`

PREPROCESSING METHOD FOR ASSISTIVE LISTENING SYSTEM

MUHAMMAD NUR ADRI BIN NAWI

BACHELOR OF ENGINEERING TECHNOLOGY (ELECTRICAL) WITH HONS

UNIVERSITI MALAYSIA PAHANG

`

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : MUHAMMAD NUR ADRI BIN NAWI

Date of Birth :

Title : PREPROCESSING METHOD FOR ASSISTIVE LISTENING SYSTEM

Academic Session : 2020/2021

I declare that this thesis is classified as:

☐	CONFIDENTIAL	(Contains confidential information under the Official Secret Act 1997)*
☐	RESTRICTED	(Contains restricted information as specified by the organization where research was done)*
❒	OPEN ACCESS	I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

(Student's Signature) (Supervisor's Signature)

New IC/Passport Number
Date:

Name of Supervisor
Date:

`

SUPERVISOR'S DECLARATION

We hereby declare that we have checked this thesis and in our opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Bachelor of Engineering Technology (Electrical) with Hons.

____

(Supervisor's Signature)

Full Name : TS. DR. ROSHAHLIZA BT M RAMLI

Position : SENIOR LECTURER & HEAD OF PROGRAM (ELECTRICAL), FACULTY OF ELECTRICAL AND ELECTRONIC ENGINEERING TECHNOLOGY, UNIVERSITI MALAYSIA PAHANG

Date :

`

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries in which have been duly acknowledged. The thesis has not been accepted for any degree and is not concurrently submitted for award of other degree.

____

(Student's Signature)

Full Name : MUHAMMAD NUR ADRI BIN NAWI

ID Number : TB17095

Date :

`

PREPROCESSING METHOD FOR ASSISTIVE LISTENING SYSTEM

MUHAMMAD NUR ADRI BIN NAWI

Thesis submitted in fulfillment of the requirements for the award of the Bachelor of Engineering Technology (Electrical) with Hons

Faculty of Electrical and Electronic Engineering Technology

UNIVERSITI MALAYSIA PAHANG

FEBRUARY 2021

ACKNOWLEDGEMENTS

Alhamdulillah. I am expressing my gratitude to Allah SWT for blessing, love, opportunity, health, and mercy to complete senior design project 2 alongside this undergraduate thesis. This undergraduate thesis entitled “PREPROCESSING METHOD FOR ASSISTIVE LISTENING SYSTEM” is submitted as one of the final requirements in accomplishing an undergraduate degree at Universiti Malaysia Pahang.

In completing this project, lots of people have provided advice, motivation, and support to me. I take this opportunity to express my gratitude and appreciation to everyone. First, the deepest appreciation goes to my beloved parents for their endless love, prayer, support, and a phone call every week i to keep going and never give up.

I also present my sincere appreciation goes to Ts. Dr. Roshahliza M. Ramli as my supervisor that without her help, support and patience, this thesis would not have been possible.
She also supervises, giving advice and guidance from the very early stage of this project throughout the past few months.

I am also very grateful to my sponsor Kerajaan Negeri Sabah for Provided me with the necessary financial support during my 4 years of study. My sincere thanks are also passed to all my lecturers, especially Dr. Waheb Abdul Jabbar, my advisor, and always takes care of his advisee. Also, friends and others also provided advice and help to me. At the same time, the advice received from everyone is also critical to this project’s success.

Finally, I want to thank everyone important to the successful completion of this paper. This undergraduate thesis is far from perfect, but I hope it will be useful not only to me but also to readers. Therefore, constructive and thoughtful suggestions and critics are welcome.

ABSTRAK

Untuk mengurangkan kebisingan, terdapat beberapa jenis kaedah yang ada, dan praktik yang paling biasa adalah menggunakan penutup telinga di industri dan alat pendengaran bagi orang yang mengalami masalah pendengaran. Dengan kemajuan teknologi, Pembatalan kebisingan diperkenalkan ke berbagai jenis perangkat, dari musik hingga industri otomotif, dan ini kerana, pada masa ini, jumlah industri meningkat dengan permintaan pengguna mendorong pengeluar untuk memasang peralatan atau mesin baru untuk membolehkan mereka beroperasi 24 jam. Ini menyebabkan pencemaran bunyi ke kawasan sekitarnya, sehingga penduduk tempatan menyuarakan keluhan. Lebih-lebih lagi, Pekerja memerlukan pekerja selama 8 jam terus dan mengalami kebisingan di kilang. Sekiranya ini berlanjutan, ini akan menyebabkan kehilangan pendengaran dan merupakan sumber gangguan deria pada manusia yang sangat kerap. Untuk mengatasi ini, *Assistive Listeninig System* (ALS) telah dibangunkan. Malangnya, penyaring adaptif yang digunakan di ALS saat ini tidak dapat menerima tahap kebisingan secara langsung, dengan menerima tahap kebisingan yang tinggi akan menyebabkan algoritma sistem rosak. fungsi ALS adalah untuk melindungi pekerja dari tahap kebisingan yang tinggi sehingga menyebabkan kehilangan pendengaran. Jadi, algoritma sistem adalah salah satu bahagian yang paling penting dalam ALS. Untuk mengatasinya, penapis diperlukan dan ia dikenali sebagai *Low Pass Filter* (LPF) adalah penapis yang melepasi isyarat dengan frekuensi yang lebih rendah daripada frekuensi pemotongan yang dipilih dan melemahkan isyarat yang lebih tinggi daripada frekuensi pemotongan dan sistem ini hanya memerlukan frekuensi yang lebih rendah untuk masuk sistem. Projek ini bertujuan untuk mengkaji bagaimana mengelakkan penapis adaptif daripada rosak oleh tahap kebisingan yang tinggi. secara khusus, selidiki dan cadangkan 4 penapis lulus rendah pada peringkat pra-pemprosesan ALS. Sebagai tambahan, nilailah saringan yang disarankan dalam peringkat pra-proses ALS dan pilih yang sesuai. Selain itu, penelitian telah dibuat dalam tinjauan literatur untuk menunjukkan bahawa saringan yang dipilih memiliki kemampuan untuk melakukan tugas lain dan menunjukkan fungsi tersebut di bidang lain. Selain daripada itu, tesis ini akan menunjukkan persamaan bagaimana penapis dibuat dan menunjukkan kod MATLAB untuk setiap penapis. Untuk maklumat, semua saringan yang digunakan dalam projek ini sudah dirancang oleh jurutera di MATLAB dan digunakan dalam projek ini untuk menguji yang paling sesuai untuk sistem ini. Hasilnya

`

mungkin kelihatan serupa kerana ia adalah *low pass filter* dan bukan pembatalan kebisingan itu sendiri. Hasilnya menunjukkan semua saringan low pass berpotensi untuk digunakan dan melindungi sistem dari rosak.

ABSTRACK

To reduce noise, there a several types of methods available, and the most common practice is using earmuff in industry and hearing aid for people who have hearing loss. With the advancement of technology, noise Cancelling was introduce to various kind of devices, from music to the automotive industry, and this is because, at present, the number of industries greatly increase with the demand for consumers pushing manufacturer to install new equipment or machine to enable them to operate 24 hours. It leads to noise pollution to surrounding neighborhoods to an extent, local voice a complaint. Moreover, Workers needed to the worker for 8 hours straight and endured the noise in the factory. If this continues, it would lead to hearing loss and indeed a very frequent source of sensory impairment in humans. To overcome this Assistive Listening System (ALS) was developed. Unfortunately, the adaptive filter used in ALS currently cannot accept a directly high level of noise. Accepting a high level of noise would lead to the system algorithm being corrupted. The ALS function is to protect workers from a high level of noise that leads to hearing loss. So, the system algorithm is one of the most important parts of the ALS. To overcome this, the filter is needed, and it is well known as a low pass filter (LPF) is a filter that passes signals with frequencies lower than the selected cutoff frequency and attenuates signals higher than the cutoff frequency, and this system needs only lower frequency to enter the system. This project aims to study how to avoid the adaptive filter from corrupted by a high level of noise. Specifically, investigate and suggest 4 low pass filters in the preprocessing stage of ALS. In addition, evaluate the recommended filter in the preprocessing stages of ALS and choose the suitable one. Besides that, research has been made in the literature review to show that the chosen filter has the capability to perform other task and shows it function in other fields. Other than that, this thesis will show the equation on how the filter was made and show the MATLAB code for each filter. As for information, all the filters used in this project are already designed by engineers in MATLAB and used in this project to test the most suitable one for the system. The result may look similar because it is a low pass filter and not the noise cancellation itself. The results show all low pass filters have the potential to be used and protect the system from being corrupted.

`

TABLE OF CONTENT

`

`

`

LIST OF FIGURES

`

LIST OF SYMBOLS

GHz	Gigahertz
MHz	Megahertz
dB	Decibel
V	Voltage
Rp	Peak-to-peak passband ripple
n	Filter Order
μ	Mu
Ω	Omega
Hz	Hertz
∞	Infinity
Wp	Passband edge frequency
Wn	Cutoff frequency

LIST OF ABBREVIATIONS

ALS	Assistive Listening System
LPF	Low Pass Frequency
WHO	World Health Organization
ABC	Artificial Bee Colony
PSO	Particle Swarm Optimization
ACO	Ant Colony Optimization
DSP	Digital Signal Processing
ADC	Analog-To-Digital Converter
SNR	Signal-To-Noise Ratio
FIR	Finite Impulse Response
SLFOR	Sidelobe Fall-Off Rate
WAV	Waveform Audio File

CHAPTER 1

INTRODUCTION

1.1 Project Background

To reduce noise, there a several types of methods available, and the most common practice is using earmuff in industry and hearing aid for people who have hearing loss. With the advancement of technology, noise Cancelling was introduce to various kind of devices, from music to the automotive industry, and this is because, at present, the number of industries greatly increase with the demand for consumers pushing manufacturer to install new equipment or machine to enable them to operate 24 hours. It leads to noise pollution to surrounding neighbourhoods to an extent, local voice a complaint. Moreover, workers need to work for 8 hours straight and endured the noise in the factory. If this continues, it would lead to hearing loss and a very frequent source of sensory impairment in humans (Pradhan et al., 2017).

The World Health Organization (WHO) describes disability as an issue in operation or form of the body that leads to limitations on action or involvement. Hearing disability was also defined as an individual's understanding of his/her hearing disabilities, impacting his/her life, family relations, and social/psychological status (Le Prell & Clavier, 2017). Hearing deficiency is a critical concern affecting the life quality of millions of people worldwide. The study shows that around 15 percent of adults in America (37.5 million) alone experience hearing difficulties.

Noise loudness is measured in decibels. Noise exposure ranges from person to person, but researchers agree that it will create road noise when the noise level is more than 85 decibels, which will impair ears. The risk of hearing loss increases as the sound gets louder. Time for exposure is also essential.

Figure 1.1 Automotive industry

Figure 1 shows the automotive industry mainly equipped with lots of automatic robotic machine to mass-produce car, with many robots working, can be done quickly but it will produce a high level of noise that can risk worker health. In the music industry, some of the music devices such as headphones already build in noise cancellation in them such as in figure 1.2, this headphone can make them clearly hear the music without noise from the surrounding. Unfortunately, it is expensive.

Figure 1.2 Sony headphone

Research also has been made to reduce the noise focusing in the industry where workers more expose to high dB noise by using a hearing aid. Unfortunately, the hearing aid may help one hear sounds, but it will not filter/ eliminate background sound. In order to tackle these problems from arising more further also with the limitation of the earmuff which it only blocks noise and makes the user difficult to communicate. Therefore, the Assistive Listening System was developed which is to reduce a high level of noise and protect workers from hearing loss.

With ALS, it could help millions of people especially for those who work in the manufacturing or other industry that requires communication form section to section while sitting in an area which produces a high level of noise, and communication can be improved because noise is being eliminated and speech can be heard clearly. In addition, this could help workers from miscommunication and avoid danger fall onto them. Besides that, ALS could improve life quality for the worker by preventing them from hearing loss.

However, the ALS also have a limitation which it's adaptive algorithm cannot directly receive high level of noise into the system which is why this project "Preprocessing Method For Assistive Listening System" was made to filter the noise signal before entering the adaptive algorithm by using basic low pass filter.

1.2 Problem Statements

A Quiet workplace is almost impossible to find, especially for those working in manufacturing. Therefore, it is very important to have a place to work that doesn't have a high level of noise so that workers can be more focus and miscommunication won't occur. The traditional way to block the noise in the industry requires earmuff. Therefore, earmuff will help workers to protect their hearing from high levels of noise.

Using earmuff not block the noise, but workers also cannot hear when someone is having a conversation with them. This could lead to miscommunication, interpretation and even danger to the workers when they don't hear the alarm. In addition, there is not suitable equipment that works perfectly as an earmuff to block noise. Besides that, earmuff is practically and being used in all kind of industry from automotive to aviation. The improvement has been made for the earmuff such as has a microphone in it so workers can communicate, but this also causes a problem because when using this type of device, they need to turn up the volume high in order to hear very clearly than the noise.

Hence, Assistive Listening System was made to tackle this problem by filtering using its adaptive filter. The adaptive filter has a really important task to perform in the ALS. Currently, the ALS doesn't have a way to directly filter noise signals above 85dB. Therefore, a direct entry of a high level of noise to the adaptive filter could corrupt the adaptive filter and makes the ALS not performing as it should. To avoid such problem from occurring, a preprocessing method needed to be applied in the preprocessing stages and act as gate for high level of noise before entering the adaptive filter.

1.3 Objectives

This project has objectives as below:

1. To avoid the adaptive filter from being corrupted by a high level of noise
2. To investigate and suggest filtering in the preprocessing stage of ALS
1. To evaluate the recommended filter in the preprocessing stages of ALS and choose the suitable one for ALS.

1.4 Scope

The scope of this project is to perform a low pass filter for high level of noise in the preprocessing stage. The filter that will be used is Butterworth, Chebyshev, Hamming, and Hanning. All these filters have a different equation to make different of its characteristic method of filtering. In addition, the low pass filters will be evaluated and then choose the suitable one for ALS

1.5 Organization of Thesis

This thesis summarizes the result of research about noise canceling algorithm on Assistive Listening System using an adaptive filter. The outlines of each chapter are as follows. Chapter 1, introduction. This chapter aims to give an overview of the project and grasp a basic understanding of preprocessing used in ALS and its importance. Other than that, in this chapter also talks about problem arises in our system if there is a high level of noise entering the system. In addition, the reader will know the objective that needed to be achieved in this chapter.

Chapter 2, literature review. This chapter provides a review on low pass filters used and what area these filters being used for. It is a very brief review of the conference paper and also journals from a recent publication. These reviews will also explain where there are using the filter for.

Chapter 3, methodology. This chapter will briefly tell about the introduction of the methodology used in the system. Besides that, it also explains about block diagram and flow chart of the system used. Other than that, what kind of input is also stated here and description of preprocessing in the ALS. This chapter also explains what filter is used with the description and formula of the filter.

Chapter 4, result and discussion. This chapter provides the result from the preprocessing method with a different type of filter used. Each result will get a different waveform, and we can conclude which one is the most suitable one for ALS.

And Lastly, chapter 5 conclusion. This is the last chapter for this thesis and concludes the objective of this project. After that, the limitation of this project will explain here some suggested improvements for this project.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In order to choose a suitable preprocessing for the system, research has been done by searching academic conferences and academic journals related to the project. Preprocessing was one of the initial steps in signal processing and is widely used in signal processing from image to audio and etc. There a lots of filtering methods available, from high to low pass filters. A low-pass filter (LPF) is a filter that passes signals whose frequencies are lower than the selected cutoff frequency and attenuates signals whose frequencies are higher than the cutoff frequency. The frequency response of the filter depended on the design of the filter.

2.2 Butterworth Filter in ABC, BA-ACO and NB-IoT Implementation

One of the most recent algorithms inspired by the intelligent nature of honey bees is the Artificial Bee Colony (ABC). (Fadloullah et al., 2017) investigates the optimum sizing of the Low Pass Butterworth filter with high optimization in short processing time using the ABC algorithm and test ABC algorithm efficiency for Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). The idea of algorithm for artificial Bee Colony was proposed by D. Karaboga and B. Basturk in 2007, and was inspired by honeybees' intellectual behavior by General control parameters such as colony size, and the maximum number of cycles are also used, ABC is an optimization approach using a population-based searching method to find the so-called food locations that artificial bees change the population during periods, and the aim of the bee is to locate food source positions with the higher quantity of nectar, which correlates to the chosen solution(Fadloullah et al., 2017). The method proposed for a filter is the frequency response of the approximation function of the Butterworth filters is referred to as a maximum flat response since the pass band is built to have a frequency response that is as flat as mathematically possible from 0 Hz to the cut-off frequency without ripples at -3d (Fadloullah et al., 2017).

Other than that, there are also similar study conducted in 2019. The research presents a BAcktracking Ant Colony Optimization (BA-ACO technique) optimization algorithm for dealing with the active analog filter architecture by (Benhala, 2019). It is based on indirect contact within a colony of simple agents, called (artificial) ants, inspired by the normal action of ants in discovering the shortest distance between their nests and food sources also the data on successful routes by a chemical agent called pheromone that accumulates and evaporates for short routes for long routes (Benhala, 2019).

There is also paper on application of Butterworth filter used in semiconductor industry in 2018. For NB-IoT implementations, (Kang et al., 2018) introduce a third-order Active-RC filter and expect that The Internet of Things (IoT) is known to be the semiconductor industry's next development driver and About 50 billion connected devices to the "The Internet of Things" in 2020 . (Bardyn et al., 2016) stated that NB-IoT is a modern protocol for enabling narrow-band IoT implementations that have been included as an essential feature of future 5G communications technology.(Kang et al., 2018) suggested third-order Butterworth low-pass filter is two phases consist of the filter. The Tow-Thomas Biquad filter is arranged after a first-order filter for better linearity.

2.3 Chebyshev Proposed in Antenna, Nano Power and S-Parameter

In the traditional way, both antenna and filter are separated, and in his paper, (He & Xu, 2016) introduce a filtering antenna with a Chebyshev 3rd-order response, the design of 3rd-order Chebyshev Bandpass Filter is constructed using a coupling matrix. An opening is then engraved on the narrow wall of the last resonator of the energy radiation filter. He continues further by showing an example of a filtering antenna with a center frequency of 9 GHz and a bandwidth of 500 MHz is shown as a demonstration. Simulated outcomes agree well with theoretical outcomes. The number of filtering antennas can be reduced to some degree compared with traditional filters and antennas (He & Xu, 2016).

(Pawarangkoon, 2019) paper presents a CMOS 4th order Chebyshev lowpass filter with nano power. The suggested circuit is based on two 2nd-order parts cascading. Six transistors and two grounded capacitors compose each portion and Simulation using 0.18-µm CMOS parameters verify the function of the suggested circuit., with the results of the simulation show that the proposed filter can work at a supply of 0.6 V and the circuit cutoff frequency can be changed from 50 Hz to 50 Hz. 300 Hz, spanning the entire ECG signal frequency spectrum (Pawarangkoon, 2019).

(Khalilabadi & Zadehgol, 2018) proposed a method for checking causality and enforcing scattering parameters (S-Parameters) that can result from computational electromagnetic devices and are often used to conduct transient electromagnetic system simulations, including antennas, (Khalilabadi & Zadehgol, 2018) in his paper suggest a procedure that uses a first-order Chebyshev filter to extract the exact truncation error expression and perform a detailed error analysis of the Hilbert transform's a numerical implementation based on frequency-normalized and magnitude-normalized absolute error plots.

2.4 Smart Irrigation System using Arduino

In 2017, signal processing tests were performed on a radar system by (Sulistyaningsih et al., 2019). In their test, digital signal processing (DSP) and analog-to-digital converter (ADC) is used inside the radar system to obtain the image detection results using filtering, sampling, and windowing methods. One of the techniques for designing a digital filter for finite impulse response (FIR) is Windowing. The window is used for the collection of the optimal frequency needed for signal analysis. Digital signal processing (DSP) incorporates two types of windows in this radar system, which are Hamming and Blackman. For both frequency ranges, the experiment is completed. Parameters that have been checked in this filter Amplitude response and signal-to-noise ratio application are (SNR). For both windows, experiment findings revealed no substantial difference (Sulistyaningsih et al., 2019).

Other than that, (Shama et al., 2017) proposed Automatic identification by digital signal processing (DSP) technique of the QRS dynamic wave of an ECG signal. The main goal of this study is to suppress sounds that conflict with the ECG signal. They stated that for the removal of power line and baseline drift sounds, the Hamming window FIR filter is used. The important part of the preprocessing is ECG signal taken from MIT/BIH is transformed into 114m.mat using the WFDB function in MATLAB and then this data is used for filtering containing power line noise and baseline wander noise using Hamming window. The hamming window answer is strong compared to the hamming window Because of the zero sidelobes present in its configuration, the design of the low pass filter is identical to the design of the bandpass filter, except the cut off frequency is different. Hamming window with BSF and LPF. The output voltage values of the BSF and LPF signal voltage values of the ECG signal also provide the impression of how the value varies as the data is filtered (Shama et al., 2017).

2.5 Convolved and Filterbank with Hanning

The convolved Hanning window is used for the FIR filter design, and the filter equipped with a convolved window is seen to exhibit higher sidelobe attenuation, and higher sidelobe falls off the intensity (Datar & Jain, 2016). The use of this filter as a low pass prototype filter in the configuration of the filter bank is also seen. It is found that the enhanced sidelobe behavior of the newly configured filter results in a substantial reduction in the filterbank aliasing error. (Datar & Jain, 2016) stated that filterbank output parameters designed using windows obtained by Hanning window self-convolution are also evaluated. Functions of the Convolution window are obtained by applying auto convolution Between window features, referred to as windows that are convoluted (CON). Such windows display low sidelobes and high sidelobe decay ratio levels. High sideband attenuation and high sidelobe drop-off rate are seen by the FIR filter constructed using convolution windows, and with the rise in the window convolution order, the sideband attenuation of the filter rises. The sidelobe fall-off rate (SLFOR) is also elevated as the filter convolution order increases. The window approach to convolution minimizes the errors caused by spectral leakage. It is found that the aliasing error indicates more declines with the rise in the number of convolutions (Datar & Jain, 2016).

2.6 Precision Water Saving Irrigation Automatic System

In this chapter, the application of four basic low pass filters is mentions in journal and conferences paper and the important role of the application of these filters in research studies. Depending on the application, these filters may be useful in other sectors too.

CHAPTER 3

METHODOLOGY

3.1 Introduction

The purpose of this chapter is to introduce the research methodology for preprocessing method regarding on what is the suitable technique to reduce the noise power generated by the noise audio itself. These approaches allowed for a deeper understanding of a few basic low pass filters and their differences, which are used in this methodology. The approaches for this study are discussed in-depth in this chapter. The research plan is including the introduction of a low pass filter, input signal, and different types of low pass filters used as primary components of this chapter.

3.2 Block Diagram and Flow Chart

Figure 3.1 shows the block diagram for the complete system from the input to the output, the part that this chapter covers only the input and the preprocessing process.

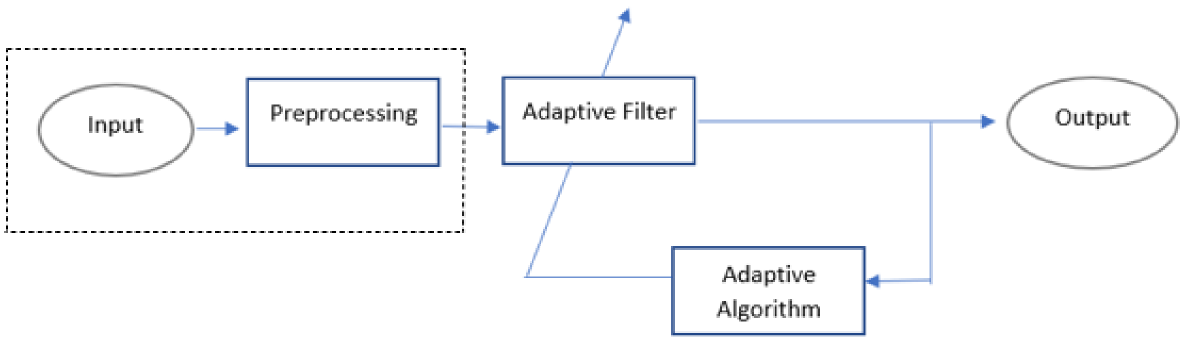


Figure 3.1 Block Diagram

Figure 3.2 shows the flow of the research methodology, starting from identifying the problem statement, which is in the introduction and as stated above it this chapter cover from start to preprocessing.

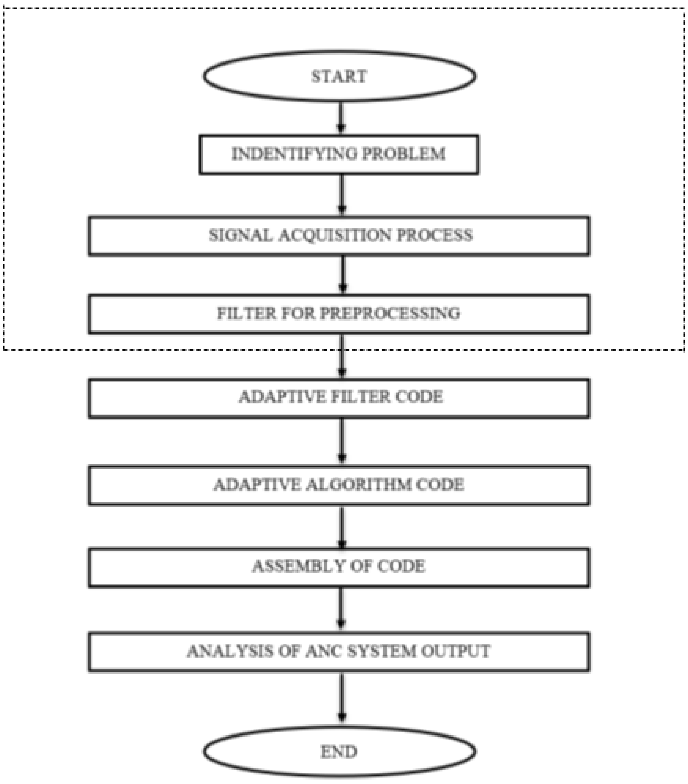


Figure 3.2 Flow Chart

3.3 Input and Signal Acquisition

The signal acquisition process is a process of physically measuring the noise power by using a sound level meter (SLM). After that, use the mobile phone to record the measured noise signal and convert the audio to Wav format, and then store audio in the MATLAB folder with preprocessing code. Before the filtering process of MATLAB software, several noise audios have been recorded, including car noise, color noise, factory noise, white noise, and people screaming. All noise audio has been modified to a waveform and WAV format with a maximum length of 1 to 3 seconds.

3.4 Preprocessing

At this stage, filtering using basic filters has been selected as the preprocessing method. The purpose of this preprocessing is to reduce the noise power generated by the noise audio itself. Without preprocessing, there is a high chance that the performance of the adaptive algorithm code will be destroyed. Butterworth, Chebyshev, Hamming window, and Hanning window low pass filter are the filter types selected for the preprocessing method. These four filters are investigated and evaluate their performance by using MATLAB software.

3.4.1 Butterworth

Butterworth filters are characterized by the property that its response to magnitude is both the passband and stopband are flat. The reaction magnitude-squared of the low pass filter of the Nth-order is given by

$$|H_a(j\Omega)|^2 = \frac{1}{1+\left(\frac{\Omega}{\Omega_c}\right)^{2N}} \qquad 3.1$$

Where N is the filter order and Ω c in rad/sec is the cutoff frequency. The magnitude-squared response plot is as follows.

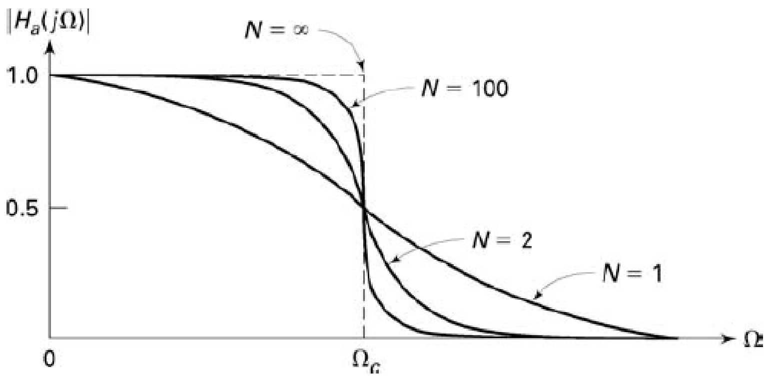


Figure 3.3

The signal At $\Omega = 0$, $|Ha(j0)|^2 = 1$ for all N and at $\Omega = \Omega c$, $|Ha(j\Omega_c)|^2 = 1/2$ for all N, which implies a 3 dB attenuation at Ωc. After That $|Ha(j\Omega)|^2$ is a monotonically decreasing function of Ω. Which $|Ha(j\Omega)|^2$ approaches an ideal low pass filter as $N \to \infty$. $|Ha(j\Omega)|^2$ is maximally flat at $\Omega = 0$ since derivatives of all orders exist and are equal to zero.

In order to evaluate the Ha(s) function of the method, we put (3.1) for receiving

$$H_a(s)H_a(-s) = |H_a(j\Omega)|^2\Big|_{\Omega=s/j} = \frac{1}{1+\left(\frac{s}{j\Omega_c}\right)^{2N}} = \frac{(j\Omega)^{2N}}{s^{2N}+(j\Omega_c)^{2N}} \quad 3.2$$

The roots of the polynomial denominator (or poles of Ha(s)Ha(−s)) from (3.2) are given by

$$p_k = (-1)^{\frac{1}{2N}}(j\Omega) = \Omega_c e^{j\frac{\pi}{2N}(2k+N+1)}, \quad k = 0, 1, \ldots, 2N-1 \quad 3.3$$

Butterworth There is an understanding of (3.3) that there are 2*N* poles of $H_a(S)H_a(-s)$, which are equally distributed on a circle of radius Ωc with the angular spacing of π/N radians and as for N odd, the poles are given by pk = $\Omega_c\, e^{jk\pi/N}$, k= 0, 1, . . . , 2N −. After that, for N even the poles are given by pk = $\Omega_c\, e^{j(\frac{\pi}{2N}+\frac{k\pi}{N})}$ and k = 0, 1, . . . , 2N – 1. With reference to the j-axis, the poles are symmetrically located, and a pole never falls on the imaginary axis and only falls on the actual axis if N is odd. As an example, 3rd- and 4th-order Butterworth filter poles are shown From Figure 3.4.

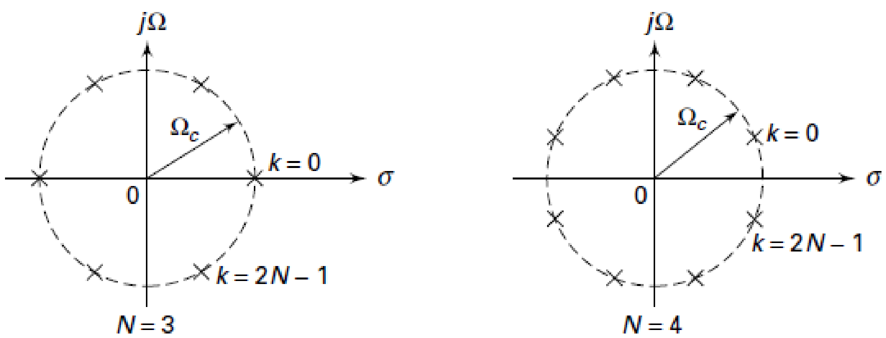


Figure 3.4 Pole plots for Butterworth filters

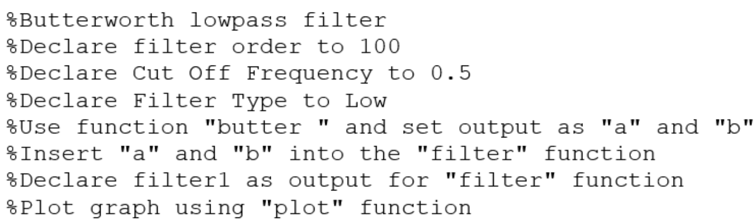


Figure 3.5 Pseudocode of Butterworth filter

3.4.2 Chebyshev

Butterworth Two kinds of Chebyshev filters are available. The filters of the Chebyshev-I have While the Chebyshev-II filters have an equiripple reaction in the passband, equiripple in the stopband reaction. Butterworth has monotonic filters in both bands, reaction we can obtain a lower-order filter by selecting a filter that has an equiripple rather than a monotone behavior. Chebyshev filters, therefore, have the same requirements in a lower order than Butterworth filters. The magnitude-squared response of a Chebyshev-I filter is

$$|H_a(j\Omega)|^2 = \frac{1}{1+\epsilon^2 T_N^2\left(\frac{\Omega}{\Omega_c}\right)} \qquad 3.4$$

Where N is the order of the filter, ϵ is the passband ripple factor, which is related to R_P, and $T_N\ (x)$ is the Nth-order Chebyshev polynomial given by

3.3.1 V

a

l

$$T_N(x) = \begin{cases} \cos\left(N\cos^{-1}(x)\right), & 0 \le x \le 1 \\ \cosh\left(\cosh^{-1}(x)\right), & 1 < x < \infty \end{cases} \quad \text{where } x = \frac{\Omega}{\Omega_c} \qquad 3.5$$

The Chebyshev filters equiripple reaction is due to this polynomial $T_N\ (x)$. It's main

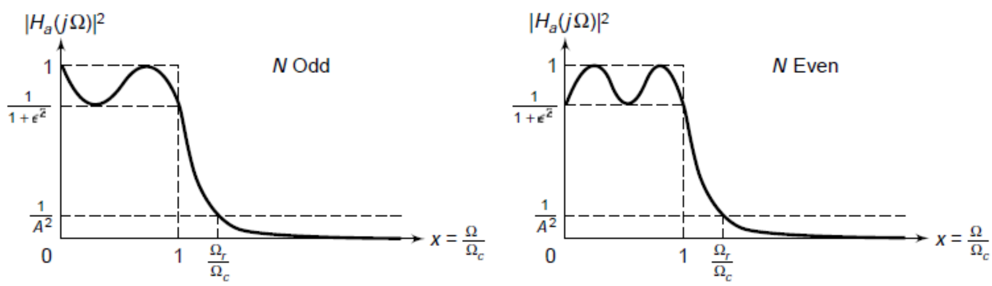


Figure 3.6 *N* Odd and *N* Even

characteristics are: (a) for $0 < x < 1$, $T_N(x)$ oscillates between $0 < x < 1$, -1 and 1, and (b) for $1 < x < \infty$, TN(x) rises to ∞ monotonically. $|Ha(j\Omega)|^2$ has two possible forms, one for N odd and one for N odd. N also as illustrated here. Remember that the normalized frequency is $x = \Omega / \Omega_c$.

From these two response plots, we observe the following properties, At $x = 0$ (or $\Omega = 0$), $|Ha(j\Omega)|^2 = 1$ for N odd and $|Ha(j\Omega)|^2 = \frac{1}{1+\epsilon^2}$ For N even. After that, at $x = 1$ (or $\Omega = \Omega_c$, $|Ha(j1)|^2 = \frac{1}{1+\epsilon^2}$ for all N. Besides that, for $0 \le x \le 1$ (or $0 \le \Omega \le \Omega_c$, $|Ha(j1)|^2$ oscillates between 1 and $\frac{1}{1+\epsilon^2}$. As for $x > 1$ (or $\Omega > \Omega_c$, $|Ha(jx)|^2$ monotonically to 0. Lastly at $x = \Omega_r$, $|Ha(j1)|^2 = \frac{1}{A^2}$. To determine the causality and stable Ha(s), we must find the poles of Ha(s) Ha(-s) and choose the poles of the left half plane of Ha(s). The pole of Ha(s) Ha(-s) is found by

$$1 + \epsilon^2 T_N^2 \left(\frac{s}{j\Omega_c} \right) \qquad 3.6$$

if it is not difficult to obtain, the solution of this equation is tedious. It can be proved that if pk = σk + jΩk, then k = 0, ..., N − 1 is the (left half plane) roots of these polynomials, then

$$\begin{aligned} \sigma_k &= (a\Omega_c)\cos\left[\frac{\pi}{2} + \frac{(2k+1)\pi}{2N}\right] \\ \Omega_k &= (b\Omega_c)\sin\left[\frac{\pi}{2} + \frac{(2k+1)\pi}{2N}\right] \end{aligned} \quad k = 0, \ldots, N-1 \qquad 3.7$$

Where,

$$a = \frac{1}{2}\left(\sqrt[N]{\alpha} - \sqrt[N]{1/\alpha}\right), \quad b = \frac{1}{2}\left(\sqrt[N]{\alpha} + \sqrt[N]{1/\alpha}\right), \quad \text{and} \quad \alpha = \frac{1}{\epsilon} + \sqrt{1 + \frac{1}{\epsilon^2}} \qquad 3.8$$

These roots fall on the ellipse of the long axis $b\Omega_r$ and the short axis $a\Omega_c$. The system function is now

$$H_a(s) = \frac{K}{\prod_k (s - p_k)} \qquad 3.9$$

where K is a normalizing factor chosen to make

$$H_a(j0) = \begin{cases} 1, & N \text{ odd} \\ \dfrac{1}{\sqrt{1+\epsilon^2}}, & N \text{ even} \end{cases} \tag{3.10}$$

```
%Chebyshev lowpass filter
%Declare filter order to 100
%Declare Cut Off Frequency to 0.5
%Declare passband edge frequency to 0.7
%Use function "cheby1" and set output as "a" and "b"
%Insert "a" and "b" into the "filter" function
%Declare filter1 as output for "filter" function
%Plot graph using "plot" function
```

Figure 3.7 Pseudocode of Chebyshev filter

3.4.3 Window

The basic idea of window design is to select a suitable ideal frequency selection filter (the filter always has an impulse response without causality and infinite duration), and then cut off (or window) its impulse response to obtain linear phase and causal FIR filter. Therefore, the focus of this method is to select a suitable window function and a suitable ideal filter. We will use H_d (ejω) to represent an ideal frequency selection filter, which has unit amplitude gain and linear phase characteristics in the passband, and zero response in its stopband. The ideal LPF with bandwidth ωc<π is given by

$$H_d(e^{j\omega}) = \begin{cases} 1 \cdot e^{-j\alpha\omega}, & |\omega| \leq \omega_c \\ 0, & \omega_c < |\omega| \leq \pi \end{cases} \tag{3.11}$$

Among them, ωc is also called cutoff frequency, and α is called sampling delay. (Please note that from the perspective of DTFT characteristics, e-jαω represents the offset in the positive n direction or the delay direction.) The impulse response of this filter has an infinite duration and is given by

$$h_d(n) = \mathcal{F}^{-1}\left[H_d(e^{j\omega})\right] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}d\omega \qquad 3.12$$

$$= \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} 1\cdot e^{-j\alpha\omega}e^{j\omega n}d\omega$$

$$= \frac{\sin\left[\omega_c(n-\alpha)\right]}{\pi(n-\alpha)}$$

Note that hd(n) is symmetric about α, which is useful for linear phase FIR filters. In order to get the FIR filter from hd(n), hd(n) must be truncated on two fulls on both sides. In order to obtain a causal linear phase FIR filter h(n) of length M, it must have

$$h(n) = \begin{cases} h_d(n), & 0 \le n \le M-1 \\ 0, & \text{elsewhere} \end{cases} \quad \text{and} \quad \alpha = \frac{M-1}{2} \qquad 3.13$$

This operation is called "window". Generally, h(n) can be considered to be formed by the product of h_d (n) and window function w(n), as shown below

$$h(n) = h_d(n)w(n) \qquad 3.14$$

Where,

$$w(n) = \begin{cases} \text{some symmetric function with respect to} \\ \alpha \text{ over } 0 \le n \le M-1 \\ 0, \text{ otherwise} \end{cases} \qquad 3.15$$

According to the way we define w(n), we get different window designs. For the Hanning window

$$w(n) = \begin{cases} 0.5\left[1-\cos\left(\frac{2\pi n}{M-1}\right)\right], & 0 \le n \le M-1 \\ 0, & \end{cases} \qquad 3.16$$

```
%set filter order to 200
%set cuttoff frequency to 0.2
%Declare filter type to LOW
%Use function window function hann
%Declare y as output for "filter" function
%Display result using wvtool
%Plot graph using "plot" function
```

Figure 3.8 Pseudocode for Hanning window

$$w(n) = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{M-1}\right), & 0 \leq n \leq M-1 \\ 0, & \end{cases} \qquad 3.17$$

```
%set SNR value from -15 to 20
%apply bandpass to the noisy signal xn
%apply bandstop to the filtered_xn
%Amplitude Spectrum k larger than 0
%Amplitude Spectrum k equal 0
%Plot graph using "plot" function
```

Figure 3.9 Pseudocode for Hamming window

3.5 Summary

The approach recommended for ALS explains in chapter 3.1. Next, 3.2 shows the block diagram and flow chart of between input and adaptive filter. Chapter 3.3 describes the input and signal acquisition on how we measure the signal. After that, chapter 3.4 describe the preprocessing method, and its subchapter explains on of the equations involves in making the filters, and it also shows the coding for each of the filters.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

The chapter shows the simulation result obtained by using MATLAB. There a four results waveforms that will be shown based on the type of filter used, and to differentiate the result, each will be put into sub-chapters to for easy recognize between low pass filters. In addition, each of the result tested by changing either the filter order, n or peak-to-peak passband ripple, Rp or passband edge frequency, Wp or cuttoff frequency and frequency constraint for Hanning, Wn depending on what the filter used and that will be told in each introduction of each subchapter.

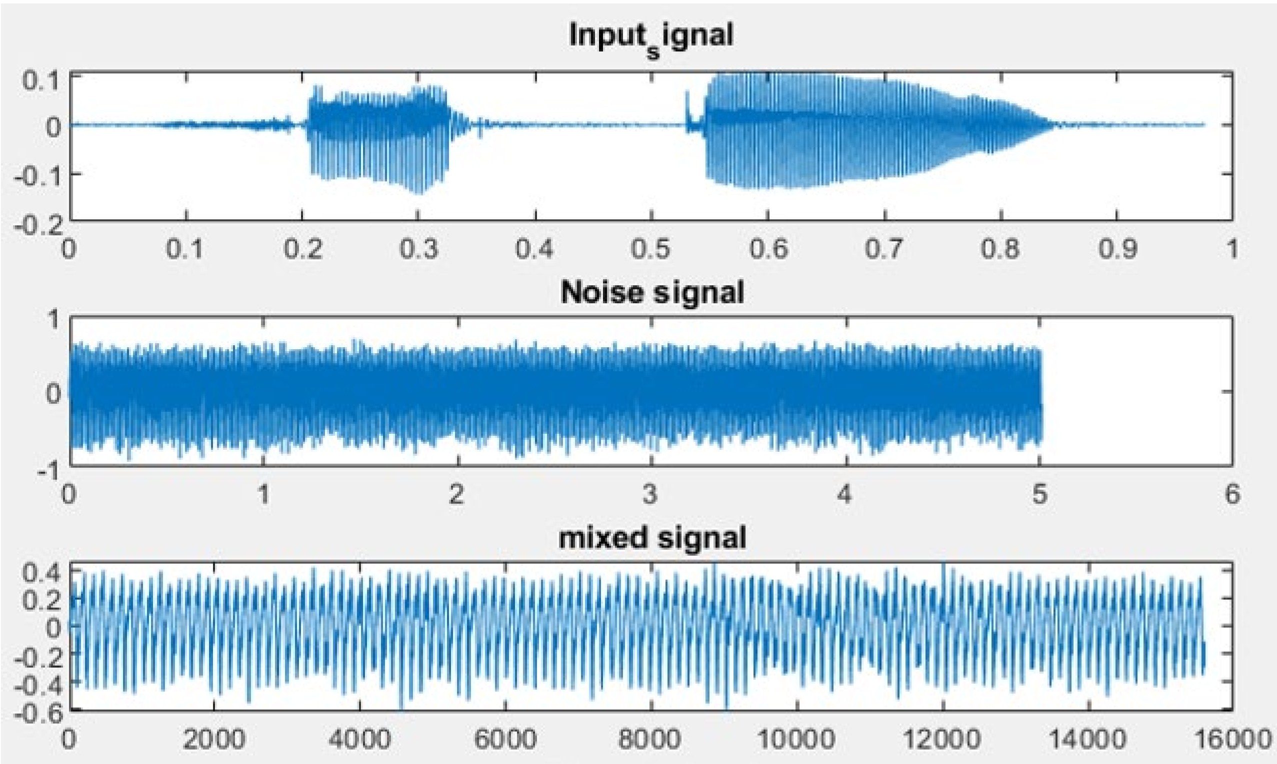
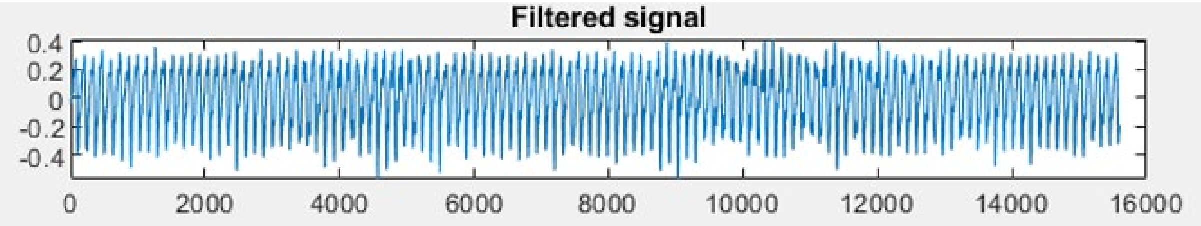


Figure 4.1 Input signal, noise, and mixed signal

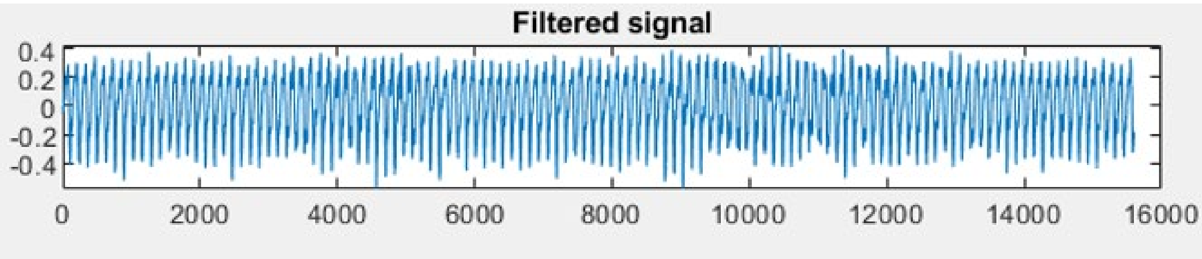
Figure 4.1 shows the input signal waveform from speech and the noise signal waveform from the car, and the third one is the mixed signal waveform, which is combinations of the input signal and noise signal, in the mixed signal where the low pass filters will perform its duty to passes signals with frequencies lower than the cutoff frequency selected and attenuates signals with frequencies greater than the cutoff frequency selected.

4.2 Butterworth

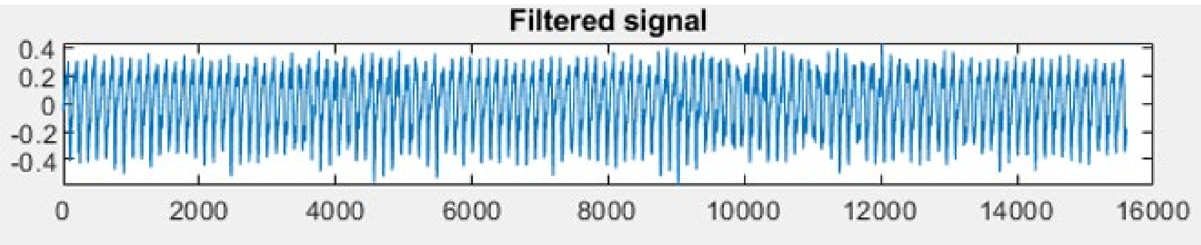
The Butterworth SNR is set to -15 and the result obtained by changing the cuttoff frequency, filter order and filter type. According to the value of ftype and the number of elements of Wn, design a low pass, high-pass, band-pass or band-stop Butterworth filter. The final bandpass and bandstop design is about 2n. the filter order will be in an increase from 10 to 100 and cutoff frequency from 0.3 to 0.6.



(a) Filter order of 10 with cutoff frequency 0.3 and filter type 'low'



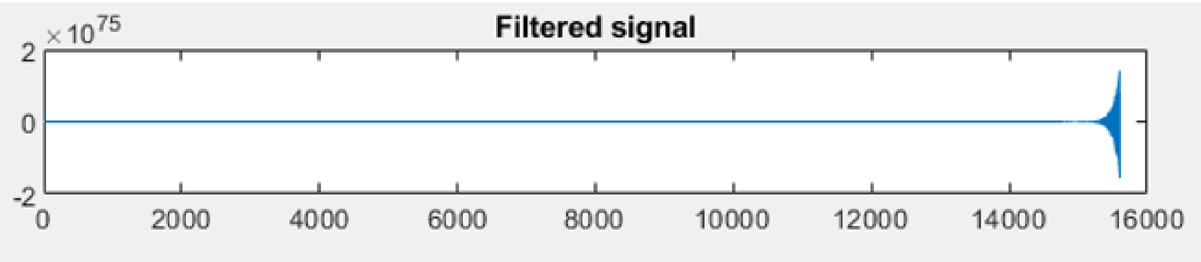
(b) Filter order of 10 with cutoff frequency 0.4 and filter type 'low'



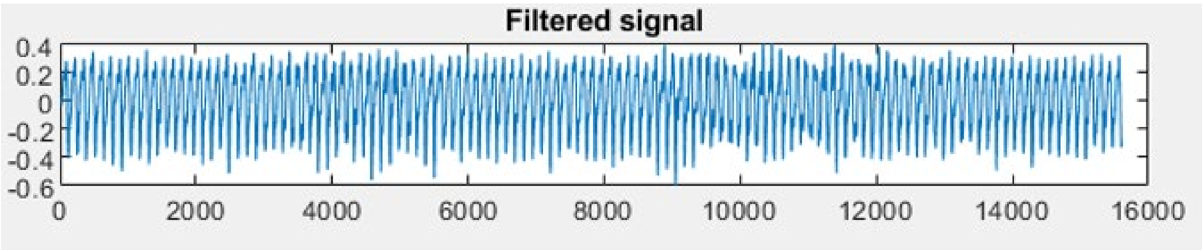
(c) Filter order of 10 with cutoff frequency 0.5 and filter type 'low'

Figure 4.2 Filter order of 10 with cutoff frequency 0.3 to 0.5 and filter type 'low'

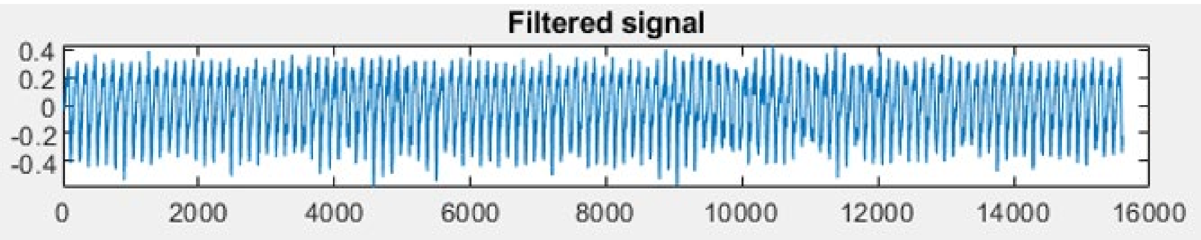
Figure 4.2 shows that the filter order of 10 is the starting filter order used in this investigation and evaluate the sound it makes through MATLAB. The filter order will gradually increase from 10 to 90. Other than that, the cuttoff in Butterworth will be different from the lowest 0.3 to 0.6 and lastly its' type 'low'. Type 'low', which is the filter is low pass filter, and that is the set by MATLAB coding for Butterworth.



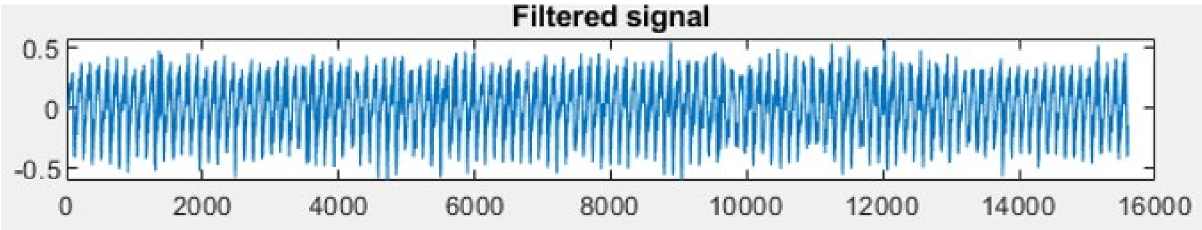
(a) Filter order of 50 with cutoff frequency 0.3 and filter type 'low'



(b) Filter order of 50 with cutoff frequency 0.4 and filter type 'low'



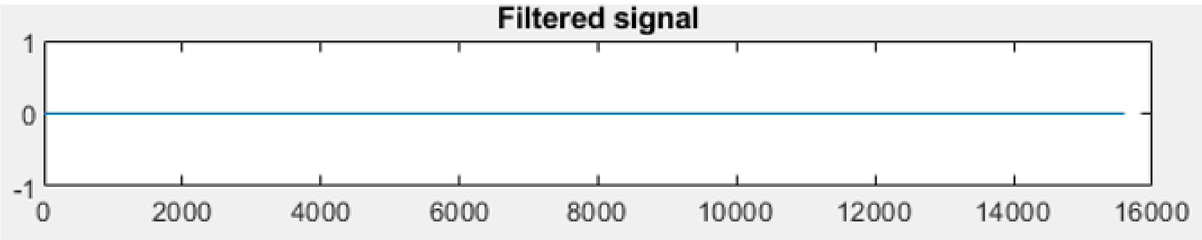
(c) Filter order of 50 with cutoff frequency 0.5 and filter type 'low'



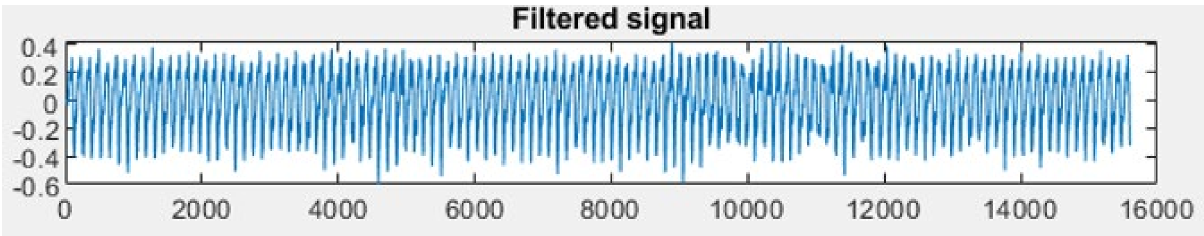
(d) Filter order of 70 with cutoff frequency 0.6 and filter type ‘low’

Figure 4.3 Filter order of 70 with cutoff frequency 0.3 to 0.6 and filter type ‘low’

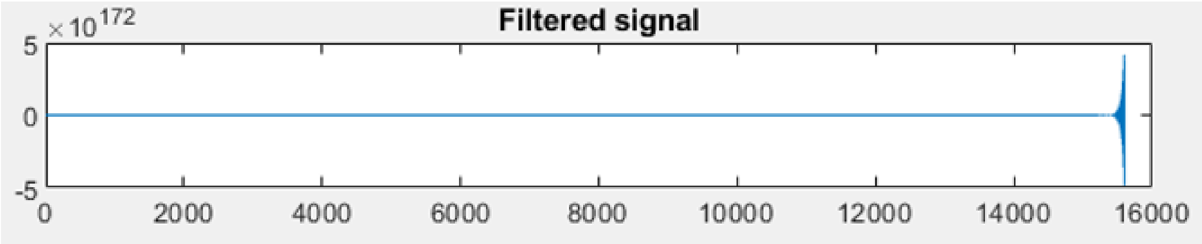
Figure 4.3 show that we jump from filter order of 10 in figure 4.2 and use a filter order of 50 in figure 4.3. From the result in figure 4.3 (a), when the filter order is higher and cutoff frequency is lower, the filtered signal will damage. This type of wave will be seen next when the filter order is higher than the cutoff frequency much lower. In addition, the sound that we heard in figure 4.3 (b), (c), and (d) still very clear.



(a) Filter order of 80 with cutoff frequency 0.3 and filter type ‘low’

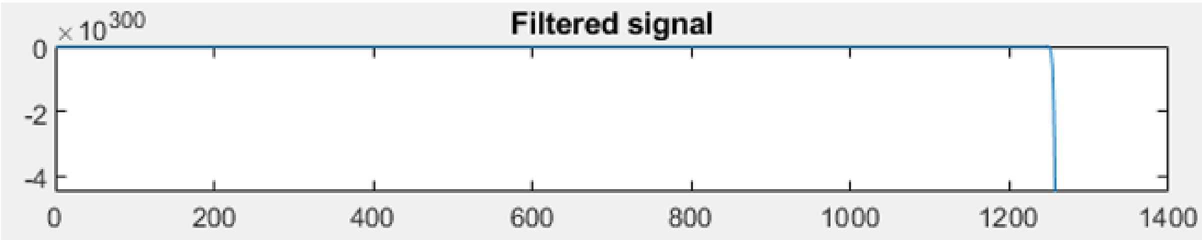


(c) Filter order of 80 with cutoff frequency 0.5 and filter type ‘low’

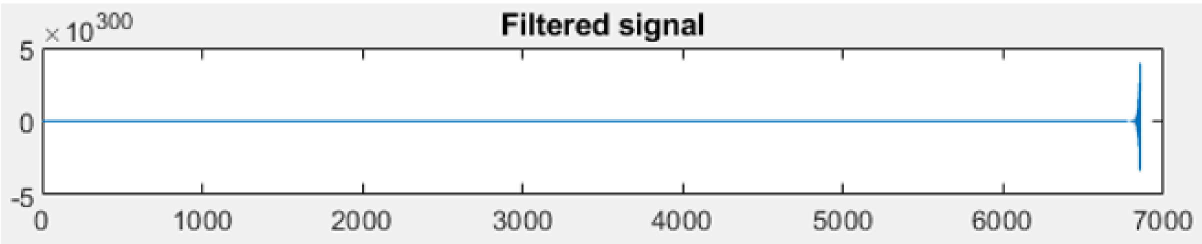


(d) Filter order of 80 with cutoff frequency 0.6 and filter type ‘low’

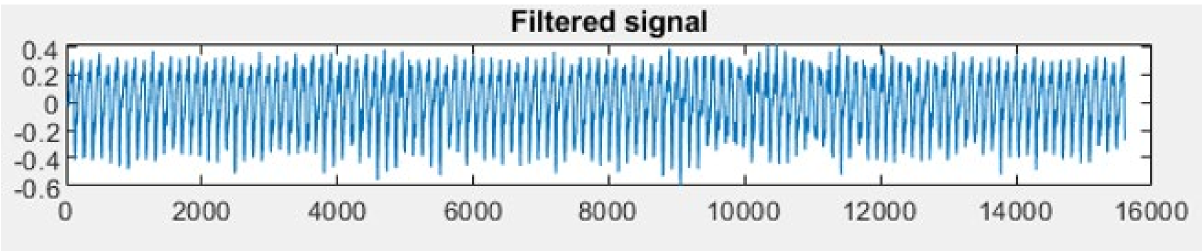
Figure 4.4 Filter order of 80 with cutoff frequency 0.3 to 0.6 and filter type ‘low’



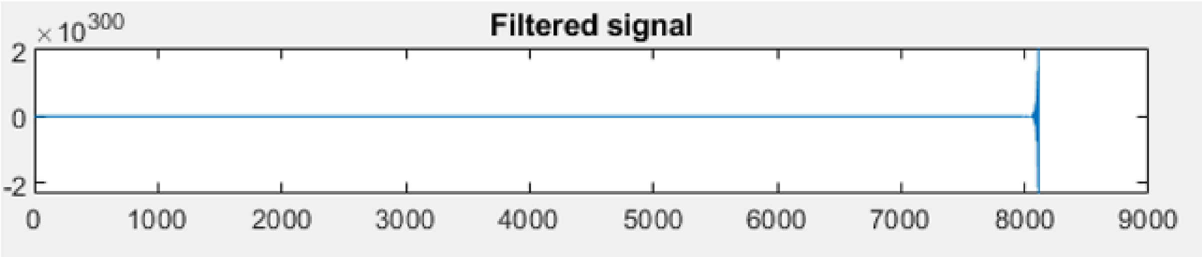
(a) Filter order of 90 with cutoff frequency 0.3 and filter type ‘low’



(b) Filter order of 90 with cutoff frequency 0.4 and filter type ‘low’



(c) Filter order of 90 with cutoff frequency 0.5 and filter type ‘low’



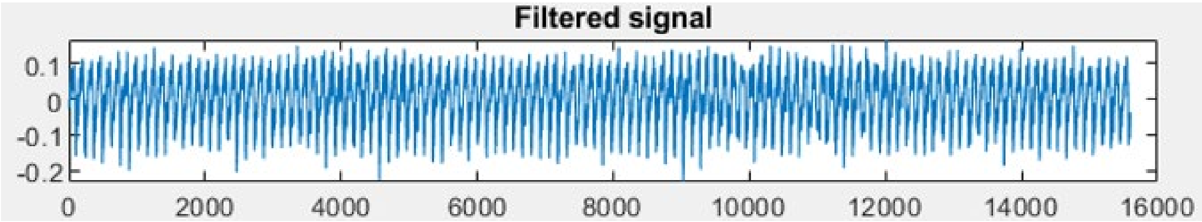
(d) Filter order of 90 with cutoff frequency 0.6 and filter type 'low'

Figure 4.5 Filter order of 90 with cutoff frequency 0.3 to 0.6 and filter type 'low'

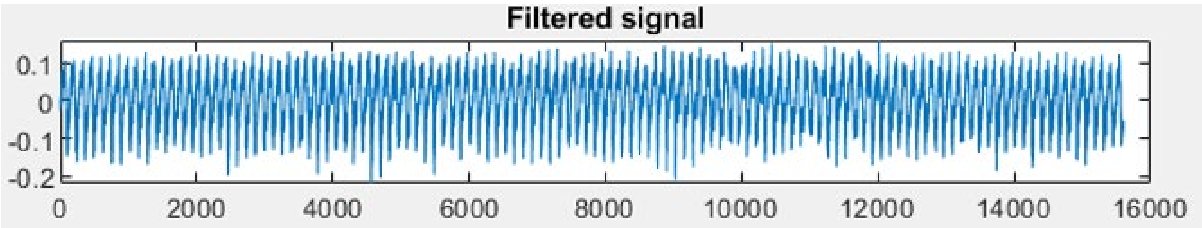
From figure 4.4 and 4.5, when the filter order getting higher closer to 1, the filtered signal will corrupt and the sound of either noise signal or input signal could be heard. Other than that, at these two figures at (c), when the cutoff frequency is at 0.5, both signal looks fine. Figure 4.4 (c) shows the simulation results using Butterworth low-pass filter. The noise used is car noise, which is first mixed with the input signal and then filtered. The maximum value of the filtered signal power is 65dB, and the average value is 20dB. Although the noise contained in the filtered signal output can still be heard, the input signal can also be clearly identified through the speaker output. The code setting of 80-order filter and 0.5 cutoff frequency is the best output we can get. Other than that, the result also shows that when the filter order is larger, and the cutoff frequency is smaller, the filtered signal will corrupt.

4.3 Chebyshev

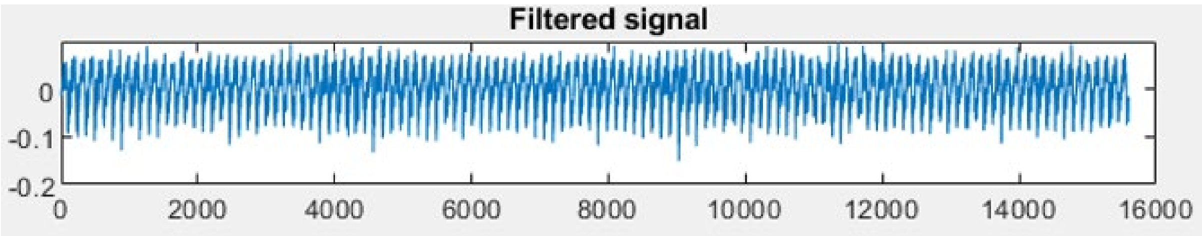
The Chebyshev is the next filter that we investigate and evaluate for the ALS. For this filter, the SNR is -15, and it is set and not changed. There are several variables that affecting the change of result obtained and that is the peak-to-peak passband ripple, filter order, and passband edge frequency.



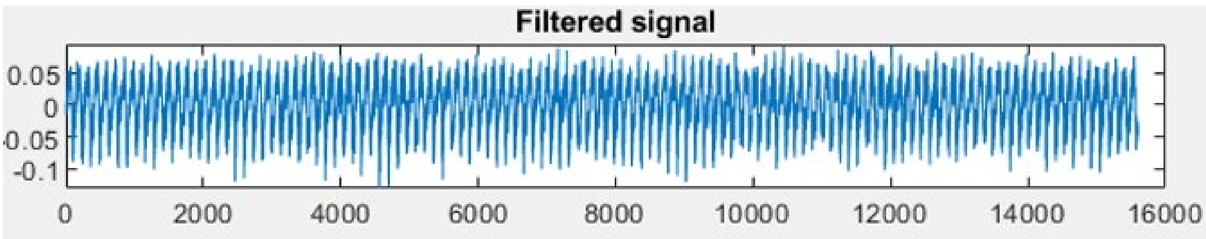
(c) Filter order of 10 with Rp of 10 and Wp of 0.6



(d) Filter order of 10 with Rp of 10 and Wp of 0.7



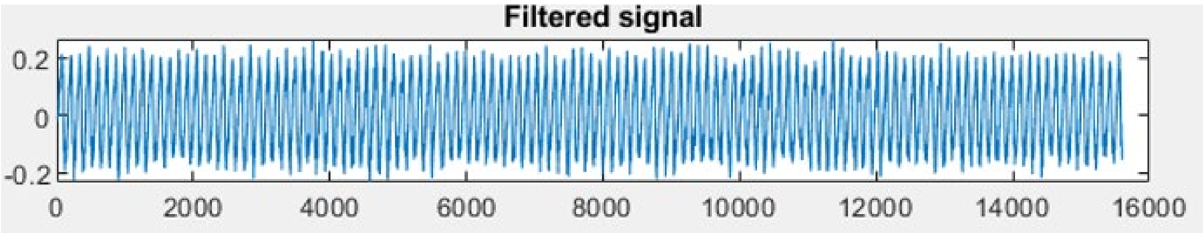
(e) Filter order of 10 with Rp of 15 and Wp of 0.6



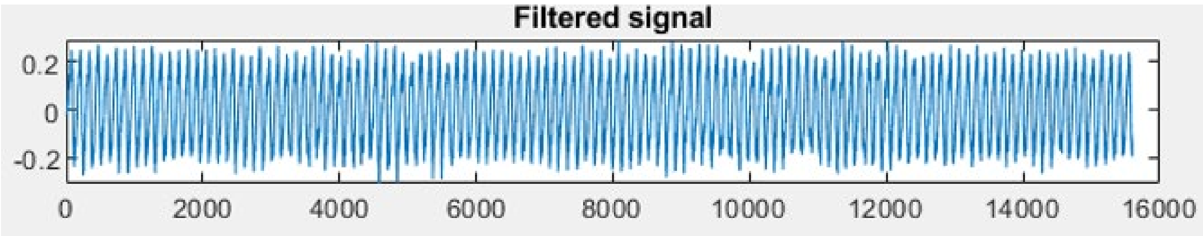
(f) Filter order of 10 with Rp of 15 and Wp of 0.7

Figure 4.6 Filter order of 10 with Rp of 1, 10, 15 and Wp of 0.6 and 0.7

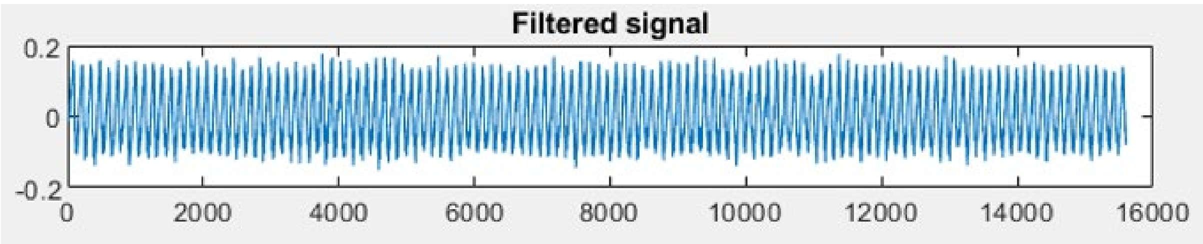
In figure 4.6, the investigation starts by using a filter order of 10, and this will gradually increase to 35 the maximum. And the Rp used from 1, 10 and 15 with Wp maintain 0.6 aad 0.7 only, and this is due to this ranger only produce the best result



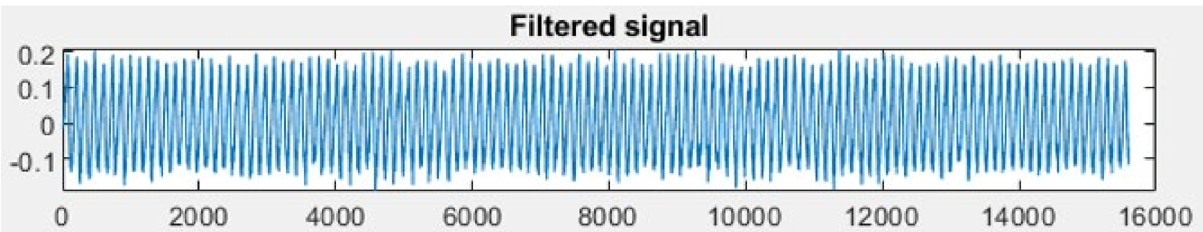
(a) Filter order of 25 with Rp of 10 and Wp of 0.6



(b) Filter order of 25 with Rp of 10 and Wp of 0.7



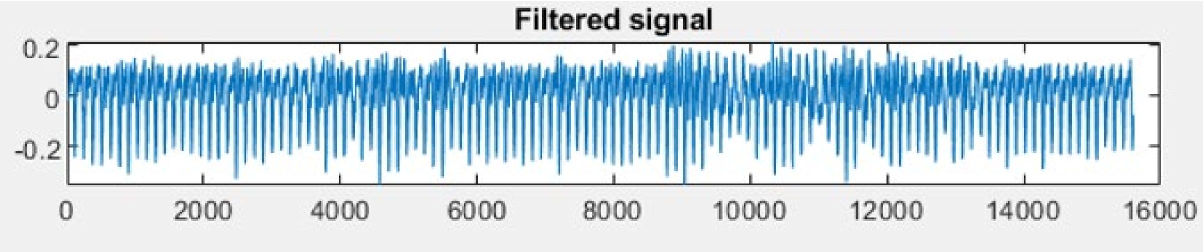
(c) Filter order of 25 with Rp of 15 and Wp of 0.6



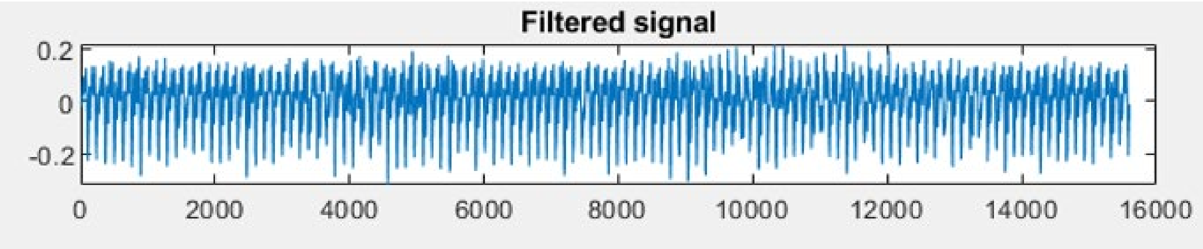
(d) Filter order of 25 with Rp of 15 and Wp of 0.7

Figure 4.7 Filter order of 25 with Rp of 1, 10 , 15 and Wp of 0.6 and 0.7

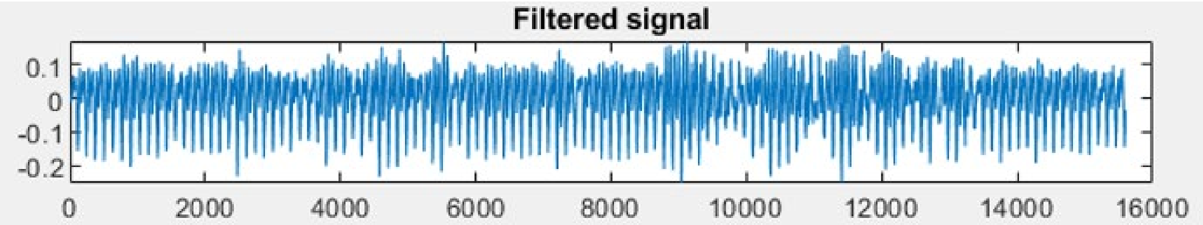
Figure 4.7 show there is not much different from the filter order 10 in the figure 4.6 and then increase it's filter order to 25. filtered signal sound played in MATLAB still could be heard deep and sink. The deep or sink sound different when using different Rp and Wn.



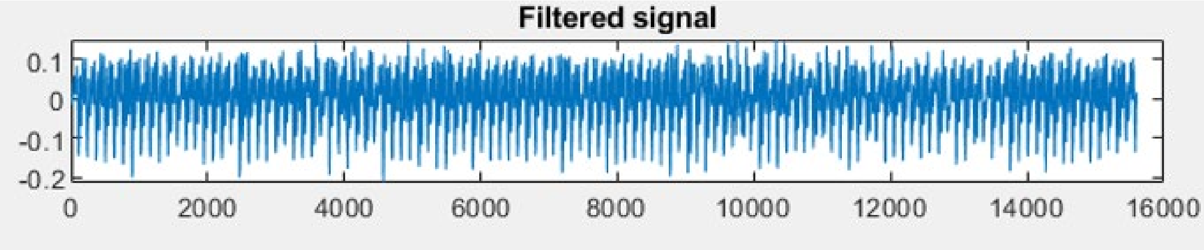
(a) Filter order of 40 with Rp of 10 and Wp of 0.6



(b) Filter order of 40 with Rp of 10 and Wp of 0.7



(c) Filter order of 40 with Rp of 15 and Wp of 0.6



(d) Filter order of 40 with Rp of 15 and Wp of 0.7

Figure 4.8 Filter order of 35 with Rp of 1, 10 , 15 and Wp of 0.6 and 0.7

Figure 4.8 (f) shows the results of simulation using Chebyshev low-pass filter. The noise used is car noise, which is first mixed with the input signal and then filtered. The maximum value of the filtered signal power is 69dB, and the average value is 24dB. The input signal can also be identified by the speaker output, but the noise from the filtered signal output can still be heard. The best output we can obtain is a code setting of filter orders of 40 with 0.7 passband frequency and 15 ripple passbands. In addition, when increasing the filter order above 40, the filtered signal could be corrupt

4.4 Hanning

The Hanning filter is the third filter that we use and it categorize ,under the windowing technique, it has the shape of one cycle of the cosine wave, and 1 is added, so it is always positive. The sampled signal value is multiplied by the Hanning function, and the result is shown in the figure. Please note that no matter what the input signal is doing, the end of the time record is forced to zero. The result obtained by SNR set to -15 and also by changing the frequency constraint and filter order from 0.2 to below 1.

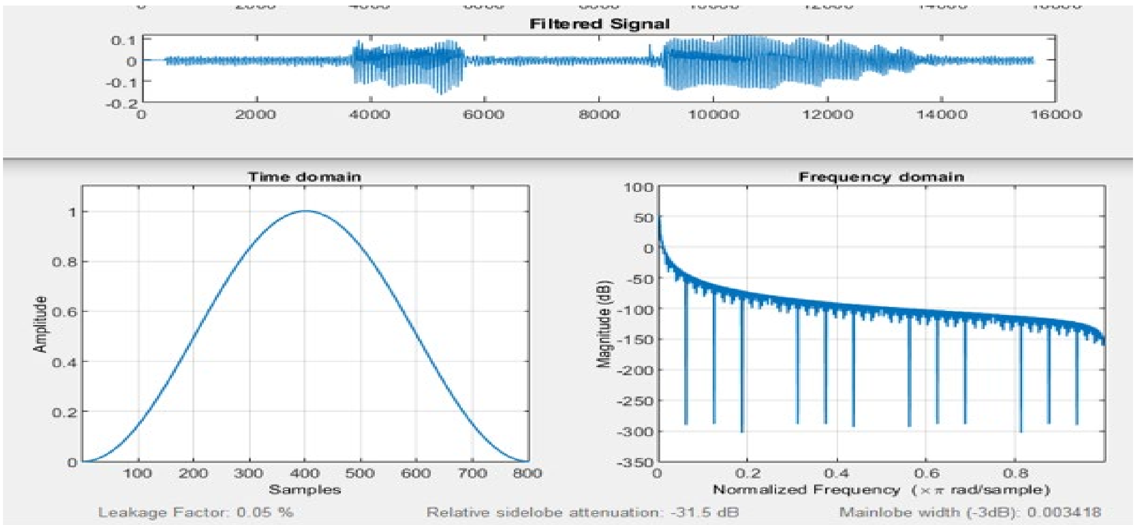


Figure 4.9 Filter order of 800 and frequency constraint of 0.6

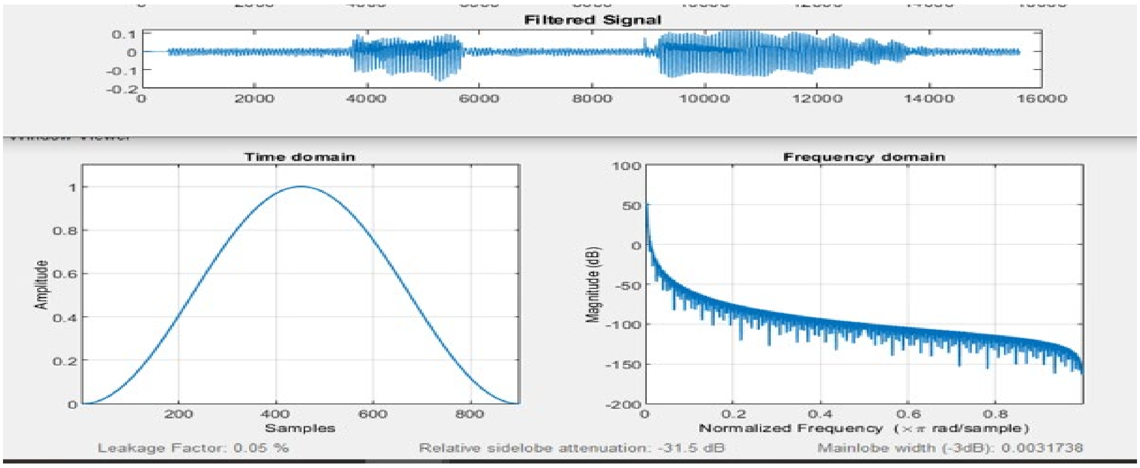


Figure 4.10 Filter order of 900 and frequency constraint of 0.6

Based on figure 4.9 and 4.10, the waveform in the filtered signal, the result don’t show much different this is because Hanning is only smoothing the signal and cannot filter greatly like the Chebyshev and Butterworth filter. This result will be the same all the way, and the output sound be filtered could not be heard clearly.

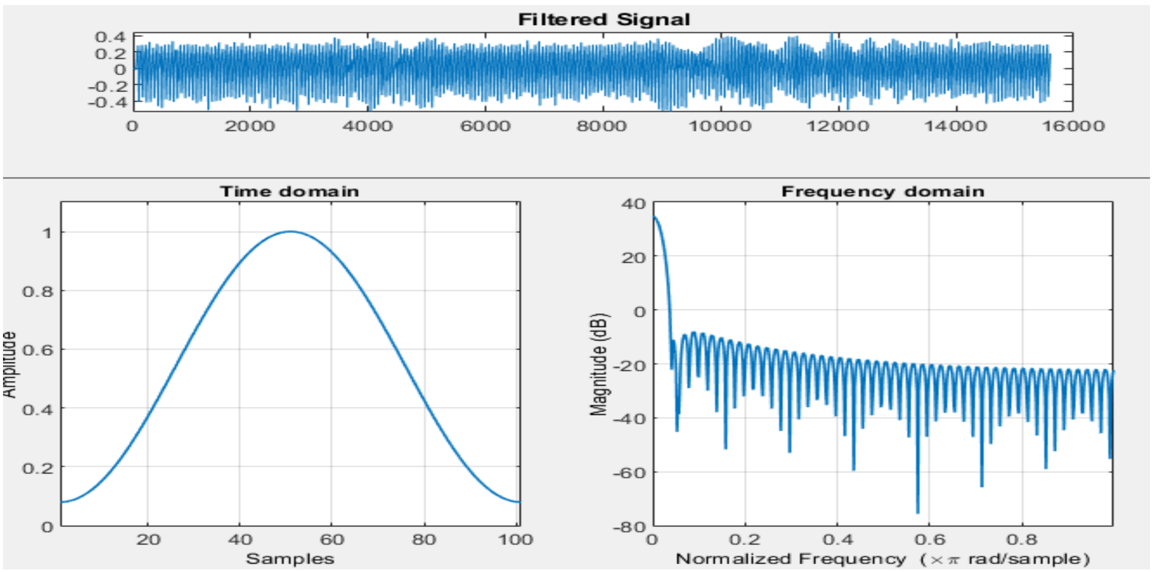


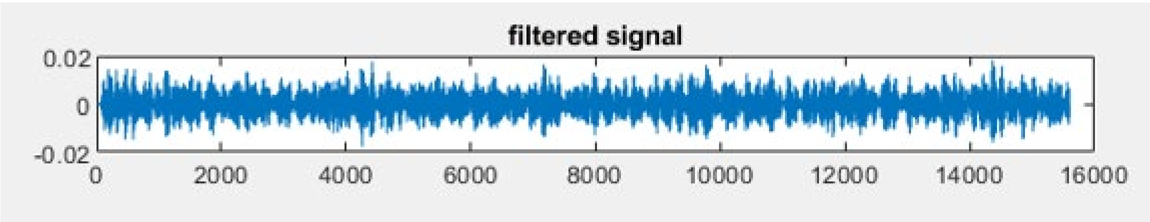
Figure 4.11 Filter order of 100 and frequency constraint of 0.2

Figure 4.11 shows the simulation results using Hanning's low-pass filter. The noise used is car noise, which is first mixed with the input signal and then filtered. The maximum noise of the filtered signal is 69dB. The input signal can also be identified by the speaker output, but the noise contained in the filtered signal output is still dominant.

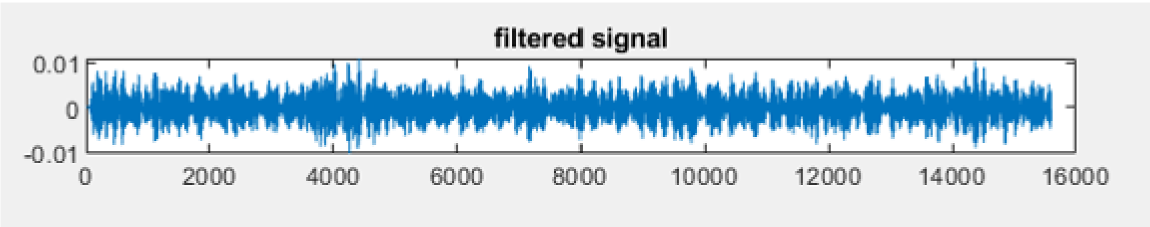
The best output we can get is 100 filter orders, 0.2 constrain frequency and Hanning filter type code setting

4.5 Hamming

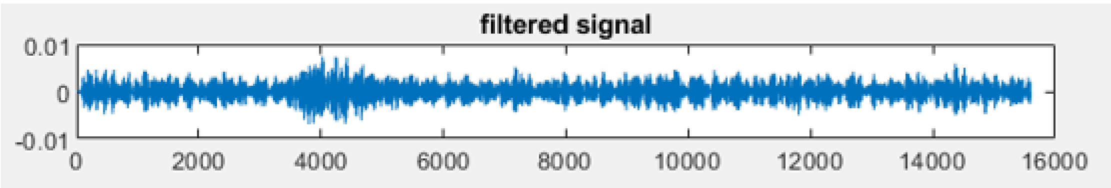
The Hamming is the last suggested filter and was investigate and evaluate. The hamming window considered as an extension of the Hanning window. The Hamming window is a taper formed by using a convex cosine with non-zero endpoints and has been optimized to minimize the nearest sidelobe the result obtained by setting the SNR to -15 and increasing or decreasing the value and results obtained by changing the filter order only.



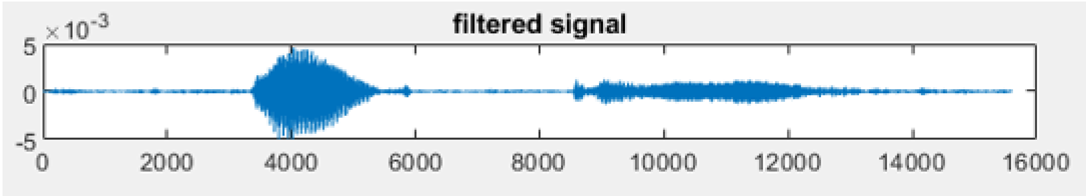
(a) Filter order of -200



(b) Filter order of -15



(c) Filter order of -10



(d) Filter order of 20

Figure 4.12 Hamming window with different of filter order.

In the time domain for Hamming was suggested for smoothing. Many of the Hamming window references come from the literature on signal processing, where it is used for smoothing values as one of several windowing features. It is often referred to as an apodization (meaning "removing the foot").

Figure 4.12 (b) shows the simulation results using the Hamming low pass filter. The noise used is car noise, which is first mixed with the input signal and then filtered. The maximum value of the filtered signal power is 17.2dB, and the average value is -46dB. It is almost impossible to recognize the input signal through the speaker output, but the noise contained in the filtered signal output can still be clearly heard. At the same time, compared with Butterworth and Chebyshev low pass filters, the output sound after filtering is very low. Code settings for -15 filter orders.

4.6 Summary

This chapter shows every output waveform that came from using all these filters with different filter order, frequency constraint and etc. need to keep in mind that this filter is basic and low pass and impossible to eliminate the noise from mixed signal. It turns out that Butterworth low pass filters show the most promising results among all basic filters.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Introduction

First of all, this senior design project already completed its objectives. In addition, basic knowledge of signal processing and MATLAB is obtained. Throughout completing this project, a low pass filter very important in order to make the ALS working properly.

5.2 Conclusion

In conclusion, the objectives of this project are achieved—this project successfully applying preprocessing method in the Assistive Listening System. In the preprocessing method, there is low pass filter that is used in the preprocessing stage, which is Butterworth filter, Chebyshev filter, and two windowing technique which is Hamming and Hanning. There are many journal and conference come out using these basic filters, and it helps a lot in this project in when searching for low pass filters that want to be applied in the preprocessing method. These filters have a different equation and different methods of filtering signal. By investigate and evaluate the performance of each of these filters, from the Butterworth filter to the Chebyshev filter, the output sound can be different clearly even the waveform looks the same.

As for the windowing technique, these two, similarly named Hamming and Hanning (more appropriately called Hann) window functions both have a sinusoidal shape. The difference between them is that both ends of the Hanning window touch zero, which eliminates any discontinuities. The Hamming window stops near zero, which means that the signal will still be slightly discontinuous, and the best result obtained from the window is from figure 4.11 and 4.12 (b), which the sound can hear but lower and not very clear. Thus, clearly windowing not good as Butterworth or Chebyshev.

Only the best result its SNR is taken for each filter. The best result in terms of SNR as an outcome can be obtained, and the SNR equation itself in the ALS system thus will not be covered in this preprocessing stage. Thus, the one with better waveform and SNR will be the most suitable one for the ALS. Then use the output signal of the Butterworth filter to filter and further filter through the adaptive filter in the adaptive algorithm. With the install this low pass filter, the ALS can avoid a high level of noise from corrupting its adaptive filter. In addition, ALS will be able to work properly as it should.

5.3 Limitation and Recommendation

Even though this preprocessing is successfully function as it should, there are some limitations that it has, such as this system cannot filter with a live input signal and noise signal. Also, when adding a new noise signal to the system, it would take a longer time to filter. In addition, for the recommendation, the system should construct an algorithm to choose the best low pass filter for the ALS.

5.4 Future work

With the completion of these projects, if there is any system that wanted to develop similar to ALS or different, hopefully, it could filter live noise signal and also fix the slow filtering. It could contribute and help with preprocessing method so that the project can run smoothly.

REFERENCES

Bardyn, J. P., Melly, T., Seller, O., & Sornin, N. (2016). IoT: The era of LPWAN is starting now. *European Solid-State Circuits Conference*, *2016-Octob*, 25–30.

Benhala, B. (2019). Optimal Low-Pass Butterworth Filter Design by an Enhanced ACO Algorithm. *Proceedings 2019 International Aegean Conference on Electrical Machines and Power Electronics, ACEMP 2019 and 2019 International Conference on Optimization of Electrical and Electronic Equipment, OPTIM 2019*, *1*, 417–422.

Datar, S., & Jain, A. (2016). Application of convolved hanning window in designing filterbanks with reduced aliasing error. *IEEE 2015 International Conference on Signal and Image Processing Applications, ICSIPA 2015 - Proceedings*, *2*, 22–26.

Fadloullah, I., Mechaqrane, A., & Ahaitouf, A. (2017). Butterworth Low Pass filter design using evolutionary algorithm. *2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems, WITS 2017*, *1*, 1–6.

He, X., & Xu, J. (2016). A filtering antenna with 3rd-order Chebyshev response. *2016 IEEE MTT-S International Microwave Workshop Series on Advanced Materials and Processes for RF and THz Applications, IMWS-AMP 2016 - Proceeding*, *1*, 3–6.

Kang, C., Chen, S., Yan, N., Yu, Y., & Min, H. (2018). A Low-power Third-Order Butterworth Filter for NB-IoT Application. *2018 14th IEEE International Conference on Solid-State and Integrated Circuit Technology, ICSICT 2018 - Proceedings*, 1–3.

Khalilabadi, A. J., & Zadehgol, A. (2018). Causality Verification using a First-Order Chebyshev Filter. *2018 IEEE Antennas and Propagation Society International Symposium and USNC/URSI National Radio Science Meeting, APSURSI 2018 - Proceedings*, *1*, 2377–2378.

Le Prell, C. G., & Clavier, O. H. (2017). Effects of noise on speech recognition: Challenges for communication by service members. *Hearing Research*, *349*, 76–89.

Pawarangkoon, P. (2019). A Nanopower 4 th -Order Chebyshev Lowpass Filter for ECG Detection System. *2018 IEEE Asia Pacific Conference on Circuits and Systems, APCCAS 2018*, 135–138.

Pradhan, S., Patel, V., Patel, K., Maheshwari, J., & George, N. V. (2017). Acoustic feedback cancellation in digital hearing aids: A sparse adaptive filtering approach. *Applied Acoustics*, *122*, 138–145.

Shama, S., Kumar, A., & Kumar, Y. (2017). Hamming filter design for ecg signal detection and processing using co-simulation. *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2017*, *2017-Janua*, 1029–1034.

Sulistyaningsih, Putranto, P., Daud, P., & Desvasari, W. (2019). Fast Fourier Transform (FFT) Data Sampling using Hamming and Blackman Method for Radar. *Proceedings of 2018 International Conference on Electrical Engineering and Computer Science, ICECOS2018*,*17*,183–1

`

APPENDIX A (Butterworth)

```
Clc
Clear all
N=length(input_signal);
t=(0:N-1)/fs ;
N/fs;
subplot(4,1,1);
plot(t,input_signal
title('Input_signal')

[input_noise, fn] = audioread('car.wav');
N2=length(input_noise);
t2=(0:N2-1)/fn ;
N2/fn;
plot(t2,input_noise)
title('Noise signal')


if length(input_noise) > length(input_signal)
    noise = input_noise(1:(length(input_signal)));
else
    noise_input = cat(1,input_noise, input_noise, input_noise,
input_noise);
    noise = noise_input(1:(length(input_signal)));
end

if (fn ~= fs)
    disp('Error = Sampling Rate mismatch of inputs');
end

if (fn == fs)    % make the fn = 8000Hz
        noise1 = downsample(input_noise,2);
        fn = fs/2;

    noise = noise1(1:(length(input_signal)));
end

snr = -15;
[d, noise] = addnoise(input_signal, noise, snr);

noisy = d;

subplot(4,1,3)
plot(d)
title('mixed signal')
sound(d)
```

`

```
[b a]=butter (40 ,0.8,'Low');
 filter1=filter(b,a,d);
 subplot(4,1,4);
 plot(filter1);
 sound(filter1)
 title('Filtered signal')
 fprintf('the filtered signal SNR is',snr)
```

`

APPENDIX B (Chebyshev)

```
clc
[input_signal,fs]=audioread('1_00.wav');
N=length(input_signal);
t=(0:N-1)/fs ;
N/fs;
subplot(4,1,1);
plot(t,input_signal);
title('Input_signal')

[input_noise, fn] = audioread('car.wav');
N2=length(input_noise);
t2=(0:N2-1)/fn ;
N2/fn;
subplot(4,1,2)
plot(t2,input_noise)
title('Noise signal')

if length(input_noise) > length(input_signal)
    noise = input_noise(1:(length(input_signal)));
else
    noise_input = cat(1,input_noise, input_noise, input_noise,
input_noise);
    noise = noise_input(1:(length(input_signal)));
end

if (fn ~= fs)
    disp('Error = Sampling Rate mismatch of inputs');
end

if (fn == fs)
        noise1 = downsample(input_noise,2);
        fn = fs/2;

    noise = noise1(1:(length(input_signal)));
end

snr= -15

[d, noise] = addnoise(input_signal, noise, snr);

noisy = d;

subplot(4,1,3)
plot(d)
title('mixed signal')
```

`

```
[b a]=cheby1(40,15, 0.7);
 filter1=filter(b,a,d);
 subplot(4,1,4);
 plot(filter1);

 title('Filtered signal')
 fprintf('the filtered signal SNR is')
```

`

APPENDIX C (Hanning)

```
clc
clear
close all


[x, fs] = audioread('1_00.wav');
r = snr(x,fs);

figure
subplot(131)
plot(x);
title('Input Signal')

L = length(x);
Y = fft(x);
P2 = abs(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);
f = fs*(0:(L/2))/L;
subplot(132)
plot(f,P1)
title('Amplitude Spectrum')


[nn, fn] = audioread('car.wav');
rn = snr(nn,fn);

nn = cat(1,nn, nn, nn, nn);

if (fn ~= fs)
        nn = downsample(nn,2);
        fn = fs/2;
end
nn = nn(1:(length(x)));

figure
subplot(131)
plot(nn);
title('Noise')

L = length(nn);
Y = fft(nn);
P2 = abs(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);
f = fn*(0:(L/2))/L;
subplot(132)
```

`

```
plot(f,P1)
title('Amplitude Spectrum')


snr = -15;
[d, noise] = addnoise(x, nn, snr);
noisy = d;
    D       =   1;
    h       =   dfilt.delay(D);
    xx      =   filter(h,d);


n = 100;
Wn = 0.2;
ftype = 'low';
window = hann(n+1);
wvtool(window)

b = fir1(n,Wn,ftype,window);
a = 1;
y = filter(b,a,d);

subplot(4,1,1)
plot(x)
title('Input Signal')
subplot(4,1,2)
plot(nn)
title('noise signal')
subplot(4,1,3)
plot(d)
title('mixed signal')
subplot(4,1,4)
plot(y);
title('Filtered Signal')
```

`

APPENDIX D (Hamming)

```
clc
lms = 0;
lms_prev = 0;
rls_prev = 0;

[input_signal,Fs]=audioread('1_00.wav');
N=length(input_signal)
[input_noise, Fn] = audioread('car.wav');
if length(input_noise) > length(input_signal)
    noise = input_noise(1:(length(input_signal)));
else
    noise_input = cat(1,input_noise, input_noise, input_noise,
input_noise);
    noise = noise_input(1:(length(input_signal)));
end
if (Fn ~= Fs)
    disp('Error = Sampling Rate mismatch of inputs');
end
if (Fn == Fs)
        noise1 = downsample(input_noise,2);
        Fn = Fs/2;
    noise = noise1(1:(length(input_signal)));
end
snr =00

;
[d, noise] = addnoise(input_signal, noise, snr);
noisy = d;


subplot(4,1,1)
plot(input_signal)
title('input signal')
subplot(4,1,2)
plot(input_noise)
title('noise')
subplot(4,1,3)
plot(noisy)
title('mixed signal')
subplot(4,1,4)
plot(filtered_xn1)
title('filtered signal')
sound(filtered_xn1)
```