can be obtained as

$$W(\mathrm{k}+1) = W(\mathrm{k}) - 2\mu\mathrm{e}(\mathrm{k})\boldsymbol{X}(\mathrm{k}) \tag{3.2}$$

Where

$$\mathrm{e}(\mathrm{k}) = \mathrm{y}(\mathrm{k}) - \boldsymbol{W}^{T}(\mathrm{k})\boldsymbol{X}(\mathrm{k}) \tag{3.3}$$

The LMS algorithm predicts the weights, and these predictions become better as the filter learns the characteristics of the signal. Finally, the weights are converged to their optimum values. The convergence is conditional given by

$$0 < \mu < \frac{1}{\lambda_{max}} \tag{3.4}$$

Where μ controls the convergence rate and the stability, λ_{max} is the maximum eigenvalue of the input data convergence matrix (Ifeachor & Jervis, 2002).

3.1.1 Implementation of LMS Algorithm

The computational steps for the LMS algorithm are summarized below (Dhull, Arya, & Sahu, 2011), (Ifeachor & Jervis, 2002)

1. First of all, set all the weights $W_{\mathrm{k}}(i),\ i = 0,1,\dots,L-1$, to zero.

   For each sampling instant, k =1,2,... , carry out step (2) to (4) shown below:

2. Calculate filter output

$$\mathrm{y(k)} = \sum_{i=0}^{L-1} W_{\mathrm{k}}(i)\,\mathrm{x(k} - i)$$

3. Calculate the error

$$\mathrm{e(k)} = \mathrm{b(k)} - \mathrm{y(k)}$$

4. Finally, update the next weights

$$W_{\mathrm{k+1}}(i) = W_{\mathrm{k}}(i) - 2\mu\mathrm{e(k)}X(\mathrm{k} - i)$$

The flowchart of the computational steps is given in Figure 3.1.

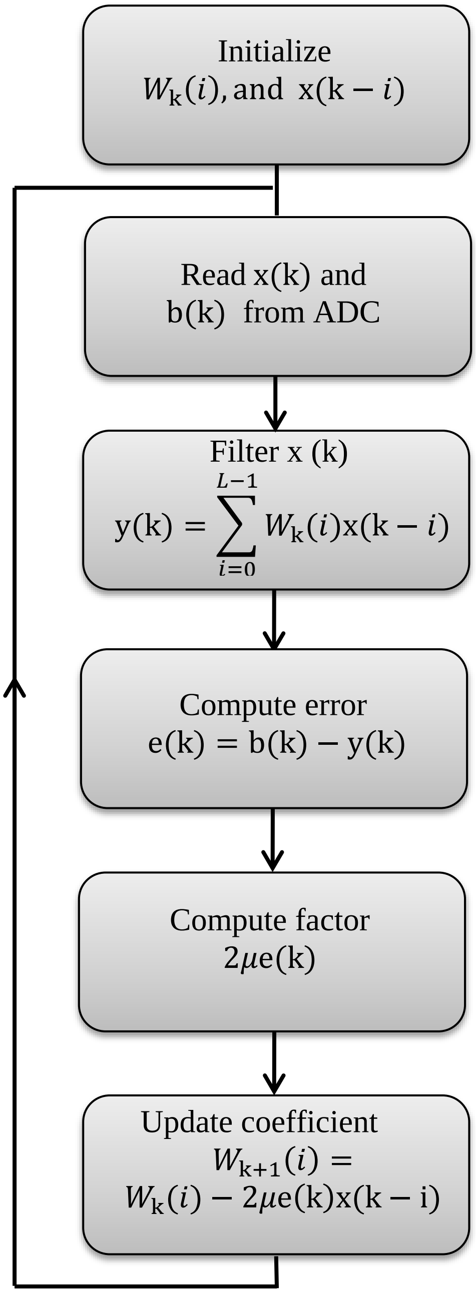


Figure 3.1: Flowchart for the LMS Algorithm

3.2 Stability of LMS Algorithm

Stability can be considered as an important performance measure for an adaptive system. Large step size may result in an undesired convergence of the LMS algorithm which may lead to an unstable operation of the adaptive system. In most cases, the adaptive systems are operated as marginally stable where the stability can be determined by the initial conditions, transfer function of the system and the step size (Haykin, 2002) (Boroujeny, 1998).

3.3 Convergence Properties of the LMS Algorithm

The convergence to the optimal weights (W_{opt}) is possible only if the step size (μ) is satisfied as in equation (3.4). The convergence from an initial weight (W_0) to the W_{opt} corresponds to the MSE convergence from $J(0)$ to J_{min}. Thus, the MSE convergence is moved toward its smallest value which is generally used as the performance criterion in adaptive systems because of its simplicity. The squared error $e^2(k)$ is non-stationary and also the weights adapt and move toward optimal weights through the adaptation process. The learning curve of a given adaptive algorithm can be obtained by plotting the MSE versus k. Since the MSE is considered as a performance criterion, the learning curve can be used to describe the transient behavior of the adaptive algorithm.

The time constant of each adaptive mode can be computed by using the step size, and the eigenvalue associated with that mode. The rate of convergence is limited by the slowest mode. Thus the overall MSE time constant can be obtained as (Haykin, 2002) (M.Kuo & Lee, 2001)

$$T_{mse} \cong \frac{1}{\mu\,\lambda_{min}} \tag{3.5}$$

Since the mathematical relationship between T_{mse} and μ are inversely related, this means that the greatest T_{mse} corresponds smaller μ, resulting in a slower speed of convergence. On the other hand, if μ has a large value, T_{mse} becomes smaller leading to a faster convergence (M.Kuo & Lee, 2001) (Vinay & Jone, 1996).

The maximum T_{mse} keeps an estimate performance of the filter, therefore only large eigenvalues will make the most important influence on the time of the convergence (M.Kuo & Lee, 2001). The ADF error convergence may be controlled by fewer modes than the number of ADF weights. Consequently, the MSE often converges more rapidly than the upper bound of equation(3.5) would suggest.

Because of the mathematical relationship between T_{mse} and λ_{min} , small valued λ_{min} gives a large T_{mse} , resulting in a slow convergence rate. If the λ_{max} is very large, it will limit the selection of μ , since small valued μ can satisfy the stability constraint.

If the maximum eigenvalue, λ_{max} , is very large and the minimum eigenvalue, λ_{min}, is very small, according to equation (3.5), T_{mse} will be very large, resulting in very slow convergence. As previously mentioned in equation (3.4), the fastest convergence occurs when $\mu = 1/\lambda_{max}$. Substituting $\mu = 1/\lambda_{max}$ into equation (3.5) gives (M.Kuo & Lee, 2001)

$$T_{mse} \leq \frac{\lambda_{max}}{\lambda_{min}} \tag{3.6}$$

When the input is stationary and μ is sufficiently small, equation (3.6) will lead to the convergence speed of the algorithm which depends on the eigenvalue spread of the matrix $\boldsymbol{R}$ (Asaad, 2008).

3.4 Definition of Mean Square Error

Firstly, the mean square error, (MSE) is defined as

$$\mathrm{MSE} = J = E[\mathrm{e}^2(\mathrm{k})] = \sigma^2 + 2\boldsymbol{P}^T\boldsymbol{W}(\mathrm{k}) + \boldsymbol{W}^T(\mathrm{k})\boldsymbol{RW}(\mathrm{k})$$

This equation was defined before in equation (2.4). Since the computation of $\nabla(\mathrm{k})$ requires information about $\boldsymbol{P}$ and $\boldsymbol{R}$, then it is estimated. The estimated gradient ($\widehat{\nabla}(\mathrm{n})$) results in the gradient estimation noise. The gradient of the true $\nabla(\mathrm{k}) \approx 0$, after the ADF algorithm converges, the gradient becomes zero $\nabla(\mathrm{n}) \approx 0$, but the estimated gradient $\widehat{\nabla}_{\mathrm{n}} \neq 0$. As it was indicated in equation (3.1), perturbing the gradient drives the weight vector $W_{\mathrm{k+1}}$ away from the optimum weight (W_{opt}) solution. Consequently, the estimation of the gradient noise avoids from remaining at the weight of the optimum solution in steady state. The result is that the weight is changed randomly about the optimum weight solution. Because the optimum weight solution corresponds to the MSE of the minimum point, when the weight moves away from the optimum weight solution, it causes a large MSE compared to the value of the minimum for it. Thus this will result in excess noise at output of the ADF (Ikuma & (Louis) Beex, 2008) (M.Kuo & Lee, 2001).

The noise in the weight vector causes excess MSE after convergence. This excess MSE is defined as

$$J_{excess} \approx \frac{\mu}{2} \; L \, R \, J_{min} \tag{3.7}$$

It is clear that the excess MSE (J_{excess}) is directly proportional to μ. It corresponds to the worst performance of the steady state after convergence for the biggest μ (Miyagi & Sakai, 2005). However, according to equation (3.5) the larger μ results in faster convergence (M.Kuo & Lee, 2001). There is a design trade-off between the convergence speed, and the J_{excess}.

Also, it is clear from equation (3.7) that the J_{excess}, is directly proportional to the filter length (L). The biggest L results in larger noise, and also, implies a smaller μ leading to a slower convergence. Finally, the biggest L means the characteristics of the filter are the best. The finite precision of the error will be influence by a chosen μ, and L (M.Kuo & Lee, 2001).

Chapter 4

ADAPTIVE LINE ENHANCER

4.1 Introduction

The ALE structure is shown in Figure 4.1. The input signal, b(k), which is converted into a digital signal from an analog signal b(t), consists of a narrowband signal s(k) and a broadband noise n(k). In the ALE structure, there are two input channels. One of them is treated as desired signal channel to the ADF. The other channel drives through a de-correlation delay (D) which is applied as a reference input signal to the ADF. The ADF adapts its weights, by using LMS algorithm, until a minimum MSE is obtained between its output y(k) and the desired input signal b(k) (Saeid, Tracey, & Abolghasemi, 2012), (Fenglin & Chris, 2004). When the adaptation process is completed, the ALE operates like a bandpass FIR filter having a peak in its frequency response at the frequency of the sinusoid. It should be noted that in the case of two sinusoids, there exists two peaks. The noise outside the frequency band of the signal is cancelled.

The performance of the ALE for a given application is determined by four parameters. These are filter length (*L*), signal to noise ratio (SNR), step size (μ), and delay (D). It is important to note that D should be greater than de-correlation time of the broadband signal n(k) and less than the de-correlation time of the narrowband signal s(k) for the correct operation of the ALE.

The ALE concept can also be applied to a number of problems such as suppression of narrowband interference in a broadband signal, and adaptive self-tuning filter. However, these applications are beyond the scope of this thesis.

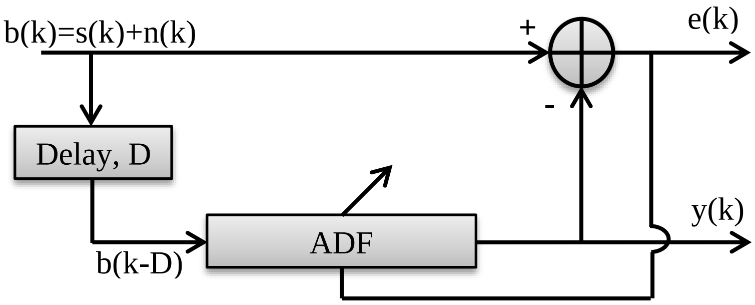


Figure 4.1: Structure of the ALE

4.2 ALE Using FIR Filter

The ADF in the ALE is usually a finite impulse response (FIR) filter as shown in Figure 4.2. The input b(k) consists of sinusoids, s(k), and white noise n(k).

The operation of ALE can be explained as follows. The delay D de-correlates the noise components of b(k) and introduces a small phase shift between the sinusoidal components. The ALE has a transfer function equivalent to that of a band-pass filter with center frequency equal to the frequency of sinusoidal components. In this case, the noise component of the delayed input is blocked. The ALE can be used to estimate and track instantaneous frequencies.

The transfer function of FIR within the ADF framework can be written as

$$\mathrm{H}_{\mathrm{ALE}}(\mathrm{z}) = \frac{Y(z)}{\mathrm{B}'(z)} = \sum_{i=0}^{L-1} W_i\, z^{-i} \tag{4.1}$$

Where $Y(z) = Z\{\mathrm{y(k)}\}$, $\mathrm{B}'(z) = Z\{\mathrm{b(k-D)}\}$.

Taking inverse z-transform of H(z) gives the impulse response of the FIR filter as

$$\mathrm{h}_{\mathrm{ALE}}(\mathrm{k}) \; = \{ W_0, W_1, W_2, \dots \; W_{L-1}\} \tag{4.2}$$

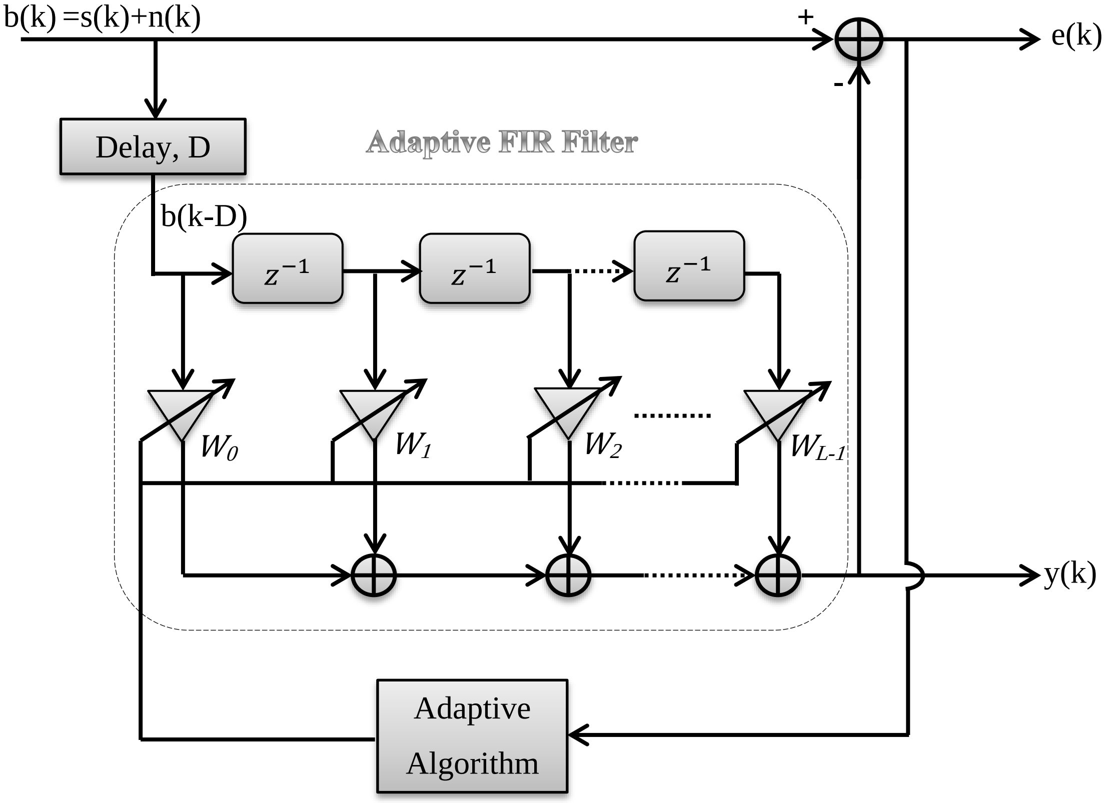


Figure 4.2: The Structure of the ALE with FIR Filter

4.3 Steady-State Behavior

When the LMS algorithm reaches its optimal value in the steady state, the weight W(k)will converge to W_{opt} (k) and the ALE output will be (Fenglin & Chris, 2004) (Widrow, et al., 1975) equal to

$$\mathrm{y(k)} = \mathbf{b}^T(\mathrm{k}-\mathrm{D})\; W_{\mathrm{opt}}\,(\mathrm{k}) \tag{4.3}$$

where **b** is input vector of the adaptive filter. Taking z transform of equation (4.3) gives

$$\mathrm{Y(z)} = \mathrm{B(z)}\,\mathrm{z}^{-\mathrm{D}}\,W_{\mathrm{opt}}\,(\mathrm{z}) \tag{4.4}$$

Therefore, the transfer function of the ALE in the steady state can be written as

$$\mathrm{H(z)} = \frac{\mathrm{Y(z)}}{\mathrm{B(z)}} = \mathrm{z}^{-\mathrm{D}}\;W_{\mathrm{opt}}\,(\mathrm{z}) \tag{4.5}$$

If the input signal has a frequency of ω_0, then the z-transformation of optimum weight vector can be written as

$$W_{\mathrm{opt}}\,(\mathrm{z}) = \frac{\mathrm{SNR}}{1 + L\,\mathrm{SNR}}\;\mathrm{z}_0^{-\mathrm{D}} \sum_{l=1}^{L} \left(\frac{\mathrm{z}}{\mathrm{z}_0}\right)^{l} \tag{4.6}$$

where $\mathrm{z}_0 = e^{j\omega_0 \mathrm{T}}$.

Substituting equation (4.6) into equation (4.5) and making use of geometric series, the transfer function of the ALE in the steady state can be written as

$$\mathrm{H(z)} = \frac{\mathrm{SNR}}{1 + L\,\mathrm{SNR}} \left(\frac{\mathrm{z}}{\mathrm{z}_0}\right)^{-\mathrm{D}} \frac{1 - \left(\frac{\mathrm{z}}{\mathrm{z}_0}\right)^{-L}}{1 - \left(\frac{\mathrm{z}}{\mathrm{z}_0}\right)^{-1}} \tag{4.7}$$

Transforming equation (4.7) into the frequency domain yields

$$\mathrm{H}(\omega, \omega_0\,) = \frac{\mathrm{SNR}}{1 + L\,\mathrm{SNR}}\,\mathrm{e}^{-\mathrm{j}(\omega - \,\omega_0)\mathrm{DT}}\,\frac{1 - \mathrm{e}^{-\mathrm{j}(\omega - \,\omega_0)L\mathrm{T}}}{1 - \mathrm{e}^{-\mathrm{j}(\omega - \,\omega_0)\mathrm{T}}}$$

$$= \frac{\mathrm{SNR}}{1 + L\,\mathrm{SNR}}\,\mathrm{e}^{-\mathrm{j}(\omega - \,\omega_0)(\mathrm{D} + \frac{L-1}{2})\mathrm{T}}\,\frac{\sin\left[\left(\frac{L}{2}\right)(\omega - \omega_0)\;\mathrm{T}\right]}{\sin\left[\left(\frac{1}{2}\right)(\omega - \omega_0)\;\mathrm{T}\right]} \tag{4.8}$$

Where T is the sampling period. When $\omega = \omega_0$, the frequency response of the ALE in the steady state can be obtained as

$$H(\omega, \omega_0\ ) = \frac{\mathrm{SNR}}{1 + L\ \mathrm{SNR}} \tag{4.9}$$

It is clear from equation (4.8) that the frequency response of the ALE varies with a change of the input characteristic frequency ω_0 meaning that the ALE possesses the property of time varying ability and frequency tracking filter functionality. It can be seen from equation (4.9) that the output signal of the ALE at the input frequency is same as the input signal with an amplitude gain of $A\mathrm{H}(\omega, \omega_0\ )$ (Fenglin & Chris, 2004). Where *A* is the amplitude of the sinusoidal signal.

4.4 The Influence of the Input SNR

The gain of the ALE with SNR and *L* can be computed by using equations (4.8) and (4.9). Three SNR ranges can be used to explain the enhancement function of the ALE (Fenglin & Chris, 2004).

i. High SNR (SNR>0 dB) . When the power of the broadband signal (noise) is less than that of narrowband signal, the gain value of the optimal filter becomes one. In this case, the broadband signal will be suppressed to a minimum value according to the LMS algorithm and the narrowband signal will through the ALE. In such a case, the output signal power can be written as

$$\begin{aligned} \mathrm{E}[\mathrm{y(k)}.\ \mathrm{y}^T(\mathrm{k})] &= \mathrm{E}\{\left[\ W^T{}_{\mathrm{opt}}(\mathrm{k}).\mathrm{b(k-D)}\right]\left[W_{\mathrm{opt}}(\mathrm{k}).\mathrm{b}^T(\mathrm{k-D})\right]\} \\ &= W^T{}_{\mathrm{opt}}\, R_{xx} W_{\mathrm{opt}}(\mathrm{k}) \end{aligned} \tag{4.10}$$

Where $W_{\mathrm{opt}}(\mathrm{k})$ is given by

$$W_{\mathrm{opt}}(\mathrm{k}) = \frac{A^2}{\alpha^2(\mathrm{k}) + LA^2}\ \mathrm{e}^{-\mathrm{j\omega DT}}\ \varphi \tag{4.11}$$

where φ is the factor vector, $\alpha^2(\mathrm{k})$ is the power of the noise, and R_{xx} is auto-correlation matrix of the filter input defined as

$$\begin{aligned} R_{xx} &= \mathrm{E}[\mathbf{b}(\mathrm{k}).\ \mathbf{b}^T(\mathrm{k})] = \mathrm{E}[\ \mathbf{b}(\mathrm{k}-\mathrm{D}).\mathbf{b}(\mathrm{k}-\mathrm{D})] \\ &= R_{ss} + R_{\mathrm{nn}} \end{aligned} \tag{4.12}$$

where R_{ss} is auto-correlation matrix of signal s(k) and R_{nn} is auto-correlation matrix of the of noise n(k).

Substituting equations (4.10) and (4.11) into equation (4.9), and simplifying gives

$$\mathrm{E}[\mathrm{y}(\mathrm{k}).\ \mathrm{y}^T(\mathrm{k})] = \frac{L\,A^2}{\alpha^2(\mathrm{k}) + LA^2}\left(A^2 + \frac{\alpha^2(\mathrm{k})}{L}\right) \tag{4.13}$$

Equation (4.13) shows that when b(k) passes through the ALE, the power of s(k) remains the same, while the noise power decreases to $\frac{1}{L}$ of its original power.

ii. Medium SNR (-10 dB < SNR < 0 dB). The gain value of the optimal filter changes from zero to one meaning that the narrowband signal can partially pass through the ALE.

iii. Low SNR (SNR<-10 dB). The gain value of the optimal filter becomes zero. In this case, the ALE cannot suppress the broadband signal or equivalently, the narrowband signal cannot be enhanced by the ALE. As a consequence of this, the broadband signal causes a large value of the mean error at the ALE output.

4.5 The Influence of the Filter Length

It is clear from equation (4.8) that the number of coefficients (*L*) in the FIR filter has an influence on the steady-state response of ALE. The value of *L* specifies how accurately a bandpass filter can be modeled by the ALE. In addition to this, it affects the convergence rate considerably. For instance, when *L* is large, the computation time of the whole system will increase which may slow down the convergence rate. When, it is small, this time the computation time will be decreased and the convergence rate will be faster (Boroujeny, 1998). On the other hand, it can affect the stability of ALE, and the MSE of the system.

Chapter 5

SIMULATION AND EXPERIMENTAL RESULTS

5.1 Introduction

The ALE configuration is simulated by using Simulink software. The performance of ALE with one and two sinusoids has been studied using various values of the parameters. An implementation of ALE configuration using a TI TMS320C6416T DSP card (see the details in Appendix) has been carried out in real time. Again, the performance of ALE in real time has been investigated by using different values of the parameters.

5.2 Simulation Results

5.2.1 One Sinusoid

5.2.1.1 The Influence of the Step Size

Consider the block diagram of the ALE shown in Figure 5.1 with D=12. The input sinusoidal signal which is defined as s(k)=sin($2\pi f_0/F_s$), where F_s=8000Hz and f_0=40Hz, is shown in Figure 5.2. The Gaussian white noise, n(k), has a zero mean with variance equal to one. The input signal b(k)=s(k)+n(k) which has a SNR of 10dB is shown in Figure 5.3.

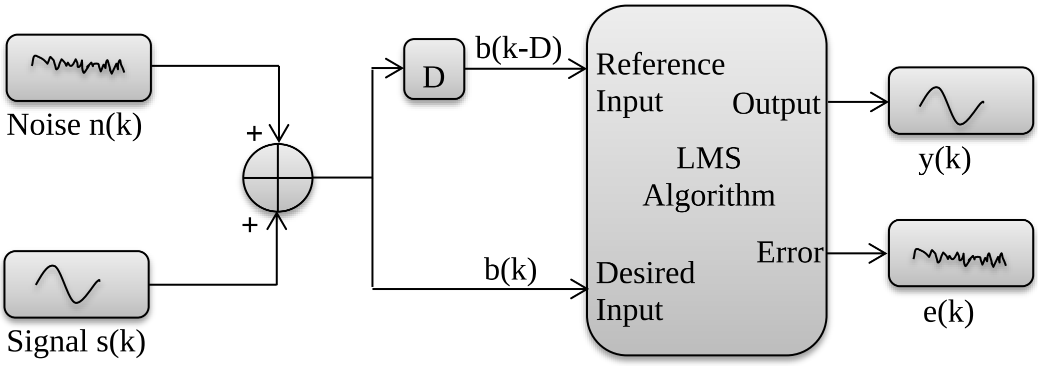


Figure 5.1: ALE Configuration with One Sinusoidal Input

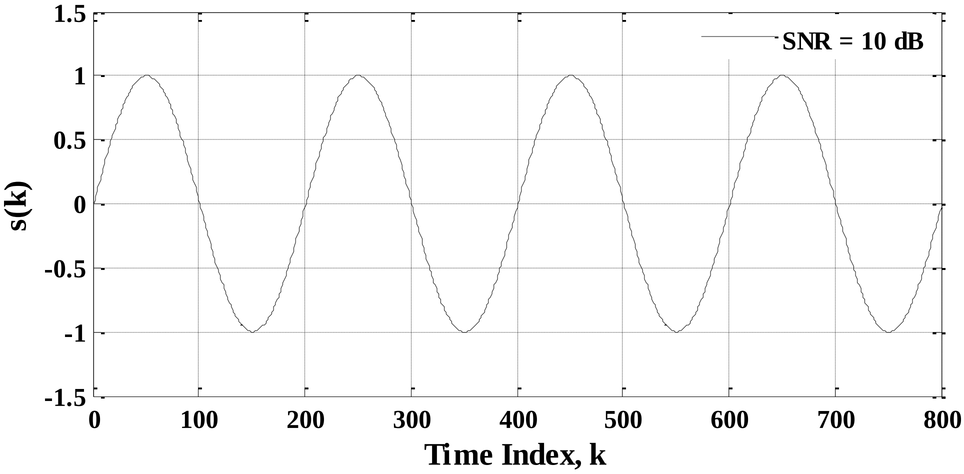


Figure 5.2: The Input Sinusoidal Signal s(k)

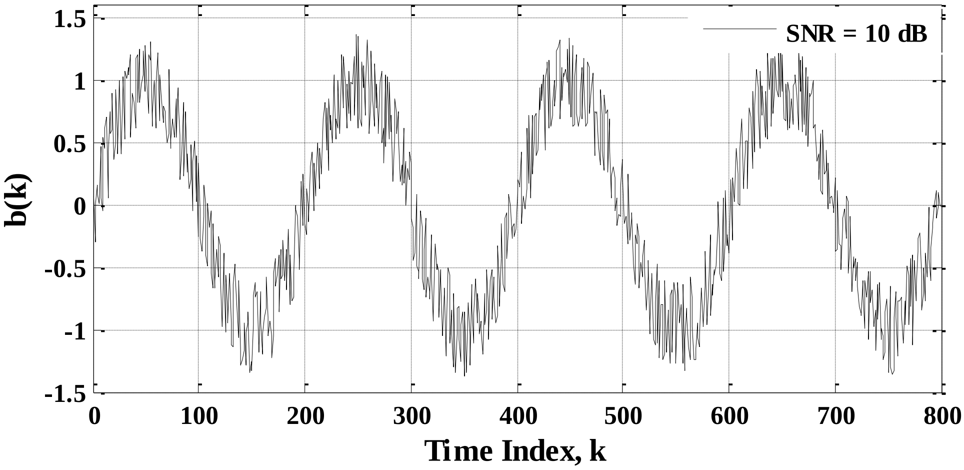


Figure 5.3: The Input Sinusoidal Signal Plus Noise b(k)

The influence of the step size on the dynamic performance of the ALE has been studied by using b(k) shown in Figure 5.3. Figures 5.4, 5.5 and 5.6 show the ALE outputs obtained by using different step size values (μ=0.0001, μ=0.001, and μ=0.01), respectively. The filter length was equal to *L*=10.

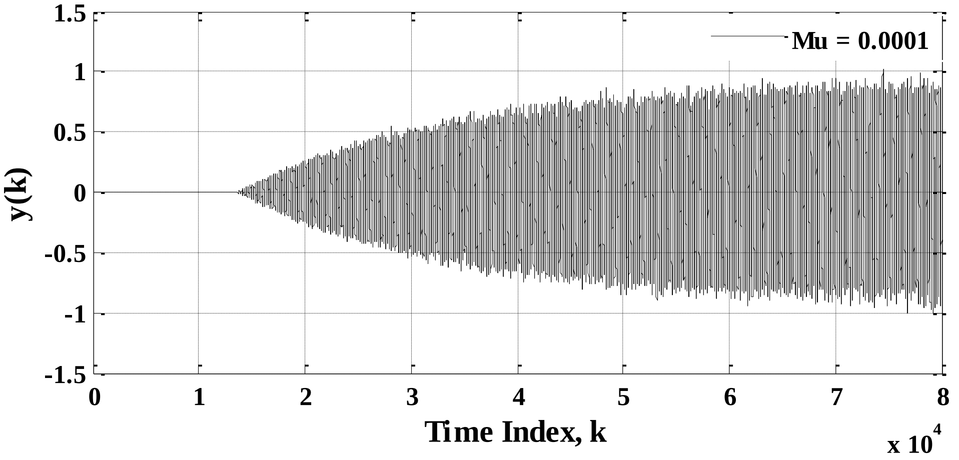


Figure 5.4: The Output of the ALE with μ = 0.0001

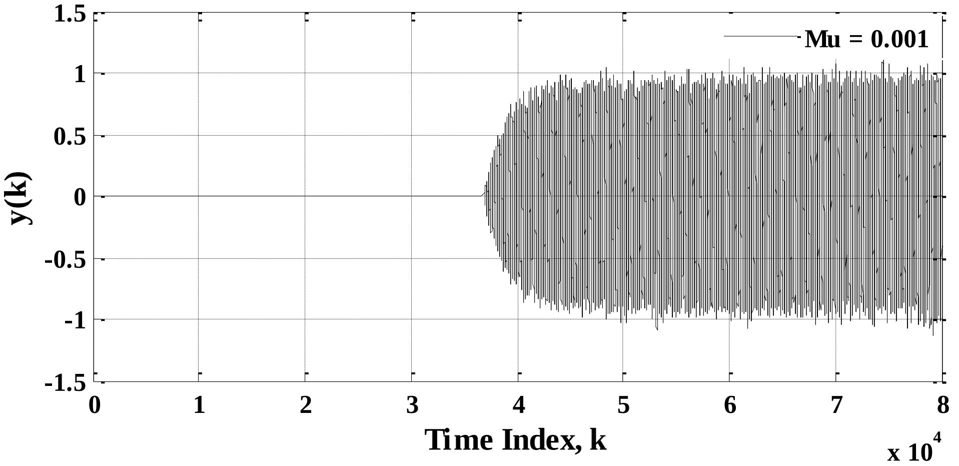


Figure 5.5: The Output of the ALE with μ = 0.001

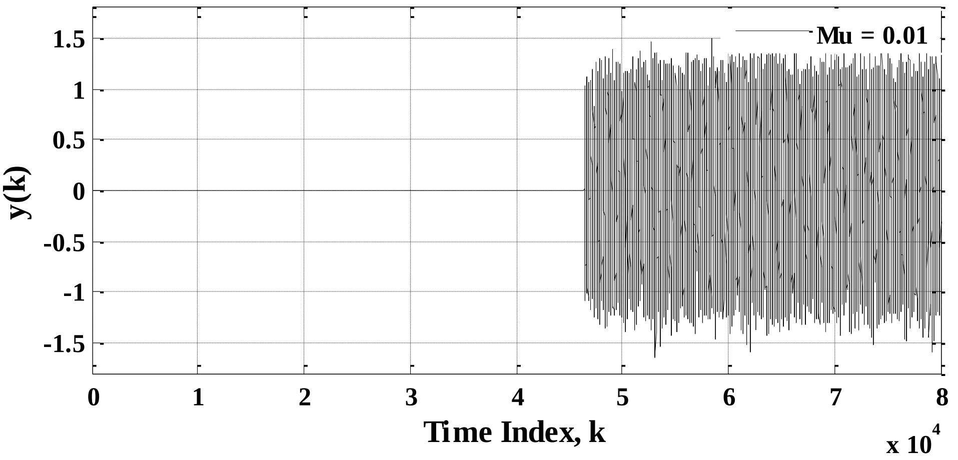


Figure 5.6: The Output of the ALE with μ = 0.01

It is clear from these results that the convergence rate of the LMS algorithm is slow when the step size is small. But, when the step size is increased, the convergence rate increases considerably and y(k) tracks its reference faster.

5.2.1.2 The Magnitude Response of the Filter

Figures 5.7, 5.8, and 5.9 show b(k) for different SNR values equal to -10dB, 0dB, and 10dB, respectively. The input sinusoidal signal s(k) has a frequency of f_0=400Hz. The sampling frequency is assumed to be F_s=8000Hz. The white noise (n(k)) has a zero mean with its variance is equal to one.

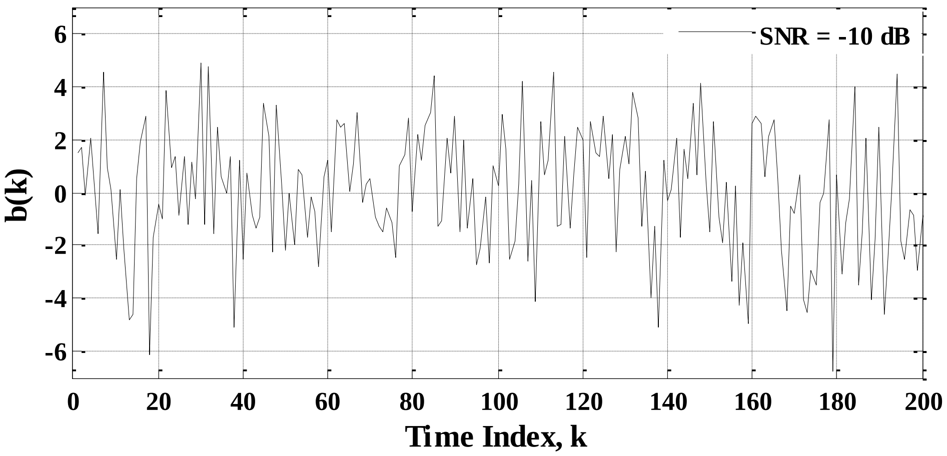


Figure 5.7: Input Signal of ALE with SNR = -10dB

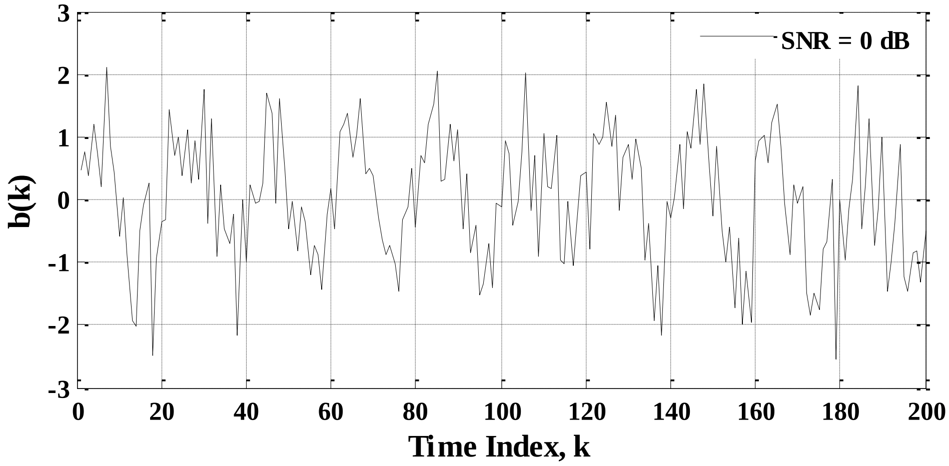


Figure 5.8: Input Signal of ALE with SNR = 0dB

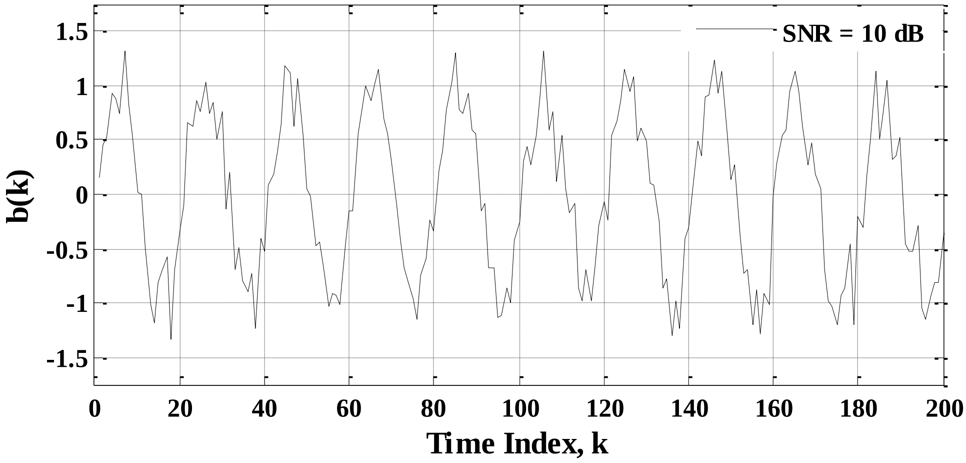


Figure 5.9: Input Signal of ALE with SNR = 10dB

When SNR is -10dB and 0dB, the sinusoidal signal s(k) is discernible in b(k) which creates a challenging condition for the ALE. But, when SNR is 10dB, the sinusoidal signal is clearly visible in b(k). The performance of the ALE for these cases is discussed below.

The magnitude response of the filter can be computed by using the following equation

$$20 \log|\mathrm{H}(\omega_o)|$$

where $\mathrm{H}(\omega_o)$ represents the transfer function of the ALE in the frequency domain.

The magnitude responses of the ALE obtained by different μ values are shown in Figures 5.10, 5.11, and 5.12. The length of FIR filter is *L*=128. The input signal to the ALE is as shown in Figure 5.9. The input signal b(k) was delayed by one sample (D=1).

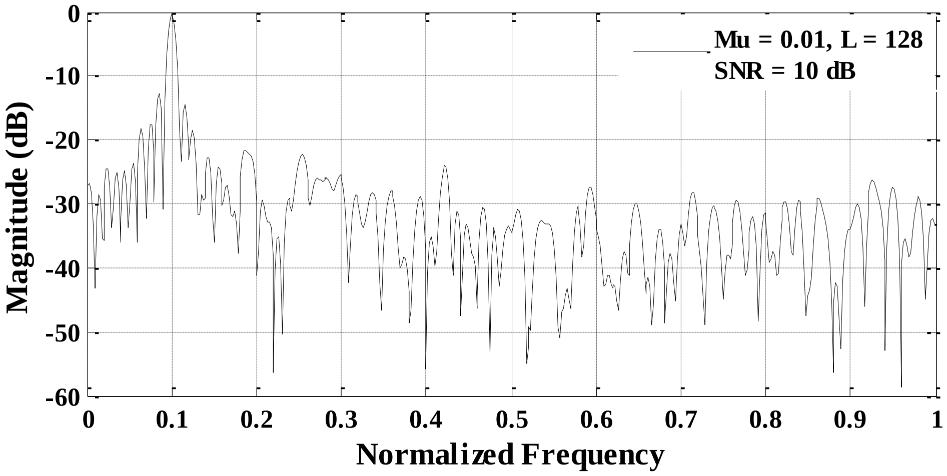


Figure 5.10: Magnitude Response of Filter with μ=0.01

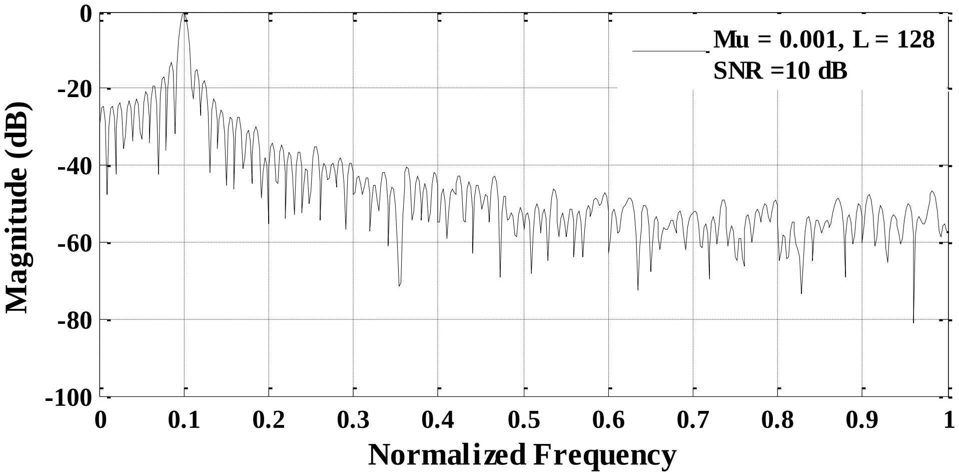


Figure 5.11: Magnitude Response of Filter with μ=0.001

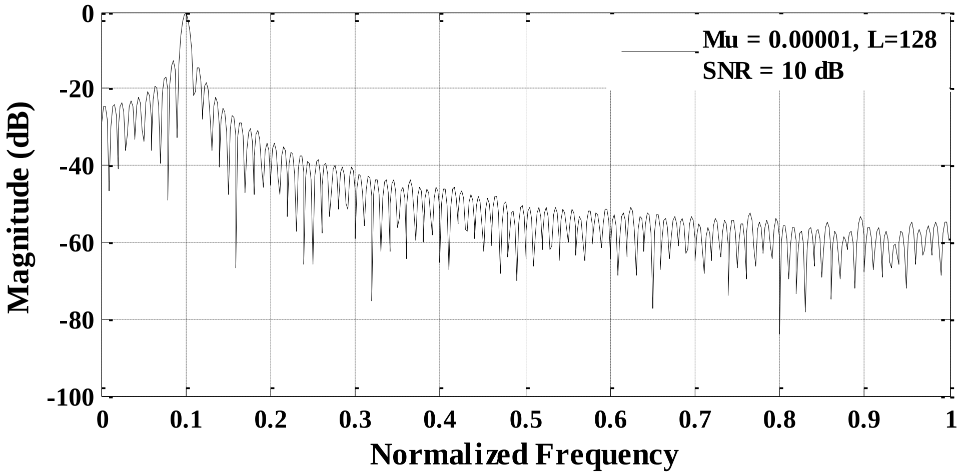


Figure 5.12: Magnitude Response of Filter with μ=0.00001

It can be seen that the ALE generates a peak at a normalized frequency of 0.1 $\left(\frac{f_0}{F_s/2}\right)$ which corresponds to the sinusoidal signal frequency of 400Hz. The ALE operates like a bandpass filter which passes sinusoidal signals with frequency of 400Hz. All other signal components with different frequencies are suppressed (or attenuated) by the filter. It is obvious that when μ is decreased the magnitude response becomes more accurate and the attenuation level increases.

The magnitude responses of the ALE obtained by different L values are shown in Figures 5.13, 5.14, and 5.15. The step size was set to $1 \times 10^{-7}$and the input signal to the ALE is as shown in Figure 5.9.

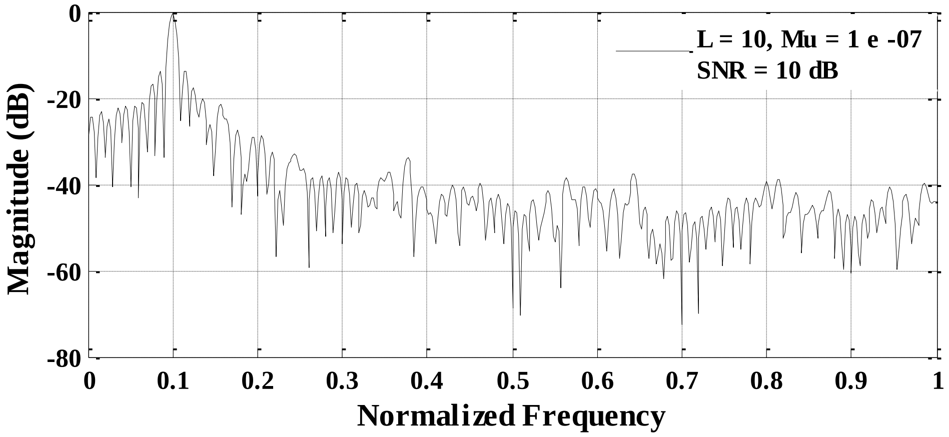


Figure 5.13: Magnitude Response of Filter with *L*= 10

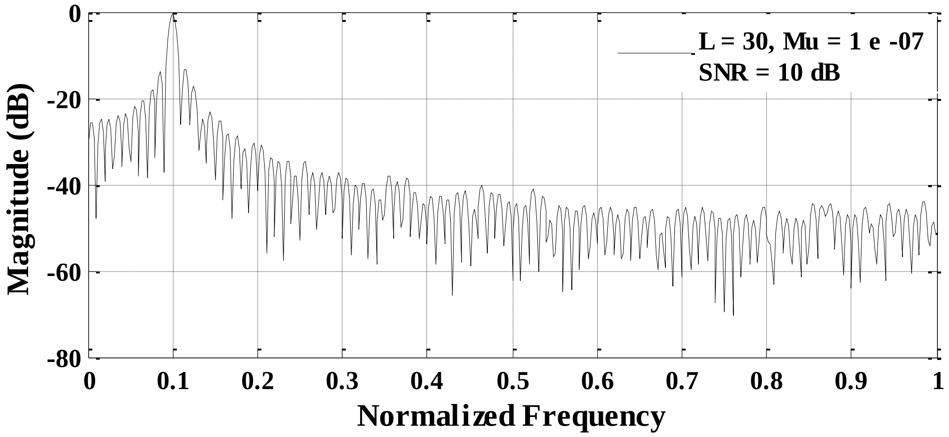


Figure 5.14: Magnitude Response of Filter with *L* = 30

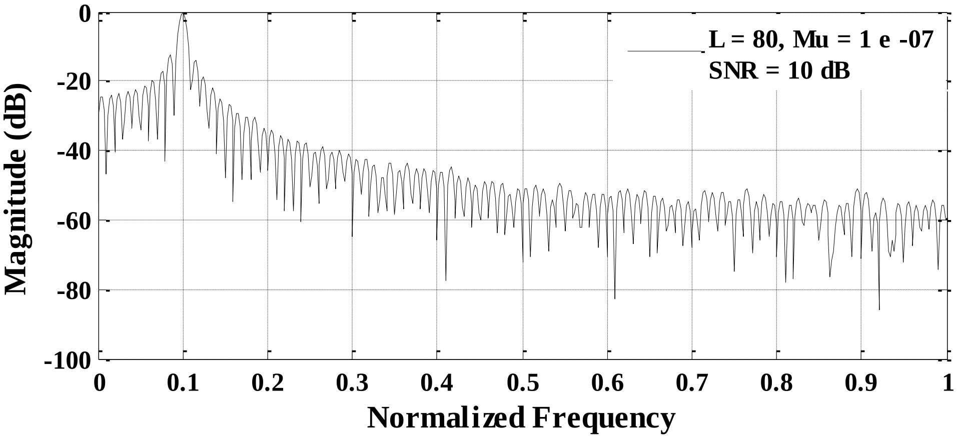


Figure 5.15: Magnitude Response of Filter with *L*= 80

Again, it can be seen that there exists a peak at a normalized frequency of 0.1. It is obvious that when *L* is increased, the magnitude response becomes more accurate and the attenuation level of the filter increases.

The magnitude responses of the ALE when b(k) has different SNR values are shown in Figures 5.16, 5.19, and 5.22. The outputs of the ALE corresponding to these results are shown in Figures 5.17, 5.20, and 5.23. The error signals of the ALE for the same cases are shown in Figures 5.18, 5.21, and 5.24. The step size and the filter length are set to 0.00001 and 128, respectively.

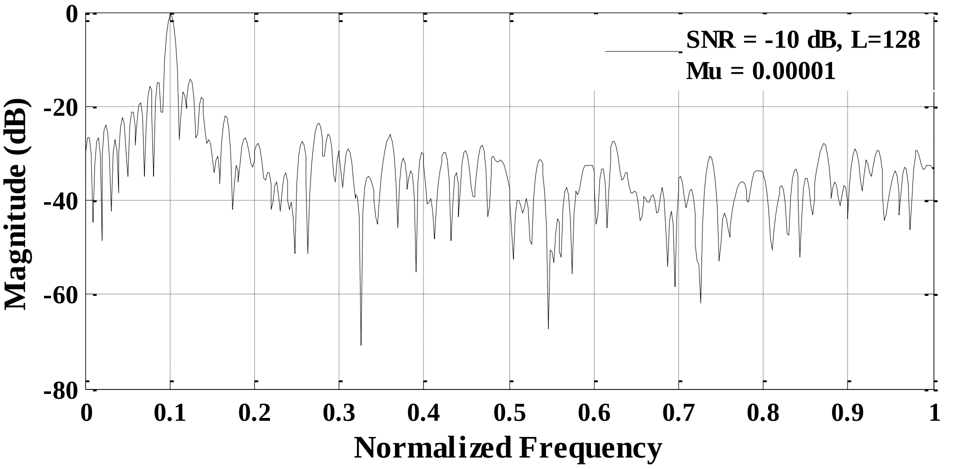


Figure 5.16: Magnitude Response of Filter with SNR = -10dB

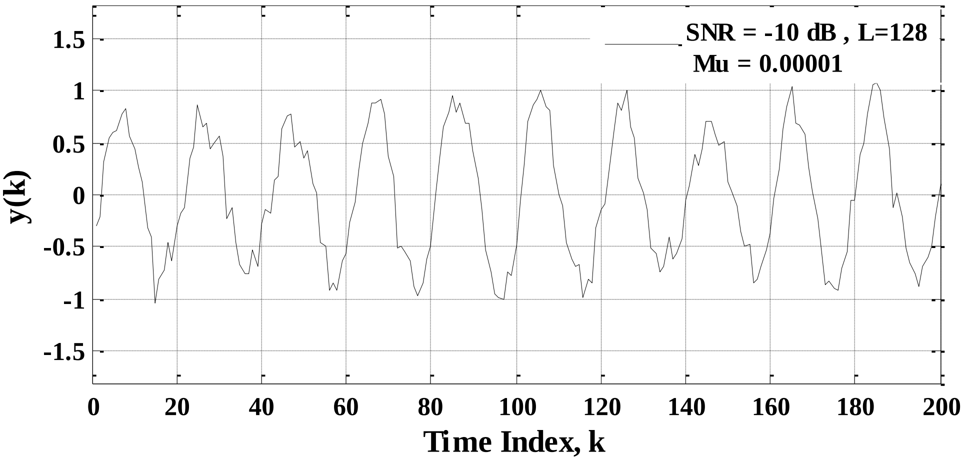


Figure 5.17: The Output of the ALE with SNR = -10dB

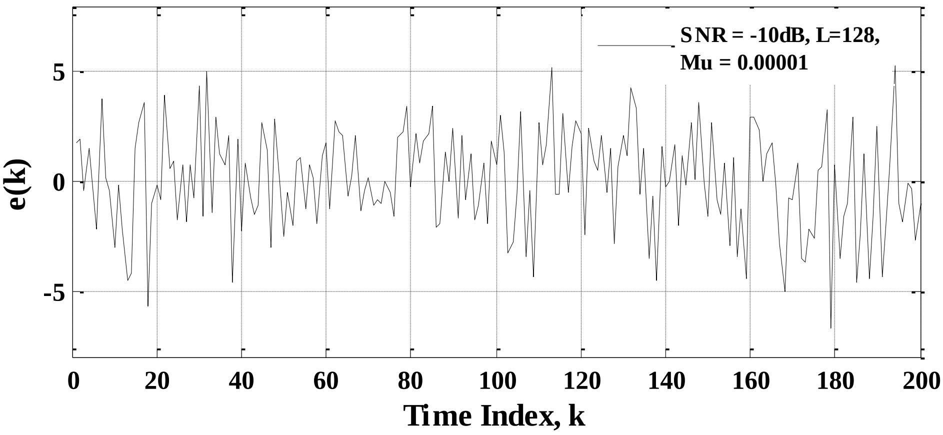


Figure 5.18: The Error of the ALE with SNR = -10dB

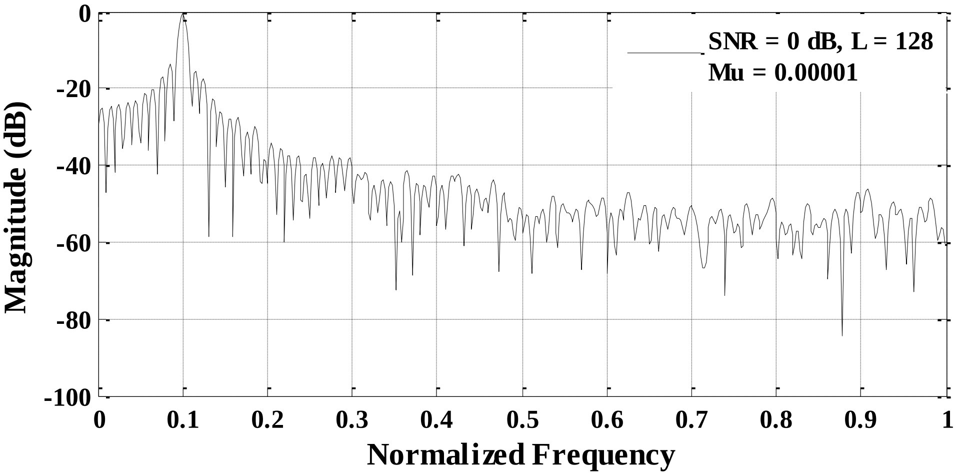


Figure 5.19: Magnitude Response of Filter with SNR = 0dB

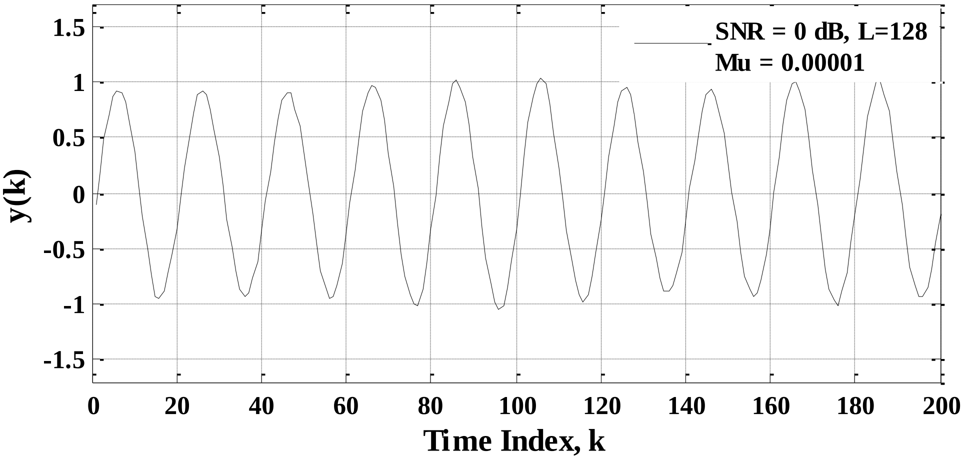


Figure 5.20: The Output of the ALE with SNR = 0dB

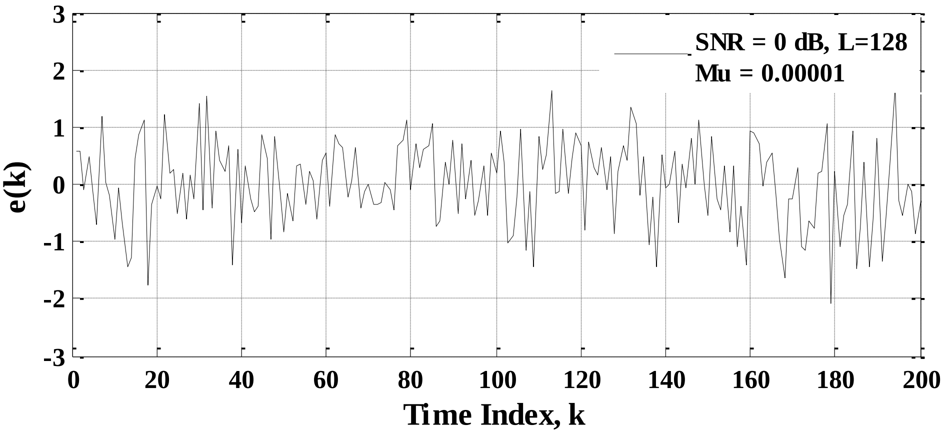


Figure 5.21: The Error of the ALE with SNR = 0dB

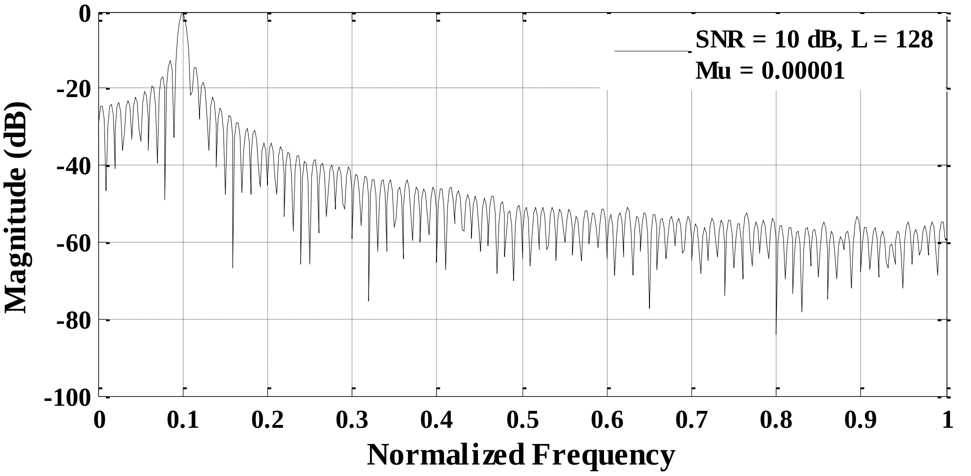


Figure 5.22: Magnitude Response of Filter with SNR = 10 dB

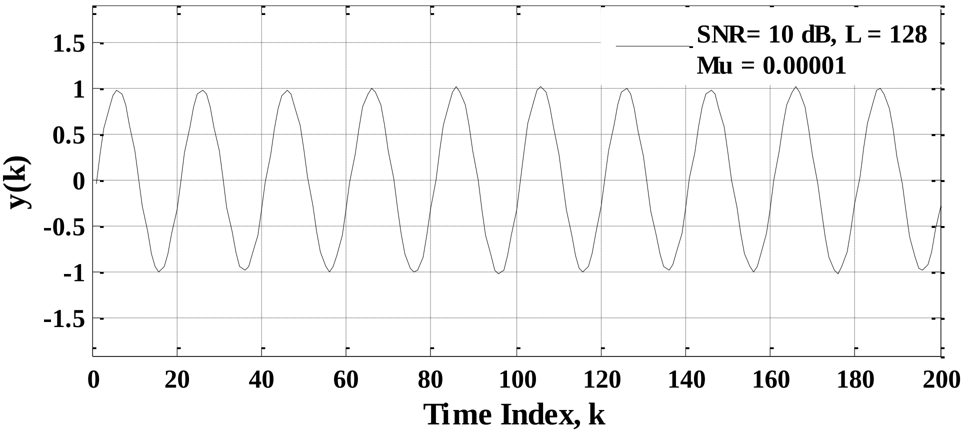


Figure 5.23: The Output of the ALE with SNR = 10dB

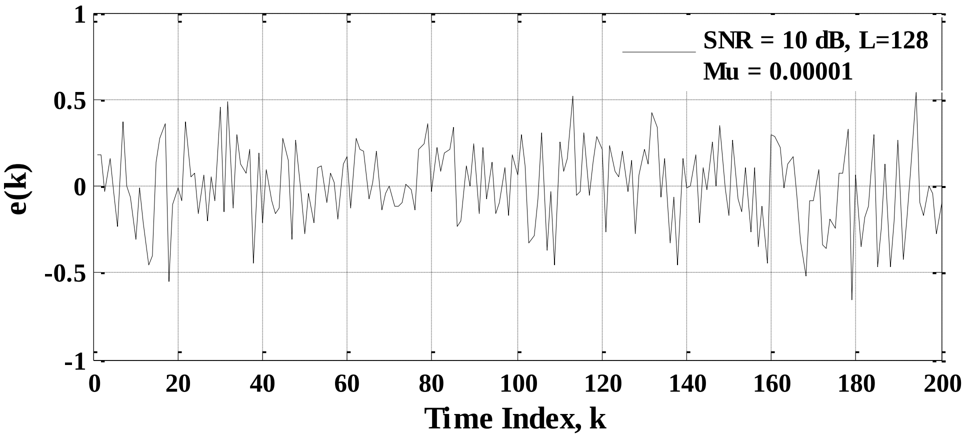


Figure 5.24: The Error of the ALE with SNR = 10dB

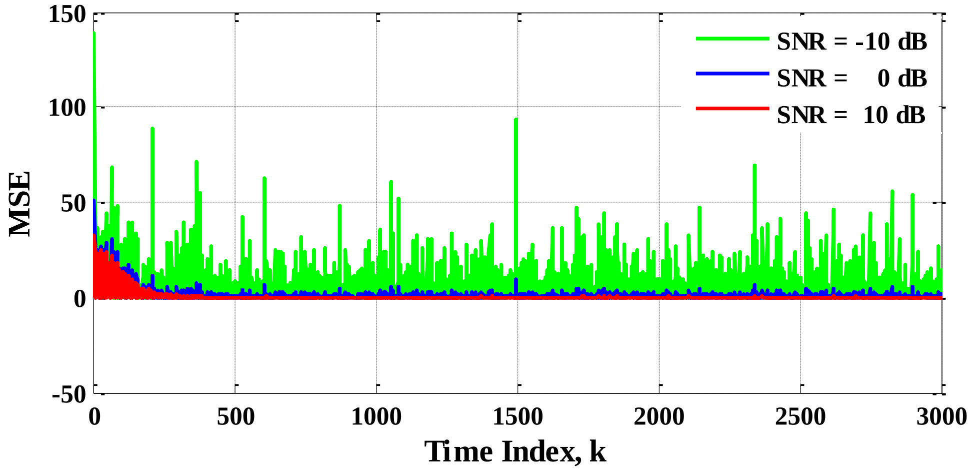


Figure 5.25: Mean Square Error of the ALE with Different SNR values

It can be easily seen from these results that the performance of the ALE is not good for the SNR value of -10dB. The ALE output y(k) is a distorted sinusoidal and the error signal e(k) is not minimized. However, when SNR value is increased (0dB and 10 dB), the performance of the ALE increases considerably which is clearly visible in Figures 5.20, 5.23 and 5.25.

5.2.2 Two Sinusoids

Consider the block diagram of the ALE shown in Figure 5.26 with D=1. The input sinusoidal signal is assumed to be s(k)=sin($2\pi f_0/F_s$)+cos($2\pi f_1/F_s$), where F_s=8000Hz, f_0=800Hz, f_1=2000Hz. The Gaussian white noise, n(k), has a zero mean with variance equal to one.

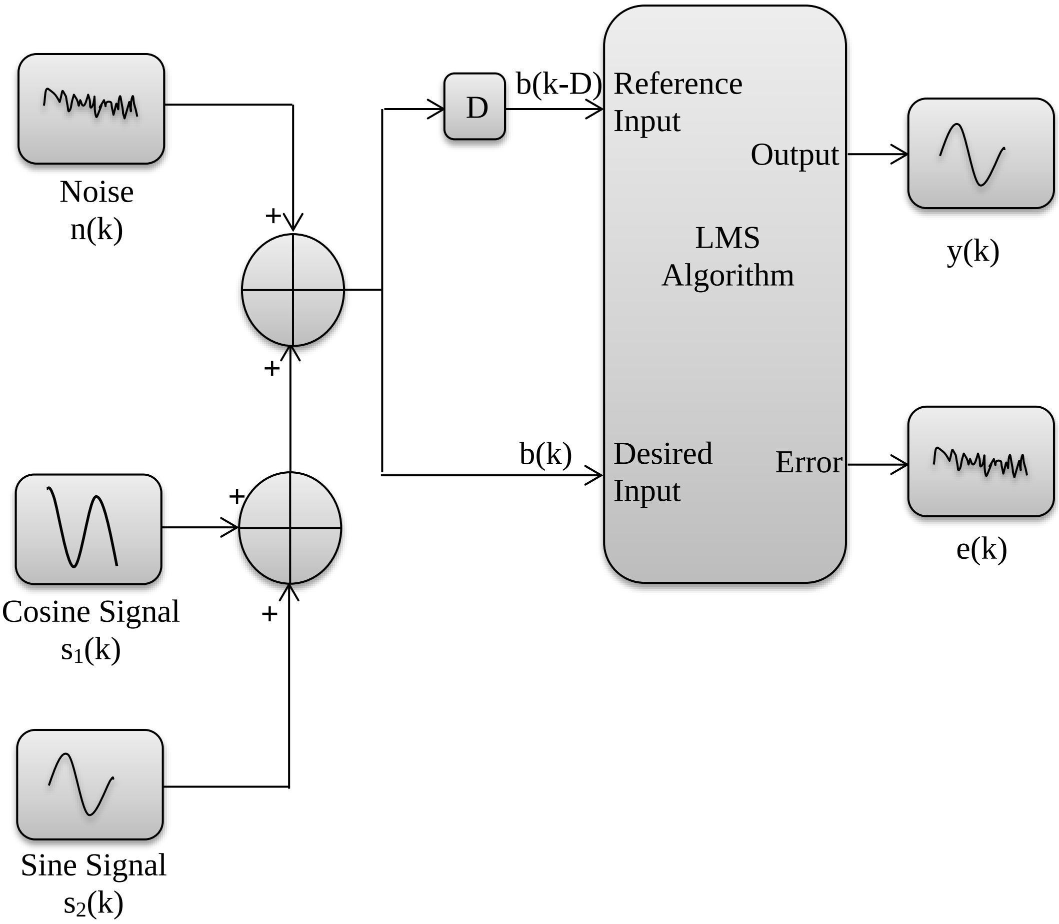


Figure 5.26: ALE Configuration with Two Sinusoidal Input

The magnitude responses of the ALE obtained by different μ values are shown in Figures 5.27, 5.28, and 5.29. The length of FIR filter is L= 128. The SNR of b(k) is 10dB.

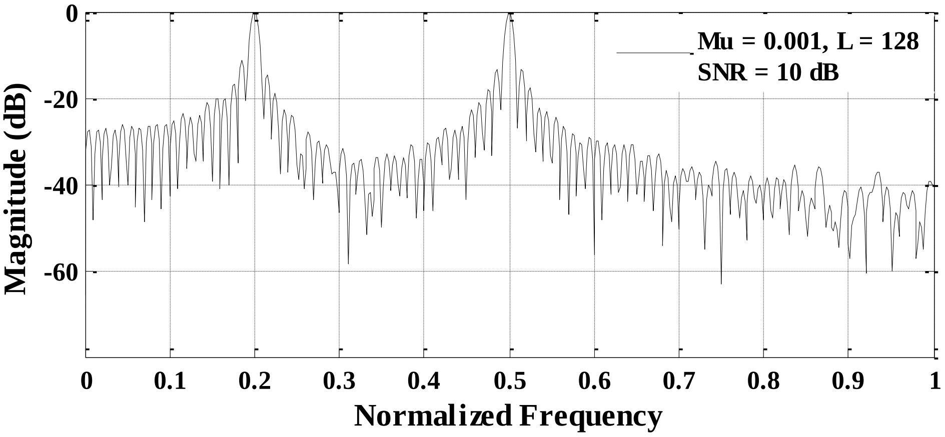


Figure 5.27: Magnitude Response of Filter with µ = 0.001

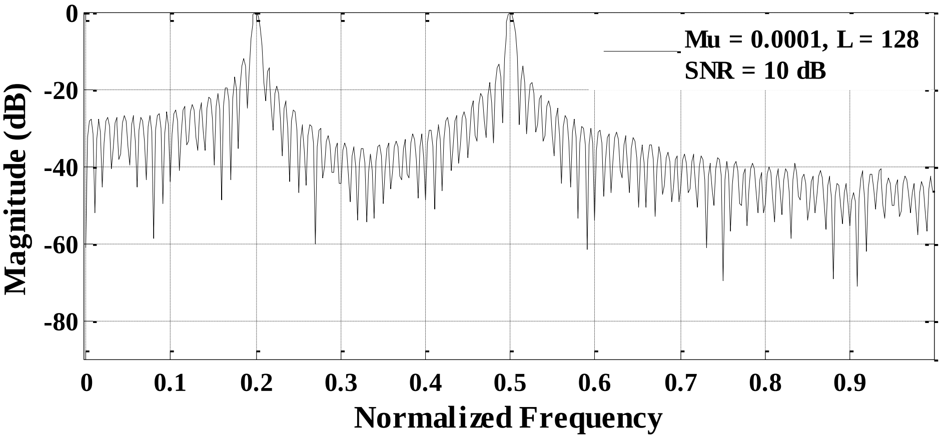


Figure 5.28: Magnitude Response of Filter with µ = 0.0001

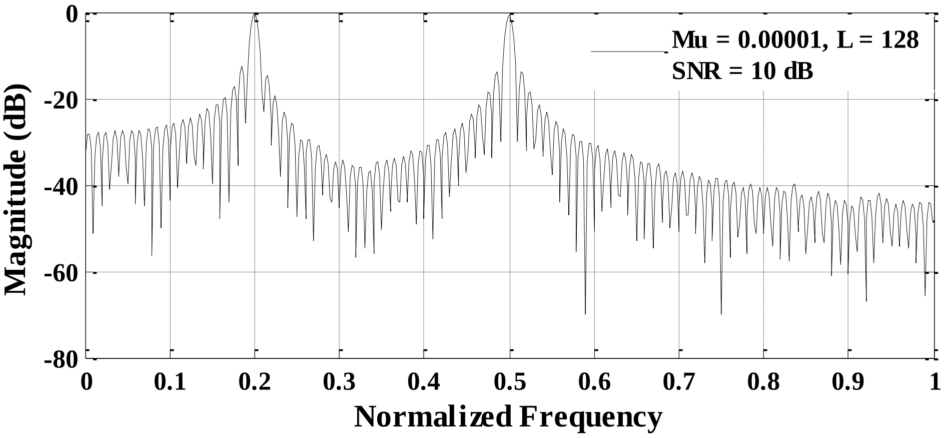


Figure 5.29: Magnitude Response of Filter with µ = 0.00001

It can be seen that the ALE generates two peaks at the normalized frequencies of 0.2 $\left(\frac{f_0}{F_S/2}\right)$ and 0.5 $\left(\frac{f_1}{F_S/2}\right)$ which corresponds to the sinusoidal frequencies of 800Hz and 2000Hz, respectively. It is clear that the ALE operates like a bandpass filter having two bandpass frequencies at 800Hz and 2000Hz. All other signal components with different frequencies are suppressed (or attenuated) by the filter. It is obvious that when μ is decreased the magnitude response becomes more accurate and the attenuation level of the filter becomes smoother.

The magnitude responses of the ALE obtained by different *L* values are shown in Figures 5.30, 5.31, and 5.32. The step size was set to 0.00001. The SNR of b(k) was assumed to be 10dB.

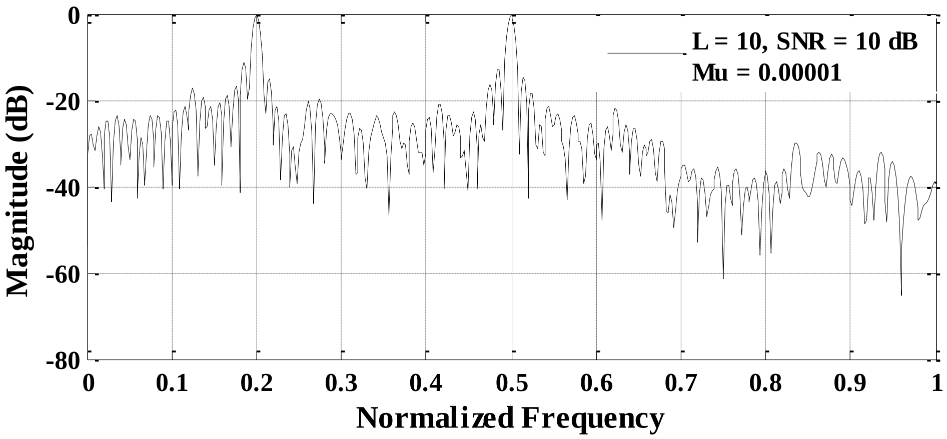


Figure 5.30: Magnitude Response of Filter *L* = 10

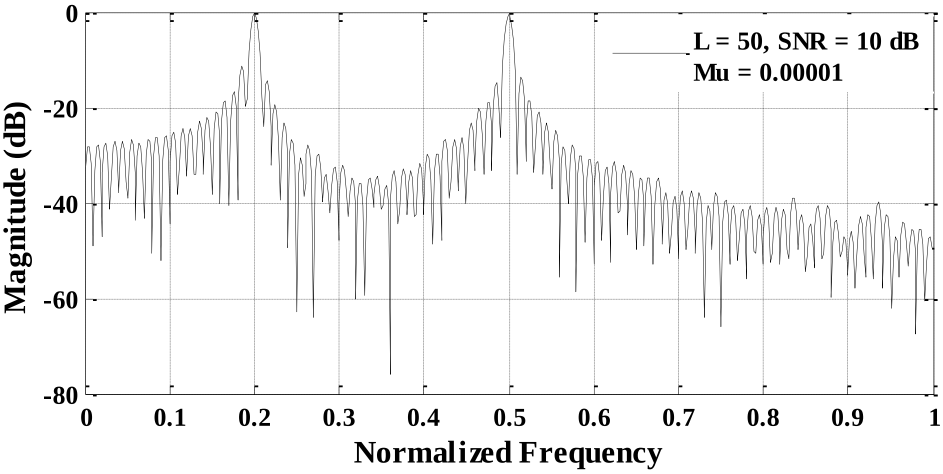


Figure 5.31: Magnitude Response of Filter *L* = 50

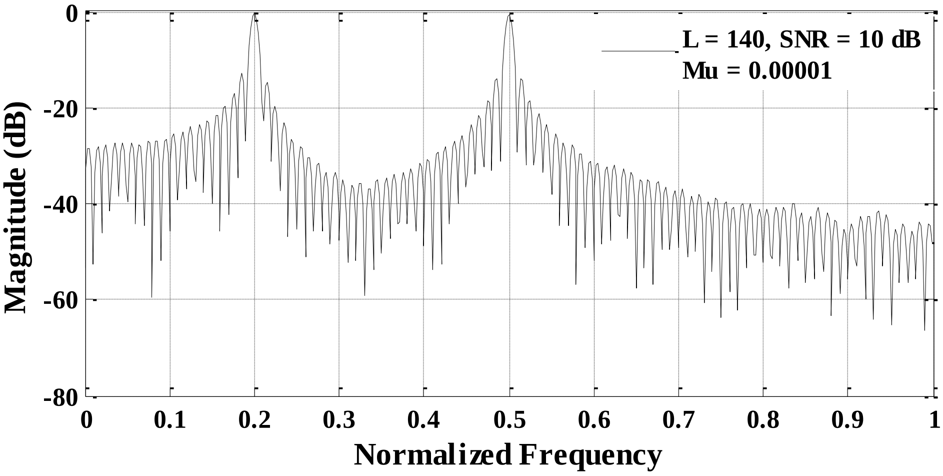


Figure 5.32: Magnitude Response of Filter *L*= 140

Again, it can be seen that there exists two peaks at the normalized frequencies of 0.2 $\left(\frac{f_0}{F_s/2}\right)$ and 0.5 $\left(\frac{f_1}{F_s/2}\right)$. It is obvious that when L is increased, the magnitude response becomes more accurate and the attenuation level of the filter increases.

5.3 Experimental Results

5.3.1 Introduction

The Simulink model of ALE configuration which was studied in section 5.3, has been loaded to the Texas Instruments (TI) TMS320C6416T DSP card. Figure 5.33 illustrates the connections of the ALE on the DSP card with external devices.

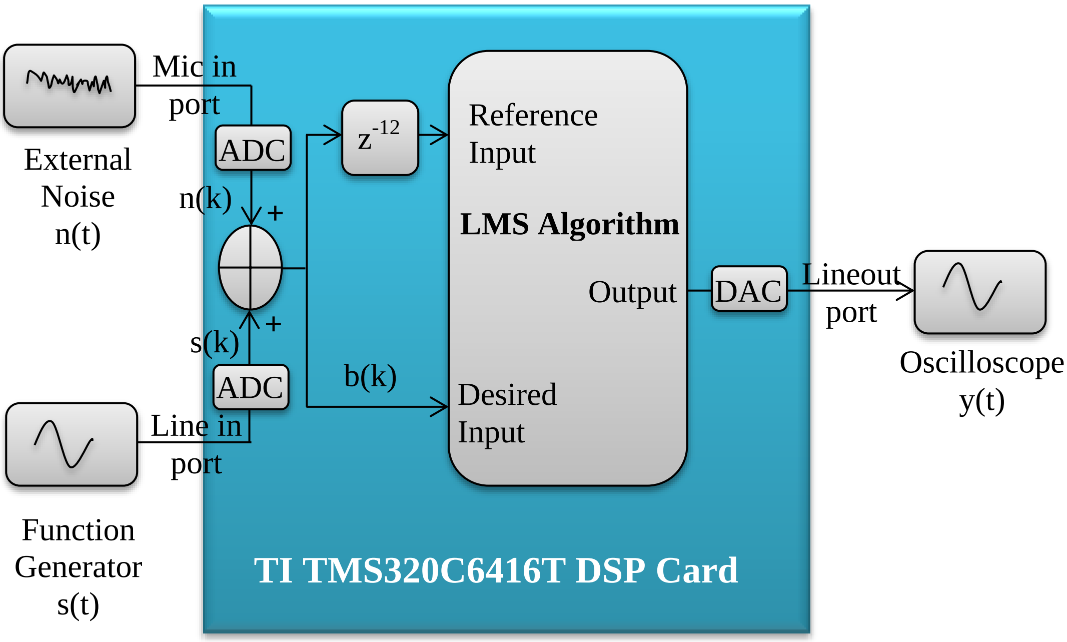


Figure 5.33: The ALE Configuration with TI TMS320C6416T DSP Card

5.3.2 Implementation of ALE in Real-Time

In the experimental study, the ALE configuration shown in Figure 5.33 was used. The sinusoidal signal s(t) generated by the function generator is shown in Figure 5.34. The frequency was equal to 40Hz, and the phase shift was equal to zero. Noise has zero mean with variance equal to one as shown in Figure 5.35. After converting these signals into digital form by using analog to digital converter (ADC), the input signal b(t) shown in Figure 5.36 is formed. It should be noted that b(t) is very similar to the waveform shown in Figure 5.3 obtained by simulation.

The delay (D) is set to 12 and the step size was set to 0.0001. The filter length was 10. The output of the ALE is shown in Figure 5.37. It is obvious that y(t) is very similar to s(t) which verifies the fact that ALE output is the estimation of s(k).

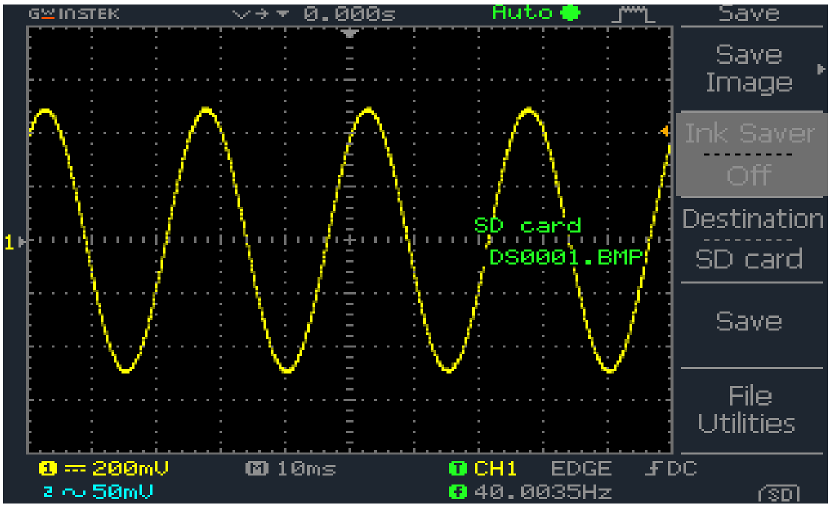


Figure 5.34: The Input Sinusoidal Signal s(t)

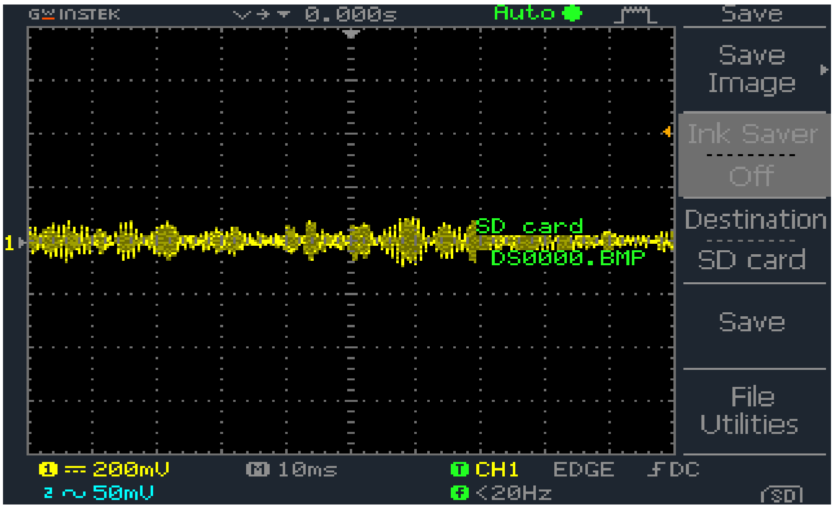


Figure 5.35: The Noise Signal n(t)

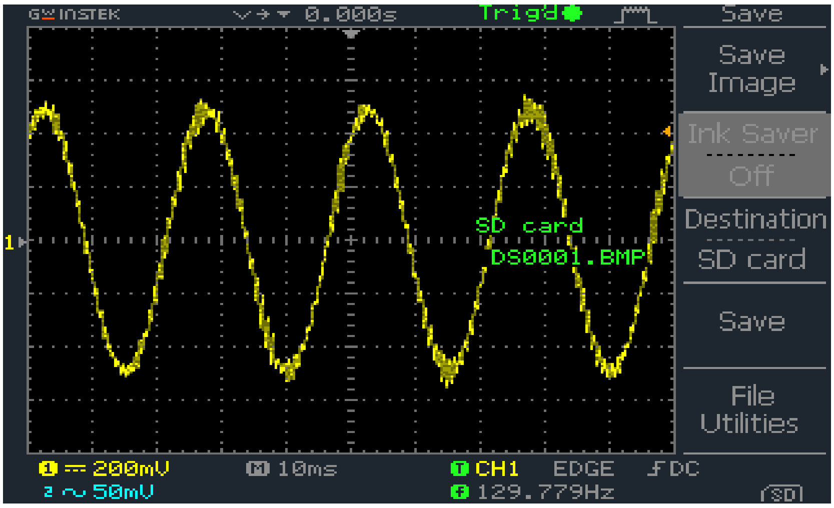


Figure 5.36: The Input Signal of ALE b(t)

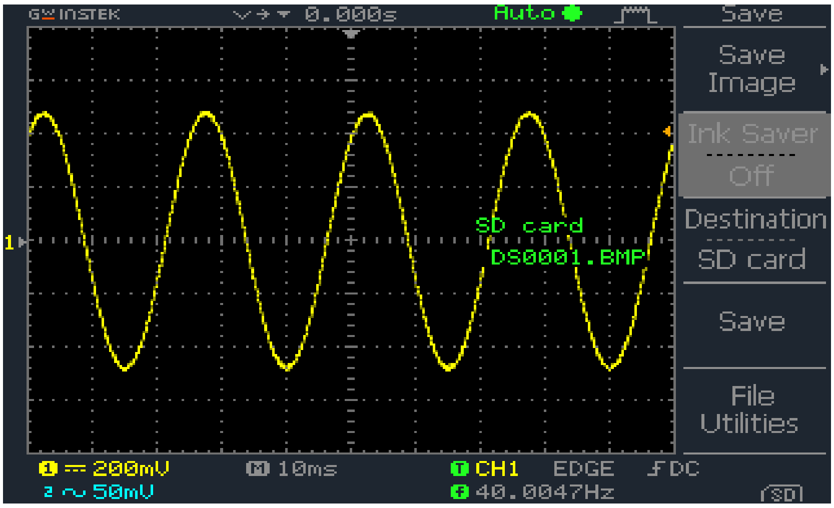
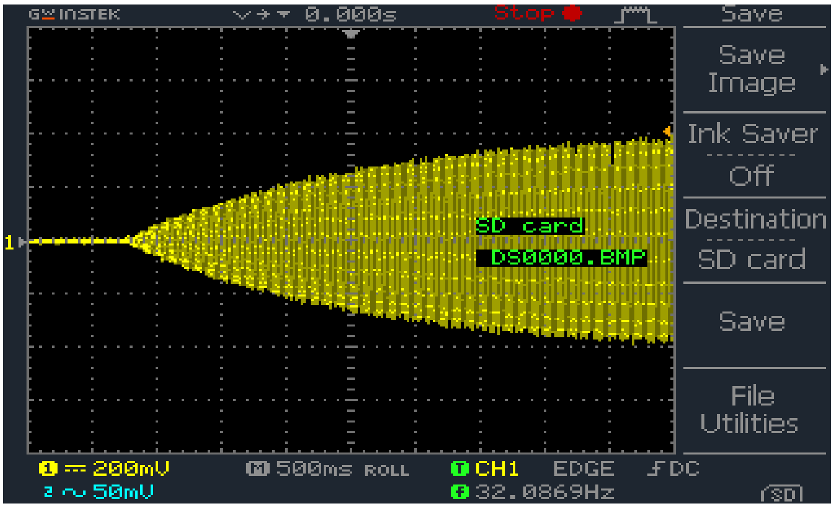


Figure 5.37: The Output of ALE with μ = 0.0001

5.3.3 The Influence of the Step Size

In this part, the influence of step size on the convergence rate is studied in real time. The input and noise signals (s(t) and n(t)) are exactly the same that were mentioned in simulation section.

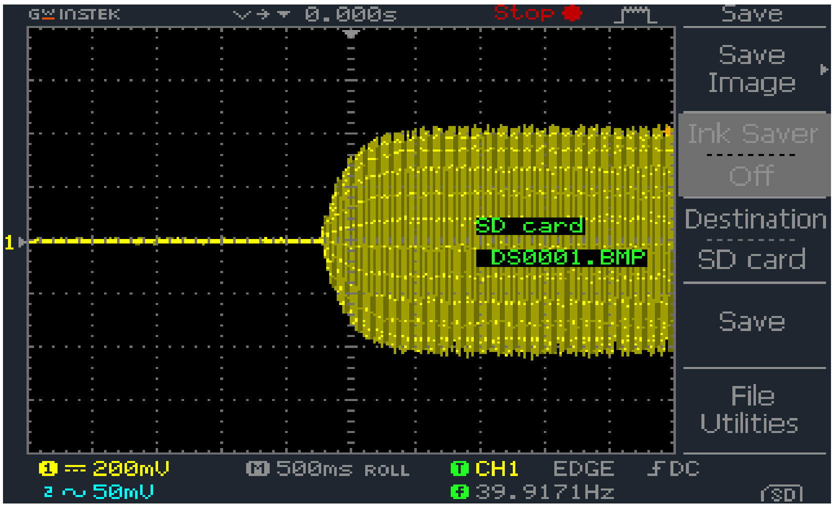
The experimental results that correspond to the simulation results (Figures 5.4, 5.5, and 5.6) are shown in Figures 5.38, 5.39, and 5.40. It is clear that the experimental results are in good agreement with the simulation results. The dynamic performance of ALE increases considerably when the step size (µ) is increased. When the step size is large, there exists an estimation error and y(k) becomes distorted. This case is shown in Figures 5.41, 5.42, and 5.43.



Ink Saver Off

Destination SD card

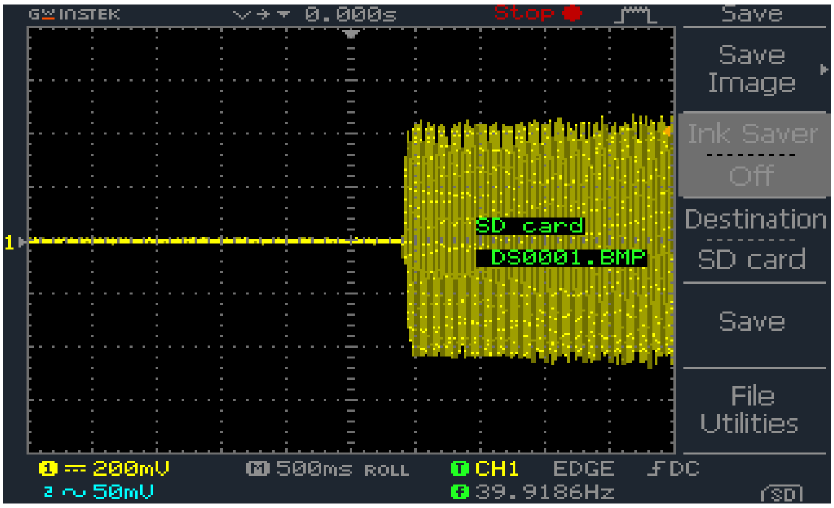
Figure 5.38: The Output of ALE with µ = 0.0001



Ink Saver Off

Destination SD card

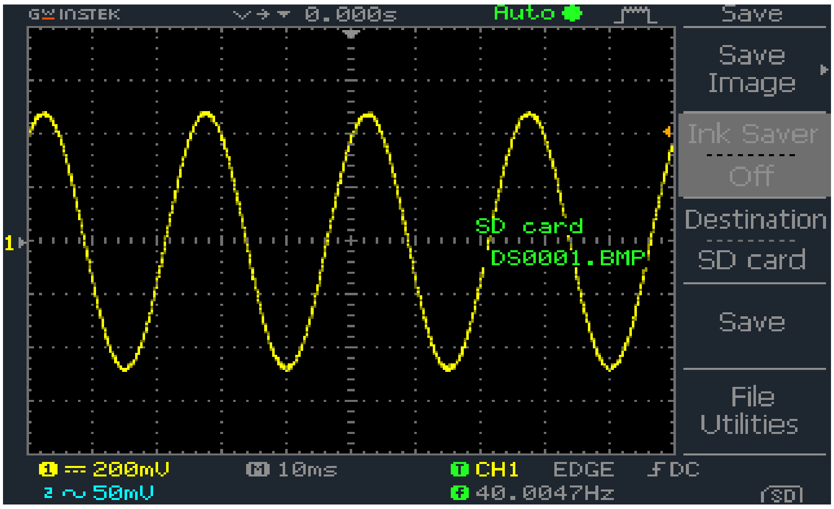
Figure 5.39: The Output of ALE with µ = 0.001



1 200mV 500ms ROLL CH1 EDGE DC
2 50mV 39.9186Hz

Figure 5.40: The Output of ALE with μ = 0.01

Obviously, the error involved in the ALE is negligibly small when the step size is very small as shown in Figure 5.41. But, when the step size is increased the convergence rate increases resulting in a faster convergence rate. However, in this case, the estimation error will be larger as shown in Figures 5.42, and 5.43. Therefore there is a trade-off between a faster convergence and lower estimation error.



1 200mV 10ms CH1 EDGE DC
2 50mV 40.0047Hz

Figure 5.41: The Output of ALE after Convergence with μ = 0.0001

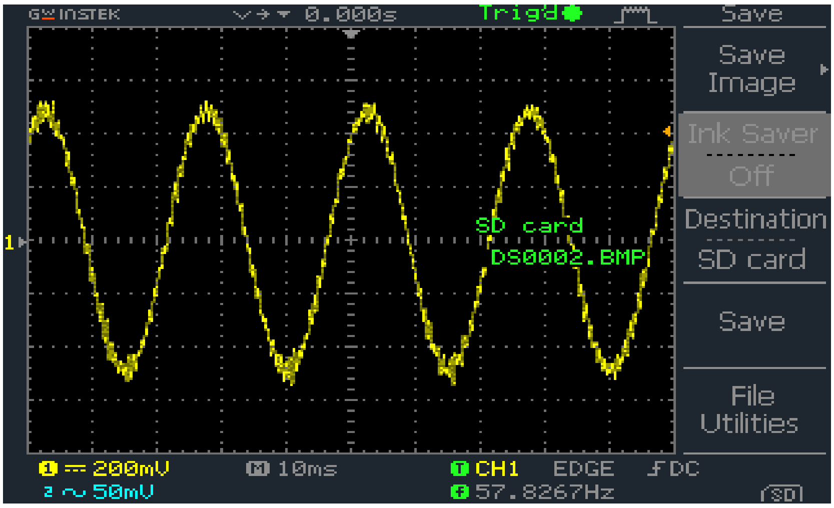


Figure 5.42: The Output of ALE after Convergence with µ= 0.001

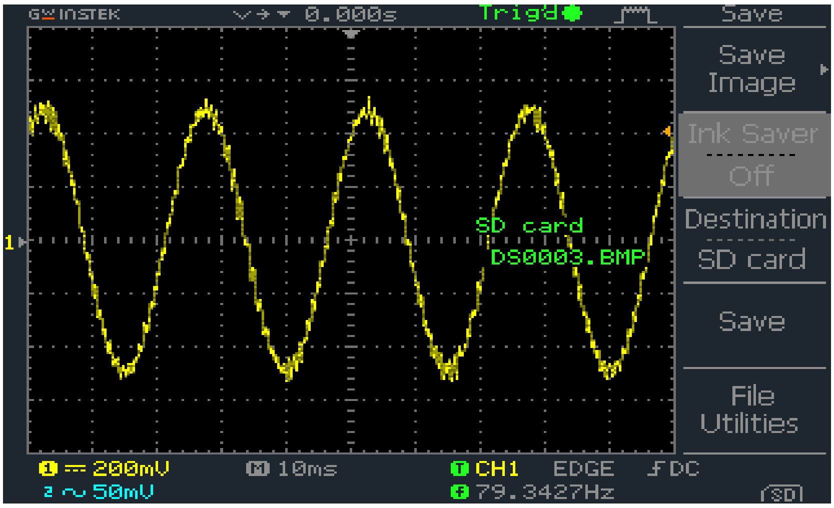


Figure 5.43: The Output of ALE after Convergence with µ= 0.01

Chapter 6

CONCLUSIONS

In this thesis, the noise cancellation problem has been studied by using an adaptive line enhancer (ALE). The ALE configuration has been simulated by using Simulink of Matlab having different parameters both in the input signal and in the ALE. The theoretical considerations have been verified experimentally by using a TI TMS320C6416T DSP card in real time.

The performance of the ALE under different parameters such as the step size, filter length, and SNR has been studied extensively by simulations and experiments when the input signal has one and two sinusoidal signal components together with a noise. The simulation and experimental results showed that whenever the value of the step size is small it leads to a good estimation of the sinusoidal signal, but the convergence speed slows down. On the contrary, whenever the value of the step size is large, the output of ALE (y(k)) becomes distorted, and the convergence speed increases considerably.

Again, the simulation and experimental results showed that when the step size and SNR are kept constant, the ALE output becomes more accurate when more coefficients are used in the filter. But, more coefficients increase the cost of the filter, and therefore, it should be selected carefully. For instance, the biggest filter length may result in a large estimation error.

Finally, the simulation and experimental results showed that when the step size and filter length are kept constant, the ALE output becomes more accurate when SNR has large values. For instance, when SNR value is -10dB, the ALE cannot estimate the sinusoidal signal and suppress the noise at its output. However, when SNR value is increased the estimation becomes more successful resulting in a small error at the ALE output.

REFERENCES

Asaad, S. F. (2008, August). White and Color Noise Cancellation using Adaptive Feedback Cross-Coupled Line Enhancer Filter. *IEEE International Symposium on Signal Processing and Information Technology, 2008. ISSPIT 2008.* , 96 - 99.

Boroujeny, B. F. (1998). *Adaptive Filters: Theory and Applications.* Chichester, New York, Weinheim, Brisbane, Singapore, Toronto: JohnWiley & Sons Ltd, BaffinsLane, Chichester, West Sussex PO19 lUD, England.

Bose, T. (2003). *Digital Signal and Image Processing.* United States of America: Wiley.

Dhull, S. K., Arya, S. K., & Sahu, O. P. (2011, May). Performance Comparison of Adaptive Algorithms for Adaptive line Enhancer. *IJCSI, 8*(3), 533.

Fenglin, W., & Chris, M. K. (2004). Frequency Properties of an Adaptive Line Enhancer. *Mechanical Systems and Signal Processing*, 797-812.

Geboren, t. U. (1992). *Adaptive Filtering Methods: On Methods to use a Priori Information in Order to Reduce Complexity While Maintaining Convergence Properties.*

Ghogho, M. (1998, Sep.). Analytic Behavior of the LMS Adaqptive Line Enhancer for Sinusoids Corrupted by Multiplicative and Additive Noise. *IEEE Transactions on Signal Processing, 46,* 2386-2393.

Griffiths, L. (1975). Rapid Measurement of Digital Instantaneous Frequency. *IEEE Transactions on Acoustics, Speech and Signal Processing, 23*(2), 207-222.

Haykin, S. (2002). *Adaptive Filter Theory* . (4th b.). Prentice Hall,Upper Saddle River, NJ.

Ifeachor, E. C., & Jervis, B. W. (2002). *Digital Signal Processing, A Practical Approach* (Second Edition b.). United States of America: Prentice Hall.

Ikuma, T., & (Louis) Beex, A. A. (2008, OCT.). Improved Mean-Square Error Estimate for the LMS Transversal Equalizer With Narrowband Interference. *IEEE Transactions on Signal Processing, 56,* 5273-5277.

Kalman, R. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transaction of the ASME - Journal of Basic Engineering*, 35-45.

Kaur, H. (2011, June). Applications and Simulation of Adaptive Filter in Noise Cancellation. *International Journal of Advanced Engineering*, 21-24.

M.Kuo, S., & Lee, B. H. (2001). *Real-Time Digital Signal Processing.*

Miyagi, S., & Sakai, H. (2005, NOVEMBER). Mean-Square Performance of the Filtered-Reference/ Filtered-Error LMS Algorithm. *IEEE Transactions on Circuits and Systems, 52,* 2454-2463.

Nehorai, A. a. (1980). On the Stability and Performance of the Adaptive Line Enhancer on Acoustics, Speech and Signal Processing. *Proc. IEEE International Conference, 5,* s. 478-481. Denver, CO, USA.

Reddy, V., Egardt, B., & Kailath, T. (1987). Optimized Lattice-form Adaptive Line Enhancer for a Sinusoidal Signal in Broadband Noise. *IEEE Transactions on Acoust Speech and Signal Processing, 35,* 1015 -1023.

Saeid, S., Tracey, L. K., & Abolghasemi, V. (2012, FEBRUARY). A New Adaptive Line Enhancer Based on Singular Spectrum Analysis. *IEEE Transactions on Biomedical Engineering, 59*, 428-434.

Sayed, A. H. (2008). *Adaptive Filters.* John Wiley & Sons, NJ.

Swami, A. (1994, Apr.). Multiplicative Noise Models: Parameter Estimation using Cumulants. *Signal Processing, 36*(3), 355-373.

Symons, F. (1978). Narrowband Interference Rejection using the Complex Linear Prediction Filter. *IEEE Transactions on Acoustics Speech and Signal Processing*, 94-98.

Treichler, J. (1980). Response of the Adaptive Line Enhancer to Chirped and Doppler-Shifted Sinusoids. *IEEE Transactions on Acoustics Speech and Signal Processing, ASSP-28*, 343-348.

Vinay, I. K., & Jone, P. G. (1996). *Digital Signal Processing using Matlab* (2nd b.). USA: Prentice-Hall. Inc.

Widrow, B., Glover, J. R., Mccool, J. M., Kaunitz, J., Williams, C. S., Hean, R. H., et al. (1975, Dec.). Adaptive Noise Cancelling: Principles and Applications. *Proceedings of the IEEE, 63*(12), 1692-1716.

Zeidler, J., Satorius, E., Chabries, D., & Wexler, H. (1978). Adaptive Enhancement of Multiple Sinusoids in Uncorrelated Noise. *IEEE Transactions on Acoustics Speech and Signal Processing, 26*(3), 240-254.

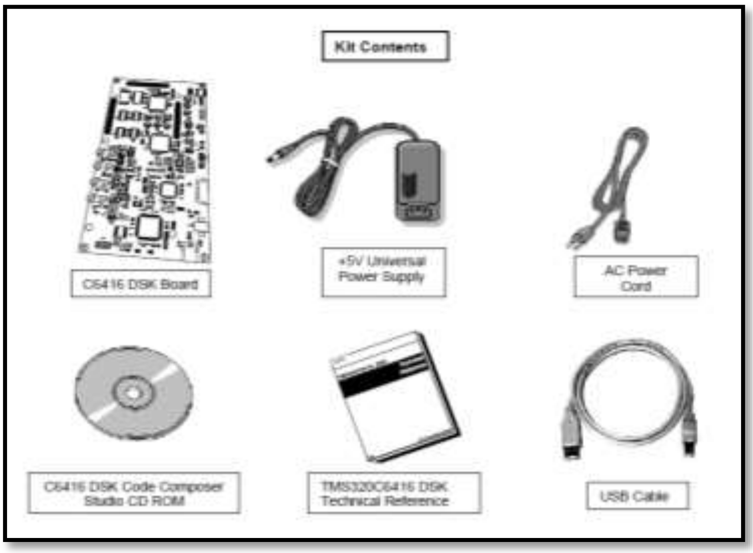
APPENDIX

Appendix A: Texas Instruments

A.1 Description

The 6416 DSP Starter Kit (DSK) is an all-in-one evaluation platform for the TMS320C6416T Digital Signal Processor from Texas Instruments. It includes a target board that can be used as a reference design for interfacing the DSP to common devices such as SDRAM, Flash and a codec as well as a special introductory version of TI's flagship Code Composer Studio development tools. An on-board JTAG emulator allows debug from Code Composer Studio through your PC's USB port.

This kit contains everything you need to get started with TI DSPs. It can also be used with the full version of Code Composer Studio and an external JTAG emulator.


Kit Contents
C6416 DSK Board
+5V Universal Power Supply
AC Power Cord
C6416 DSK Code Composer Studio CD ROM
TMS320C6416 DSK Technical Reference
USB Cable

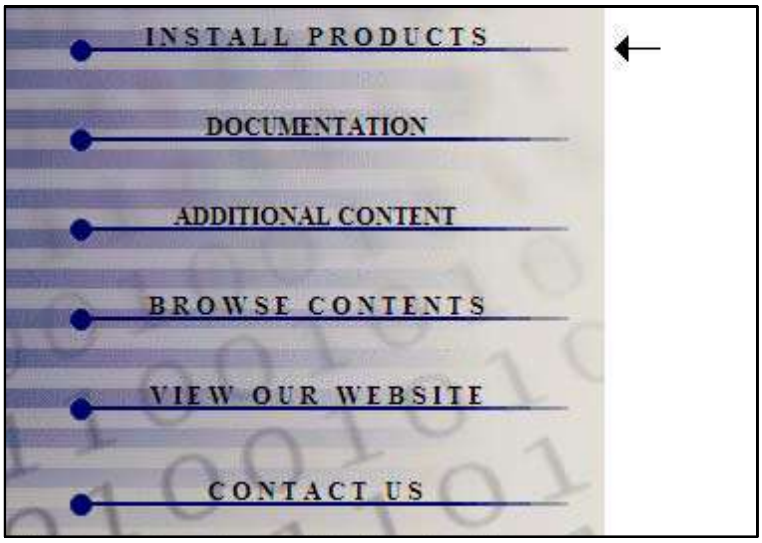
A.2 System Requirements

- 500MB of free hard disk space
- Microsoft Windows™ 2000/XP
- 128MB of RAM
- 16-bit color display
- CD-ROM Drive

A.3 Install DSK Content from the CD-ROM

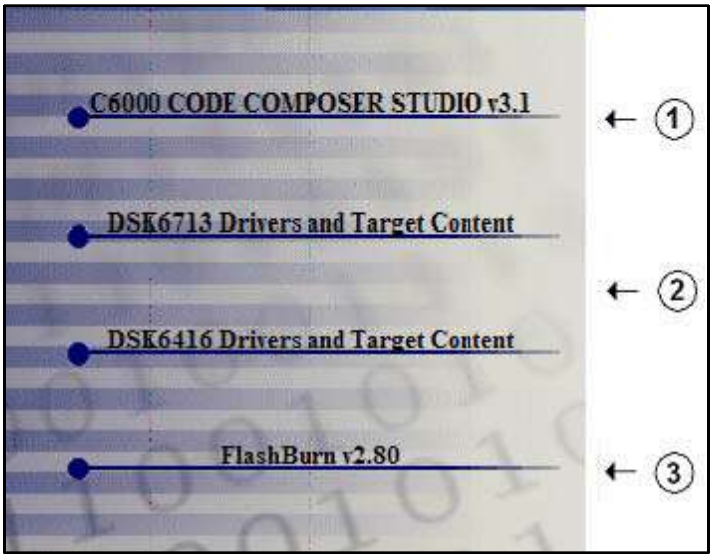
Before you install the DSK software, please make sure you are using **Administrator privileges** and any **virus checking** software is turned off. The DSK board should **not be plugged in** at this point.

1. Insert the Code Composer Studio installation CD into the CD-ROM Drive. An install menu (see below) should appear. If it does not, manually run Launch.exe from the CD-ROM. Select the **Install Products** option from the menu.



2. Install any components you need. To debug with the DSK you must have 1) a copy of **Code Composer Studio**, 2) the **target content** package for your board and 3) a copy of the **FlashBurn** plug-in. Users of the full Code Composer Studio

package can skip the DSK Code Composer installation and simply install the target content packages.


C6000 CODE COMPOSER STUDIO v3.1
①
DSK6713 Drivers and Target Content
DSK6416 Drivers and Target Content
②
FlashBurn v2.80
③

3. The installation procedure will create two icons on your desktop:

6416 DSK CCStudio v3.1

6416 DSK Diagnostics Utility v3.1

A.4 Connect the DSK to Your PC

1. Connect the supplied USB cable to your PC or laptop. We recommend that anyone making hardware modifications connect through a USB hub for safety.

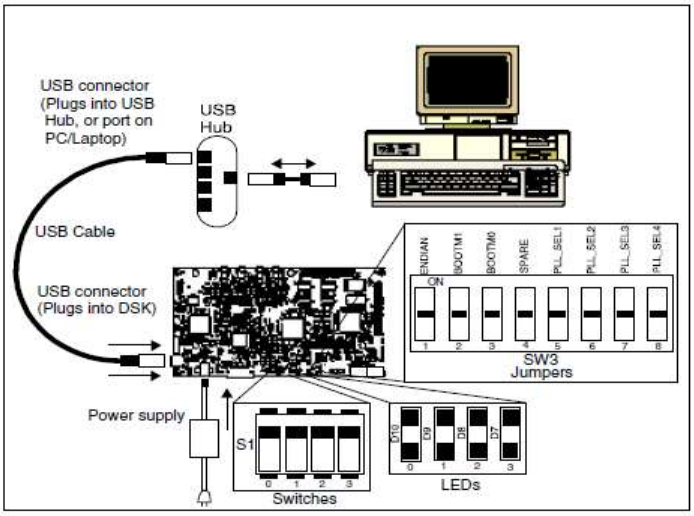
2. If you plan to connect a microphone, speaker, or expansion card these must be plugged in properly before you connect power to the DSK board.

3. Connect the included 5V power adapter brick to your AC power source using the AC power cord.

4. Apply power to the DSK by connecting the power brick to the 5V input on the DSK.

5. When power is applied to the board the Power On Self Test (POST) will run. LEDs 0-3 will flash. When the POST is complete all LEDs blink on and off then

stay on. At this point your DSK is functional and you can now finish the USB driver install.



6. Make sure your DSK CD-ROM is installed in your CD-ROM drive. Now connect the DSK to your PC using the included USB. After few seconds Windows will launch its "Add New Hardware Wizard" and prompt for the location of the DSK drivers.

7. Follow the instructions on the screens and let Windows find the USB driver files “dsk6416.inf” and “sdusb2em.sys” on the DSK CD-ROM. On XP systems Windows will find the drivers automatically.

A.5 Testing Your Connection

If you want to test your DSK and USB connection you can launch the C6416 DSK Diagnostic Utility from the icon on your desktop.

6416 DSK Diagnostics Utility v3.1

From the diagnostic utility, press the start button to run the diagnostics. In approximately 30 seconds all the on-screen test indicators should turn green.

A.6 Starting Code Composer

To start Code Composer Studio, double click the 6416 DSK CCStudio icon on your desktop.

A.7 Running the DSK Tutorial

The on-line help included with the DSK contains in-depth information about the hardware and software that comes with the kit. It also contains a tutorial that will help you get started with your DSK and learn about its features. To access the on-line help and run the tutorial, follow these steps:

1. Start Code Composer Studio (ignore this if CCS is already running) by double-clicking on the C6416 DSK icon on your desktop.

2. Use the **Debug --> Connect** menu option to open a debug connection to the DSK board.

3. Launch the DSK help file by opening the following file using Windows Explorer.

   C:\CCStudio_v31.\docs\hlp\c6416dsk.hlp

4. Look in the section entitled "**Welcome to Your 'C6416 DSK**". You will find the tutorial and other introductory material there. Note that older versions of CCStudio use c:\ti as the default install path instead of c:\CCStudio_v31. so you may have to adjust the path names to match the actual location of files on your PC.

A.8 Related Documentation

A complete list of related documentation is available in the on-line help. The path to this documentation is (assuming you choose c:\CCStudio_v3.1 as the default path)

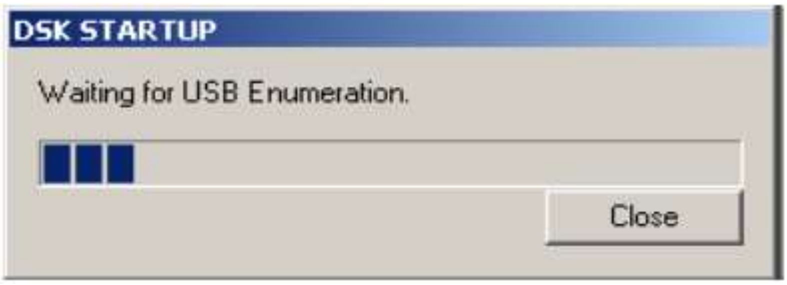
c:\CCStudio_v3.1\docs\pdf\release_ccs_full_master.html

A.9 Debug Hints and Trouble Shooting

1. If installing on Windows XP and your PC is connected to the internet through a firewall the USB install may take up to 15 minutes if you let it complete normally. The work-around for this issue is to simply disconnect your network cable during the USB hardware install.
2. Make sure all of the Configuration Switches (SW3) are set in the off position. This conFigures the DSK for the factory default settings of little endian processor mode booting out of the on-board Flash memory.
3. Some of the Help Files are links to Adobe Acrobat PDF files. If you intend to access these files you must have Adobe Acrobat installed on your system.
4. If you want to verify a successful USB driver install, open your device manager by right clicking on the My Computer icon on your desktop and selecting Properties --> HW --> Device Manager. You should see a new class "SD USB Based Debug Tools" and one Spectrum Digital TMS320C6416 DSK installed.
5. The BUSY LED above the USB connector comes on when power is applied to the DSK. Do not launch Code Composer until the LED is off.

A.10 Error Messages

Many of the error messages below refer to USB enumeration issues. The following window will appear when launching CCS or the Diagnostic Utility indicating the enumeration status**.**



Waiting for USB Enumeration.

1. Message: Failed: Requires Win98, Win2K or Win XP.

   Problem: Your operating system does not support USB.

2. Message:

   Failed: USB device is NOT enumerated or plugged in.

   Problem: Windows cannot find the DSK. Check power and USB Cable.

3. Message:

   Failed: Load USB Application.

   Problem: Loader could not download the DSK emulation application. The on-board emulation controller could be in a bad state. Cycle power on the DSK.

4. Message:

   Failed: USB Channel in use by another app.

   Problem: Another application is using the DSK USB communication channel. Only one application is allowed to communicate to the DSK over the USB channel at the same time. Check for active applications using the Windows Task Manager if open application is not obvious.

5. Message:

   Waiting for USB Enumeration.

   Status: This is a status message which indicates that Code Composer Studio is waiting for communication with the on-board JTAG emulator to be established. This should take less than 10 seconds. Do not hit the close button unless it is clear that something is wrong (the progress bar runs forever).