Offering at Your Lotus Feet

Acknowledgements

I would like to express my deepest gratitude to my Project supervisor, Prof. Siva Sankara Sai, for his constant support during the course of the project work.

I would also like to extend my sincere thanks to Mr. Sivaramakrishnan S, Analog Devices Inc., Bangalore for his valuable guidance and inspiration.

I am deeply indebted to the Department of Physics and its staff for providing all the necessary facilities.

I am also grateful to Sudheer Babu brother for his valuable suggestions and insightful discussions which always motivated me to achieve more than what I can do alone.

Special thanks to Boopala Krishnan, who inspired and guided me whenever I found myself at crossroads. Many Thanks to Research Scholar Murali Ravi for his timely help.

At this juncture, I would like to acknowledge the assistance of Sravan, Guna Sekar, Veera Ganesh, Akhilesh Pandey and Sai Chand who helped at various stages of this project.

Completion of this project would not have been possible without the support and nurturing of my friends, from the Department of Physics and the Department of Computer Science, who added more color to my stay here.

I am grateful to all the hostel staff for making my life easy and comfortable and making me feel hostel as a home away from home.

I thank my parents and brother for their constant support and belief in my abilities.

Last but not the least, I am thankful to Swami for giving this opportunity to me for doing his work.

Abstract

Acoustic Noise problems are increasing continuously as industrial equipment such as engines, transformers and compressors are getting employed at almost all the workplaces. Therefore, the Noise Cancellation is becoming a requirement rather than an extravagance. Active Noise Control techniques play an important role in handling these problems. It also refers to the methods, which generate anti-noise signals to reduce the strength of noise signals. Our project involves the study and implementation of these Active Noise Control algorithms. This project was also aimed to implement Active Noise Control techniques to low frequency signals in confined spaces such as car cabins. Two such Active Noise Cancellation algorithms, Least Mean Square and Recursive Least Squares were implemented at the software level during the course of this project work.

Contents

Dedication i

Acknowledgements iii

Abstract iv

List of Figures vii

List of Tables xi

1 Introduction 1
1.1 Motivation 1
1.2 Thesis Outline 2

2 Filters 4
2.1 Definition of Filter 4
2.2 Applications of Filters 5
2.3 Types of Filters 5
2.3.1 Analog Filters 5
2.3.2 Digital filters 6
2.3.3 Types of Digital filters based on impulse response 6
2.3.4 Types of Digital filters based on the nature of coefficients . . 8
2.3.5 Underlying Structures of Adaptive Filters 8
2.3.6 Weight Updating Section of Adaptive Filters 9
2.3.7 Configurations of adaptive filters 11

3 Literature Survey 13
3.1 System Identification 13
3.2 Applications of System Identification 15
3.3 Noise Cancellation 15
3.3.1 Passive Noise Cancellation 16
3.3.2 Active Noise Cancellation 16
3.4 Methods of Active Noise Cancellation 17

3.4.1 Adaptive Cancellation Method . . . 17
3.4.2 Synthesis method . . . 18
3.5 Limitations of ANC Systems . . . 19
3.6 Applications of ANC Systems . . . 20
3.7 Hardware platform . . . 20
3.7.1 Development tools . . . 22
3.7.2 Keil MDK . . . 23
3.7.3 Communication protocols . . . 23
3.7.4 Hardware Bottlenecks . . . 24

4 Implementation of Adaptive Algorithms 25
4.1 System Identification . . . 25
4.1.1 Direct approach . . . 27
4.1.2 Iterative Approach . . . 28
4.1.3 Pseudo code for System Identification . . . 31
4.2 Active Noise Cancellation . . . 32
4.2.1 Least Mean Square algorithm . . . 33
4.2.2 Recursive Least Square algorithm . . . 33
4.2.3 Pseudo code for Active Noise Cancellation . . . 34

5 Results 35
5.1 System Identification using deconvolution . . . 35
5.2 System Identification using adaptive filters . . . 37
5.3 Noise Cancellation for non-real time signals . . . 41
5.4 Noise cancellation for real time signals . . . 45
5.4.1 Noise coming from fan motor . . . 45
5.5 Noise Cancellation (considering interfering signals) . . . 47

6 Conclusions and Future Work 49
6.1 Conclusions . . . 49
6.2 Future work . . . 49

Bibliography 50

List of Figures

2.1 Generic structure of a filter 4

2.2 Structure of a FIR filter 7

2.3 Structure of an IIR filter 7

2.4 Basics Structure of transversal adaptive filter 9

2.5 Learning process according to Gradient Descent algorithm [1] 11

2.6 Block Diagram for System Identification 11

2.7 Block diagram for Inverse Modelling 12

2.8 Block diagram for Interference Cancellation 12

3.1 Block Diagram for System Identification 14

3.2 Comparison of noise and anti-noise waves 16

3.3 Block diagram for noise cancellation using Feed-forward Control . . 18

3.4 Block diagram for noise Cancellation using Synthesis method 19

3.5 STM32F407VG microcontroller[2] 20

3.6 Wolfson Pi audio card[2] 21

3.7 Structure of a codec 21

3.8 Connections between microcontroller and audio card[2] 21

3.9 Connection between PC and microcontroller 22

4.1 Block Diagram for System Identification 26

4.2 Block diagram for Active Noise Cancellation 32

5.1 Block diagram for System Identification using deconvoltuion 35

5.2 Correct Identification of system using deconvolution 36

5.3 Incorrect Identification of system using deconvolution 36

5.4 Comparison of Plant weights and adaptive Filter weights before training in block based processing 38

5.5 Comparison of Plant weights and adaptive filter weights after training in block based processing 39

5.6 Comparison of Plant weights and adaptive filter weights before training in sample based processing 39

5.7 Comparison of Plant weights and adaptive Filter weights after training in sample based processing 40

5.8 Frequency domain representation of Input signal 41

5.9 Frequency domain representation of plant 41

5.10 Frequency domain representation of secondary path 42

5.11 Comparison of error signal with step-sizes according to LMS algorithm 42

5.12 Comparison of plant output and controller output in LMS algorithm 43

5.13 Comparison of LMS algorithm performance with different number of filter weights 43

5.14 Error graph in RLS algorithm 44

5.15 Comparison of plant output and controller output in RLS algorithm 44

5.16 Frequency domain representation of fan noise 45

5.17 Error graph obtained in LMS algorithm with input as fan noise . . 45

5.18 Comparison of plant output and controller output LMS algorithm with input as fan noise 46

5.19 Error graph obtained in RLS algorithm with input as fan noise . . . 46

5.20 comparison of plant output and controller output using RLS algorithm with input as fan noise 47

5.21 Frequency domain representation of input signal 47

5.22 Frequency domain representation of input signal with interfering signals 48

5.23 Frequency domain representation of error signal 48

List of Tables

5.1 Results obtained for System Identification using deconvolution method 37

5.2 Results obtained for system identification using adaptive algorithms 40

Chapter 1

Introduction

1.1 Motivation

One of the traditional approaches followed to cancel these noise signals is by using passive materials like sound absorbers. However, it was observed that these techniques work best for the middle and high frequencies (above 500 Hz), but not for frequencies less than 500 Hz. It is because, at low frequencies, the acoustic wavelengths become large compared to the thickness of an acoustic absorber. It is also difficult to stop the transmission of low - frequency signals from one place to another unless there is a heavy interference barrier. Since most of the acoustic noise problems are because of the low-frequencies, this method of cancelling noise won't be effective as the solutions will become expensive and bulky[3].

A modern approach known as Active Noise Control (ANC), can solve this problem by emitting anti-noise signals to cancel the noise signals. The phase of this anti-noise signal should be made opposite to that of the noise signal in order to obtain Noise Cancellation. This technique works best for the low frequencies, where the traditional techniques were proven to be less effective [4]. Therefore, in order to get better results in terms of overall Noise Cancellation, both the traditional and modern approaches should be used.

This Idea of controlling the noise signals using anti-noise signals was first introduced by Lueg [4] in the 1930s. As the characteristics of the environment and acoustic noise sources are time-varying, the perfect design of ANC systems which can handle all these variations is still the need of the hour. This requires complex adaptive algorithms to run on hardware units. Continuous progress of Active Noise Cancellation includes the development in signal processing algorithms and DSP hardware.

As more powerful processing chips are available today, the system performance can directly be improved with the algorithmic approach. Therefore, the main motive of this project is to design and implement an ANC system in a car, by which the noise generated by the car engine can be cancelled at each passenger location inside the cabin, for a better travel experience.

1.2 Thesis Outline

As filters play a crucial role in any signal processing algorithms, we discuss about filters in **Chapter 2**. As the environment inside the car cabin changes with time, the emphasis was made on the adaptive filters, which can change its functionality with time.

Chapter 3 provides technical details on the concepts of System Identification, Active Noise Cancellation and the hardware used to implement the same. In order to cancel the noise signals using active noise control techniques, the incoming anti-noise signals and the noise signals should have a phase difference of 180 degrees at the target location. Here, target location refers to the place, noise signals are to be cancelled. Since the source of noise is physically separated from the target location in most of real-time applications, the frequency of the anti-noise signals emitted by the source will be affected by the surrounding environment, which in turn changes the phase. Therefore, the characterization of this environment needs to be done prior to release of the anti-noise signals. This process of characterizing the environment is known as System Identification. Anti-noise signals should exhibit 180-degree phase difference with noise signals after passing through this environment, at target location.

In Chapter 4, different adaptive algorithms were implemented to obtain Noise Cancellation.

Chapter 5 explains the results obtained for System Identification and Active Noise Cancellation methods.

Chapter 6 gives the conclusions and the future work.

Chapter 2

Filters

In this chapter, we discuss different types of filters. First, we have a brief introduction to filters followed by their classification. Next, we have a few machine learning concepts such as the cost function and the Gradient descent techniques. Lastly, we look at the adaptive filters and their importance.

2.1 Definition of Filter

Generically, anything that changes the input is known as a filter, which is sometimes referred to as a system as well. These filters are used in various applications, where specific features are removed while some are retained.

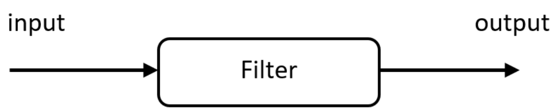

Figure 2.1: Generic structure of a filter

2.2 Applications of Filters

(a) In Image processing, filters are used to enhance certain features of an image in addition to the removal of less interesting features. In general, filters are used for smoothing, sharpening and edge enhancement in image processing applications.

(b) In Signal processing, filters are mainly used to manipulate the frequencies present in the signal. They are used in Analog to Digital Converters (ADC) for removing aliased frequencies and also in Digital to analog converters (DAC) during the reconstruction of the analog signal.

(c) In Radio communications, filters are used mainly in radio receivers to capture only the desired signal, rejecting all other interfering signals.

(d) In Audio electronics, a network of filters is used to separate low frequencies, mid-frequencies and high frequencies so that they can be sent to woofers, mid-range speakers and tweeters respectively.

2.3 Types of Filters

Filters are classified mainly into two categories, based on the domain in which filtering takes place.

2.3.1 Analog Filters

Filtering action in analog filters is done by using passive components like resistors, capacitors and active components like operational amplifiers. Here the signal stays in its actual form throughout the process of filtering. One of the major drawbacks of these filters is its inability to adapt. Filter components have to be redesigned for desired changes at the output of the filter.

2.3.2 Digital filters

Filtering action in digital filters is done on quantized, sampled signal values of the input signal for reducing or enhancing certain features of a signal, by using Processors. ADCs have to be used in order to obtain samples of the signal, whereas DACs have to be used while reconstructing the signal. Therefore, the performance of a digital filter is mainly decided by its ADC, DAC and Processor Specifications. Since these components are programmable, changes can be made without disturbing the filter circuitry. The major advantage of these filters is its ability to adapt to the changes in the environment. Moreover, the filtering process can be modified easily by changing the filter coefficients.

One of the important terms to be considered while using digital filters is the number of filter coefficients. Exact frequency responses cannot be obtained by digital filters unless we use a large number of filter coefficients. For example, a low pass filter designed to attenuate frequencies greater than 1000Hz, cannot attenuate the frequencies right from 1000Hz. Hence, a large number of filter coefficients need to be used to the exact response to our specification.

2.3.3 Types of Digital filters based on impulse response

Impulse Response is the output of a filter when an impulse is given as the input. As the impulse function is made up of a very large number of frequencies, changes made by the system on the input signal can be easily noted at the output which helps in characterizing a filter.

(a) FIR Filters

Filters which compute their output based on the current and the previous input are often referred to as FIR filters[5], as shown in figure 2.2. Response from FIR filters is considered to be stable always and we can expect a linear phase response always from FIR filters, which is an important property while passing signals through different systems. These filters have finite impulse response.

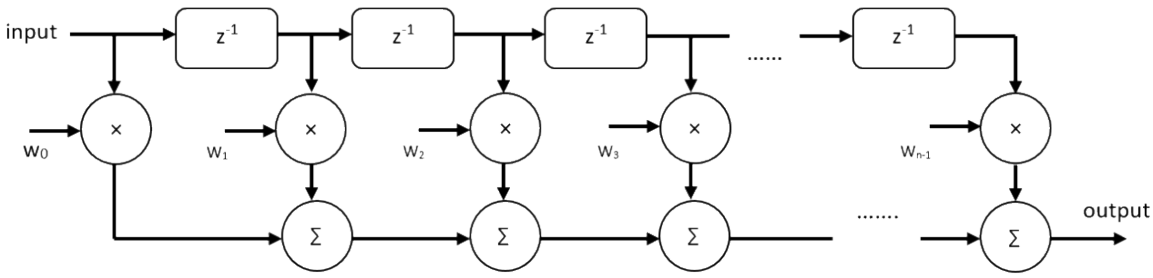

FIGURE 2.2: Structure of a FIR filter

(b) IIR Filters

Filters which compute their output based on the current and previous input samples, and on the previous output samples, are often referred to as recursive filters as shown in figure 2.3. One major advantage of IIR filters over FIR filters is that we can characterize a filter with less number of coefficients. These filters have infinite impulse response, due to presence of feedback.

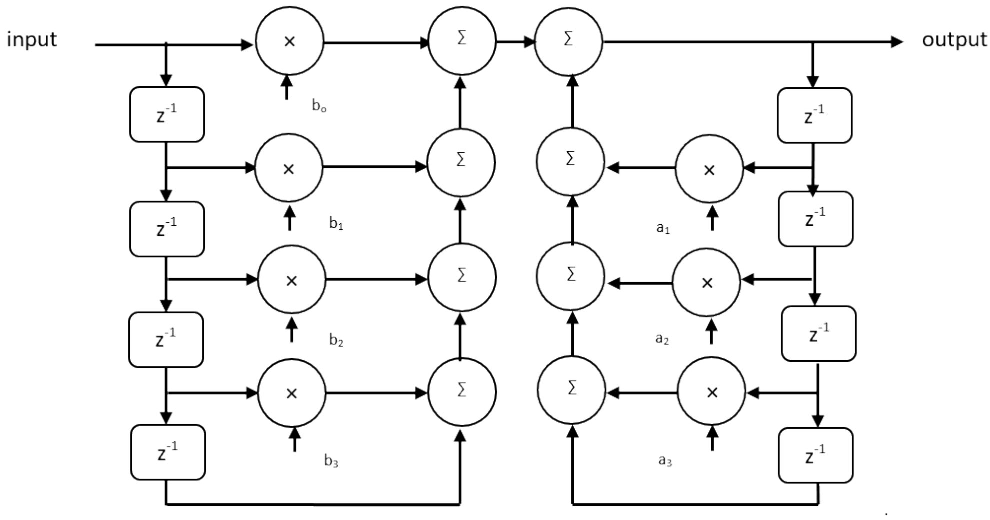

FIGURE 2.3: Structure of an IIR filter

2.3.4 Types of Digital filters based on the nature of coefficients

Based on the nature of weights, filters are classified into mainly two categories.

(a) Fixed Coefficient Filters

For the filters of this kind, coefficients/weights do not change with time. They will be constant throughout the filtering operation. These filters are useful only when the parameters of the signal are stationary. Design of fixed filters is based on prior knowledge of input signals.

(b) Variable Coefficient Filters

Adaptive filters are those digital filters, whose impulse response can be changed over time with the help of a class of algorithms called adaptive algorithms. Design of adaptive filters requires a little or no prior knowledge of input signals[6]. In general, adaptive FIR filters are much widely used than adaptive IIR filters because FIR filter structures are guaranteed to be stable for any set of filter coefficients and it is also simple to update FIR filter Coefficients compared to IIR filter coefficients[7]. Adaptive filters consist of two parts, a) filtering part, b) weight updating part. The Function of the former is to get filter output, while the function of latter is to adjust the filter coefficients under the control of adaptive algorithms to optimize certain performance criterion.

2.3.5 Underlying Structures of Adaptive Filters

The underlying structures of adaptive filters can be of different types, that include transversal structures and recursive structures. Among these, the transversal structures are widely used in the implementation of adaptive filters.

(a) Transversal Structure

It is a temporal filter structure which processes on the temporal samples of its input signal to get the output as shown in figure 2.4.

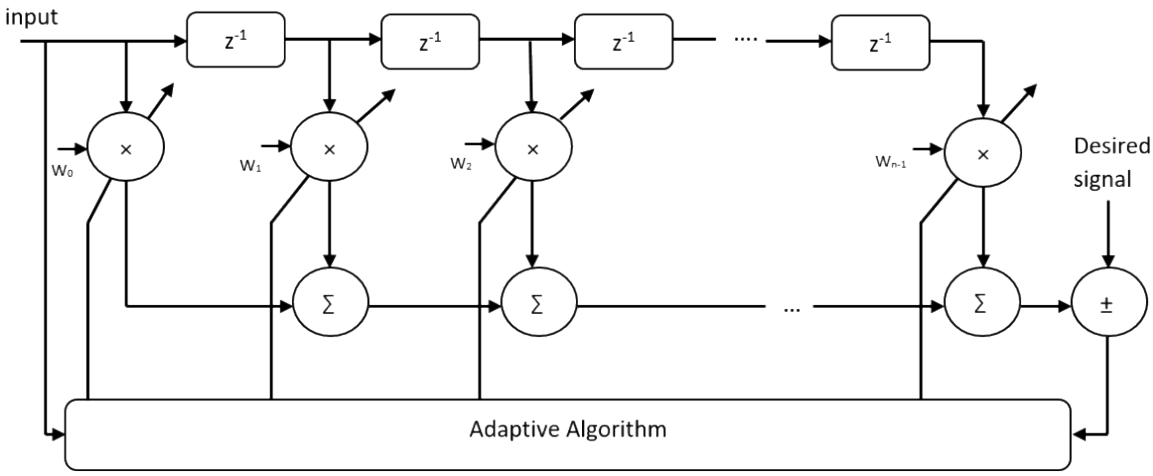

FIGURE 2.4: Basics Structure of transversal adaptive filter

(b) Recursive Structures

Filter structures that compute their output based on current and past input samples and past output samples are often referred to as recursive filters. These filter structures are also used in the IIR filters. Here, the digital filtering section is divided into two parts. One set of coefficients represents the system in forward direction i.e. from input to output of a system, and the other set is to represent reverse direction, which is often called as feedback coefficients. Considering N feed forward coefficients and M feedback coefficients, the adaptive filter must update (N + M) filter coefficients to optimize the performance.

2.3.6 Weight Updating Section of Adaptive Filters

The basic function of the weight updating section of adaptive filters is to update/adjust the weights with the help of adaptive algorithms until certain performance criterion is met. One such performance criterion used is to obtain the

Least Mean Square Error value at the output, which can be computed in a single step/iteration or in multi-steps, i.e., after running the algorithm for few iterations. This idea of meeting desired condition in a single step is not practically feasible, but it is the basis for all other iterative methods. One such method is the Weiner filter concept.

(a) Direct Approach

Weiner filters, also known as Optimal filters are one of the fundamental concepts in adaptive signal processing. Weiner solution finds the filter weight vector in such a way that mean square value of the error signal becomes minimum, and it is also known as Mean Square Error (MSE) criterion.

(b) Iterative Approach

Iterative methods use Cost functions and Gradient Descent concepts. Cost function or loss function gives the ability to the system to estimate the relationship between two parameters y and Y, in terms of their difference[1]. This function runs in an iterative fashion to compare the predicted value against the true value.

Systems learn by minimizing the cost function. In order to minimize the cost function, gradient descent concepts are used[8]. Gradient descent techniques enable the system to learn the direction that the model should take in order to reduce the error. As shown in the figure 2.5, gradient descent algorithm basically uses the difference between two parameters y and Y. Mean Square Error function is the cost function considered in the next sections. Mathematically,

$$MSE = \frac{1}{N}\sum_{i=0}^{N}(y_i - Y_i)^2 \tag{2.1}$$

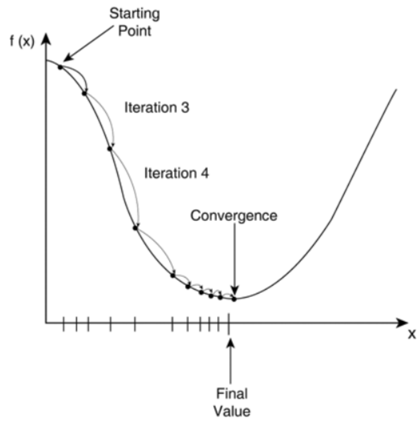

FIGURE 2.5: Learning process according to Gradient Descent algorithm [1]

2.3.7 Configurations of adaptive filters

Adaptive filters have a wide range of applicability such as System Identification, Inverse System Modelling and Interference Cancellation [9].

(a) System Identification

System Identification is a method of building the mathematical form of a dynamic system using the systems input and output signals. Here, the adaptive filter is used to replicate the model of an unknown system. Block diagram for System Identification is shown in figure 2.6. This configuration of adaptive filters is mainly used in acoustic echo cancellation and adaptive noise cancelling applications.

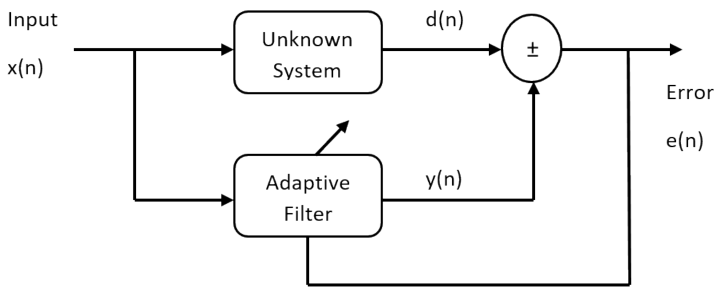

FIGURE 2.6: Block Diagram for System Identification

(b) Inverse Modelling

Adaptive filters in Inverse Modelling are used to find the inverse model of the plant. As the plant and the adaptive filter are connected in series, the impulse response of the adaptive filter will be reciprocal of the plants impulse response, making the path as unity Gain channel. The block diagram for the Inverse Modelling is shown in figure 2.7. This configuration of adaptive filters is used in channel equalization techniques.

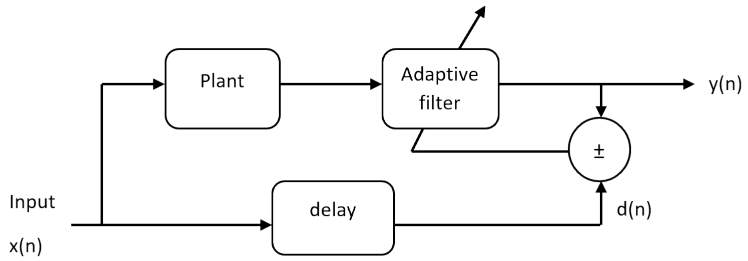

FIGURE 2.7: Block diagram for Inverse Modelling

(c) Interference Cancellation

In this configuration, an adaptive filter is used to reduce the interference contained in the desired signal d(n) as shown in the figure 2.8. Beam-forming applications employ this kind of configuration.

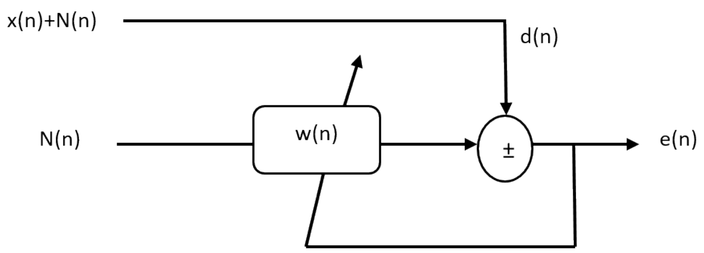

FIGURE 2.8: Block diagram for Interference Cancellation

Chapter 3

Literature Survey

This chapter mainly deals with three concepts. Firstly, we have a brief introduction to System Identification concept followed by its applications. Next, we discuss different types of Noise Cancellation methods to cancel the noise signals. In the next section, we discuss the limitations and applications of ANC Systems. At the end, a briefing on hardware implementation were presented.

3.1 System Identification

System Identification is the method of characterizing the unknown system by using the systems input and output signals.

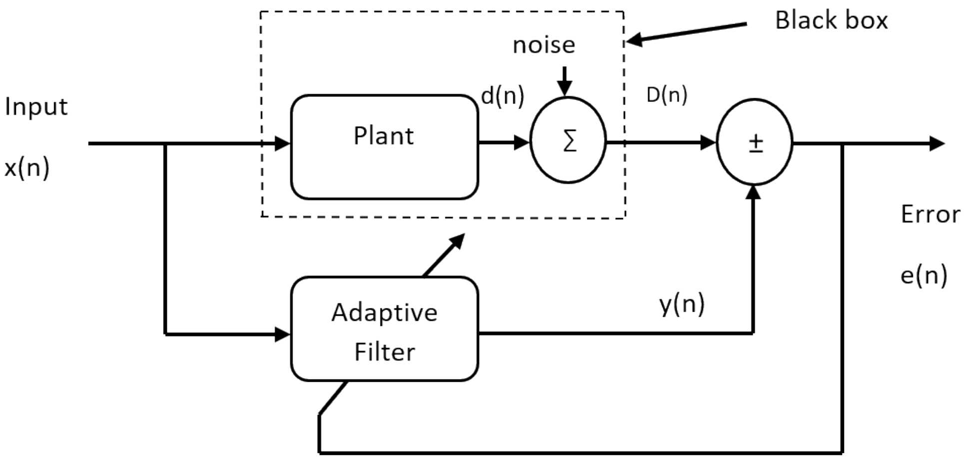

FIGURE 3.1: Block Diagram for System Identification

Two things inside the black box shown in figure 3.1, are
(a) Filter coefficients which can characterize the unknown system and
(b) Noise signal, which corrupts the plant output signal d(n).
Representing x(n) as input to plant and d(n) as the response of the plant, D(n) can be given as

$$D(n) = d(n) + noise \tag{3.1}$$

In this type of configuration, an adaptive filter is used to replicate the signal d(n) accurately at its output. Considering both plant and adaptive filter as FIR Filters, output response obtained near additive block is given by,

$$D(n) = w_{opt}^T x(n) + noise \tag{3.2}$$

where w_{opt}^T represents the best possible vector of filter coefficients for the plant at time index n. Adaption procedure should adjust the weights of adaptive filter i.e., w(n), in such a manner that w(n) should be equal to w_{opt}^T(n).

3.2 Applications of System Identification

(a) Identification of the medium in Communication systems

In general, data is transmitted through media like a fiber optic cable or a radio link etc., which are having the property of distorting the signals. This makes deciphering the information a difficult task at the receiving end. Adaptive filters are used to characterize the effects of media, which can make deciphering the information at the receiver easier.

In this case, the transmitter chooses and transmits the data that is known to the receiver, before transferring the original information. On receiving this data, the receiver starts comparing it with the known signal, in order to identify the effects of the medium[7]. By using this information, receiver decodes the original information sent in the future.

(b) Adaptive Noise Cancelling

Measurement of certain signals like the heartbeat of an unborn child is a difficult task, as the childs heartbeat interferes with the mothers heartbeat [7]. However, if a reference version of mothers heartbeat is available or if it can be sensed accurately, we can measure the childs heartbeat accurately by using an adaptive filter, as it can characterize the relationship between the mothers heartbeat x(n) and the interfered signal d(n). The residual signal e(n) is the heartbeat of the child. In a similar manner, adaptive filters are used in many medical applications.

3.3 Noise Cancellation

The elimination/reduction of noise signal strength comes under the category of Noise cancellation. There are two categories which are commonly used to reduce the strength of the noise signals, a) Passive Noise Cancellation b) Active Noise Cancellation.

3.3.1 Passive Noise Cancellation

It is the method in which noise signal strength is made less by allowing it to pass through sound absorbing materials[4]. An example is an ear muff of an earphone. However, it was found that these methods are not effective with respect to low-frequency signals.

3.3.2 Active Noise Cancellation

It is the method in which the anti-noise waves are used to reduce the strength of the noise signal. This method uses the concept of destructive interference principle, which states that the superposition of the two signals that are perfectly out of phase, cancel each other when they meet at the same place and at the same time as shown in the figure 3.2.

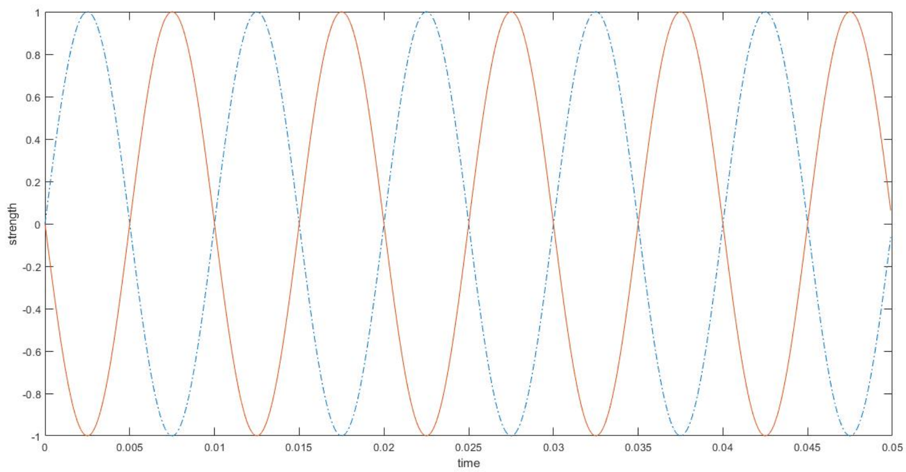

FIGURE 3.2: Comparison of noise and anti-noise waves

This method offers effective solutions in attenuating lower frequency noise signals (i.e., below 500Hz) where passive systems are proven to be less effective. As environment and the noise source are varying with time, frequency content and amplitude of these noise signals will continuously change. Therefore, an Active Noise Cancellation System must be able to handle these variations.

Two things which contribute to the success of the ANC systems are,

1 Anti-Noise signals shape and frequency with respect to the noise signal.

2 Phase difference between noise and anti-noise signals at target location[10].

In order to be useful in real time applications, ANC systems should possess certain characteristics which are:

1 ANC systems should be efficient over a large frequency band for cancelling wide range of noises.

2 The capability of the system to handle variations in parameters like temperature.

3 Reliability and robustness of different components present in the system.

3.4 Methods of Active Noise Cancellation

Two methods are commonly followed in the implementation of ANC systems [11], which are as follows:

3.4.1 Adaptive Cancellation Method

In this method of noise cancellation, ANC Systems consist of a few sensors for measuring the noise field at the source and noise fields near the target location. Sensor, which is used at source is referred as reference sensor, whereas sensor present at target location is called as error sensor, as it measures the environment that consists of noise and anti-noise signals. In addition, reference sensor is useful for giving input to the adaptive filter[12], whereas error sensor is used for adjusting adaptive filter coefficients according to the residual acoustic field.

In this case, ANC System generates the anti-noise waveform after detecting noise signals. This approach is useful for cancelling both periodic and non-periodic noises. In the case of periodic noise environment, a few noise cycles will be stored and then the anti-noise waves will be generated and emitted into the environment to cancel the noise . However, in the case of non-periodic noise, feed-forwarding is used to predict the noise before it reaches the target as shown in the figure 3.3.

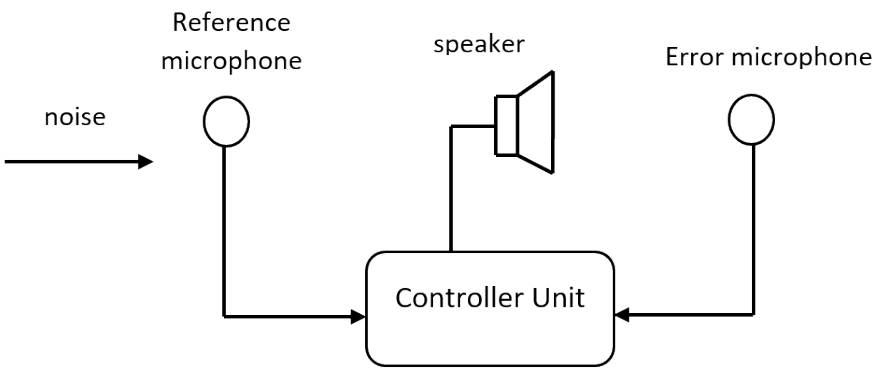

Figure 3.3: Block diagram for noise cancellation using Feed-forward Control

3.4.2 Synthesis method

Noise generated by fans, engines, and propellers is generally a periodic signal. Therefore, by observing the mechanical motion of these sources, we can get a reference signal in the electrical domain, which consists of the fundamental frequency and several harmonics. This method is more suitable for cancelling the periodic noise as it includes sampling, storing the received noise signal and then generating an anti-noise signal to suppress the noise[10].

In this case, Anti-noise waveform is generated by associating a digital pulse train with the noise cycle and the same is used to synchronize the anti-noise waveform with the emitted noise as shown in the figure 3.4. In addition, pulse train can be obtained from non-acoustic sources like engines odometer.

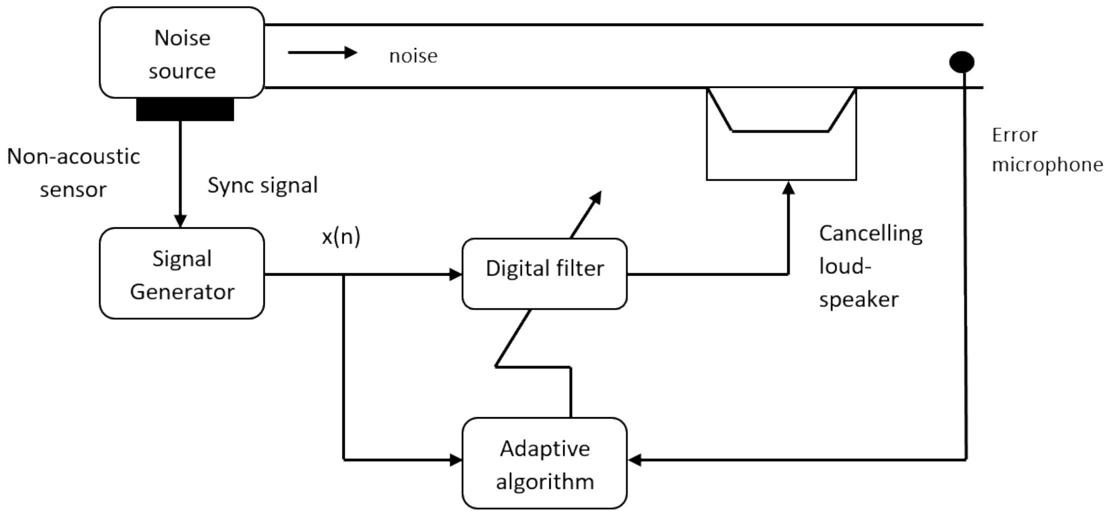

FIGURE 3.4: Block diagram for noise Cancellation using Synthesis method

Advantages of synthesis method:

1 Acoustic feedback from loudspeakers to reference microphones can be avoided.

2 There are no aging problems as reference microphones are not used in this technique.

3 Each harmonic signal can be controlled individually as the reference signal is generated internally.

3.5 Limitations of ANC Systems

1 ANC systems are limited to small, confined spaces as the phase of anti-noise signal wont be constant within the whole volume.

2 Amplification of noise signals happens instead of attenuation if anti-noise signal properties like frequency and amplitude does not matches with the properties of noise signal.

3.6 Applications of ANC Systems

1 Active Noise Cancellation systems are used in mufflers for reducing the noise emitted from exhaust of an combustion engines, and for noise cancellation inside vehicle passenger compartments.

2 In appliances like air conditioners, high end refrigerators, washing machines, vacuum cleaners, ANC is being used.

3 It is also used in many of headphones and ear protectors.

3.7 Hardware platform

Microcontroller STM32F407, developed by STMicroelectronics, that is shown in the figure 3.5 and Wolfson pi audio card that is shown in the figure 3.6 were used to implement the System Identification and noise cancellation concepts in a real-time environment.

The Wolfson Pi Audio Card provides the best quality audio i/o as it supports audio qualities up to 192 kHz/24bit. STM32F407VG belongs to F4 family whose operating frequency ranges are in the range of 168MHz. In general, cortex-M processors are used in the embedded applications as they are the most energy efficient groups.

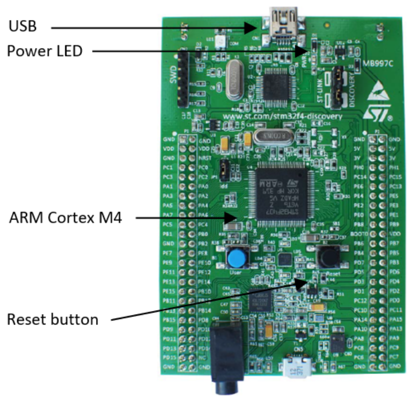

FIGURE 3.5: STM32F407VG microcontroller[2]

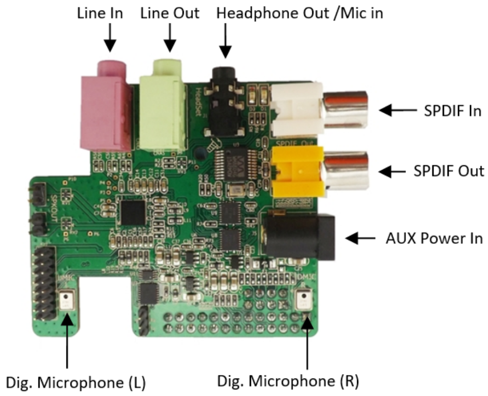

FIGURE 3.6: Wolfson Pi audio card[2]

In general, to process audio frequency signals, the controller should consist of a digital signal processor and an analog interface[13]. Such system can be realized by using the floating point processor, STM32F407, on the discovery board and the codec WM5102 on the Wolfson Audio Card. Codec refers to a component which consist of both ADC and DAC shown in the figure 3.7. Connections between Wolfson pi audio card and STM32 board are shown in the figure 3.8.

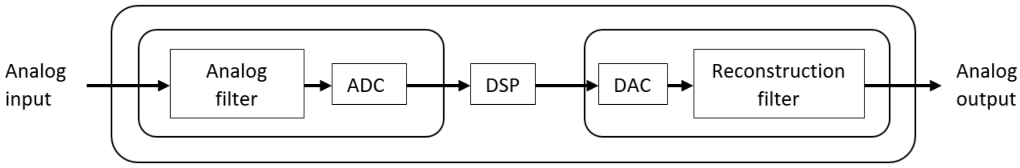

FIGURE 3.7: Structure of a codec

Wolfson Audio Card	function	STM32F407 Discovery
J7 pin #1	5V from Discovery	P2 pin #3
J4 pin #15	LDO ENA (3V from Discovery)	P2 pin #5
J7 pin #7	GND	P1 pin #49
J7 pin #13	GND	P2 pin #1
J7 pin #11	I2C SDA (PB11 on Discovery)	P1 pin #35
J7 pin #9	I2C SCL (PB10 on Discovery)	P1 pin #34
J4 pin #11	GPIO PD11	P1 pin #43
J7 pin #3	I2S WCLK (PB12 on Discovery)	P1 pin #36
J7 pin #6	I2S BCLK (PB13 on Discovery)	P1 pin #37
J7 pin # 4	I2S TX (PC3 on Discovery)	P1 pin#9
J7 pin #5	I2S RX (PC2 on Discovery)	P1 pin #10

FIGURE 3.8: Connections between microcontroller and audio card[2]

Some of the requirements for Active Noise Cancellation include:

1 Few sensitive microphones.

2 High-resolution ADCs and DAC$'$s for better accuracy.

3 A reasonable amount of Flash Memory and RAM to store the data.

4 A processing unit that can operate on floating point numbers.

5 High-quality speakers.

The combination of STM32 board and Wolfson pi meets all the above requirements and some of them are:

1 Core: 32-bit ARM Cortex M4 Processor capable of operating operate at 168 MHz clock.

2 Memory: 1 Megabyte Flash Memory and 192Kb RAM.

3 Peripherals: ADCs of 12-bit resolution, two DACs.

4 Two high-quality MEMS microphones for capturing audio signals.

3.7.1 Development tools

Certain tools like the compiler, debugger, and In-circuit Serial Programmer are required to write the code as shown in the figure 3.9.

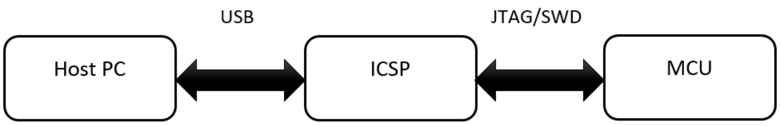

FIGURE 3.9: Connection between PC and microcontroller

Integrated Development Environments (IDE) like Keil combine all of the necessary tools into an integrated environment for developing code on microcontrollers. In addition to these IDEs, an ICSP is required to program and test the code on the microcontroller. It acts as an interface between the microcontroller and personal computer via a USB port. These ARM Cortex-M microcontrollers support two programming protocols such as JTAG and SWD, that are supported by many ICSPs such as Keil U-Link 2 and ST-Link.

Using the ST-LINK programming and debugging tool, the STM32 board connects to the PC via a USB cable. The Keil MDK - ARM development environment that runs on the host PC allows compiling, linking and downloading the software written in C to run on the discovery board.

3.7.2 Keil MDK

Keil Microcontroller Development Kit (MDK) is the complete software development environment for a wide range of cortexM based microcontrollers. This MDK is used to create, build and debug embedded applications. This IDE combines project management, run-time environment, source code editing and the program debugging in a single powerful environment.

3.7.3 Communication protocols

Communication protocols are the set of rules and regulations that allow two electronics devices to exchange data with one another. Wolfson pi audio card communicates with the STM32 board by using I2C bus for control, for writing into codec registers and I2S for the audio data transfer.

Full duplex communication between the WM5102 codec and the ARM processor is implemented on the STM32F407 microcontroller using two I2S instances, SPI/I2S and I2S2_ext. SPI/I2S2 is configured as receiver and I2S2_ext is configured as transmitter. The WM5102 codec operates in master mode as it generates I2S word and clock signals, whereas the STM32F407 I2S peripheral receives this information by operating in slave mode. In this way, the real time operations are controlled by timing of I2S interface, which in turn decided by codec.

3.7.4 Hardware Bottlenecks

The signal data is getting stored as large floating point numbers when MEMS microphone is used that is available on WM5102 board. The analysis in this regard could not be achieved due to brevity of completing the software implementations. Therefore, hardware implementation could not be completed during the course of this project work.

Chapter 4

Implementation of Adaptive Algorithms

In this chapter, we see the implementation of the adaptive algorithms such as Least Mean Square (LMS) and Recursive Least Squares (RLS) for solving System Identification and Noise Cancellation Problems. Initially, we see the fundamental steps that are involved in any adaptive filtering algorithms. Later in the chapter, we see the implementation of algorithms for finding unknown environments. Finally, we see the integration of system identification with active noise control techniques to obtain Noise cancellation.

4.1 System Identification

The terminology used in the system identification concept as shown in the figure 4.1 is as follows:

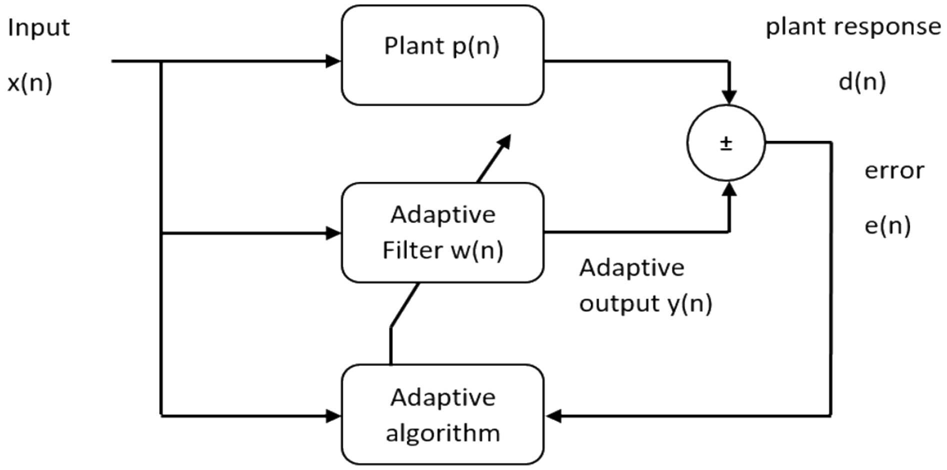

FIGURE 4.1: Block Diagram for System Identification

N: the order of the filter,

x(n): input signal vector of size N*1 in nth iteration,

w(n): adaptive filter weight vector of size N*1 in nth iteration,

y(n): output of FIR filter in nth iteration,

d(n): value of desired signal in nth iteration,

e(n): error in nth iteration,

R: auto-correlation matrix of input x(n),

p: Cross-Correlation vector between input x(n) and desired signal d(n),

and μ represents step-size

Any adaptive algorithms consist of two process [9]

1. Filtering Process:

The filtering process is composed of two steps

(a) Computation of plant output d(n) and the adaptive filter output y(n)

$$y(n) = y(n)^T x(n) \tag{4.1}$$

$$d(n) = p(n)^T.x(n) \tag{4.2}$$

(b) Computing the error

$$e(n) = d(n) - y(n) \tag{4.3}$$

2. Adaptive Process:

The task of modifying the adaptive filter coefficients is done by using the error obtained in the equation (3.3). Ideally, optimum filter weights can be obtained in a single step by using the Weiner filter concept. Due to the practical constraints such as latency and memory requirement, we need to adapt an iterative approach to compute the Weiners solution by using adaptive algorithms. These algorithms are the procedures for modifying the weights of the adaptive filters in order to reach the minimum of the chosen cost function. Moreover, these adaptive algorithms are classified into different categories based on the procedure they adapt to modify the filter coefficients.

4.1.1 Direct approach

Weiner filter characterizes the unknown system in such a manner that, mean square value of the error signal (MSE) is minimum. The cost function is given by,

$$J(n) = E\left[e^2(n)\right] \tag{4.4}$$

which basically represents the mean square error. Therefore, by substituting the error obtained from equation (3.3) in equation (3.4), we can get

$$J(n) = E\left[d^2(n) - 2d(n).y(n) + y^2(n)\right] \tag{4.5}$$

Considering the desired signal of zero mean, equation (3.5) can be rewritten as,

$$J(n) = \sigma_d^2 - 2w^T p \ + w^T R\, w \tag{4.6}$$

As cost function J(n) obtained in equation (3.6) is in the form of quadratic function, a unique minimum point can be obtained, which is the goal of the Weiner filter [14][15]. Therefore, optimum filter weights that represent the unknown system, can be obtained by computing the gradient of the equation (3.6) and then equating it to zero.

$$w_{opt} = R^{-1}p \tag{4.7}$$

Equation (3.7) indicates that Weiner solution requires inverse of the auto-correlation matrix and cross-correlation vector in order to compute optimum filter weights. However, the computation of the matrix inversion is a complex issue. Therefore, we need to use certain methods which can compute the Weiner solution in an iterative fashion. Thus, we need to select an adaptive algorithm based on the application, computational complexity, and latency. One such algorithm is steepest descent algorithm, which does approximations on intermediate parameters in deriving the weight updating equation and it is given as,

$$w(n+1) = w(n) - (\mu/2).gradient(J(n)) \tag{4.8}$$

Substituting equation (3.6) in the equation (3.8), we get

$$w(n+1) = w(n) + \mu[\, p - R.w(n)\,] \tag{4.9}$$

However, the problem of estimating the parameters $\boldsymbol{R}$ and $\boldsymbol{p}$ continues to persist. Therefore, the LMS algorithm was devised to solve this problem, by using instantaneous estimates of $\boldsymbol{R}$ and $\boldsymbol{p}$ instead of using the mean values.

4.1.2 Iterative Approach

(a) Least Mean Square (LMS) algorithm

LMS algorithm identifies the unknown system by finding the adaptive filter coefficients in an iterative manner which corresponds to minimum value of mean square error signal (MSE). The adaptive filter coefficients are modified based only on the error at the current timestamp[9].

The algorithm starts by initializing all the adaptive filter weights to zeros. At each step, the adaptive weights will be modified based on the gradient of the MSE.

Basic weight updating equation is given by,

$$w(n+1) = w(n) - \mu.gradient(J(n)) \tag{4.10}$$

μ is the step size controlling the size of the steps taken by each gradient. By approximating parameters R and p, equation (3.10) is calculated as,

$$w(n+1) = w(n) + \mu.e(n).x(n) \tag{4.11}$$

where w(n+1) refers to the weights of the adaptive filter in the (i+1)th iteration, and w(i) refers to the weights in the ith iteration.

Stability and speed of the LMS algorithm are mainly dependent on the convergence rate parameter called step-size. Misadjustment is one of the parameters indicating how far the obtained iterative solution is from the optimal solution of the used cost function.

(b) Recursive Least Squares (RLS) algorithm

Recursive Least square algorithm finds the optimum filter coefficients, resulting in the least value of the weighted linear squares of the error function given by,

$$J(n) = \sum_{i=0}^{n} \lambda^{n-i} e^2(i) \tag{4.12}$$

λ represents the memory of the algorithm and e(i) is the value of the error at time i, which can be computed by,

$$e(i) = d(i) - w^T(n)x(i) \tag{4.13}$$

where w(n) represents adaptive filter weight vector of the L^{th} order and x(i) is a function of all the input data vectors up to the present time and current value of w(n).

By Substituting these values in the equation (3.12), we get

$$J(n) = \sum_{i=0}^{n} \lambda^{n-i} - 2w^T(n) \sum_{i=0}^{n} d(i)x(i) + w^T(n) \sum_{i=0}^{n} \lambda^{n-i} x(i)x^T(i)w(n) \tag{4.14}$$

By using auto-correlation matrix R(n) and cross correlation vector p(n), equation (3.14) can be rewritten as

$$J(n) = \sum_{i=0}^{n} \lambda^{n-i} d^2(i) - 2p^T(n)w(n) + w^T(n)R(n)w(n) \tag{4.15}$$

where

$$R(n) = \sum_{i=0}^{n} \lambda^{n-i} x(i)x^T(i) \tag{4.16}$$

$$p(n) = \sum_{i=0}^{n} \lambda^{n-i} d(i)x(i) \tag{4.17}$$

In order to minimize the cost function, gradient of cost function should be made zero, which results in

$$R(n)w_{opt}(n) = p(n) \tag{4.18}$$

where w_{opt}(n) represents the optimum adaptive filter weight vector at time *n*. But the computation of R^{-1}(n) is quite complex, as the time index could increase to a very large value in real time. This problem can be solved by using a recursive approach i.e. computing R(n) and p(n) from R(n-1) and p(n-1). Therefore, R(n) and p(n) can be rewritten as,

$$R(n) = \lambda R(n-1) + x(n)x^T(n) \tag{4.19}$$

$$p(n) = \lambda p(n-1) + d(n)x(n) \tag{4.20}$$

Even now, computing $R^{-1}(n)$ is a complex issue. Therefore, in order to reduce the computational requirements, fast RLS algorithm is developed which uses a time-recursive approach to compute R−1(n) from R−1(n-1), instead of finding R(n) and then finding R−1(n). By using the matrix inversion lemma, We can write $R^{-1}(n)$ as

$$R^{-1}(n) = \lambda^{-1}R^{-1}(n-1) - \frac{(\lambda^{-1}R^{-1}(n-1)x(n)x^T(n))}{x^T(n)\lambda^{-1}R^{-1}(n-1)x(n)+1} \tag{4.21}$$

Representing $R^{-1}(n)$ by Q(n) and then defining k(n), we get

$$k(n) = \frac{z(n)}{x^T(n)z(n)+1} \tag{4.22}$$

$$Q(n) = \lambda^{-1}Q(n-1) - k(n)z^T(n) \tag{4.23}$$

where z(n) = λ^{-1}Q(n-1) x(n) Therefore, filtering process and weight updating process in RLS algorithm are given by,

$$e(n) = d(n) - w^T(n)x(n) \tag{4.24}$$

$$z(n) = \lambda^{-1}Q(n-1)x(n) \tag{4.25}$$

$$k(n) = \frac{z(n)}{x^T(n)z(n)+1} \tag{4.26}$$

$$w(n+1) = w(n) + k(n)e(n) \tag{4.27}$$

$$Q(n) = \lambda^{-1}Q(n-1) - k(n)z^T(n) \tag{4.28}$$

4.1.3 Pseudo code for System Identification

1 Initialize plant to some known response

2 Initialize adaptive weights to zeros

3 Initialize the input delay line to zeros

4 Shift the delay line

5 Generate a single random value in (+1,-1)

6 Add the value to head of the delay line

7 Normalize the delay line

8 Run algorithm update routine by using error

9 Repeat the process from step 4 to step 8 until adaptive filter weights converges to plant

4.2 Active Noise Cancellation

Secondary path which was identified using system identification, has to be incorporated into Noise Cancellation system as shown in 4.2.

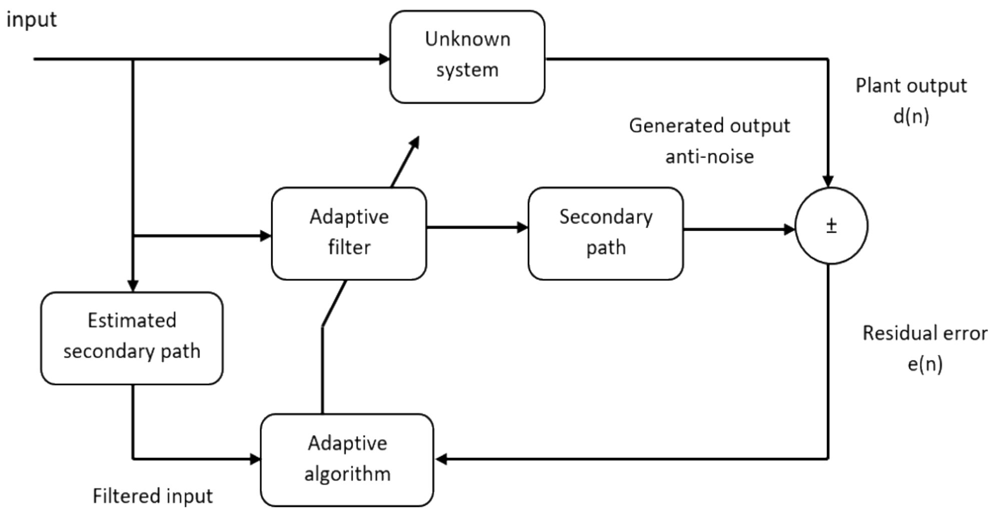

FIGURE 4.2: Block diagram for Active Noise Cancellation

This type of configuration is known as the Filtered-x LMS algorithm[4], which removes the effect of secondary path on anti-noise signals at the target location. The error signal is slightly modified compared to equation (3.3) and it is given by

$$e(n) = d(n) - s(n) * [w^T(n)x(n)] \tag{4.29}$$

where n represents the time index, s(n) represents the impulse response of the secondary path function S(z), and * indicates the convolution.

4.2.1 Least Mean Square algorithm

As LMS algorithm assumes the cost function as mean square function, adaptive filter tries to minimize this instantaneous squared error. Hence, the weight updating equation is given by

$$w(n+1) = w(n) - (\mu/2). \bigtriangledown (J(n)) \tag{4.30}$$

By solving for instantaneous estimate of the mean square error, the weight updating equation can be rewritten as

$$w(n+1) = w(n) + \mu(x(n) * s(n))\ e(n)) \tag{4.31}$$

in which * represents convolution.

4.2.2 Recursive Least Square algorithm

As RLS algorithm assumes the cost function as weighted linear least square function, adaptive filter tries to minimize this weighted squared error.

$$J(n) = \sum_{i=0}^{n} \lambda^{n-i} e^2(i) \tag{4.32}$$

$$z(n) = \lambda^{-1} Q(n-1).(x(n) * s(n)) \tag{4.33}$$

$$k(n) = \frac{z(n)}{x^T(n)z(n) + 1} \tag{4.34}$$

$$J(n) = \sum_{i=0}^{n} \lambda^{n-i} e^2(i) \tag{4.35}$$

$$w(n+1) = w(n) + k(n)e(n) \tag{4.36}$$

$$Q(n) = \lambda^{-1}Q(n-1) - k(n)z^T(n) \tag{4.37}$$

Where $x^{'}(n)$ is the filtered input sequence and * represents the convolution.

4.2.3 Pseudo code for Active Noise Cancellation

1 Initialize plant to some known response

2 Initialize adaptive weights to zeros

3 Initialize secondary path weights and estimated secondary path to the coefficients found in System Identification

4 Initialize all delaylines to zeros

5 Shift all delaylines by 1 unit

6 Generate a input value

7 Add the value to head of plant, adaptive filter and estimated secondary path delaylines.

8 Compute adaptive filter output

9 Add this output to secondary path delayline

10 Compute the error by using outputs of plant and controller

11 Run algorithm update routine

12 Repeat the process from step 6 to step 11 for whole length of the input signal

Chapter 5

Results

5.1 System Identification using deconvolution

Mathematically, deconvolution is carried out by using a polynomial division, which involves large number of operations. In order to implement deconvolution in real-time systems, a high-speed processor and a sufficient memory space on the hardware unit are necessary. The block diagram for system identification using deconvolution is shown in the figure 5.1.

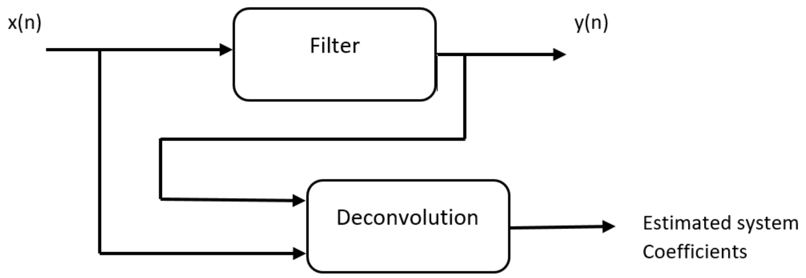

Figure 5.1: Block diagram for System Identification using deconvoltuion

The deconvolution process was implemented using an existing Matlab function. A low pass filter is considered as the system to be identified. The figure 5.2 and the figure 5.3 show the results obtained using deconvolution process. The table 5.1 explains the drawback of the deconvoltuion method.

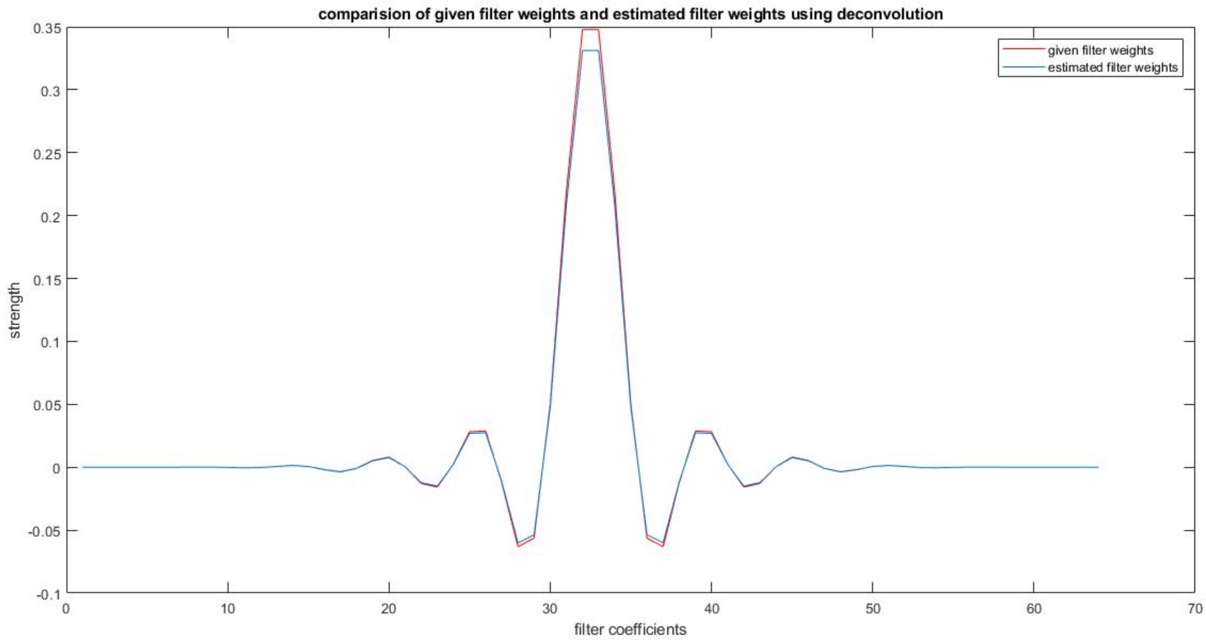

FIGURE 5.2: Correct Identification of system using deconvolution

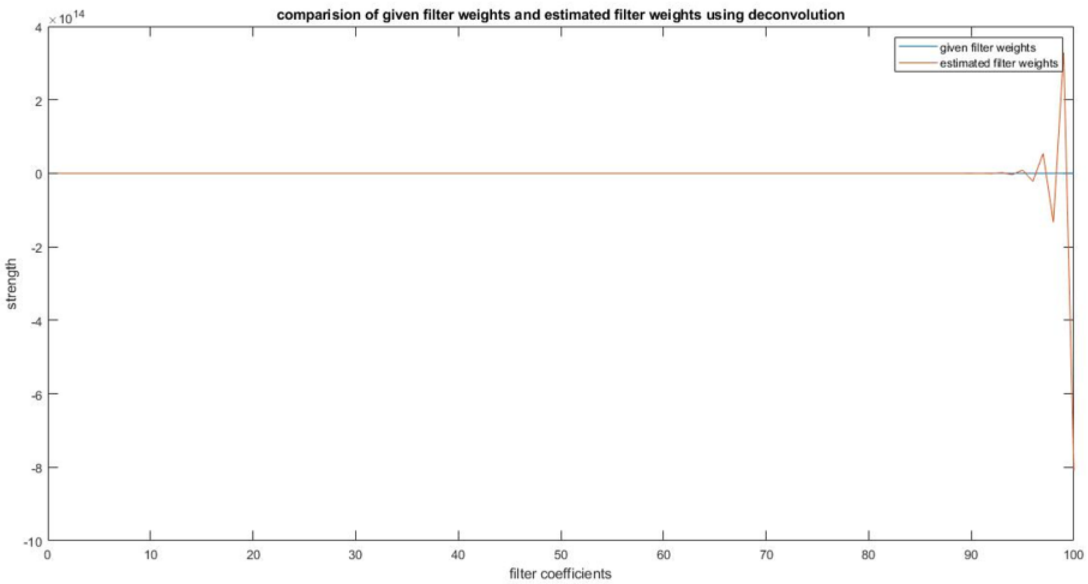

FIGURE 5.3: Incorrect Identification of system using deconvolution

Length of input signal (in samples)	Filter order: 9	Filter order: 99	Filter order: 255
100	Correct identification	Incorrect identification	Incorrect identification
1000	Correct identification	Incorrect identification	Incorrect identification
10000	Correct identification	Incorrect identification	Incorrect identification
100000	Correct identification	Incorrect identification	Incorrect identification

TABLE 5.1: Results obtained for System Identification using deconvolution method

5.2 System Identification using adaptive filters

From the table 5.1, it is clear that the deconvolution method is not the perfect method to identify an unknown system. Hence, we tried to identify the system using adaptive filters, whose block diagram is shown in the figure 4.1. The LMS algorithm is used as the adaptive algorithm to update the weights of the adaptive filter. Input signals can be passed to the filter in two different forms Block based processing and Sample based processing. For comparing these two techniques, some parameters were chosen as common. Those are as follows:

Chosen Parameters:

1 Plant as a low pass filter of 64 weights, with cut off frequency of 2500 Hz.

2 Adaptive filter with all coefficients initialized to zeros.

3 White Gaussian Noise signal as input to the system.

4 Stopping the execution of algorithm when error between the outputs is less than 0.0001.

Block based Processing: A block of sample values in the time domain will be given as input to the algorithm at every single iteration and the adaptive filter's weights will be updated only after processing a block. The figure 5.4 shows the comparison between adaptive filter weights and plant coefficients before training the system. The figure 5.5 gives the comparison between adaptive filter weights and plant weights after training.

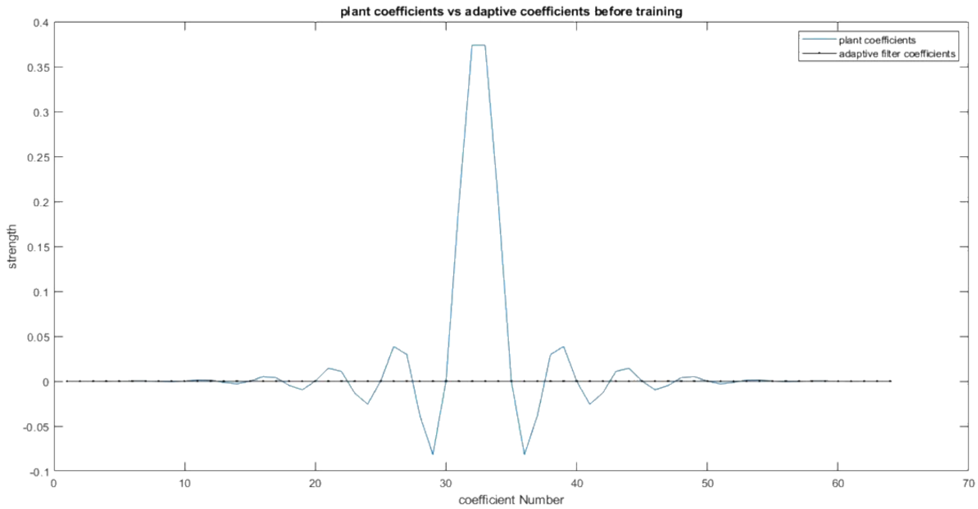

FIGURE 5.4: Comparison of Plant weights and adaptive Filter weights before training in block based processing

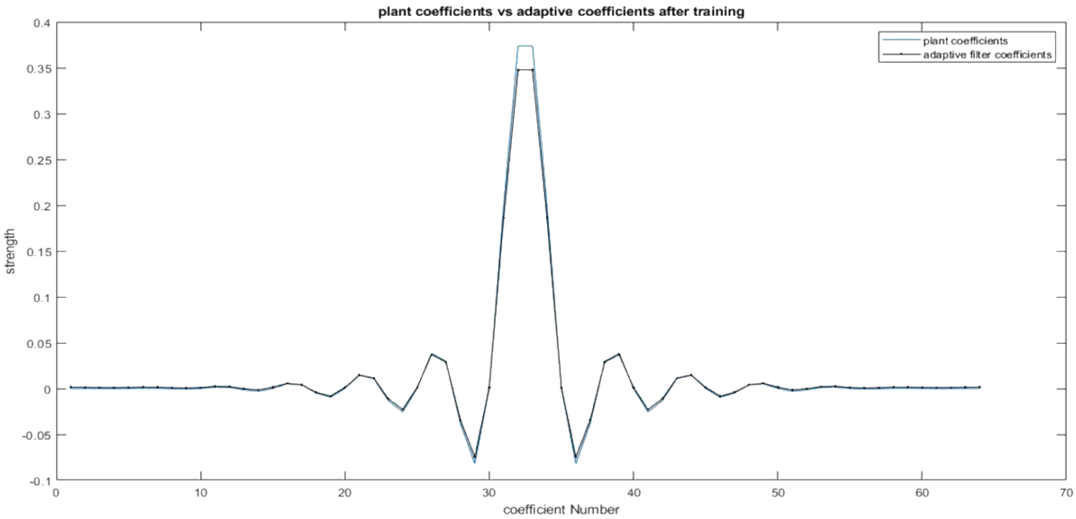

FIGURE 5.5: Comparison of Plant weights and adaptive filter weights after training in block based processing

(b) Sample based processing

The weights of the adaptive filter will be updated after processing on each sample. The figure 5.6 shows the comparison between adaptive filter weights and plant coefficients before training the system. The figure 5.7 gives the comparison between adaptive filter weights and plant weights after training the system.

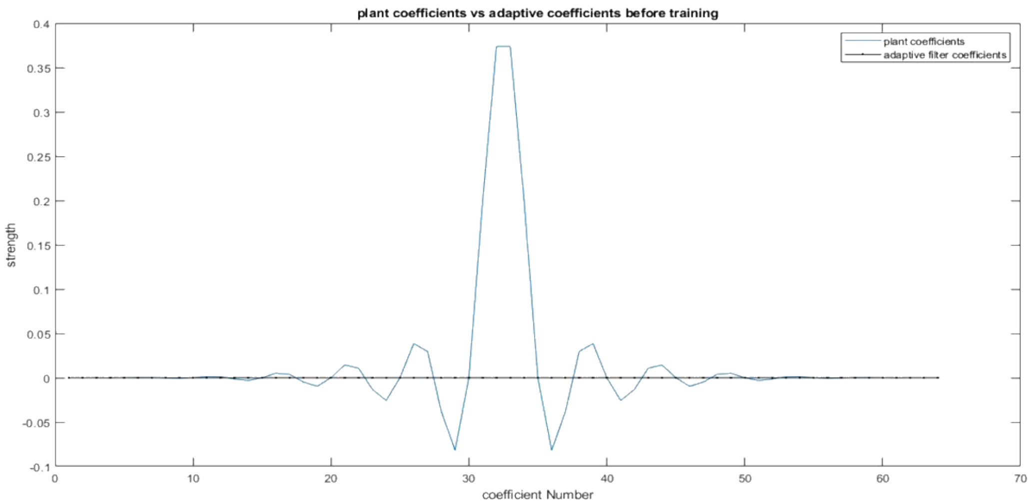

FIGURE 5.6: Comparison of Plant weights and adaptive filter weights before training in sample based processing

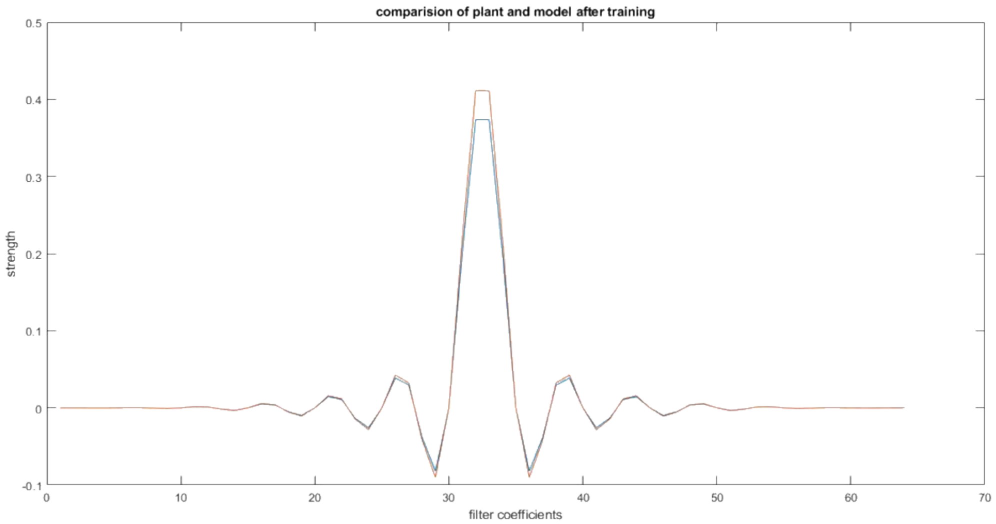

FIGURE 5.7: Comparison of Plant weights and adaptive Filter weights after training in sample based processing

The table 5.2 shows the comparison of results between the sample based processing and the block based processing. The length of the input signal (in samples) considered by the algorithm to meet a specified error condition is referred as the length of input to identify plant in this table.

Parameter	Sample Based processing	Block based processing		
		Block size:16	Block size:32	Block size:64
Length of input signal considered to identify system (in samples)	6300 samples	2477 blocks (almost 40000 samples)	1264 blocks (almost 40000 samples)	638 blocks (almost 40000 samples)

TABLE 5.2: Results obtained for system identification using adaptive algorithms

Conclusion: From the table 5.2, we observed that the sample based processing works faster in comparison to the block based processing. Therefore, only the sample based processing is implemented in the Noise Cancellation algorithms, whose results are shown in the next sections.

5.3 Noise Cancellation for non-real time signals

For comparing the performance of two algorithms, a few parameters like input signal, plant coefficients and secondary path coefficients were fixed for both the algorithms and used in the implementation of the algorithms.

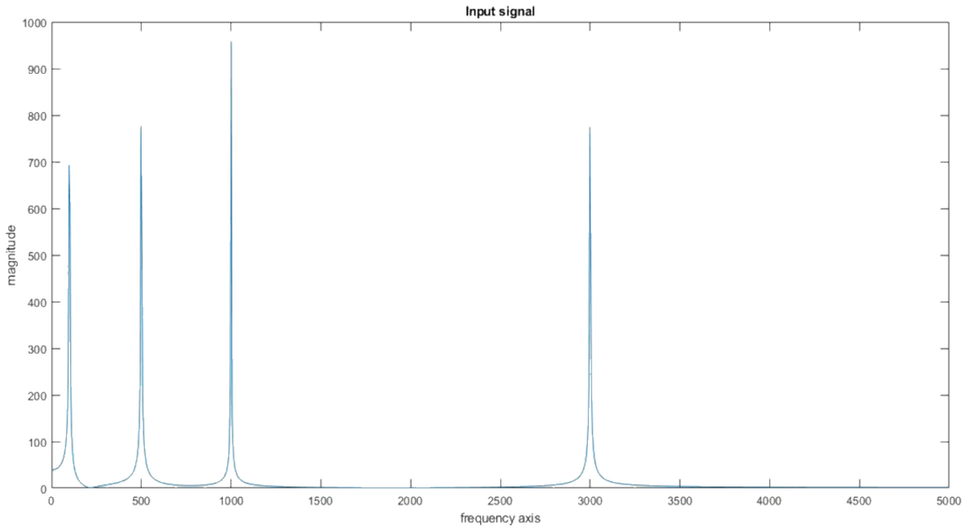

FIGURE 5.8: Frequency domain representation of Input signal

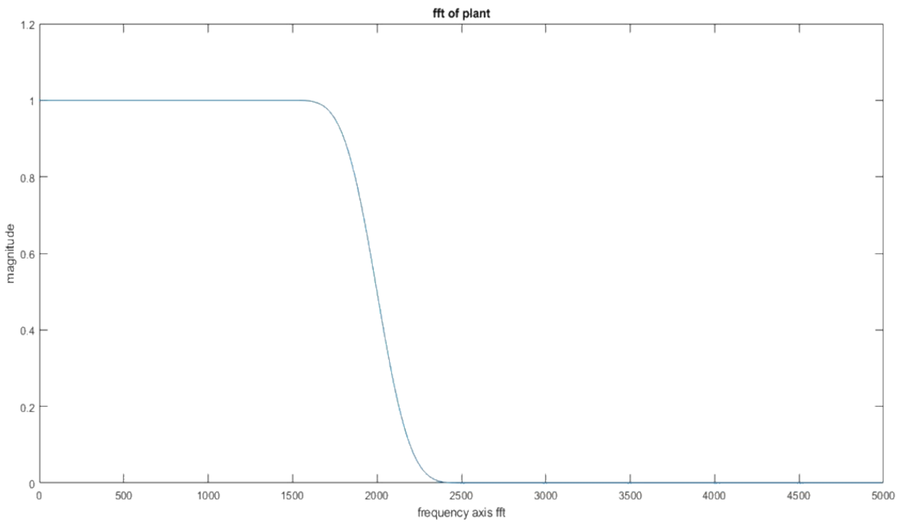

FIGURE 5.9: Frequency domain representation of plant

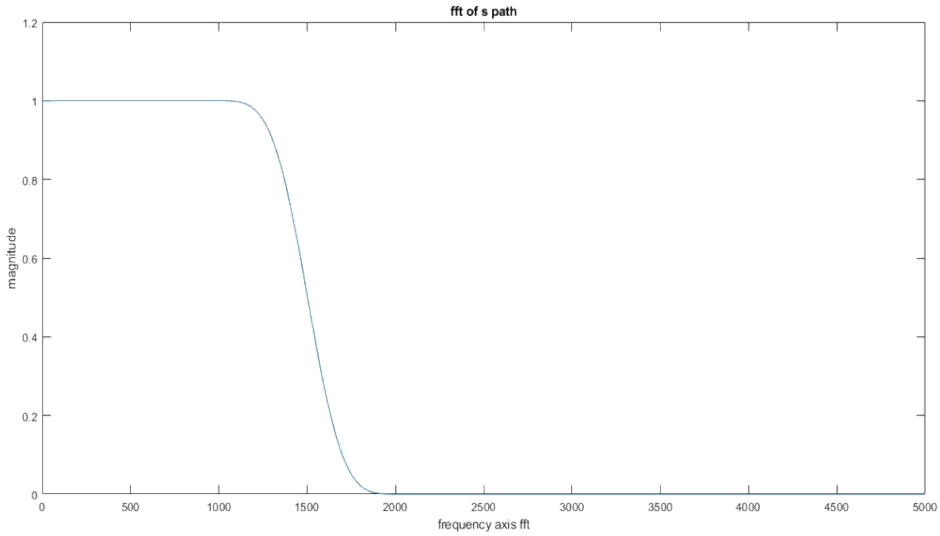

FIGURE 5.10: Frequency domain representation of secondary path

(i) Least Mean Square algorithm:

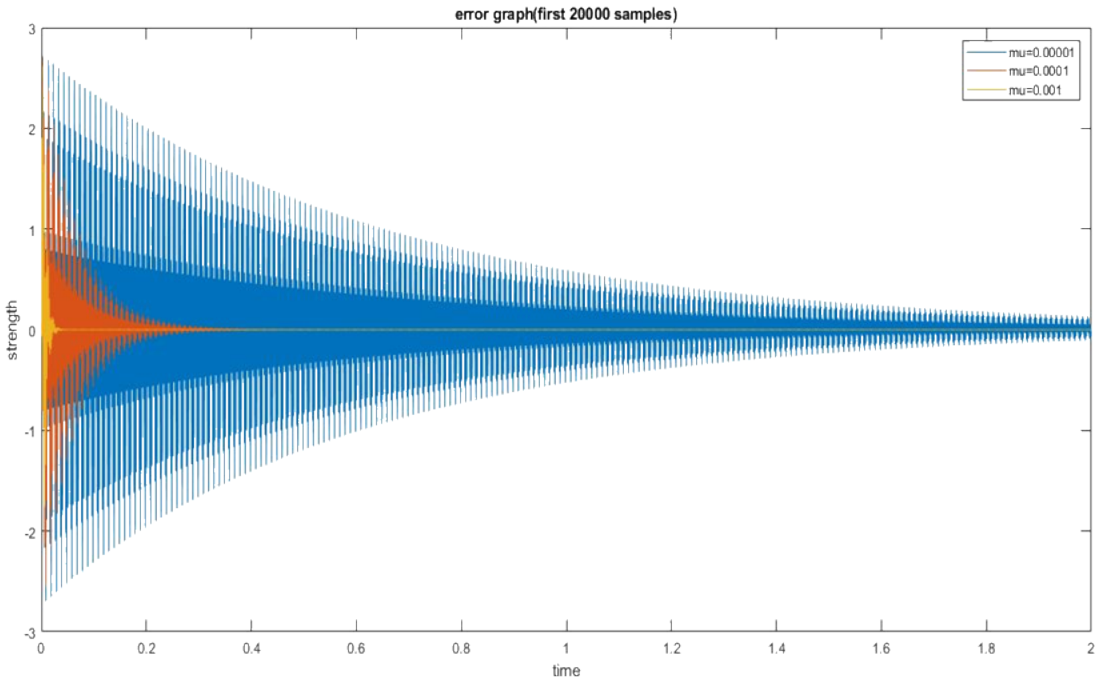

FIGURE 5.11: Comparison of error signal with step-sizes according to LMS algorithm

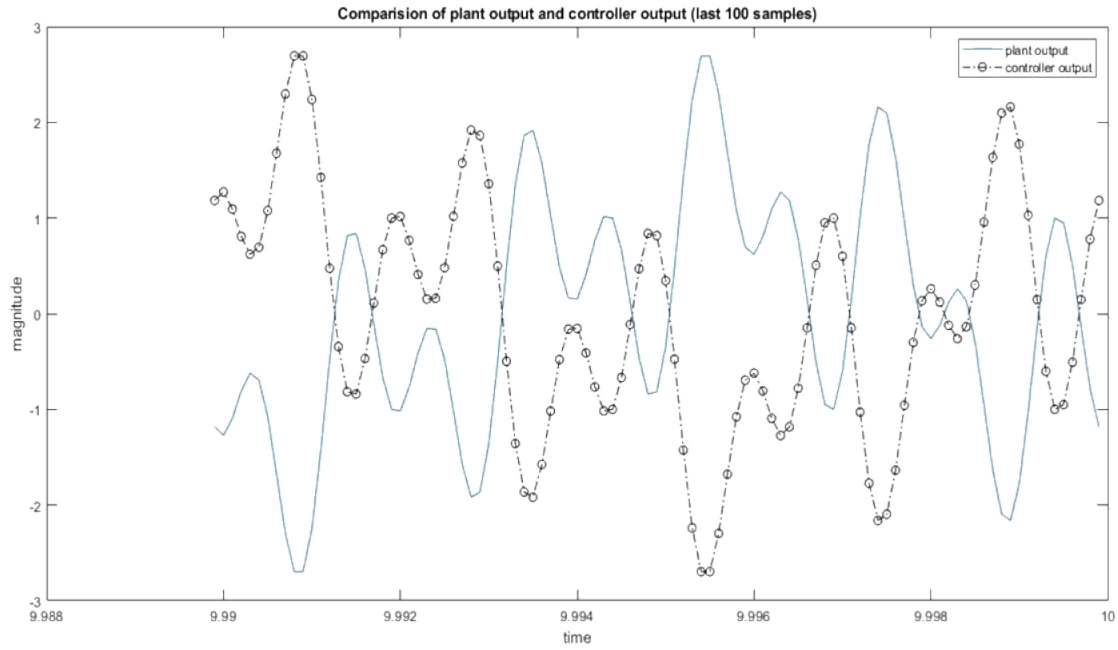

FIGURE 5.12: Comparison of plant output and controller output in LMS algorithm

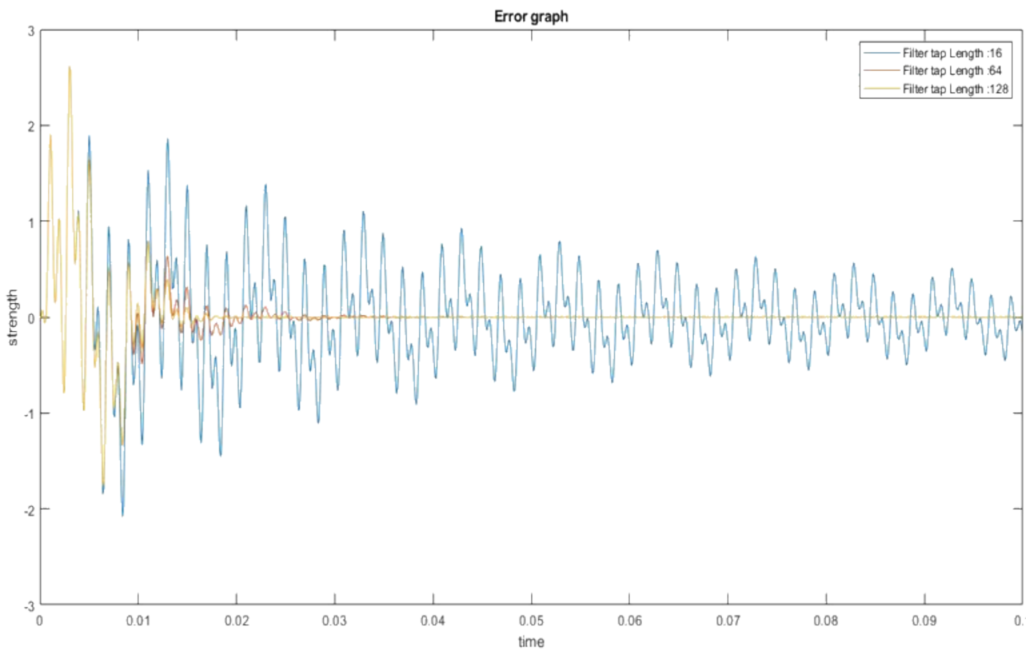

FIGURE 5.13: Comparison of LMS algorithm performance with different number of filter weights

(ii) Recursive Least Squares algorithm:

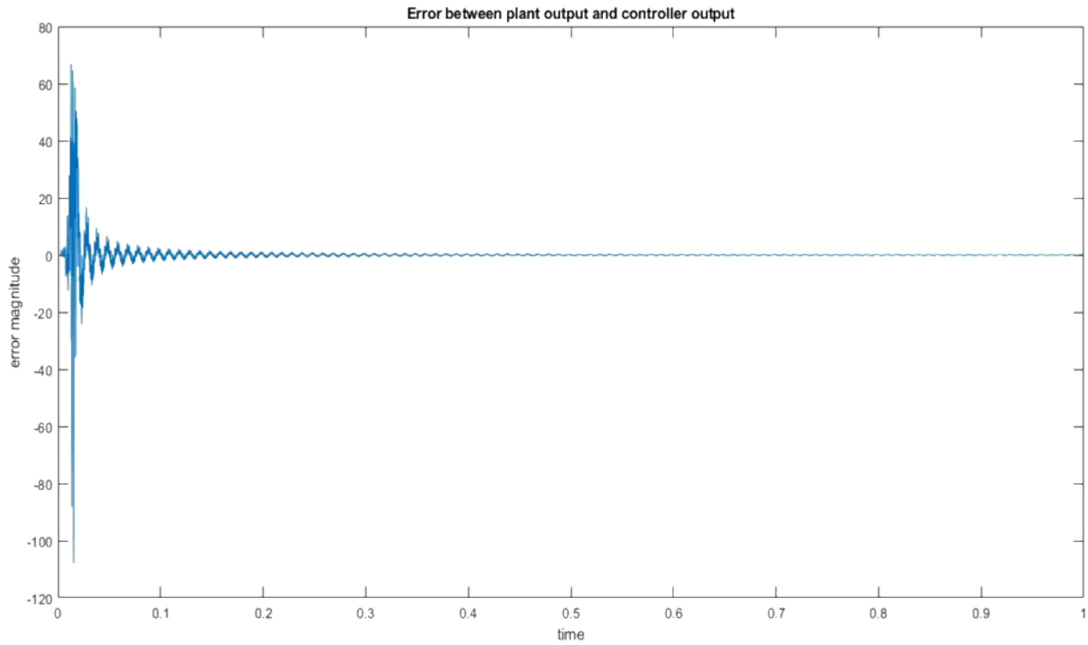

FIGURE 5.14: Error graph in RLS algorithm

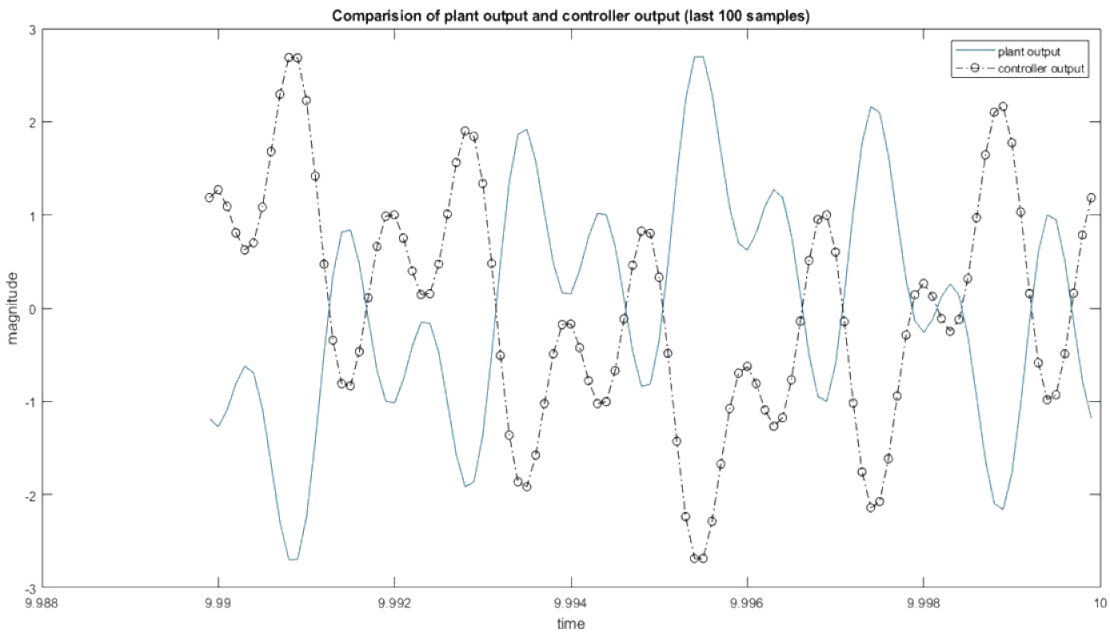

FIGURE 5.15: Comparison of plant output and controller output in RLS algorithm

5.4 Noise cancellation for real time signals

5.4.1 Noise coming from fan motor

Sampling frequency:2000 Hz

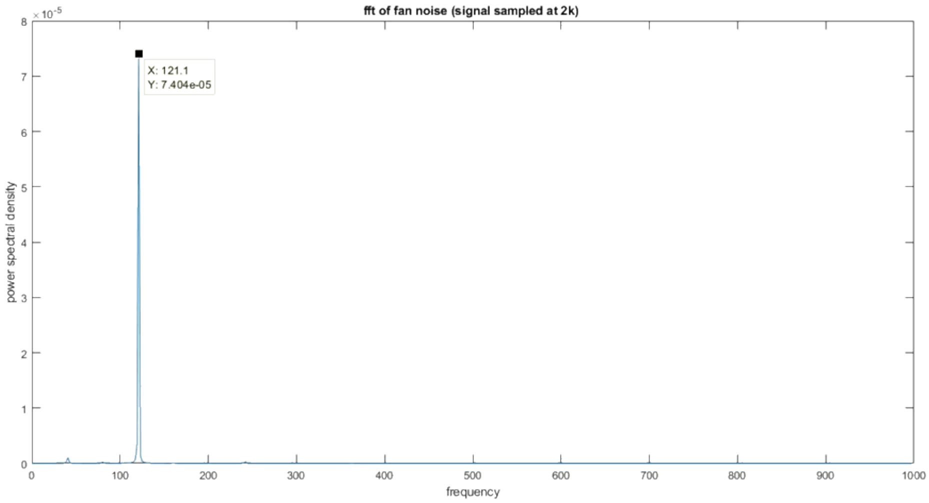

FIGURE 5.16: Frequency domain representation of fan noise

(i) Least Mean Square algorithm

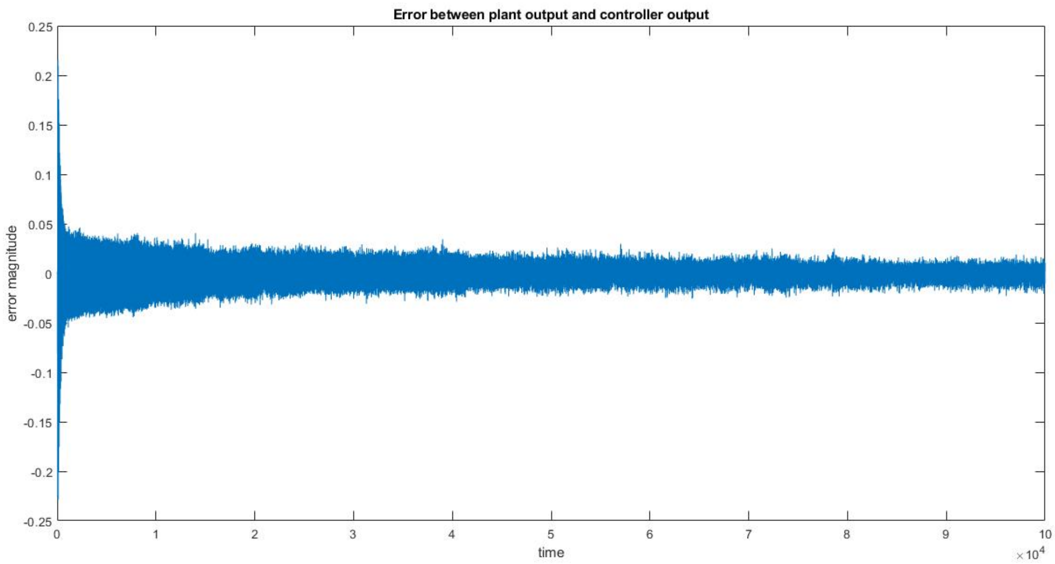

FIGURE 5.17: Error graph obtained in LMS algorithm with input as fan noise

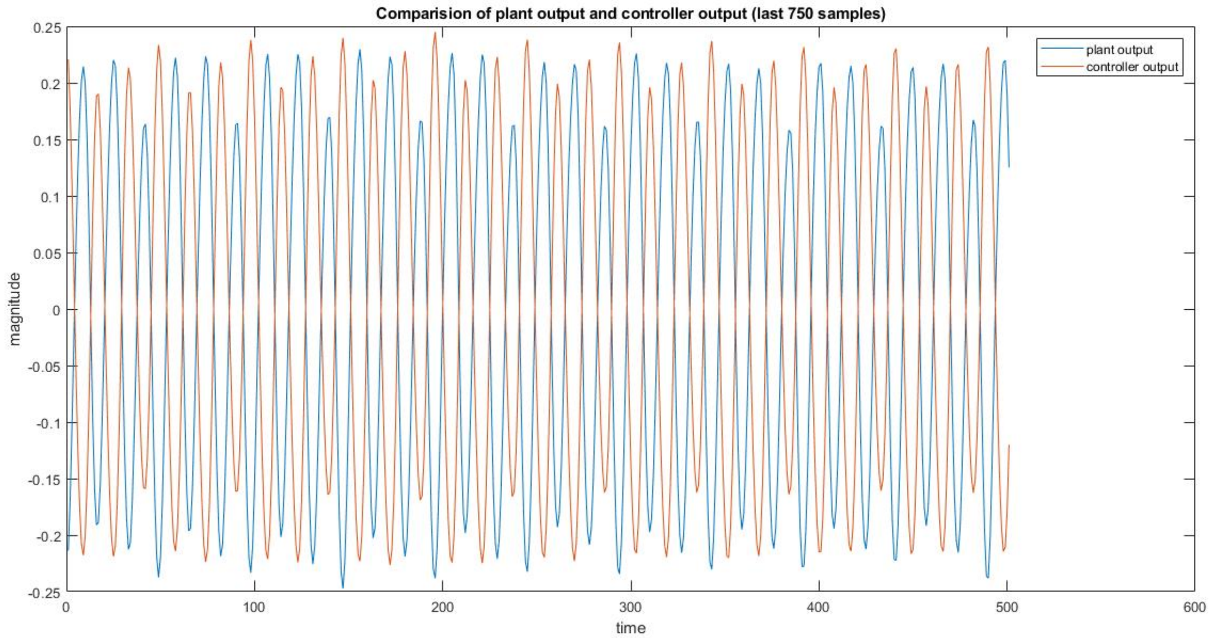

FIGURE 5.18: Comparison of plant output and controller output LMS algorithm with input as fan noise

(ii) Recursive Least Square algorithm

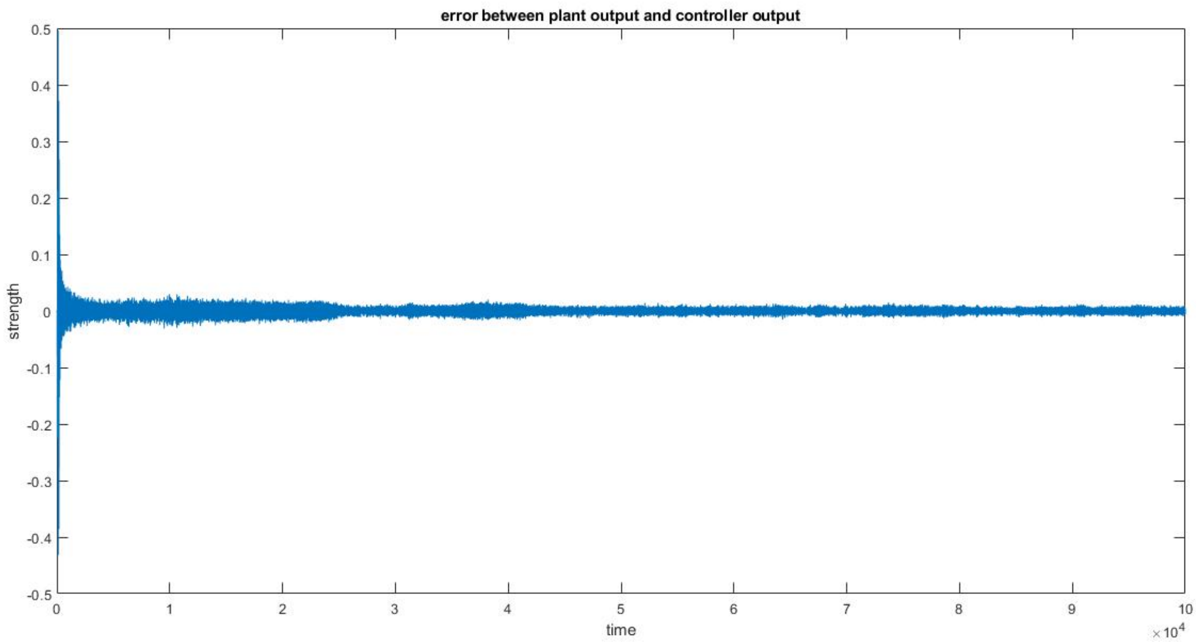

FIGURE 5.19: Error graph obtained in RLS algorithm with input as fan noise

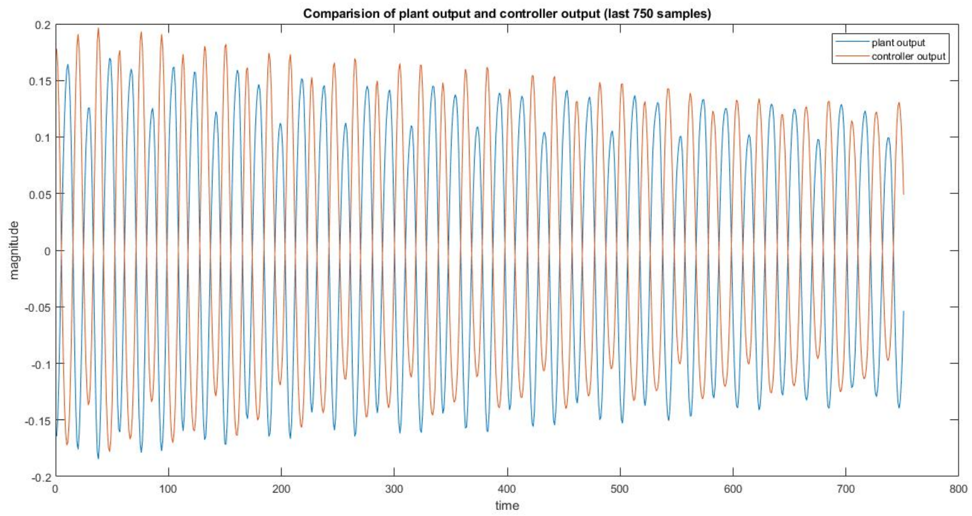

FIGURE 5.20: comparison of plant output and controller output using RLS algorithm with input as fan noise

5.5 Noise Cancellation (considering interfering signals)

Parameters chosen:

1. Input to the Plant: 100Hz, 300Hz, 500 Hz, 750 Hz, 1000 Hz, 3000 Hz

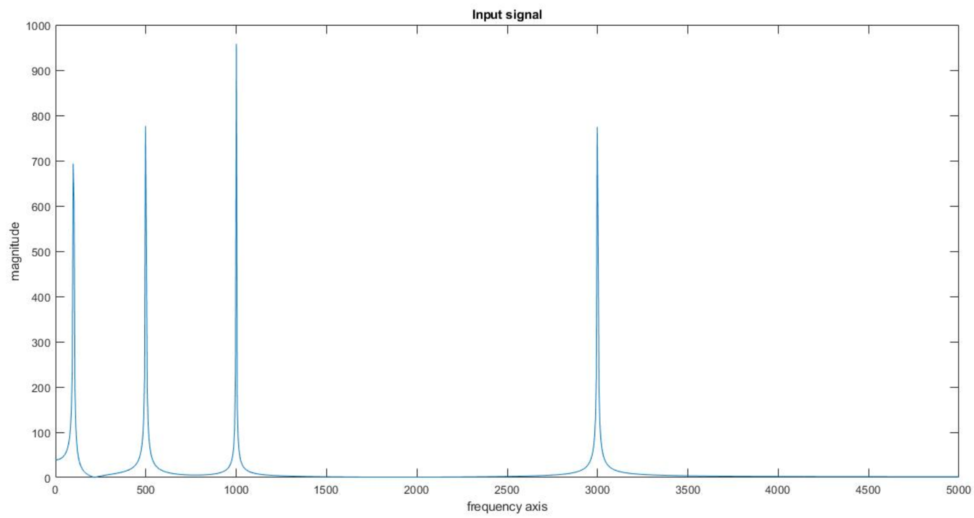

FIGURE 5.21: Frequency domain representation of input signal

2. Input to the ANC System: 100Hz, 500 Hz, 1000 Hz, 3000 Hz

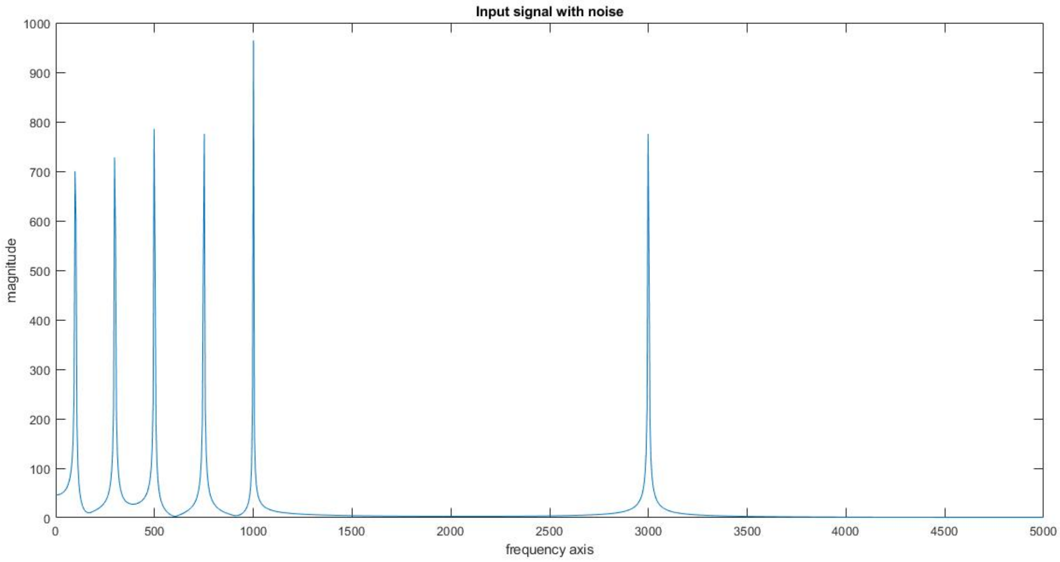

FIGURE 5.22: Frequency domain representation of input signal with interfering signals

3. Error spectrum indicating that 300 Hz and 750 Hz remain uncancelled

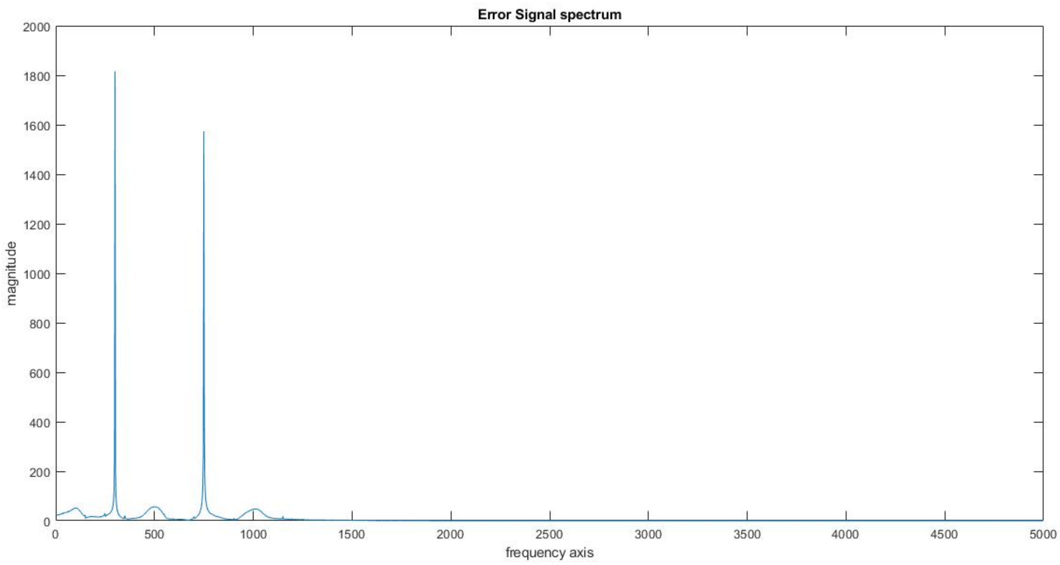

FIGURE 5.23: Frequency domain representation of error signal

Chapter 6

Conclusions and Future Work

6.1 Conclusions

1. We observed that sample based processing works fast than block based processing while identifying an unknown system.
2. We observed that at software level, the Recursive Least square algorithm takes more time than the Least Mean Square algorithm to meet expectations.
3. We observed that Recursive Least Square algorithms only work when the lambda value is between 0.999 to 1.
4. At software level, adaptive algorithms are accurately generating anti-noise wave forms.

6.2 Future work

1. Test the performance of Least Mean Squares and Recursive Least Squares algorithms with engine noise recordings that are non-stationary.
2. Identify the limitation of these algorithms, with respect to, how rapidly they can track changes in the noise.
3. Understand the implementation of these algorithms on SHARC DSP platform.

Bibliography

[1] Cost function, Gradient descent and Univariate linear regressions. URL https://bit.ly/2xc0Jp6.

[2] Information regarding STM32 board and several other development tools is gathered from https://www.keil.com/download/file/.

[3] S.J.Elliott and P.A.Nelson. Active Noise Control, Oct 1993. ISSN 1053-5888.

[4] Sen Maw Kuo and Dennis R.Morgan. Active Noise Control: a tutorial review. 1999.

[5] Elena Punskaya. Design of FIR filters. URL https://bit.ly/2CcUPXv.

[6] Bernard Widrow, John R Glover, John M McCool, John Kaunitz, Charles S Williams, Robert H Hearn, James R Zeidler, JR Eugene Dong, and Robert C Goodlin. Adaptive Noise cancelling: Principles and applications. volume 63, pages 1692–1716. IEEE, 1975.

[7] Vijay k.Madisetti & Douglas B.Williams. Digital Signal Processing fundamentals. CRC Press, 2010.

[8] Machine Learning fundementals: Cost functions and Gradient descent. URL https://bit.ly/2VQwdeD.

[9] Sajjad Ahmed Ghauri and Muhammed Farhan Sohail. System Identification using LMS, NLMS and RLS. 2013.

[10] Reza Hashemian. Design of an Active Noise Control System using combinations of DSP and FPGAs. 1996.

[11] Jeffrey Denenber. Quieting the Environment: Noise Cancellation. 2005.

[12] A.V.Oppenheim, E.Weinstein, K. C.Zangi, M.Feder, and D.Gauger. Single-Sensor Active Noise Cancellation. volume 2, pages 285–290, April 1994. doi: 10.1109/89.279277.

[13] Donald Reay. Digital Signal Processing and applications using the ARM Cortex M4 by. 2016.

[14] Saurabh R. Prasad and Bhalchandra B.Godbole. Optimization of LMS algorithm for System Identification. *CoRR*, abs/1706.00897, 2017. URL http://arxiv.org/abs/1706.00897.

[15] Danilo Mandic. Adaptive Estimation and Filtering.