PPU College Of Engineering and Technology

The Home of Competent Engineers and Researchers

Electrical and Computer Engineering Department

Communication and Electronics Engineering

Bachelor Thesis

Graduation Project

Wheelchair Movement controlled by Voice Commands

Project Team

Fares M. Ideas   Moawiya W. Fataftah   Raneem T. Masri

Project Supervisor

Eng. Ahmad Qudaimat

Hebron – Palestine

2011-2012

جامعة بوليتكنك فلسطين

الخليل– فلسطين

كلية الهندسة و التكنولوجيا

اسم المشروع

التحكم بحركة الكرسي المتحرك باستخدام الاوامر الصوتية

اسماء الطلبة

فارس ادعيس   معاوية فطافطة   رنيم مصري

بناء على نظام كلية الهندسة والتكنولوجيا واشراف و متابعة المشرف المباشر على المشروع و موافقة أعضاء اللجنة الممتحنة تم تقديم هذا المشروع الى دائرة الهندسة الكهربائية و الحاسوب و ذلك للوفاء بمتطلبات درجة البكالوريوس في الهندسة تخصص هندسة كهربائية فرع هندسة اتصالات و الكترونيات.

توقيع المشرف

....

توقيع اللجنة الممتحنة

....

توقيع رئيس الدائرة

....

PALESTINE POLYTECHNIC UNIVERSITY

COLLEGE OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Graduation Project

Wheelchair Movement Controlled By Voice Commands

Project Team

Fares Ideas **Moawiya Fatafta** **Raneem Masri**

According to the system of the College of Engineering and Technology, and to the recommendation of the Project Supervisor, this project is presented to Electrical and Computer Engineering Department as a part of requirements of B.Sc. degree in Electrical Engineering – Communication and Electronic Engineering.

Project Supervisor signature

…………………………...

Testing Group signature

………...…...……….. ……………….…………..……... ………...…………….

Department Headmaster signature

………………………………….

Hebron-Palestine

DEDICATION

To our fathers

To our mothers

To our brothers and sisters

To our teachers

To our friends

To everyone who helped us

To whom we love

We dedicate our humble effort

Acknowledgment

We want to thank all the people who have direct or indirect contributions in our project.

Also we want to thank

Eng. Ahmad Qudaimat the supervisor for his efforts …

Project Team

Abstract

In this project, a wheelchair controlled by voice commands has been constructed to help people who can’t walk or moving freely. The system which will be built will control the wheelchair's movement in four directions, front, behind, right and left, based on voice commands. It will use MFCC technique to extract features of each voice command. This system generally consists of many subsystems, such as a microphone which will receive voice commands and convert it to electrical signal to pass it to DSP kit for signal processing, after that a microprocessor become as interface to control wheelchair’s movement and direction. To prevent accidents with obstacles, sensors were added to a wheelchair to find the range from any obstacles and avoided.

الملخص

يهدف هذا المشروع الى بناء كرسي متحرك يتم التحكم به وبحركته باستخدام الأوامر الصوتية، وذلك لمساعدة وتسهيل حياة ذوي الاحتياجات الخاصة الذين لا يستطيعون الحركة بحريه دون مساعدة الآخرين. النظام الذي سيتم بناءه سيكون قادر على التحكم بحركة الكرسي المتحرك في الاتجاهات الأربعة ، أمام، خلف، يمين، يسار، وسيستخدم تقنية ال MFCC لإستخراج المميزات والصفات الخاصة بكل أمر صوتي لتمكين النظام من التعرف على الصوت والأمر الصادر. ويتكون النظام بشكل عام من عدة اجزاء كالميكروفون الذي يستقبل اشارة الصوت ويحولها الى اشارة كهربائية يمررها الى DSP kit والتي ستقوم بمعالجتها واستخراج المميزات الخاصة به ومقارنتها مع الكلمات المخزنة مسبقا لمعرفة الكلمة المنطوقة، وإرسال الأمر اللازم للمتحكم ليقرر تحريك الكرسي واتجاه الحركة ام لا. ولتجنب الاصطدام بالعوائق التي قد تواجه الكرسي المتحرك سيتم اضافة بعض المجسات المنوط بها تحديد بعد العوائق من امام الكرسي وتفادي الاصتدام بها وتجنبها.

List of contents

Chapter Three

Chapter Four

Chapter Five

Chapter Six

List of Tables

List of Figures

List of Abbreviations

A/D	: Analog to Digital converter
ALU	: Arithmetic Logic Unit
ASIC	: Application Specific Integrated Circuit
ASR	: Automatic Speech Recognition
CCS	: Code Composer Studio
CTFT	: Continuous Time Fourier Transform
DC	: Direct Current
DCT	: Discrete Cosine Transform
DFT	: Discrete Fourier Transform
DIP	: Dual In-line Package
DSK	: DSP starter kit
DSP	: Digital Signal Processing
DTFT	: Discrete Time Fourier Transform
EEPROM	:Electrically Erasable Programmable Read Only Memory
FFT	: Fast Fourier Transform
FPGA	: Field-Programmable Gate Arrays
FT	: Fourier Transform
HMM	: Hidden Markov Model
IC	: Integrated Circuit
IDE	: Integrated Development Environment
LED	: Light Emitting Diode
LPC	: Linear Predictive Coding
MAC	: Multiply Accumulate
MFCC	: Mel Frequency Cepstral Coefficients

MFLOPS : Million Floating Point Operations Per Second

MIPS : Million Instructions Per Second

PC : Personal Computer

RAM : Random Access Memory

ROM :Read Only Memory

SDRAM : synchronous dynamic RAM

USB : Universal Serial Bus

VLIW : Very Long Instruction Word

VQ : Vector Quantization

1

Chapter One

Introduction

Chapter One

Introduction

1.1 Introduction

If we look at the world around us, we find that the technology has embraced the human being and took over most areas of his life, especially with the progress of science and innovation in new ways in which access to what meet the human needs and desires that cannot be broken.

Perhaps the most important fields of this technology is voice recognition that is human needed in most areas of his life, especially those who suffering from physical disability, such as not being able to move, and who cannot act like a human being normal, and now with the use of voice recognition has become a unable human life is easier as he be able to control his movement alone and without help others.

The field work and researches in voice recognition subject is not limited around facilitating the movement of humans only, but may be used in solving other problems such as lighting homes or control the movement of a robotic car …etc.

And here, we began our thinking and our search to find an effective system capable of providing service to the community, whose member suffers from a deficit in the movement and thus achieve psychological comfort and reassurance to them.

1.2 Project Objectives

This project aims to achieve the following:

1. Building a system capable of controlling the movement of a wheelchair in the four directions in properly way.
2. To facilitate the lives of individuals who suffer from a deficit in the movement by controlling the movement by themselves.
3. Save time through this person self-control and reach what he need with required speed, and the provision of effort by not having to help another human being in motion, so do not be a burden on others.
4. Apply the digital signal processing and what we studied concepts in previous years in the establishment of an effective system.

1.3 General Idea

The idea in the design of this project focuses on the construction of five voice commands are used directly to move the wheelchair.

The function of this project is to build an intelligent and accurate system in order to be able to identify the words entered which are in Arabic and gives the best performance in response to these voice commands entered.

The commands which will used:

- Front { أمام }
- Behind { خلف}
- Stop { قف }
- Left { يسار}
- Right {يمين }

This project consists of two main parts:

Software components that will be built to carry out the actual speech recognition and we will use the MATLAB program in the field of software. The hardware components are the wheelchair, which is designed to carry out voice-activated, microphone, microcontroller, DSP kit, motor and amplifiers that control the movement of the wheelchair.

When the user speaks a particular word, the electrical signal emitted by the microphone is digitize by A/D converter then stored in the DSP kit to recognize the voice command input, the DSP kit attempt to match the input with digitize voice sample,if the matching occur ,then the accepted command transfer to microcontroller to analysis the signal and send a command to wheelchair to be move.

There is difficulty in distinguishing in particular the noise signals, which play a major role in influencing the sound, may be there an interfere with the signals from the surrounding depending on the area located by the person incapable of movement, but will be here as mechanisms to overcome that.

1.4 Literature Review

The principle of voice recognition was presented many years ago, a lot of people and researcher worked and still works in this field to use it in different applications. By group search in the internet and university library they found different documents and papers in this field as following.

The first project : Voice Recognition Security System was done by Xiaowen Lu (xl76) and Shihjia Lee (sl362), at 2006: The function of this speech recognition security system is to have a system that will only unlock upon recognizing a voice password spoken by the administrator or password holder. In this project team work used a tiny Mega32 microcontroller to process voice to discrimination between different words.

The second one : was done by Khalid T. Al-Sarayreh, Rafa E. Al-Qutaish, Basil M. Al-Kasasbeh, Using the Sound Recognition Techniques to Reduce the Electricity Consumption in Highways: This paper aims at using the sound recognition techniques in order to turn on the lights only when there are cars on the highway and only for some period of time. Linear Predictive Coding (LPC) method and feature extraction used to apply the sound recognition. Furthermore, the Vector Quantization (VQ) used to map the sounds into groups in order to compare the tested sounds[Journal of American Science 2009].

finally: Speech Control for Car Using the TMS320C6701 DSP was done by MartinPetriska, Dušan Považanec, Peter Fuchs. This paper presents a design of voice module designed for using in cars. The module communicates with driver by human speech. It informs driver about the state of car equipments and recognizes his voice commands. This feature makes it easy to control a lot of car equipment by human voice. The system consists of DSP board with TMS320C6701, large memories and analog codec. Accordingly actual situation in speech recognition and speech synthesis with aspect for application in car technology.

1.5 Project Team

Project team consists of two groups of different specializations Engineering , Communications group and Biomedical Group. Communications engineering team will be responsible for processing sound signals and convert sound to the orders and then the Biomedical teams will analyze such orders responsible for moving the wheelchair.

1.6 Time Plan

The following tables define the main tasks in the project introduction and project itself:

T1	Project Definition	1 Week
T2	Collecting data	11 Weeks
T3	Analysis	7 Weeks
T4	Theoretical calculation	4 Weeks
T5	Documentation	10 Weeks
T6	Prepare for presentation	2 Weeks

Table 1.1: Time scheduled table for project introduction

The time of the project introduction is scheduled over 16 weeks, table 2 shows how the work was scheduled over this time:

Week / Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
T1																
T2																
T3																
T4																
T5																
T6																

Table 1.2: Time plan table for project introduction

The following table defines the main tasks in the project:

T1	Collecting data	3 Week
T2	Implementation	10 Weeks
T3	Analysis	5 Weeks
T4	Building and testing the system	8 Weeks
T5	Documentation	10 Weeks
T6	Prepare for presentation	2 Weeks

Table 1.3: Time scheduled table for project

The time of the project is scheduled over 16 weeks, table 4 shows how the work was scheduled over this time:

Week / Task	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
T1	■	■	■													
T2			■	■	■	■	■	■	■	■	■	■				
T3				■	■	■	■	■								
T4							■	■	■	■	■	■	■	■		
T5					■	■	■	■	■	■	■	■	■	■		
T6															■	■

Table 1.4: Time plan table for project

1.7 Estimated Cost

The following is a list of different costs needed to implement the system**:**

Number	**Object**	**Cost ($)**
1	Wheelchair	200
2	Motor #2	750
3	DSP Kit	500
4	Controller	550
5	Microphone	4
6	Ultrasonic Sensor #2	80
Total cost ($)		2084

Table 1.5: Estimated cost

1.8 Project Benefits

Many people exposed for many accident in their life, that's may be cause an Impair mobility prevent them to lives their life in normally way, because of that a wheelchair was invention to help those people in movement. Unfortunately, many of them cannot even move their hands. So that, they need other people to help them in mobility even when they used a wheelchair. To get rid from these Restrictions and give disabled people more freedom in movement Without the help of others , this project aims to control a wheelchair by different voice commands not by hands and thus, ease the lives of disabled and this is a very useful thing .

1.9 Project Risk Management

1. **Hardware**: This project contains many of hardware components such as Wheelchair, DC motor, Microcontroller, DSP Kit, and Microphone. Team work faced some difficulties in obtaining some of these pieces.
2. **Software:** MATLAB program used in signal processing part of this project, and downloads this program on DSP kit to process voice signal practically. One of problems faced the team was DSP kit didn't support all blocks used in MATLAB SIMULINK that’s forcing team to build a system as DSP kit support.
3. **Team Risk:** one of the problems suffered by the team, no similarity in the study programs that's effected in connection team members. Also, lack of a suitable places for meetings between team members.
4. **Project Risk:** Team work faced difficulty in determine a good idea for a graduation project and many of ideas that offered on team was reject as it's not applicable here.

1.10 Project Contents

The project is divided up to six chapters; the chapters follow each other logically to get the complete idea about the project.

Chapter 1: discussed the definition of the project from multiple sides, its objectives and importance. Then taking about the time plane and the estimated cost of the system components that are needed to implement the designed system. Finally, the difficulties that the group had been faced .

Chapter 2: Talk about the theoretical background of the project, hardware and software we needed to implement the project.

Chapter 3: Project conceptual design and the detail objective of the project . We dealt with way of implement the project and draw the block diagram.

Chapter 4: system implementation: This chapter discus how we implement and built the system as we design in chapter 3.

Chapter 5: system testing: In this chapter we test each part of the system as we implemented in chapter 4 and verify from result.

Chapter 6: provides the concluding after working the system, and suggestion for Future work.

2

Chapter Two

Theoretical background

Chapter Two

Theoretical background

2.1 Introduction

This chapter show the theoretical background about our project, we starting by submit a detailed information about voice recognitions, sound neutrality, how its propagate in materials, the affect of noise, and interference on sound, then show signal processing, microphones describe its circuit and how its work, finally we show the DSP kit specification, objective, performance and there download, the ultrasonic sensor and microcontroller.

2.2 Voice Recognition

2.2.1 What is Voice Recognition?

Voice recognition is the technology by which sound, words or phrases spoken by humans are converted into electrical signals, these signals are transformed into coding patterns that can be identified by a computer or DSP processor.

Processor usually converting a speech waveform into a sequence of words involves several essential steps. First, a microphone picks up the acoustic signal of the speech to be recognized and converts it into an electrical signal. A modern speech recognition system also requires that the electrical signal be represented digitally by means of an analog-to-digital (A/D) conversion process, so that it can be processed with a digital computer or a microprocessor. This speech signal is then analyzed (in the analysis block) to produce a representation consisting of features of the speech.[1]

2.2.2 Why do we want Speech Recognition?

The main goal of speech recognition is to get efficient ways for humans to communicate with computers or other devices .Bloomberg [2] enumerates some of the applications of speech recognition, some advantages that can be achieved by speech recognition. For example, personal computer that can be voice-controlled and used for dictation. Another application is environmental control, such as turning on the light, controlling the TV etc.

2.2.3 Why is Speech Recognition Difficult

There are many factors that affect voice recognition process such as:

1. Isolated Words

The system that operates single words at a time requires a pause between saying each word.

2. Continuous speech

Speech has no natural pauses between the word boundaries, the pauses mainly on a syntactic level, such as after a phrase or a sentence. This introduces a difficult problem for speech recognition - how should we translate a waveform into a sequence of words?

One way to simplify this process is to give clear pauses between the words. This works for short command-like communication, but as the possible length of utterances increases, clear pauses get cumbersome and inefficient.

3. Speaker variability.

All speakers have their special voices, due to their unique physical body and personality, the voice is not only different between speakers; there are also wide variations within one specific speaker.

4. Realization

If the same words were pronounced over and over again, the resulting speech signal would never look exactly the same. Even if the speaker tries to sound exactly the same, there will always be some small differences in the acoustic wave .

5. The Gender of speaker

Men and women have different voices, the main reason for that; women have in general shorter vocal tract than men. The fundamental tone of women's voices is roughly two times higher than men's because of this difference.

6. Environment and Noise

Speech is uttered in an environment of noise, a clock ticking, compute, radio playing somewhere, another human speaker in the background etc. Unwanted information in the speech signal, in voice recognition systems must identify and filter out from speech signal.

Another kind of noise is the echo effect, which is the speech signal bounced on some surrounding object, and that arrives in the microphone few milliseconds later. If the place in which the speech signal has been produced is strongly echoing, this may give raise to a phenomenon called reverberation, which will effects in the recognition quality.

2.3 Nature of Sound

This section show more detail about sound nature.

2.3.1 Sound Characteristics

Sound is a mechanical wave, composed of frequencies within range of hearing, and a level sufficiently strong to be heard.

The perception of sound in any organism is limited to a certain range of frequencies. For humans, hearing is normally limited to frequencies between 20 Hz and 20,000 Hz, although these limits are not definite. The upper limit generally decreases with age. Other species have a different range of hearing.

The speed of sound depends on the medium that waves transmitted in it. In general, the speed of sound is proportional to the square root of the ratio of the elastic modulus (stiffness) of the medium to its density. Those physical properties and the speed of sound change with ambient conditions. The speed of sound V(m/s) depend also on temperature as the following equation

$$V = 331 + 0.6T \tag{2.1}$$

In 20 °C (68 °F) air at sea level, the speed of sound is approximately 343 m/s. The speed of sound is also slightly sensitive to the sound amplitude, which means that there are nonlinear propagation effects, such as the production of harmonics and mixed tones not present in the original sound.[3]

2.3.2 The Propagation of Sound

Sound is a sequence of waves of pressure which propagates through compressible media such as air or water. Sound can propagate through solids as well, but there are additional modes of propagation. During their propagation, waves can be reflected, refracted, or attenuated by the medium.

All media have three properties which affect the behavior of sound propagation:

1. A relationship between density and pressure. This relationship, affected by temperature determines the speed of sound within the medium.

2. The motion of the medium itself, if the medium is moving, the sound is further transported.

3. The viscosity of the medium. This determines the rate at which sound is attenuated. For many media, such as air or water, attenuation due to viscosity is negligible.

One way that the propagation of sound can be represented is by the motion of wave fronts- lines of constant pressure that move with time. Another way is to hypothetically mark a point on a wave front and follow the trajectory of that point over time. This latter approach is called ray-tracing and shows most clearly how sound is refracted.

The speed of sound is determined by the properties of the air, and not by the frequency or amplitude of the sound. Sound waves, as well as most other types of waves.[4]

2.3.3 Noise and Interference Affect

This phenomenon play a crucial role in influencing the sound and it's considered an important issue in speech recognition.

Noise

Noise is a random fluctuation in an electrical signal, Noise generated by electronic devices varies greatly, as it can be produced by several different effects. In communication systems, noise is an error or undesired random disturbance of a useful information signal, introduced before or after the detector and decoder. The noise is a summation of unwanted or disturbing energy.

The noise level in an electronic system is typically measured as an electrical power N in watts or dBm. Noise may also be characterized by its probability distribution and noise spectral density $N_0(f)$ in watts per hertz. A noise signal is typically considered as a linear addition to a useful information signal. Typical signal quality measures involving noise are signal-to-noise ratio (SNR or S/N).

Noise is a random process, characterized by stochastic properties such as its variance, distribution, and spectral density. The spectral distribution of noise can vary with frequency, so its power density is measured in watts per hertz (W/Hz). Since the power in a resistive element is proportional to the square of the voltage across it, noise voltage (density) can be described by taking the square root of the noise power density. Noise power is measured in Watts or decibels (dB).

Noise levels are usually viewed in opposition to signal levels, and so are often seen as part of a signal-to-noise ratio (SNR). Telecommunication systems strive to increase the ratio of signal level to noise level, in order to effectively transmit data. In practice, if the transmitted signal falls below the level of the noise in the system, data can no longer be decoded at the receiver. Noise in telecommunication, is a product of both internal and external sources to the system.[5]

Interference

The interference happened when signal or more affect on the desired signal, and this caused attenuation in the amplitude and phase of the desired signal, wasted the information that contained, interference could happen in time or in frequency domain, in time by send data at same time for two station or more, and its happened in frequency by using the same frequency between stations ,we can consider two type of interference that are: intentional interference for example a base stations broadcast information on the same frequency band, or non-intentional that comes from two signal can't knowing the frequency of each other, in a communication system we can measure the quality of a signal, when its face interference by (Signal to Noise-Interference ratio) as the following:[6]

$$SINR = \frac{signal\ power(p)}{Noise\ power(N) + Interference\ power(I)} \qquad (2.2)$$

2.4 Signal Processing

2.4.1 Time Domain and Frequency Domain

In a communication system the signal can be manipulated, make a processing on its domain, and there’s a two domain: time domain and frequency domain as the following:

Time Domain

Time domain is a term used to describe the analysis of mathematical functions, physical signals or time series of economic or environmental data, with respect to time. In the time domain, the signal or function's value is known for all real numbers, for the case of continuous time, or at various separate instants in the case of discrete time. An oscilloscope is a tool commonly used to visualize real-world signals in the time domain. A time domain graph shows how a signal changes over time.[7]

Frequency Domain

Frequency domain is a term used to describe the domain for analysis of mathematical functions, or signals with respect to frequency, rather than time whereas a frequency domain graph, shows how much of the signal lies within each given frequency band over a range of frequency. A spectrum analyzer is the tool commonly used to visualize real-world signals in the frequency domain.

A given function or signal can be converted between the time and frequency domains, with a pair of mathematical operators called a (transform). An example is the Fourier transform, which decomposes a function into the sum of a (potentially infinite) number of sine wave frequency components. The "spectrum" of frequency components is the frequency domain representation of the signal. The inverse Fourier transform converts the frequency domain function back to a time domain.[7]

2.4.2 Sampling Process

This is the conversion a continuous time signal into a discrete time signal, obtained by taking samples of the continuous time signal at discrete time instants. Thus, if $X_a(t)$ is the input to the sampler ,the output is $X_a(nT) \equiv X(n)$,where T is called sampling interval .[8]

The Nyquist–Shannon sampling theorem states that a signal can be exactly reconstructed from its samples, if the sampling frequency is greater than twice the highest frequency of the signal; but requires an infinite number of samples.[9]

People can hear different sounds with frequency range (20- 20,000) Hz. But human sounds contain frequencies with 4 KHz maximum, so by Nyquist theorem to recover original signal from its sampling, we need to sample human voice by:

$$2 * F_{max} = 2 * 4000 = 8KHz \tag{2.3}$$

2.4.3 Window and Frames

In all practical signal processing applications, it is necessary to work with short terms or frames of the signal, unless the signal is of short duration. This is especially true if we use conventional analysis techniques on signals (such as speech) with non stationary dynamics. In this case it is necessary to select a portion of the signal that can reasonable be assumed to be stationary.

The window (time domain window), say $w(n)$, is a real, finite length sequence used to select a desired frame of the original signal say $x(n)$, by a simple multiplication process. We will assume windows to be causal sequence beginning at time $n = 0$. The duration will usually be denoted N. Most commonly used window are symmetric about the time$(N-1)/2$ where this time may be halfway between two sample points if N is even. That means that the windows are linear phase sequences have DTFTs that can be written.[10]

$$W(\omega) = |W(\omega)|e^{-jw\,(\frac{N-1}{2})} \tag{2.4}$$

2.4.4 Fourier Transform

Since frequency is one of the important parts of information, we must find a method to give the equivalent frequency spectrum of such signal in time domain and this method capable to back the time domain signal from its spectral. The Fourier transform is a suitable method and it's representation of the frequency of that signal. It calculates for a periodic signal either its continuous or discrete and cause continuous signal in frequency domain.

Continuous Time Fourier Transform (CTFT)

The Fourier transform (FT) is used to represent a continuous-time non periodic signal as a superposition of complex sinusoidal. Thus the FT representation for a time signal involves an integral over frequency, as shown by

$$X(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \mathrm{X(j\omega)e^{j\omega t}d\omega} \qquad (2.5)$$

Where

$$X(j\omega) = \int_{-\infty}^{\infty} X(t)e^{-j\omega t}\ dt \qquad (2.6)$$

The transform X(jω) describes the signal X(t) as a function of sinusoidal frequency ω and is termed the frequency-domain representation for X(t) .The second equation is termed the FT of X(t) since it converts the time-domain signal into its frequency-domain representation .The first equation is termed the inverse FT since it converts the frequency-domain representation X(jω) back into time domain.

Discrete Time Fourier Transform (DTFT)

The DTFT representation is expressed as

$$X[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\Omega})\ \mathrm{e^{j\Omega n}d\Omega} \qquad (2.7)$$

Where

$$X\left(e^{j\Omega}\right) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n} \qquad (2.8)$$

The transform X($e^{j\Omega}$) describe the signal X[n] as a function of sinusoidal frequency **Ω** and is termed the frequency-domain representation of x[n]. The second equation is the DTFT of X[n] since it converts the time-domain signal into its frequency-domain representation. The first equation is termed the inverse DTFT since it converts the frequency-domain representation back into the time domain. X($e^{j\Omega}$) its periodic signal with 2π period. [8]

2.4.5 Mel scale

For each tone with an actual frequency, f, measured in Hz, a subjective pitch is measured on a scale called the 'Mel' scale. The Mel-frequency scale is linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. As a reference point, the pitch of a 1 kHz tone, 40 dB above the perceptual hearing threshold, is defined as 1000 mels. A popular formula to convert ***f*** hertz into ***m*** Mel is: [11]

$$Mel(f) = 2595\log_{10}\left(1 + \frac{f}{700}\right) \qquad (2.9)$$

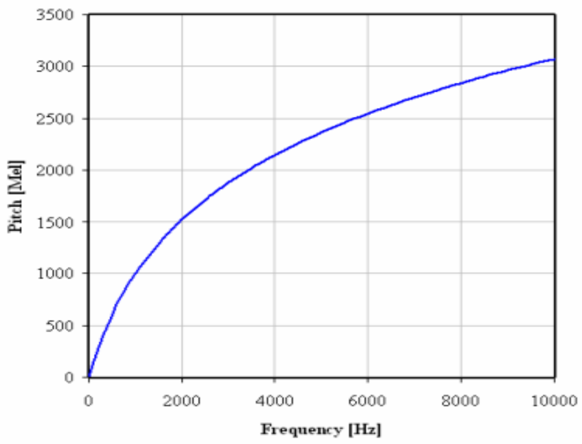


Figure 2.1: Mel scale (mel) versus Frequency scale [Hz].

2.4.6 Mel-frequency filter bank

In order to represent the static acoustic properties, the Mel-Frequency Cepstral Coefficient (MFCC) is used as the acoustic feature in the cepstral domain. This is a fundamental concept which uses a set of non-linear filters to approximate the behavior of the auditory system.

Equation (2.10) is the filterbank with **M** filters (**m**=1,2… **M**), where filter m is the triangular filter given by:

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \leq k \leq f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \leq k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases} \qquad (2.10)$$

Which satisfy:

$$\sum_{m=0}^{M-1} H_k(m) = 1$$

The central frequency of each mel-scale filter is uniformly spaced below 1kHz and it follows a logarithmic scale above 1 kHz as shown in Eq(2.9) and Figure (2.2) More filters process the spectrum below 1 kHz since the speech signal contains most of its useful information such as first formant in lower frequencies

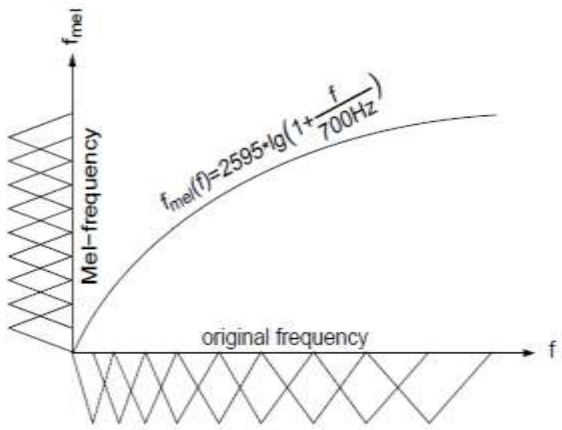


Figure 2.2: Filters are either uniformed distributed at mel scale
or non uniformed at original spectrum

If we define f_l and f_h be the lowest and highest frequencies of the filter banking Hz, Fs the sampling frequency in Hz, M the number of filters and N the size of FFT, the centering frequency f (m) of the mth filter bank is:

$$f(m) = \left(\frac{N}{F_s}\right) Mel^{-1}\left(Mel(f_l) + \ m\ .\ \frac{Mel(f_h) - \ Mel(f_l)}{M+1}\right) \qquad (2.11)$$

Where the Mel-scale Mel is given by Equation(2.9) and Mel^{-1} is its inverse is :[11]

$$Mel^{-1}(b) \quad = 700\ .\left(\ 10^{b/2595} - 1\ \right) \qquad (2.12)$$

2.4.7 Euclidian distance

Is a term almost used when we want to find a distance between two vectors, or even the difference between two patterns. When we say Euclidian distance its gives us different meaning such that:

- The Euclidian distance between two point (i.e. point p and point q) tell us the length of line segment connecting them
- The Euclidian distance between two sequences, give the degree of similarity between them, and match each sequence (pattern) to appropriate other pattern.

The function of the Euclidian distance as the following: if $= [x1\ x2 \ldots \ldots xn]\ and\ Y = [y1\ y2 \ldots \ldots yn]$. [12]

Then:

$$d(X,Y) = \sqrt{\sum_{i=1}^{n}(X_i - Y_i)^2} \qquad (2.13)$$

2.5 Project Components

2.5.1 Microphone

2.5.1.1 What Is Microphone

Microphone is an acoustic-to-electric transducer or sensor that changes information from one form to another; it converts sound into an electrical signal. Microphones are transducers which detect sound waves and produce an electrical image of the sound; they produce a voltage or a current which is proportional to the sound signal. [13]

There are many different types of microphone dynamic, ribbon, or condenser microphones. Nowadays the most microphone use for audio, the electromagnetic induction (dynamic microphone), capacitance change (condenser microphone, pictured right), piezoelectric generation, or light modulation to produce the signal from mechanical vibration.

2.5.1.2 The Dynamic Microphone

Dynamic microphones are also called moving-coil microphones. The most important features, that give the Dynamic microphones preference than other type that is dynamic microphone is heavier, larger, and less sensitive than other. They are extremely durable, dependable, and can withstand high sound pressure levels. Dynamic microphones are versatile because they don't require a power source.[13]

2.5.1.3 Specifications of Microphone

When choosing the right microphone must be considered taking into account some characteristics of:

1. Sensitivity: mean how much electrical output is produced by a given sound. This is a vital specification for recording very tiny sounds. A very sensitive mice's, might overload the input electronics of the mixer or tape deck, producing distortion.

2. Overload characteristics: Any microphone will produce distortion when it is over driven by loud sounds. Sustained over driving or extremely loud sounds can permanently distort the diaphragm, degrading performance at ordinary sound levels.

3. Linearity or Distortion: The distortion characteristics of mice's are determined mostly by the care with which the diaphragm is made and mounted. High volume production methods can turn out an adequate microphone, but the distortion performance will be a matter of luck. No mice's is perfectly linear, but we can find mice's with distortion that complements the sound which will record.

4. Frequency response**:** A frequency response has been the main goal of microphone. Console manufacturers add equalizers to each input, to compensate to improve the level of performance

5. Noise: Microphones produce a very small amount of current, which makes sense when you consider just how light the moving parts, must be to accurately follow sound waves. To be useful for recording, the signal must be amplified by a factor of over a thousand. Any electrical noise produced by the microphone will also be amplified, so even slight amounts are intolerable.

Dynamic microphones are essentially noise free, but the electronic circuit built into condenser types, is a potential source of trouble Noise, and also includes unwanted pickup of mechanical vibration through the body of the microphone. The most common source of noise associated with microphones, is the wire connecting the mice's to the console or tape deck. A mice's preamp is very similar to a radio receiver, so the cable must be prevented from becoming an antenna. The basic technique is to surround the wires that carry the current to and from the mice's with a flexible metallic shield, which deflects most radio energy. [14]

2.5.2 Digital Signal Processing Starter Kit

Digital signal processors is a specialized form of microprocessor, such as the TMS320C6x of processors are like fast special-purpose microprocessors, with a specialized type of architecture and an instruction set appropriate for signal processing .

The most important features of the TMS320C6713 Digital Signal Processing Starter Kit are:

1. Very well suited for numerically intensive calculations.

2. Based on a very-long-instruction-word (VLIW) architecture.

3. The most powerful processor.

4. Low cost development platforms for real time digital signal processing applications.

Digital Signal Processing Starter Kit comprises a small circuit board containing aTMS320C6713 floating - point digital signal processor, and a TLV320AIC23 analog interface circuit (codec), can connects to a host PC via a USB port. PC software in the form of Code Composer Studio (CCS), is provided in order to enable software written in C or assembly language, to be compiled and/or assembled, linked, and downloaded to run on the DSK.[15]

The architecture and instruction set of a DSP are optimized for real time digital signal processing. Typical optimizations include:

1. Hardware multiply - accumulate (MAC) provision,
2. Hardware circular and bit reversed addressing capabilities (for efficient implementation of data buffers, and fast Fourier transform computation),
3. Harvard architecture (independent program and data memory systems).

In many cases, DSPs resemble microcontrollers insofar, as they provide single chip computer solutions incorporating onboard volatile, and nonvolatile memory, a range of peripheral interfaces, and have a small footprint, making them ideal for embedded applications. In addition, DSPs tend to have low power consumption requirements.

DSP processer's very suitable for the audio-frequency range. Common applications using these processors have been for frequencies from 0 to 96 kHz. Speech can be sampled at 8 kHz (the rate at which samples are acquired), which implies that each value sampled is acquired at a rate of 1/(8 kHz) or 0.125ms. A commonly used sample rate of a compact disk is 44.1 kHz. Analog/digital (A/D)-based boards in the megahertz sampling rate range, are currently available.[15]

2.5.2.1 Key Features

Digital signal processors are used for a wide range of applications, from communications and controls to speech and image processing. These processors have become the products of choice for a number of consumer applications, since they have become very cost-effective. They can handle different tasks, since they can be reprogrammed readily for a different application. DSP techniques have been very successful, because of the development of low-cost software, and hardware support.

DSP processors are concerned primarily with real-time signal processing. Real time Processing requires the processing to keep pace with some external event, where non-real-time processing has no such timing constraint. The external event to keep pace with is usually the analog input. Whereas analog-based systems with discrete electronic components, such as resistors can be more sensitive to temperature changes, DSP-based systems are less affected by environmental conditions.DSP processors enjoy the advantages of microprocessors. They are easy to use, flexible, and economical [16]

2.5.2.2 DSK support tools:

The following tools are used:

1. A Texas Instruments DSP starter kit (DSK). The DSK package includes:

   a. Code Composer Studio (CCS), which provides the necessary software support tools. CCS provides an integrated development environment (IDE), bringing together the C compiler, assembler, linker, debugger.

   b. A circuit board (the TMS320C6713 DSK) , its containing a digital signal processor, and a 16 - bit stereo codec for analog signal input and output.

   c. A universal Serial bus (USB), its cable that connects the DSK board to a PC.

   d. A +5 V universal power supply for the DSK board.

2. A PC. The DSK board connects to the USB port of the PC through the USB cable included with the DSK package.

3. An oscilloscope, spectrum analyzer, signal generator, headphones, microphone, and speakers .[17]

Texas Instrument TMS320C6713 DSP. It has the following properties:

- 32 bit instruction cycle.
- Typical 225 MHz, up to 300 MHz clock frequency.
- 2400 MIPS / 1800 MFLOPS instruction processing.
- 4xALU and 2x32-Bit MAC functional unit.
- 4KB Level1 program cache, 4KB Level1 data cache, 256KB Level2 data cache.
- 32 bit external memory interface up to 512 MB addressable external memory space .
- 2xI2C ports, 2xMCBSP (SPI) ports, 2x32-Bit timers. [17]

2.5.2.3 DSK Board

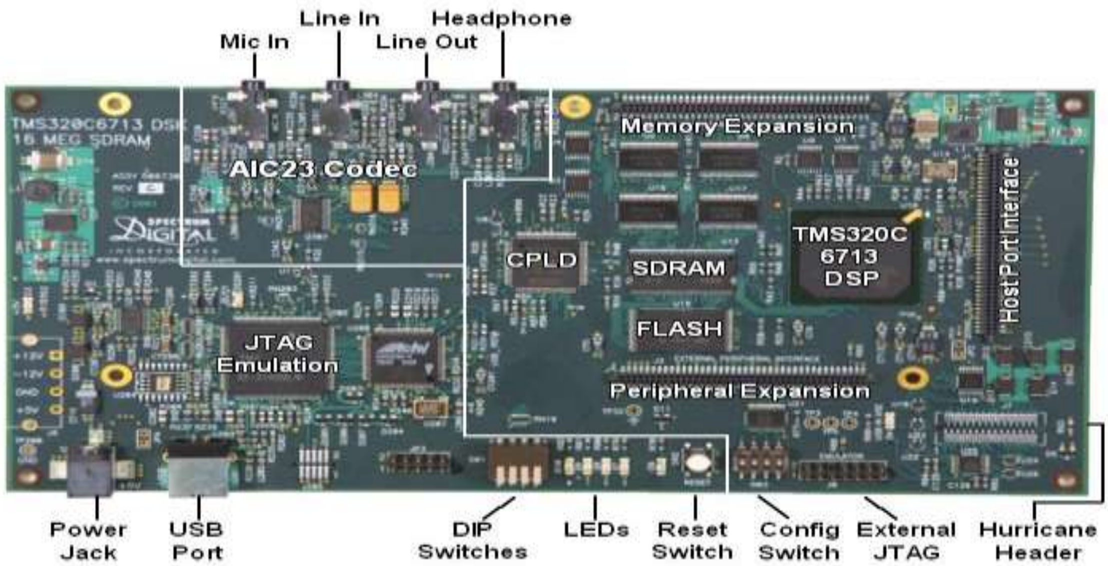


Figure 2.3: The DSK Board.

The DSK packages are powerful, relatively inexpensive, with the necessary hardware and software, support tools for real time signal processing. They are complete DSP systems. The DSK boards, which measure approximately 5 ×8 inches, include either a 225 - MHz C6713 floating - point digital signal processor, or a 1 - GHz C6416 fixed - point digital signal processor, and a 32 - bit stereo codec TLV320AIC23 (AIC23) for analog input and output.

The onboard codec AIC23 uses sigma delta technology that provides analog -to - digital conversion (ADC), and digital - to - analog conversion (DAC) functions. It uses a 12 - MHz system clock, and its sampling rate can be selected from a range of alternative settings from 8 to 96 kHz. A daughter card expansion facility is also provided on the DSK boards, Two 80 - pin connectors provide for external peripheral and external memory interfaces.

The DSK boards each include:

1. 16 MB (megabytes) of synchronous dynamic RAM (SDRAM).

2. 512 KB (kilobytes) of flash memory.

3. Four connectors on the boards provide analog input and output:
   (a) .MIC IN for microphone input.
   (b) LINE IN for line input.
   (c) LINE OUT for line output.
   (d) HEADPHONE for a headphone output (multiplexed with line output).

The status of four user DIP switches on the DSK board, can be read from within a program running on the DSP, and provide the user with a feedback control interface. The states of four LEDs on the DSK board can be controlled from within a program running on the DSP. Also onboard the DSKs are voltage regulators that provide 1.26 V for the DSP cores and 3.3 V for their memory and peripherals.[15]

2.5.2.4 Code Composer Studio:

CCS is a powerful integrated development environment that provides a useful transition between a high-level (C or assembly) DSP program and an on-board machine language program. CCS consists of a set of software tools, and libraries for developing DSP programs, compiling and linking them into machine code, and writing them into memory on the DSP chip, and on-board external memory. It also contains diagnostic tools, for analyzing and tracing algorithms, as they are being implemented on-board. We will use CCS to develop, compile, and link programs that will be downloaded from a PC to DSP hardware.[18]

2.5.3 Ultrasonic Sensor

Ultrasonic sensors can be used to detect the presence of something in the range of the sensor and the distance between the sensor and that body, sometimes called range finders. There are many types of these sensors; the type implemented in this system is the LV-MaxSonar-EZ0 (seefigure2.4).

The LV-MaxSonar-EZ0 sensor operates at range of voltage between 2.5V to 5.5V provide ranging up to 6.45 meter (255 inch) with a controlled beam width. This sensor output ranges in three manners, as a Pulse Width Modulation (PWM) representing the range with scaling factor of 147uS per inch. Another output as a serial data represented in ASCII code. The third output as an analog voltage with sensitivity depending on the voltage source, for example with a supply of 5V yields ~9.8mV/in. and 3.3V yields ~6.4mV/in. The analog output is the type used in this system, due to the simplicity of reading the range in analog signal using the PIC microcontroller that provides up to 13 analog inputs. [19]

Figure 2.4: LV-MaxSonar-EZ0 Range Finder.

2.5.4 Microcontroller

A microcontroller (also microcontroller unit, MCU or µC) is a small computer on a single integrated circuit consisting of a relatively simple CPU combined with support functions such as a crystal oscillator, timers, watchdog timer, serial and analog I/O etc. Program memory in the form of EEPROM (Electrically Erasable Programmable Read Only Memory) or ROM (Read Only Memory) is also often included on chip, as well as a typically small amount of RAM (Random Access Memory).

Microcontrollers have long been a convenient interface for embedded systems; they represent the core of the control system for electronic devices in dedicated applications. Thus, in contrast the microprocessors that are used in general purpose applications like personal computers that need high-performance, and multitasking. Microcontrollers contain data and program memory, serial and parallel I/O, timers, external and internal interrupts, and peripherals. These make them a strong choice when implementing control systems.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, remote controls, office machines, appliances, power tools, and toys. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems. [20]

Figure 2.5: PIC Microcontroller

3

Chapter Three

System Design

3.1 Introduction

3.2 Design Option

3.3 General block diagram of system

3.4 Automatic Speech Recognition

3.5 Feature Extraction Principles

3.6 Pattern Recognition

3.7 Sensor Design

Chapter Three

System Design

3.1 Introduction

The conceptual design is the most important thing in any project, because this part usually describes the heart of project, and gives the reader a vision about the project and make him familiar with the work. This chapter specifies the design procedure of project and gives a description for all ideas included in this project.

We will start this chapter by showing the design option, then we will put a block diagram for the project to show how the parts of the project interconnected and interact with each other. After that we will show a flowchart which summarizes all steps of Mell Frequency Cepestral Coefficients (MFCC) method, in order to get the feature extraction of the words, and how these steps related with each other to give the final result of the project. Finally, we will show the method used to compare the sequence from MFCC with the stored sequence.

3.2 Design Option

In this project we follow a strategy to choose the right hardware for speech recognition, Such as flexibility, reusability, high performance, short development time, and cheap.

The following is the comparison of some alternative available for Digital Signal Processing with DSP Kit (hardware applied in our project):

- The FPGA alternative

Field-Programmable Gate Arrays has become the key components in high performance digital signal processing systems, FPGA has advantages of short development cycle, low-risk and has the capability of being reconfigurable within a system.

However, FPGAs are significantly more expensive and typically have much higher power dissipation than DSPs with similar functionality.

- The ASIC alternative

Application-specific ICs can be used to perform specific functions extremely well like speech recognition, and can be made quite power efficient. However, since ASICs are not fielded reprogrammable, they cannot be changed after development. Consequently, every new version of the product requires a redesign and has to be reproduced .

On the other hand, programmable DSPs can merely change the software program without changing the hardware, which greatly reduces the development cost and makes aftermarket feature enhancements possible with mere code, downloads.

3.3 General block diagram of system

Acoustical signals resulting from a voice commands, will receive by a microphone, which will convert to electrical signals and pass to DSP kit which will process this signals and extract main features to make a decision about voice recognition. Depending on DSP kit signals processing and its decision will send as a special pulse to a microcontroller if it found that matching occur in features between incoming signal from a microphone and features for some special words store previously in kit's RAM.

Microcontroller is very important in this system, its work as interface between a kit and wheelchair, and it's responsible about the movement and direction of wheelchair.

Communication group will focus and specialize in signal processing with DSP kit to extract the features of speech commands to make voice recognition. While biomedical group will programming a microcontroller and building a wheelchair.

The following figure generally views the main component in system and how they were connected to each other.

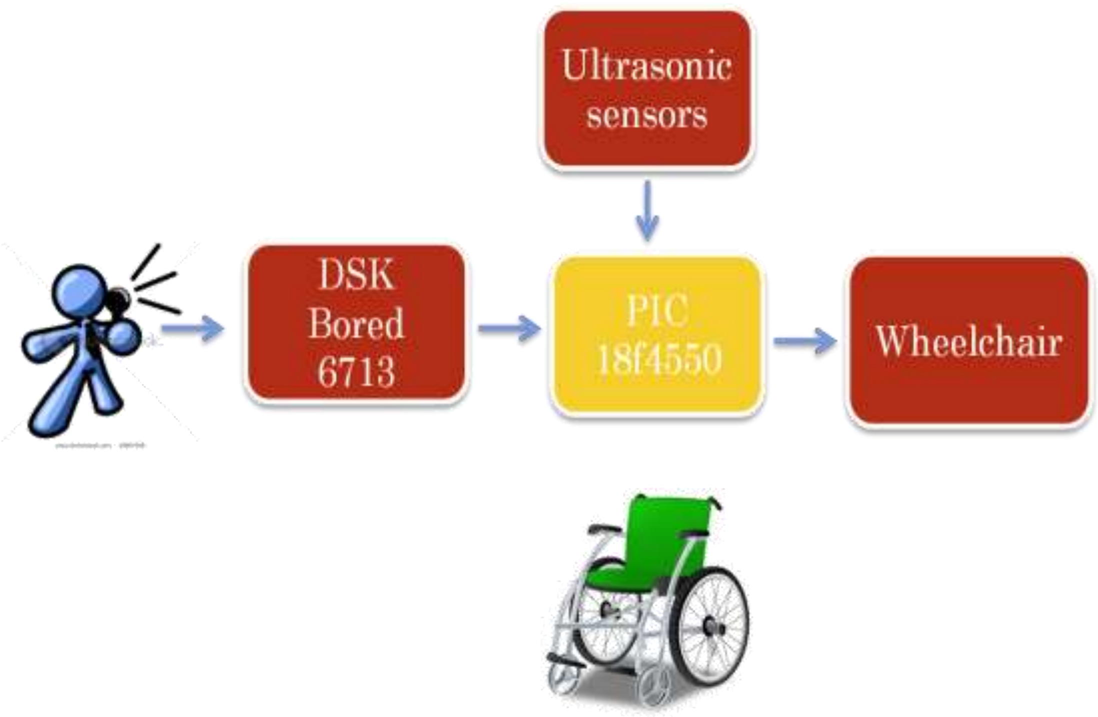


Figure 3.1: General block diagram of the system

3.4 Automatic Speech Recognition

Generally, any speech recognition mainly consist of two parts, front-end part and pattern recognition part as shown in figure (3.2).

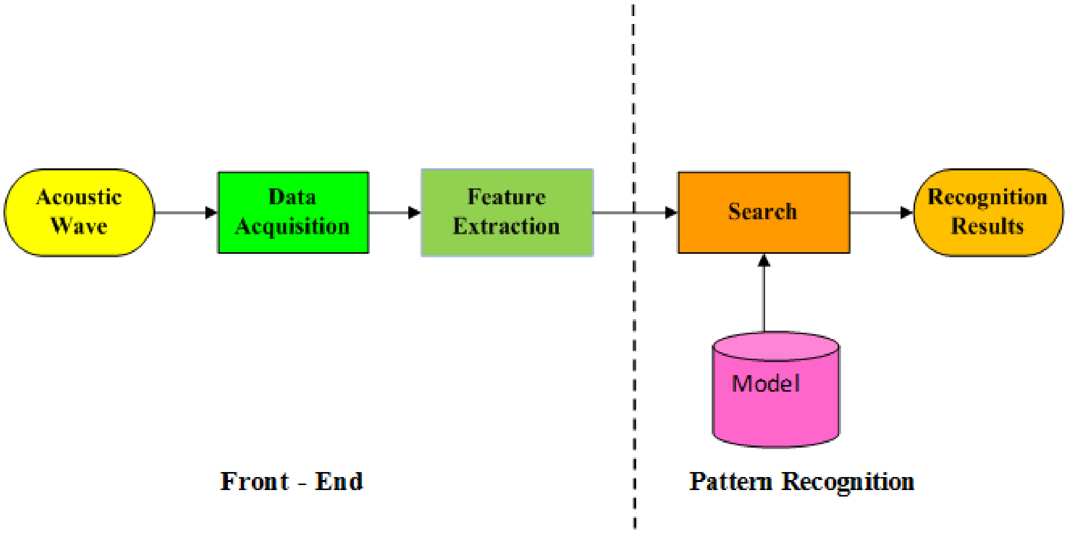


Figure 3.2: Block diagram of ASR system

Typically, front- end part includes two models, namely data acquisition and feature extraction. The data acquisition consists of a microphone which will receive a speech signal, and codec to generate digital speech data. While feature extraction means extract acoustic features from the speech waveform and represent it as a patterns of numbers. There are many techniques to do feature extraction such as, linear predictive coding (LPC), perceptual linear prediction (PLP), and mel-frequency cepstral coefficients (MFCC) the one which will be used it in this project.

Pattern recognition means select an efficient way to compare between pattern produce in feature extraction step of present command, and patterns store previously of special commands to identify matching occurred or not.

3.5 Feature Extraction Principles

Figure (3.3) is the detailed block diagram of the feature extraction processing. Feature extraction is done on short time. The speech signal is divided into frame each frame overlapped with the next frames. From each frame, feature extraction parameters form a pattern, the pattern will compare with the pattern which stored in a model, and then a decision will make to give pattern recognition.

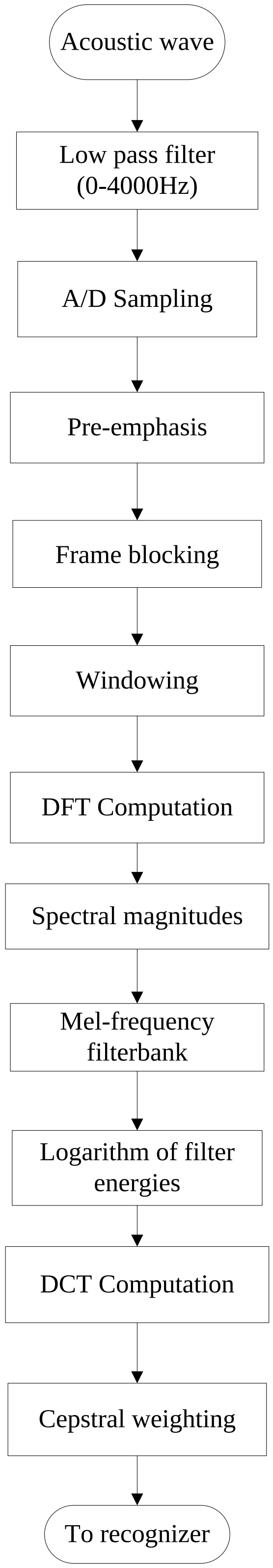


Figure 3.3: Block diagram of MFCC

3.5.1 Low Pass Filter

Humans can hear voices contain frequencies lies between 20-20000Hz, also they can produce voices with high frequencies, but nature speech contains low frequencies and most features located at that band between 0-4000Hz, so that we need a low pass filter to accept low frequencies and reject other high frequencies as a microphone response to all frequencies up to 18 KHz.

3.5.2 A/D Sampling

The first step for the user to speak a word or phrase into a microphone. The electrical signal from the microphone is digitized by an "analog-to-digital (A/D) converter". Because speech is relatively low bandwidth and the most feature of speech in 4 kHz, so we need at least 2 samples per cycle, 8000 samples/sec (8kHz) is sufficient for most feature extracting "Nyquist Theorem".

3.5.3 Pre-emphasis

The spectrum for voiced segment has more energy at lower frequencies so we use pre-emphasis to boosting the energy of high frequency, pre-emphasis using to enhance the specific information in the higher frequencies of speech, and negate the effect of energy decrease in higher frequencies in order to enable proper analysis on the whole spectrum of the speech signal. The speech signal s(n) is sent to a high-pass filter:

$$S_2(n) = s(n) - \alpha * s(n-1) \qquad (3.1)$$

Where $S_2(n)$ is the output signal and the value of α is usually between 0.9 and 1.0. The z-transform of the filter is

$$\mathrm{H(z)} = 1 - \alpha * \mathrm{z}^{-1} \qquad (3.2)$$

By using this method, the spectrum magnitude of the outgoing pre-emphasized speech will have a 20 dB boost in the upper frequencies and a 32dB increase at the Nyquist frequency.

3.5.4 Frame Blocking

Speech signal stay stationary in a short period of time interval. For this reason, speech signals are processed in short time intervals (frame). It is divided into frames of N sample, the sample rate is 8kHz and the frame size N_0=200 sample point, then the frame duration is 200/8000=25ms. Each frame overlaps its previous frame by 80 point (10ms), then the frame rate is 8000/(200-80)=67 frame per second. The goal of the overlapping scheme is to smooth the transition from frame to frame. "The first frame consists of the first N_0 samples. The second frame begins M samples after the first frame and overlaps it by N_0- M samples and so on".

3.5.5 Windowing

Each frame then multiplied by a fixed length window $[h(n)]_{n=0}^{N0-1}$, where N_0 is the length of a frame. Window functions are signals that are concentrated in time, often of limited duration N_0. This is done in order to minimize the signal discontinuities at the beginning and end of each frame. The concept here is to minimize the spectral distortion by using the window to taper the signal to zero at the beginning and end of each frame. We define the window as $h(n), 0 \leq n \leq N_0$, where N_0 is the number of samples in each frame, and then the result of windowing is the signal

$$x(n) = s(n) * h(n) , \qquad 0 \leq n \leq N_0 \tag{3.3}$$

Hamming window is used, which has the form:

$$h(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) , \qquad 0 \leq n \leq N_0 \tag{3.4}$$

3.5.6 Discrete Fourier Transform (DFT) computation

Spectral analysis shows that different timbres in speech signals corresponds to different energy distribution over frequencies. So, that after windowing the speech signal, Discrete Fourier Transform (DFT) is used to transfer these time-domain samples into frequency-domain ones. There is a family of fast algorithms to compute the DFT, which are called Fast Fourier Transforms (FFT). Direct computation of the DFT from Eq.3.5 requires N^2 operations. Meanwhile, the FFT algorithm only requires on the order of N $\log_2$ N operations, so it is widely used for speech processing to transfer speech data from time domain to frequency domain.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j 2\pi n k}{N}} \qquad 0 \leq k < N \tag{3.5}$$

If the number of FFT points, N, is larger than the frame size N_0, N - N_0 zeros are usually inserted after the N_0 speech samples. So that, each frame of window size N_0=200 samples (25ms) is zero padded to form an extended frame of 256 samples for 8 kHz sampling rate.

3.5.7 Spectral Magnitudes

Generally speaking, the signal *X (k)* is a complex value containing the real and image parts. But in the speech recognition system which deals with the real speech signal, the complex value is always ignored by researchers. Therefore, only the magnitude of the complex value X(k) is utilized in this situation. If we assume the real and image parts of *X (k)* are *Re (X (k))* and *Im (X (k)),* then the spectral magnitude of the speech signal should be

$$|X(k)| = \sqrt{Re(X(k))^2 + Im(X(k))^2} \tag{3.6}$$

3.5.8 Mel-frequency filter bank

Human's ears are more sensitive and have higher resolution to low frequency compared to high frequency. Therefore, the filter bank should be defined to emphasize the low frequency over the high frequency. We multiply the magnitude frequency response by a set of 20 mel frequency filter bank as in equation(2.10). These overlapping mel frequency filters bank are made such that their center frequencies are equidistant on the Mel scale, and each filter starts and ends at the centre of the adjacent filter.

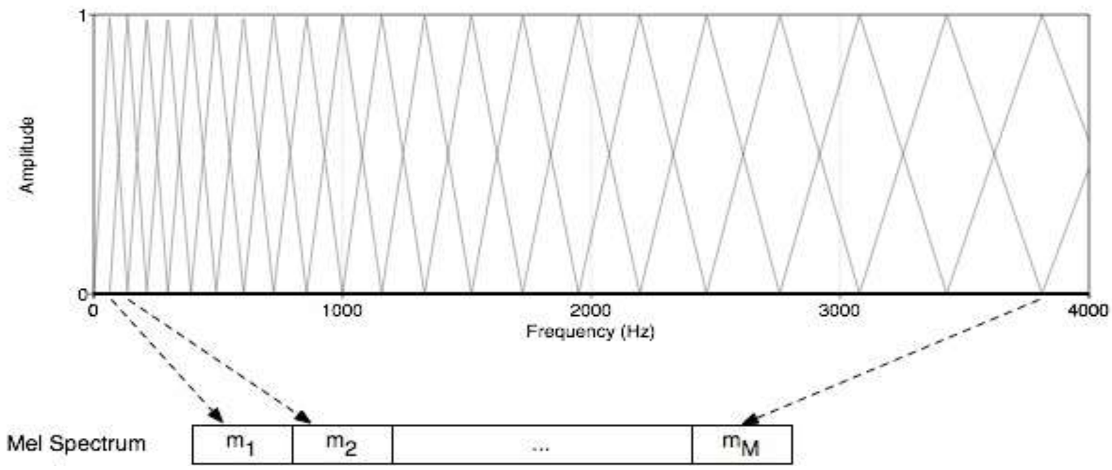


Figure 3.4: Filter Bank in Mel frequency scale

Note that this filter bank is applied in the frequency domain; therefore it simply amounts to taking those triangle-shape windows in the Figure 3.4 on the spectrum. A useful way of thinking about this Mel-wrapping filter bank is to view each filter as a histogram bin (where bins have overlap) in the frequency domain. Each filter output is the sum of its filtered spectral components as shown in Figure (3.5).

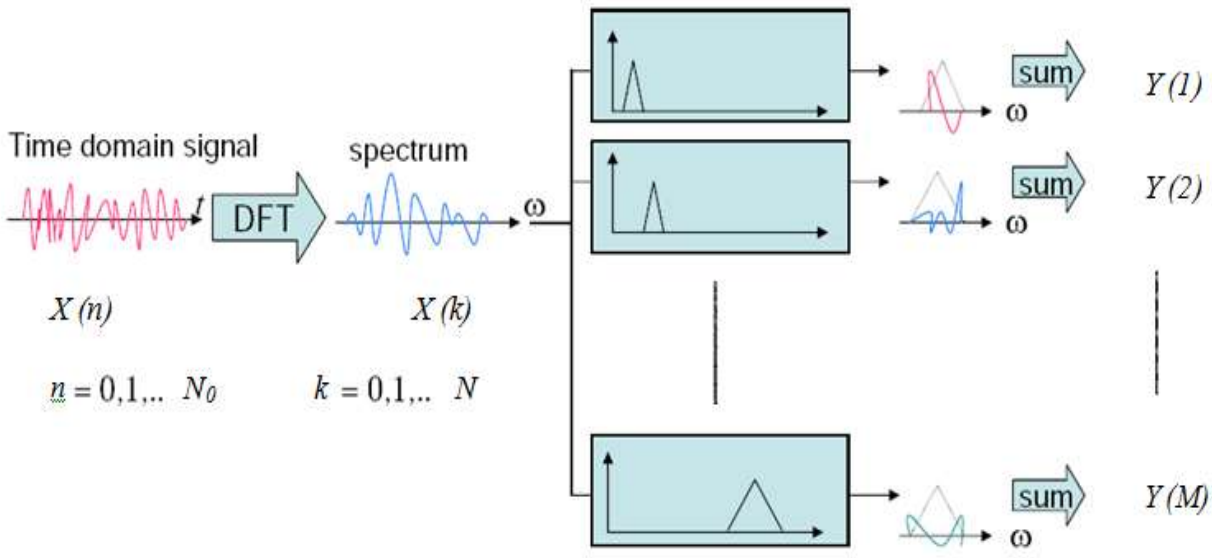


Figure 3.5: Mel Frequency Filter Bank Output

3.5.9 Logarithm of Filter Energies

The log energy of each filter's output calculated as following:

$$S(m) = log_{10}\left(\sum_{k=0}^{N-1} |X(k)|^2 \cdot H_m(k)\right) , \qquad 0 \le m < M \qquad (3.7)$$

We do that as logarithm scale compresses dynamic range of values, also because Human response to signal level is logarithmic not linear. Beside that humans less sensitive to slight differences in amplitude at high amplitudes than low amplitudes; in addition, phase information not important in speech recognition.

3.5.10 Discrete Cosine Transform (DCT)

A discrete cosine transform (DCT) expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies. It's used to compress the spectral information into a set of low order coefficients by gathers most of the information in the signal to its lower order coefficients, by discarding the higher order. This representation is called the Mel-cepstrum.

Therefore complexity and computational cost can be reduced. Typically the number of coefficients, K, for recognition ranges between 8 and 13 the equation is as following:

$$C(k) = \sum_{m=0}^{M-1} S(m) COS\left(\pi . \frac{k\left(m+\frac{1}{2}\right)}{M}\right), \qquad 0 \le k < K \qquad (3.8)$$

3.5.11 Cepstral Weighting

A weighting window (named liftering window) is applied after decorrelating the cepstral coefficients. In our project, the sinusoidal lifter as shown in equation is utilized to minimize the sensitivities by lessening the higher and lower cepstral coefficients.

$$\hat{\mathrm{C}}(\mathrm{k}) = \mathrm{C}(\mathrm{k}).\left[1 + \frac{\mathrm{K}}{2}.\sin\left(\frac{\mathrm{k}\pi}{\mathrm{K}}\right)\right], \qquad 0 \le \mathrm{k} < K \qquad (3.9)$$

3.6 Pattern Recognition

To find a way for make a comparison between the stored sequence and the entered sequence we use the Euclidian distance and support our comparison by using cross correlation.

3.6.1 Euclidian Distance

This is an effective method to compare two pattern that include a non-digital digit (i.e. integer number) according to the following equation if

$$X = [x_1\ x_2 \ldots \ldots x_n]\ , \quad Y = [y_1\ y_2 \ldots . . y_n]$$

Then

$$d(X,Y) = \sqrt{\sum_{i=1}^{n} (X_i - Y_i)^2} \qquad (3.10)$$

Where X represent input pattern, Y represent pattern stored previously for special commands in model.

3.7 Sensor Design

We will use the (LV-MaxSonar-EZ0) sensor, and the operation principle of this sensor is relates the distance that measure away from it with the yield output voltage according to the relation (1 inches =9.8mV) , in this section we will discuss two cases that are; the straight direction case and the slope direction case.

3.7.1 Straight Direction Case

Here, we assuming that the distance between the wheelchair and the obstacle around (d=70cm=27.55inch) as it shown in figure (3.6), according to this distance, sensor will output voltage Vout=(27.55inch)*9.8mv/1inch)=0.27v.

The following figure describes some measurement, and the calculation above built according to this measurement.

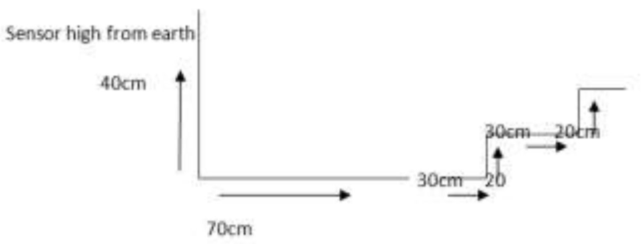


Figure 3.6: Distance Measurements for Straight Direction Case.

There's a flowchart in the figure (3.7) that show the behavior of the system at the straight direction case, and specify detecting object mechanism.

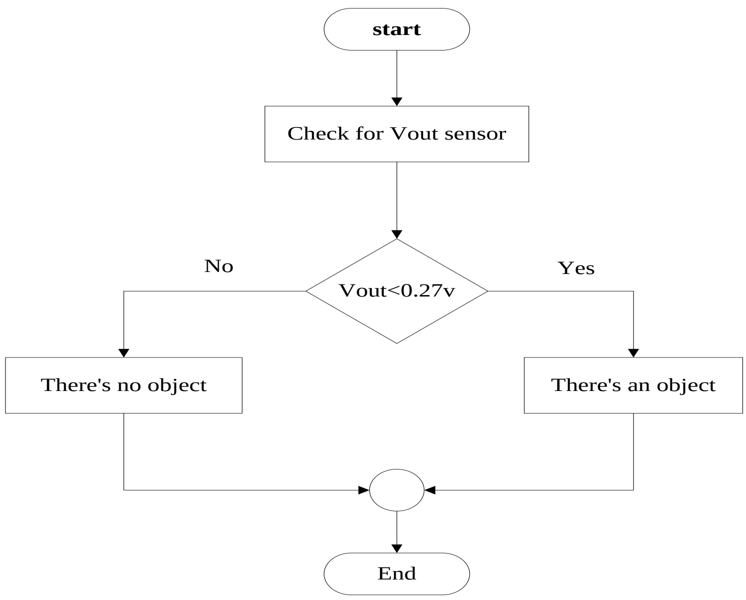


Figure 3.7: System Behavior for Straight Direction Case

3.7.2 Slope Direction Case

According to figure (3.8), the vertical distance (d=20cm=7.87inch), so the output voltage Vout=(7.87 *9.8mv/1inch)=0.077v,(the sensor located on the arm).

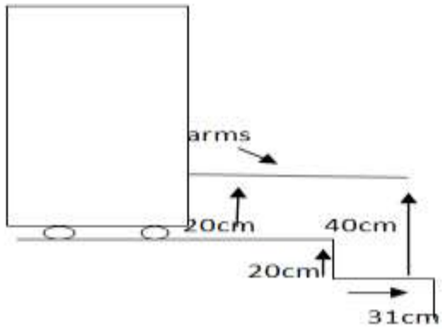


Figure 3.8: Distance Measurements for Slope Direction Case.

The following flow chart describes the system behavior in the slope direction case.

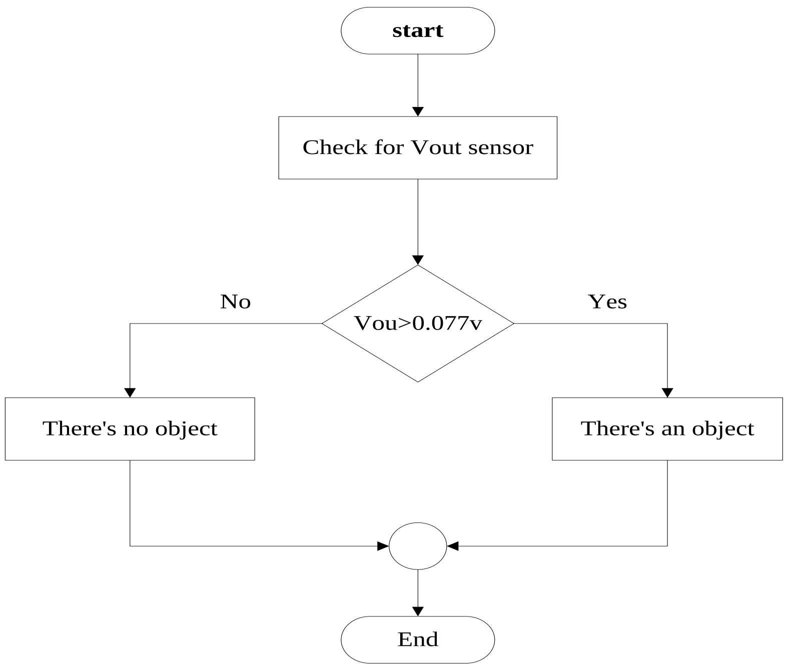


Figure 3.9: System behavior for slope direction case.

4

Chapter Four

System Implementation

Chapter Four

System Implementation

4.1 Introduction

In the previous chapter we presented the design principles for feature extraction from every voice command which controlled wheelchair by using mel-frequency cepestral coefficients, In this chapter we will discuss the detailed software design and interfacing connection between sensors and microcontroller.

This chapter provides a description about project details which can be summarized by understanding the way how a system builds to achieve MFCC.

The following figure (4.1) show the general block diagram for every subsystem build to achieve MFCC.

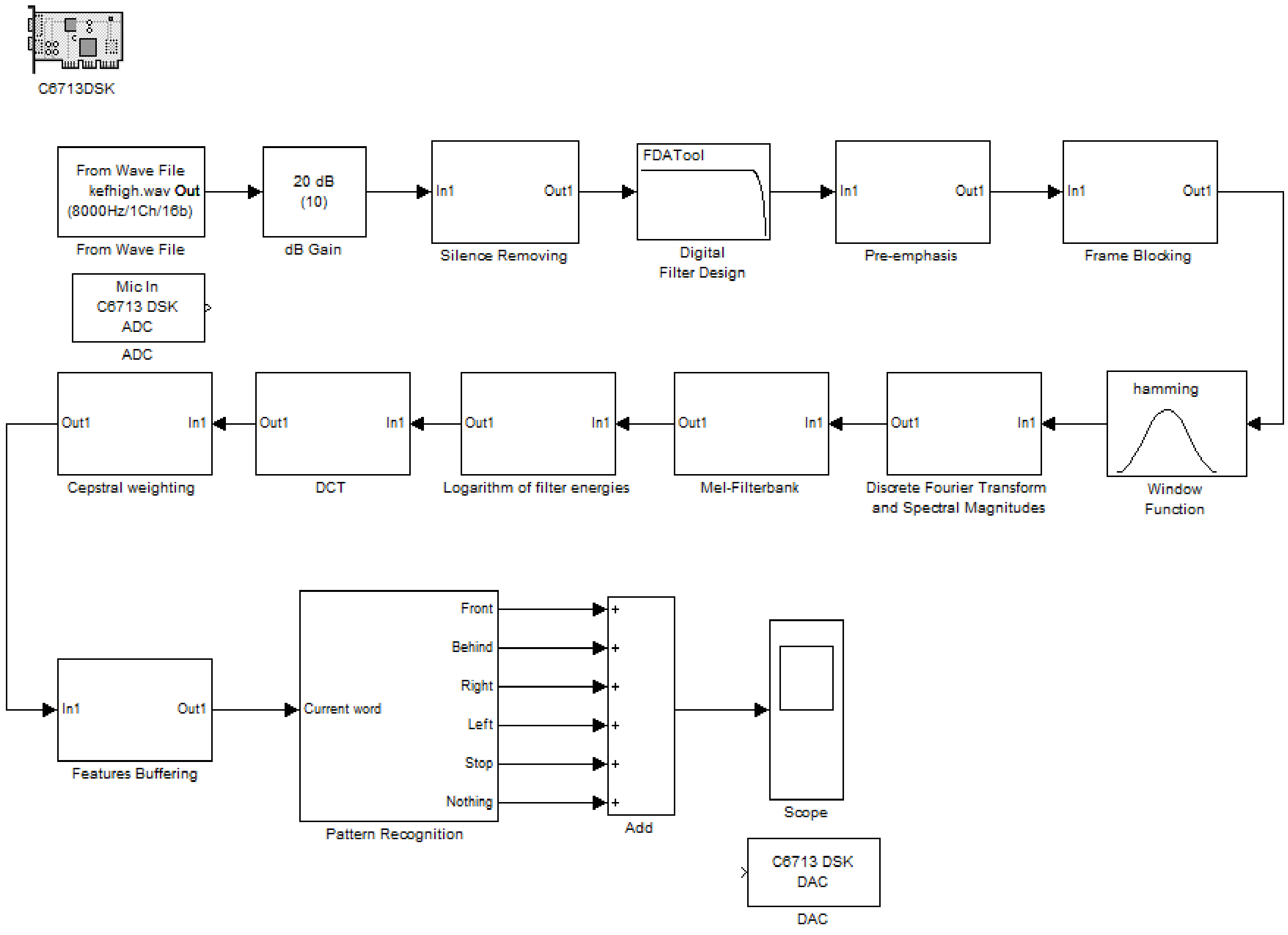


Figure 4.1: General Block Diagram Matlab System

4.2 Input Voice Commands

A voice command enter the DSK Kit from MIC, then system will extract the feature for each voice command by using the suitable matlab simulink blocks, which will describe in detailed in the next section.

Each command represent the neutrality of the human voice, most concentration of information for each command lies below 4 KHz.

In this project we have a five command word which control the wheelchair, Front{ أمام } Behind{ خلف},Stop{ قف }, Left{ يسار},Right{يمين }, the figures below show the plot for each word, x-axis shows the time , y-axis shows the amplitude.

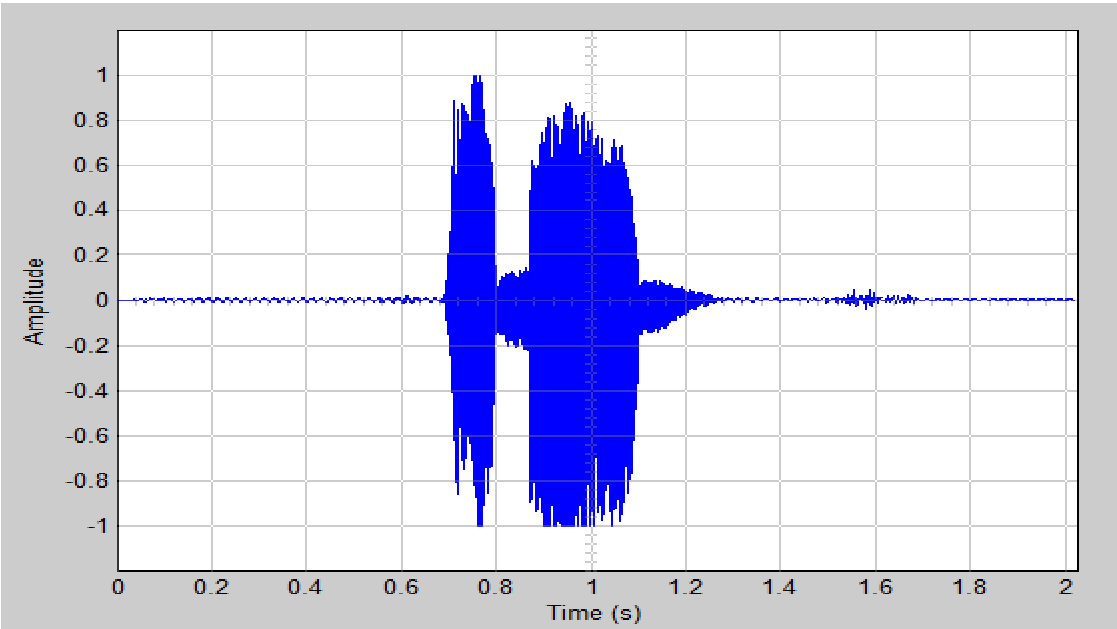


Figure 4.2: Front (أمام) word shape

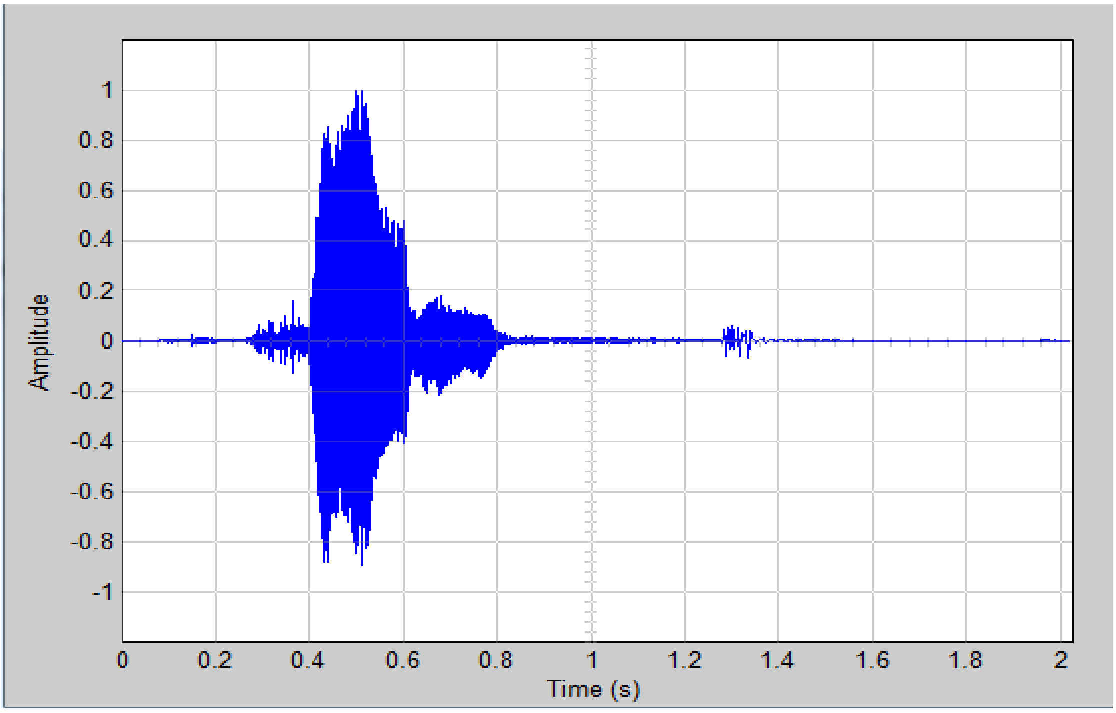


Figure 4.3: Behind (خلف) word shape

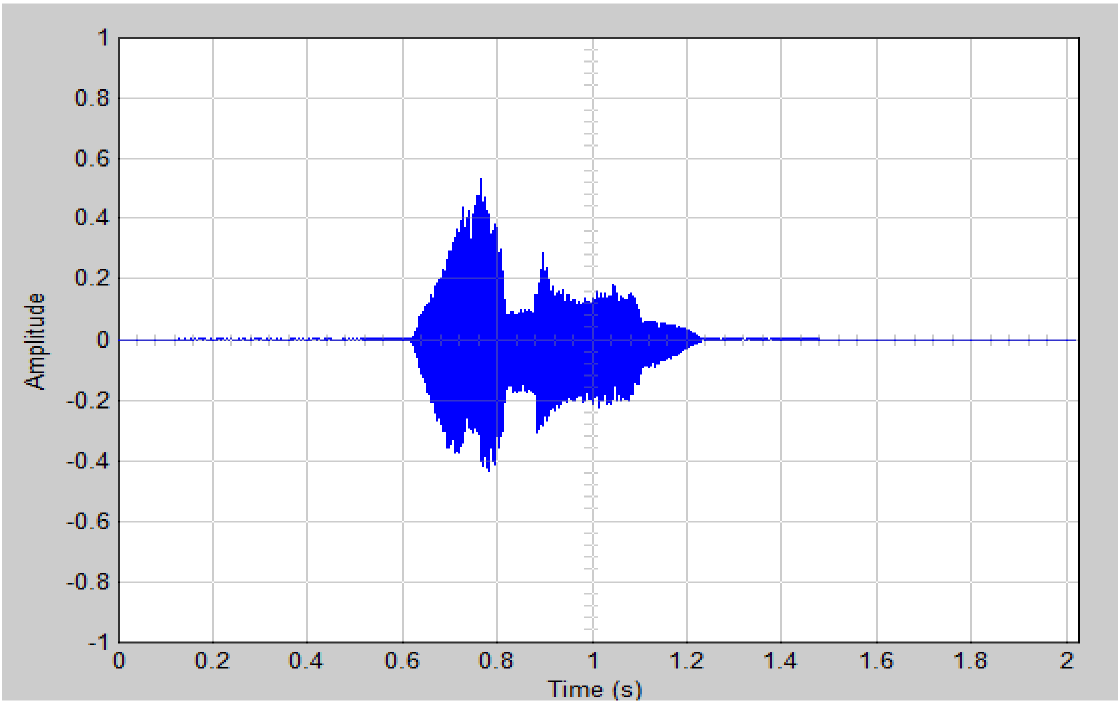


Figure 4.4: Right (يمين) word shape

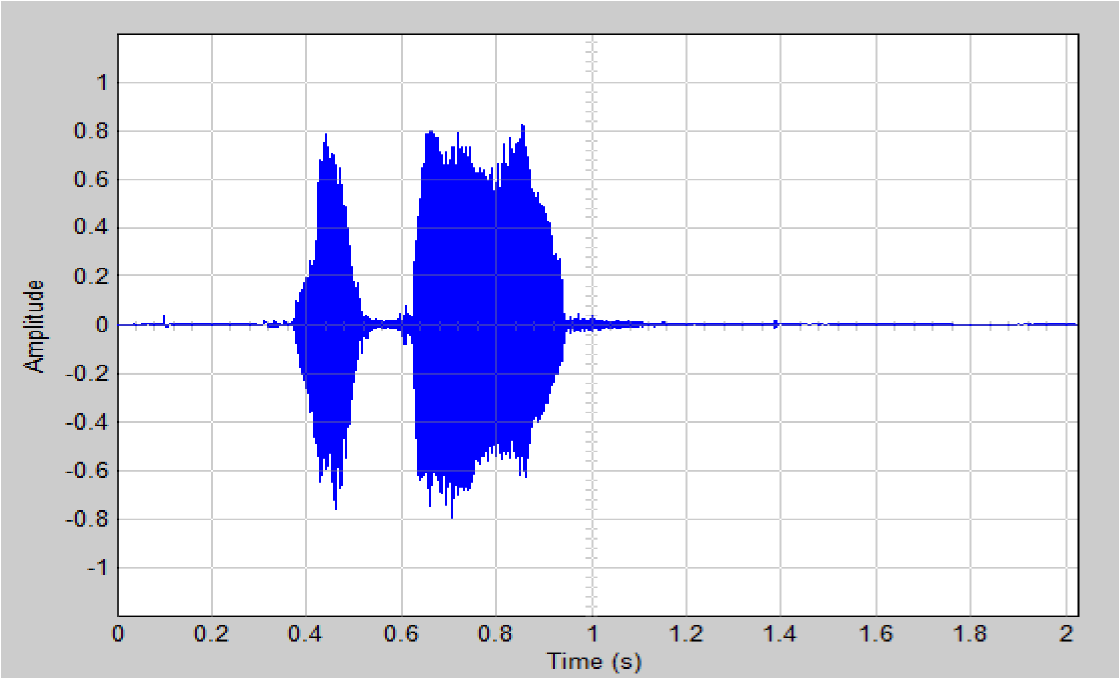


Figure 4.5: Left (يسار) word shape

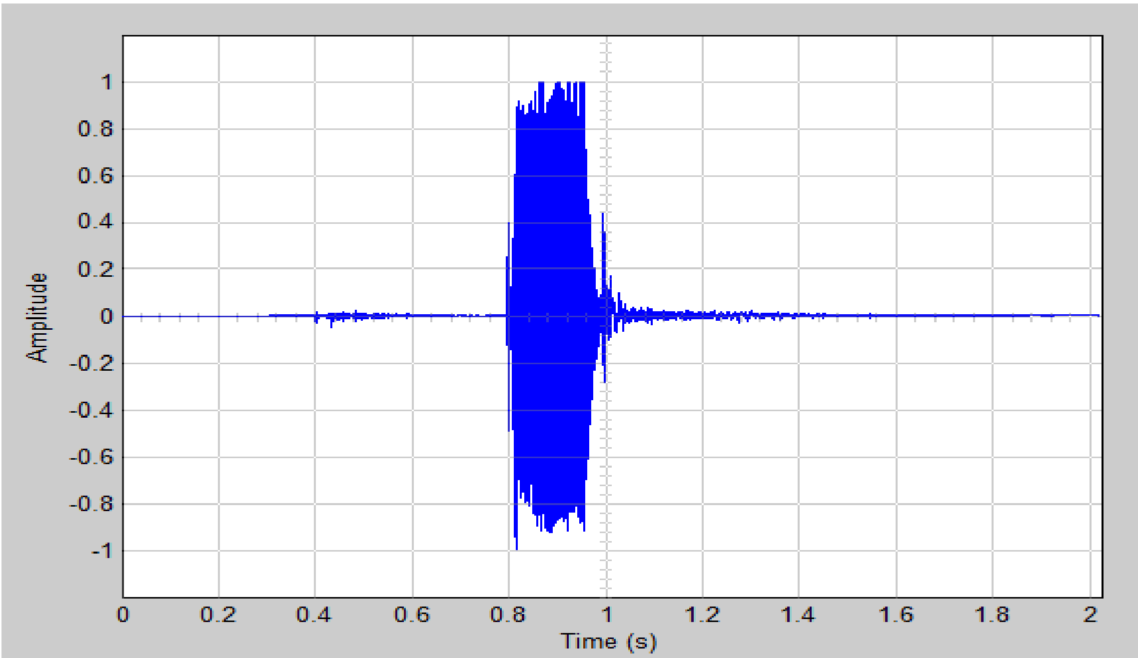


Figure 4.6: Stop (قف) word shape

4.3 Matlab Simulink Blocks

By using SIMULINK in MATLB program a system build to extract the feature for each input command, the system consist many subsystem collecting together to build the whole system, every subsystem do a specific function and contain many block, each block have a specific parameter.

The figure below (4.7) shows the whole system for MFCC method of feature extraction. Blocks collecting together to contain subsystem, each subsystem has its own process on voice command, so we need many subsystem to achieve our goal in modeling the system.

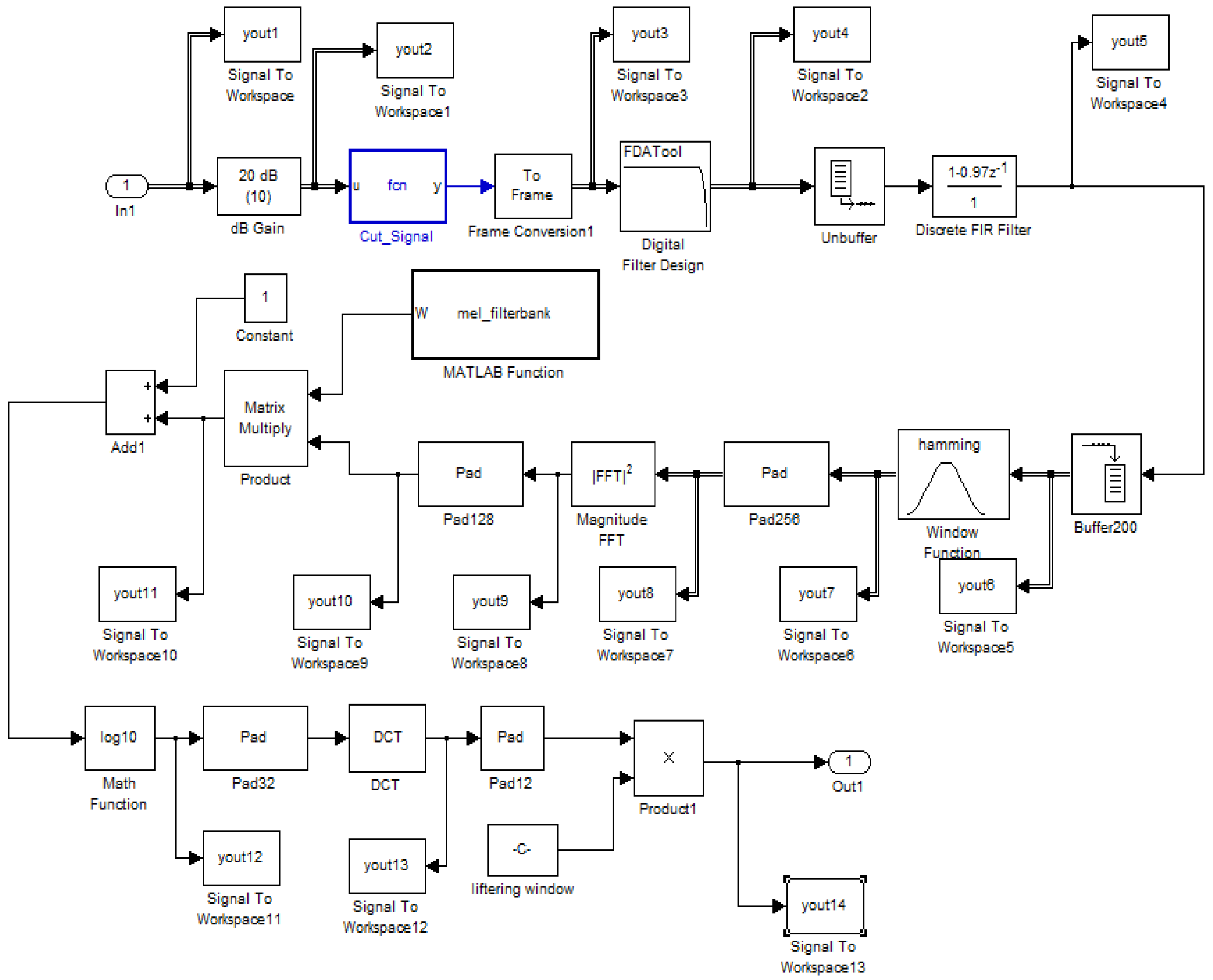


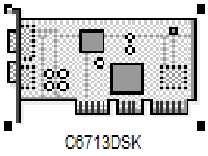
Figure 4.7: Main system blocks of MFCC in MATLAB

Every block in subsystem do its function and pass the result to the next block which do another function, blocks adapted by specifying its parameter to match its main function.

This section shows a detailed description for each subsystem, blocks used in system.

1. Target Preferences block:

This block add to SIMULINK model to provides access to the processor hardware settings we need to configure when generate a project from a Simulink software model.

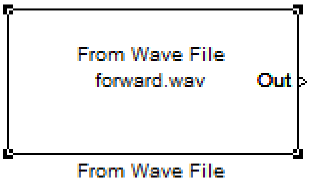


Target Preferences block

This block must be in the model at the top level and not in a subsystem. It stands alone to set the target preferences for the model. Simulink software returns an error when the model does not include a target preferences block or has more than one.

2. From Wave File block:

This block used in the model only during building and testing stage, it will be replaces in the final form when complete whole system with a C6713 DSK ADC Block.



From Wave File block

This block streams audio data from a Microsoft Wave (.wav) file and generates a signal with one of the data types and amplitude ranges.

Parameter specification for this block as following:

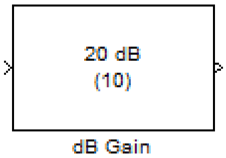
- File name: name of file.
- Samples per output frame: the number of samples in each output frame is 16200.
- Minimum number of samples for each read from file: 16200 number of consecutive samples to acquire from the file.
- Data type: The data type setting determines the output's amplitude range, double ±1 .

3. dB gain block:

This block used to enhance the gain of voice command, dB Gain block multiplies the input by the decibel values specified in the Gain parameter.

Parameter specification for this block as following:

- Gain: 20 dB.
- Input signal: amplitude.



dB gain Block

4. C6713 DSK ADC Block:

This block used in the final model instead of From Wave File block, C6713 DSK ADC (analog-to-digital converter) block used to capture and digitize analog signals from external sources, such as audio devices. This block lets us use the audio coder-decoder module (codec) on the C6713 DSK to convert an analog input signal to a digital signal for the digital signal processor.

Parameter specification for this block as following:

- ADC source: Mic In.
- Sampling rate (Hz): 8 kHz.
- Output data type: double.
- Sample per frame: 16200.

C6713 DSK ADC Block

5. Removing silent and noise from speech signal subsystem:

This subsystem used in our system to cut the noise and silent signal at the beginning and end in command word before any process in the signal, this subsystem consist of two Frame conversion block and embedded MATLAB function block.

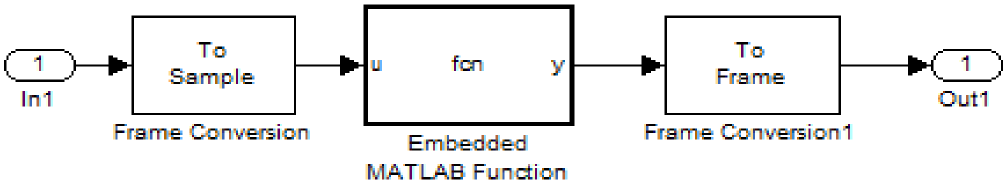


Figure 4.8: Removing silent and noise from speech signal subsystem

The parameter specification for each block in the subsystem as following:

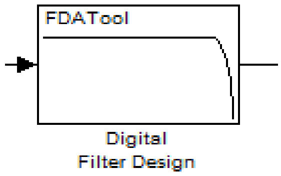
- ✓ Frame conversion block: The Frame Conversion block passes the input through to the output and sets the output sampling mode to the value of the sampling mode of output signal parameter, which can be either frame-based or sample-based.
  - Sampling mode of output signal: Sample-based.

- ✓ Embedded Matlab Function block: is a block allow us to write a matlab code within, we use this block to write a Matlab code which cut the signal at the beginning when the signal amplitude is less than .85, and cut the signal at the end when the amplitude less than .75, also this signal decrease number of samples in each output frame from 16200 point to be 6000 point, if the user saying nothing which means silent state this block generate zeros matrix with length 6000 point.

- ✓ Frame conversion block: The Frame Conversion block passes the input through to the output and sets the output sampling mode to the value of the sampling mode of output signal parameter, which can be either frame-based or sample-based.
    - Sampling mode of output signal: Frame-based.

4.3.1 Feature extraction subsystem

As we mention in chapter three (3.5) which describe MFCC process, this section will show all subsystem used to achieve MFCC in SIMULINK MATLAB Blocks.



Digital filter design block

1. Digital filter design block used to build an IIR low pass filter which will pass all frequency below 4 KHz for each command word. FDATool used to design a filter by adjusting its specific parameters as following:

- Response Type: lowpass.
- Design Method: IIR Butterworth.
- Filter Order: Minimum order.
- Option Match exactly: stop band.
- Frequency Specifications: Unit (kHz), Fs (8), Fpass (3.5), Fstop (3.9).
- Magnitude Specifications: Units (dB), Apass(1),Astop(80).

The Digital Filter Design block after adjust the parameters represent a low pass filter with a discrete form IIR transposed structure, filter pass all frequency up to 4KHz, by Nequist frequency (half sampling frequency),and stop frequencies greater than or equal to 4KHz theoretical, but in practical it must defined as fpass=3.5kHz and fstop=3.9KHz to a void aliasing.

Magnitude Specifications control the order of the filter, this filter is seven orders and it’s stable.

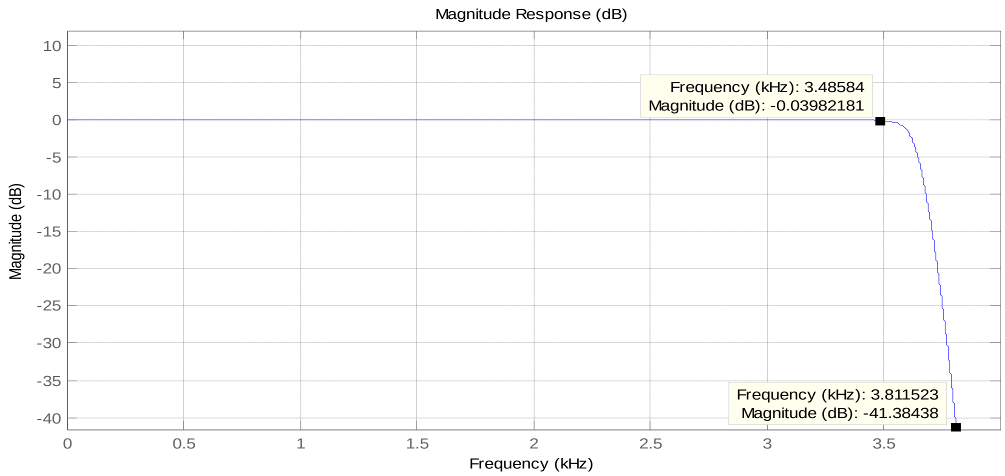


Figure 4.9: Low Pass Filter Magnitude Response

2. Pre-emphasis subsystem

To pre-emphasis the signal, a discrete filter is used, this block can implement a time varying filter with coefficients that change over time. Because this filter treating each element of the input as an individual so we use an unbuffer Block before it to convert the continuous signal to discrete mode.

Discrete filter does not change the dimension of the input signal.

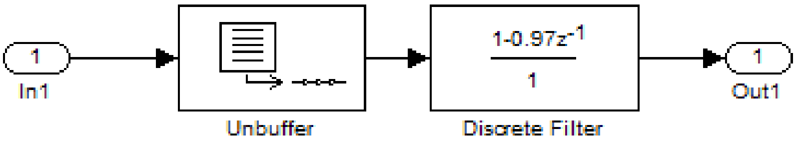


Figure 4.10: Pre-emphasis subsystem

The parameter specification for each block in the subsystem as following:

✓ Unbuffer Block: its convert a frame to a scalar sample output
  - Initial conditions: 0.

✓ Discrete Filter:
  - Coefficient source: Dialog parameters(Specify whether we want to input the filter coefficients on the block mask)
  - Numerator coefficient: [1 -0.97] (Specify the vector of numerator coefficients of the filter's transfer function).
  - Sample time: -1(Specify the time interval between samples).

3. Frame Blocking subsystem

A Buffer Block used to sequence the input in a frame size of 200 point as we say in chapter (3), overlap between two successive frame is 80 point.

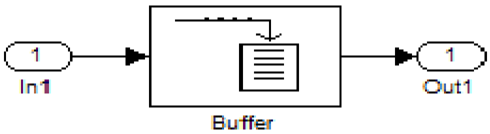


Figure 4.11: Frame blocking subsystem

The parameter specification for Buffer block as following:

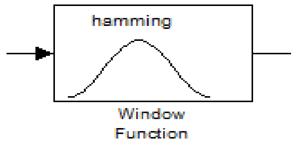
- Output buffer size: 200.
- Buffer overlap: 80.
- Initial conditions: 0.

4. Windowing Block:

A Window Function Block is used to apply a window to an input signal.

The parameter specification for block as following:

- Operation: Apply window to input.
- Window type: Hamming.
- Sampling: Symmetric.



Windowing Block

5. Discrete Fourier Transform and Spectral Magnitudes subsystem

This subsystem consists of two Pad Blocks and Magnitude FFT block. Magnitude FFT block computes a separate estimate for each of the N independent channels and generates an N_{FFT}-by-N matrix output, N_{FFT} is specified by the frame size of the input, which must be a power of 2.

In our case the frame length is 200 point. If we want to apply the FFT, frame size must be 256 point, to achieve that a Pad Block used to pad each frame of window size 200 point to 256 point by inserted 56 zeros to each frame.

Pad block extends the dimension of the input by padding along its dimension. Magnitude FFT block used to take the magnitude square of the FFT samples, and the second pad block to cut the FFT frame and take the first half of each frame.

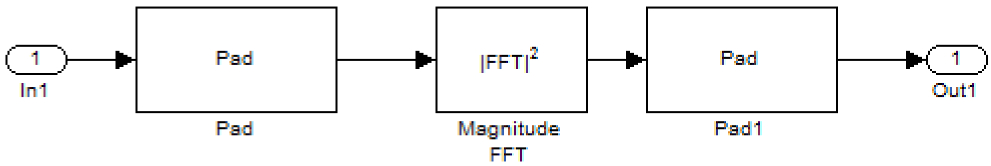


Figure 4.12: Discrete Fourier Transform and Spectral Magnitudes subsystem

The parameter specification for each block in the subsystem as following:

✓ Pad Block: Pad block append or prepend a constant value to the input along specific dimension
- Pad over: columns (Specify the dimensions over which to pad).
- Pad value source: Specify via dialog (Specify the one-based dimensions over which to pad (the pad value can come from an input port or from the dialog).
- Pad value: 0(Specify the constant scalar value with which to pad the input).
- Column size: 256(specify the column length of the output).
- Pad signal at: End (Specify how many values to add to the end of the input signal along the specified dimension).

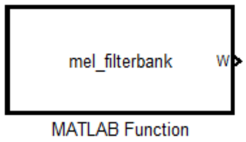
✓ Magnitude FFT block

- Output: Magnitude squared (Specify whether the block computes the magnitude FFT or magnitude-squared FFT of the input).
- Inherit FFT length from input dimensions (Select to use the input frame size as the number of data points, N_{FFT}, on which to perform the FFT).

✓ Pad1 Block: This block used because 256 point from FFT is repeated after 128 point so we cut frame size to be 128 point only

- Pad Over: Columns.
- Pad Value: 0.
- Column Size: 128.
- Pad signal at: end.

6. Mel Frequency Filter Bank Subsystem



Embedded Matlab Function block is a block allow us to write a matlab code within, we use this block to write a function that give us a matrix with (20X128) which represent 20 filters along each FFT frame. Unfortunately DSK bored did not accept this block, to solve this problem, team work build this matrix come from code within embedded function by blocks that shown in figure (4.13).

This Subsystem build to give matrix with(128X20) which represent 20 filters along each FFT frame. We build this subsystem by using 20 constant block that handle the coefficient of Mel Frequency filter, then a matrix concatenate block used to collect constant blocks to form a matrix (128X20) then a transpose block to inverse the matrix to be (20X128).

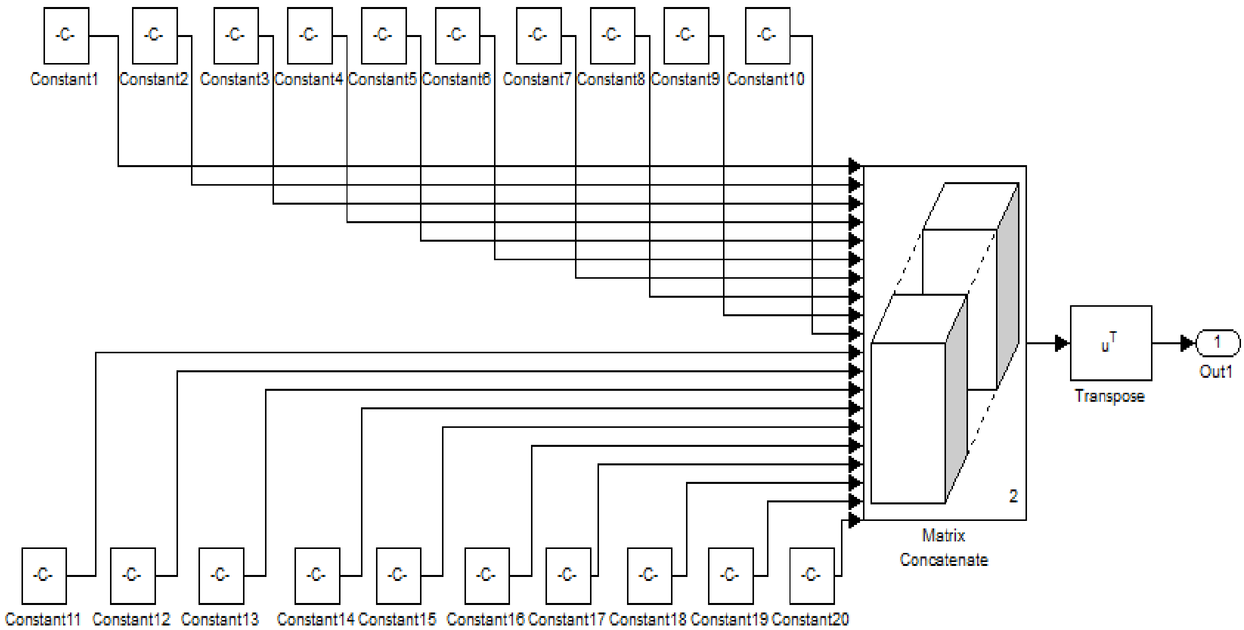


Figure 4.13: Mel Frequency Filter Bank blocks

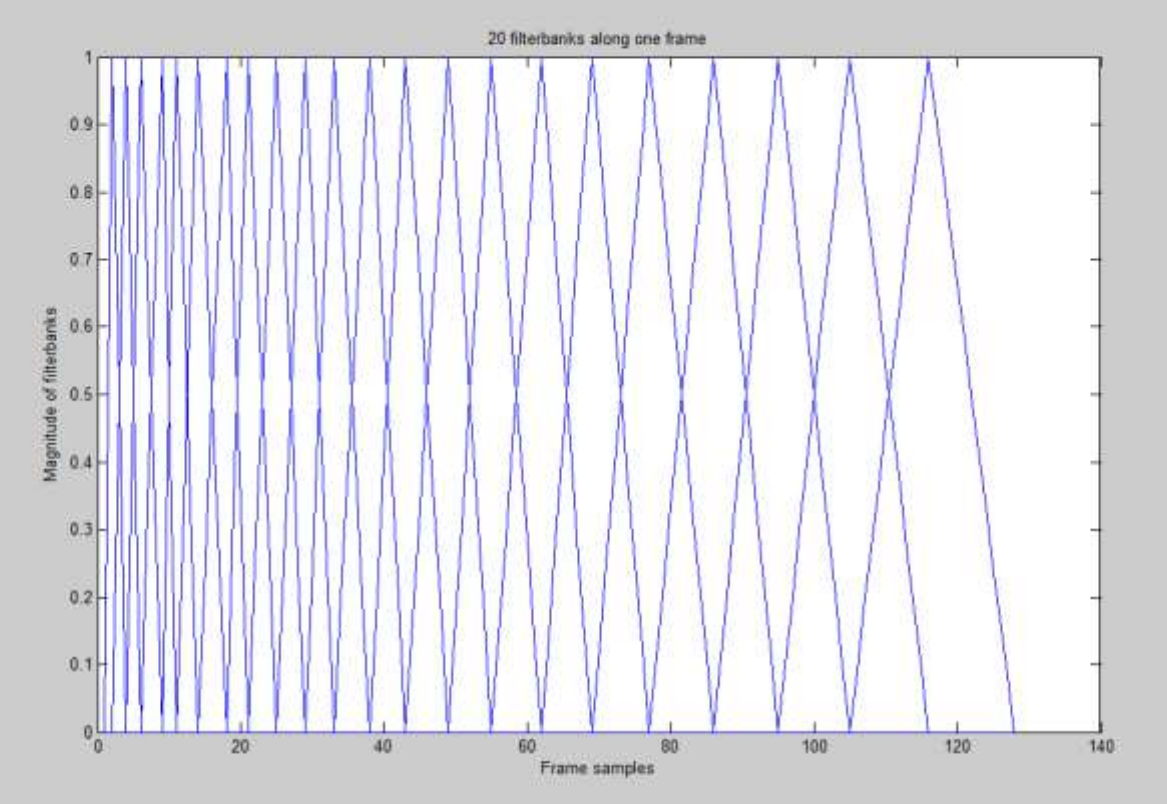


Figure 4.14: Filter Bank along one frame.

- ✓ Constant Block:
  - Constant value: taken from the Mel Frequency Filter bank.
- ✓ Matrix Concatenate block: The Concatenate block concatenates the signals at its inputs to create an output signal whose elements reside in contiguous locations in memory.
  - Number of inputs: 20.
  - Mode: Multidimensional array.
  - Concatenate dimension: 2. (Specifies the output dimension along which to concatenate the input arrays)
- ✓ Transpose block: The Transpose block transposes the M-by-N input matrix to size N-by-M.
  - Hermitian: not selected

After we get Mel Frequency Filter we use a Product block to performs multiplication between the signal and the filter

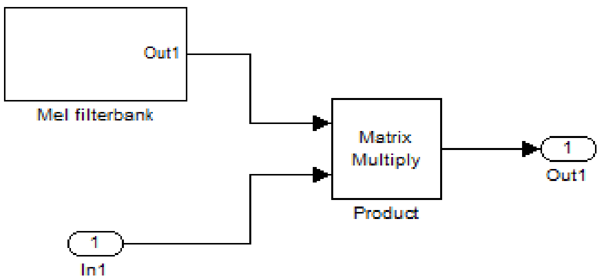


Figure 4.15: Mel Frequency Filter Bank Subsystem

- ✓ Product Block
  - Number of inputs: 2.
  - Multiplication: Matrix

7. Logarithm of filter energies subsystem

By Math function block a $\log_{10}$ is taken for the signal.

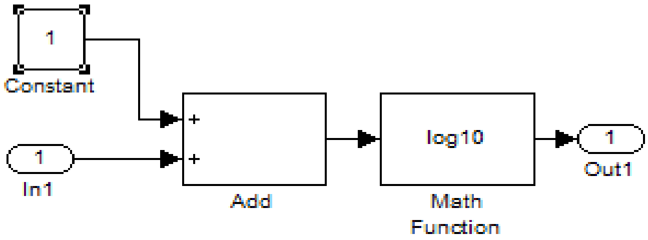


Figure 4.16: Logarithm of filter energies subsystem

The parameter specification for each block in the subsystem as following:

- ✓ An add block used to add one for all element, because there is a zero number in the signal which will enter log10 Block and then a cosine will be taken for it.

  as we know $\log_{10} 0= -\infty$, $\cos(-\infty)$=NAN (Not A Number) in Matlab, this will make an error in pattern recognition subsystem, to avoid this error we add one to the whole signal, this addition will shift the signal up by one and not affected in final result.

- ✓ Math Function Block
  - Function: log10.
  - Output signal type: auto.

8. Discreet Cosine Transform subsystem

The DCT Block computes the unity discreet cosine transform (DCT) for frame base input; the block assumes that each input column is a frame containing M consecutive samples from an independent channel. The frame size M must be a power of two. To work with other frame sizes, use the Pad block to pad the frame size to a power-of-two length.

Each frame contain 20 point, if we take 16 point (2^4) a 20% of the data will be lost, so a lot of features will be lost.

To solve this problem, a pad block used to make frame length 32 point (2^5). Pad frames with zeros at the end of each frame do not change the result as the following equation shows:

$$y(k,l) = w(k) \sum_{m=1}^{M} u(m,l) \cos \frac{\pi(2m-1)(k-1)}{2M}, \quad k = 1,\ldots,M$$

Where

$$w(k) = \begin{cases} \frac{1}{\sqrt{M}}, & k = 1 \\ \sqrt{\frac{2}{M}}, & 2 \le k \le M \end{cases}$$

After take DCT for the signal we want to take 12 point from each frame this 12 point is the final coefficient.

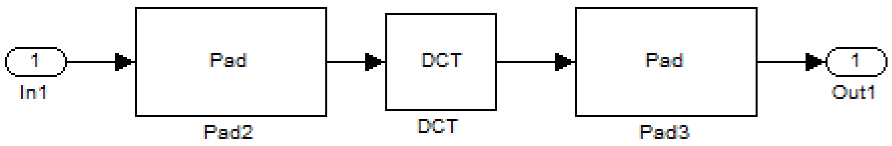


Figure 4.17: Discreet Cosine Transform subsystem

The parameter specification for each block in the subsystem as following:

✓ Pad Block:

This block used because 20 point pad with zeros to be 32 point in each frame

- Pad Over: Columns.
- Pad Value: 0.
- Column Size: 32.
- Pad signal at: end.

✓ DCT Block:

The DCT block computes the unitary discrete cosine transform (DCT) of each frame

- Sine and cosine computation: Table lookup

✓ Pad Block:

This block used to crop a 32 point to be 12 point coefficients from each frame

- Pad Over: Columns.
- Pad Value: 0.
- Column Size: 12.
- Pad signal at: end.

9. Cepstral weighting subsystem

Liftering window Block: it's a constant block contains valuable taken from the equation mention in chapter three (3.5.11).

$$\hat{C}(k) = C(k).\left[1 + \frac{K}{2}.\sin\left(\frac{k\pi}{K}\right)\right] \qquad\qquad 0 \le k < K$$

.

Liftering window multiply with 12 point coefficient to get the final features.

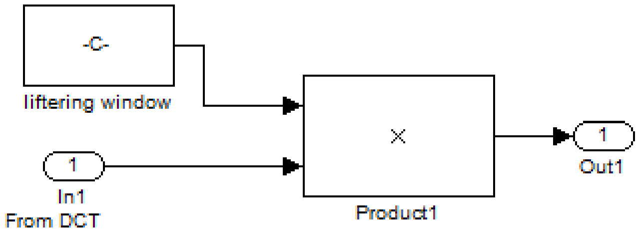


Figure 4.18: Cepstral weighting subsystem

The parameter specification for each block in the subsystem as following:

- ✓ Constant Block:
  - Constant values: taken from the figure below.

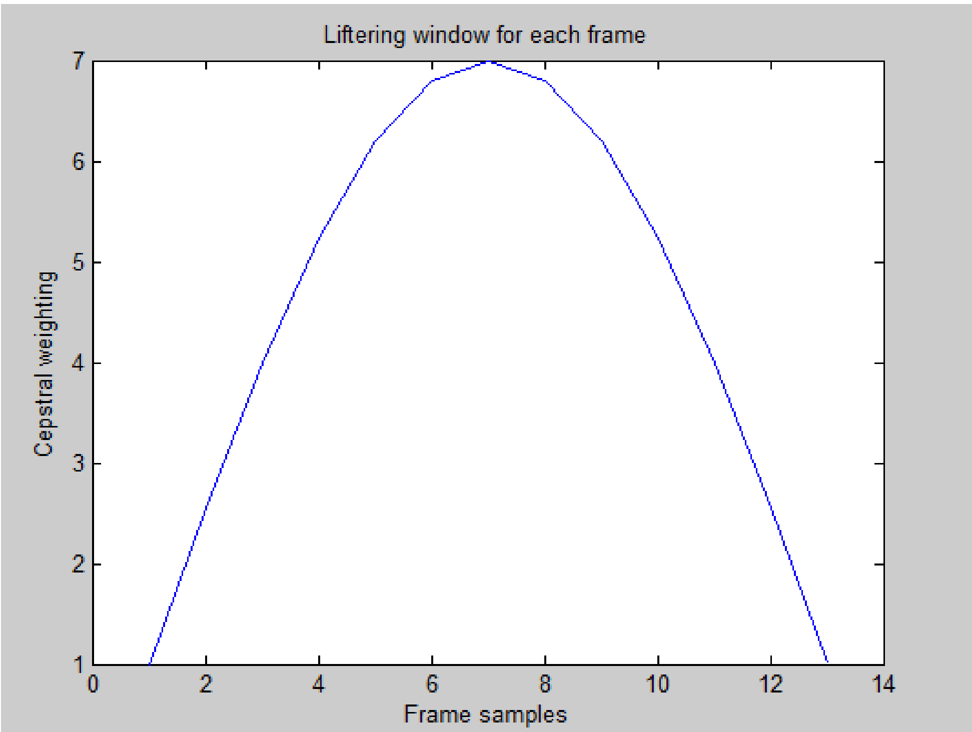


Figure 4.19: Liftering window for each frame

- Interpret vector parameter as 1-D.

✓ Product Block:
  - Number of inputs: 2.
  - Multiplication: Element-wise.

4.3.2 Features buffering subsystem

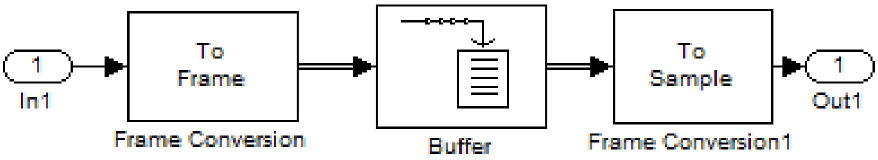


Figure 4.20: Feature buffering subsystem

This subsystem collects and save the data – comes from feature extraction subsystem- for a current word. In a current word, each frame pass in feature extraction subsystem and generate 12 MFCC at the end of that subsystem, these values collect in buffer block with size 1188 point and when this buffer become full it pass its values to pattern recognition subsystem to make decision.

The parameter specification for each block in subsystem as following

✓ Frame conversion block:
  - Sampling mode of output signal: Frame-based.

✓ Buffer block:
  - Output buffer size: 1188.
  - Buffer overlap: 0.
  - Initial conditions: 0.

✓ Frame conversion block:
  - Sampling mode of output signal: Sample-based.

4.3.3 Pattern Recognition subsystem

After we finished Feature extraction process and extract the features for the five words, Front{ أمام } Behind{ خلف},Stop{ قف }, Left{ يسار}, Right{يمين }, by take MFCC for each one, the final stage in this project is to make the decision any words uttered by the person?

In this stage a comparison made between the stored features for the five words and the spoken word.

This subsystem contain many blocks like, constant block, Euclidean distance block, MinMax Block, the figure below show this subsystem.

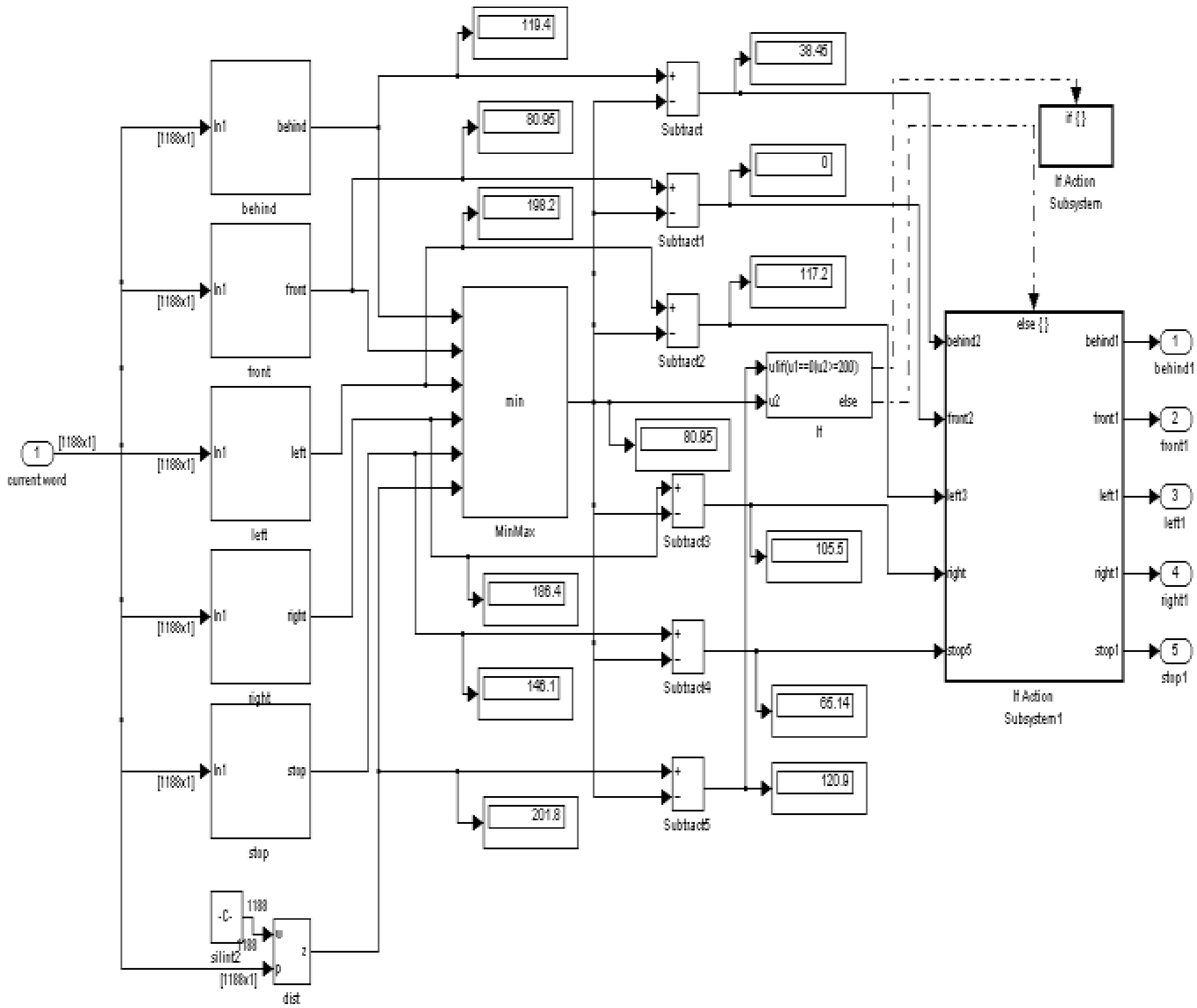


Figure 4.21: Pattern recognition subsystem blocks

In this subsystem we stored the features which extract for the five main words in constant blocks. Each word stored in four different cases to enhance the system and made it more accuracy and suitable for the real life, as figure below shows.

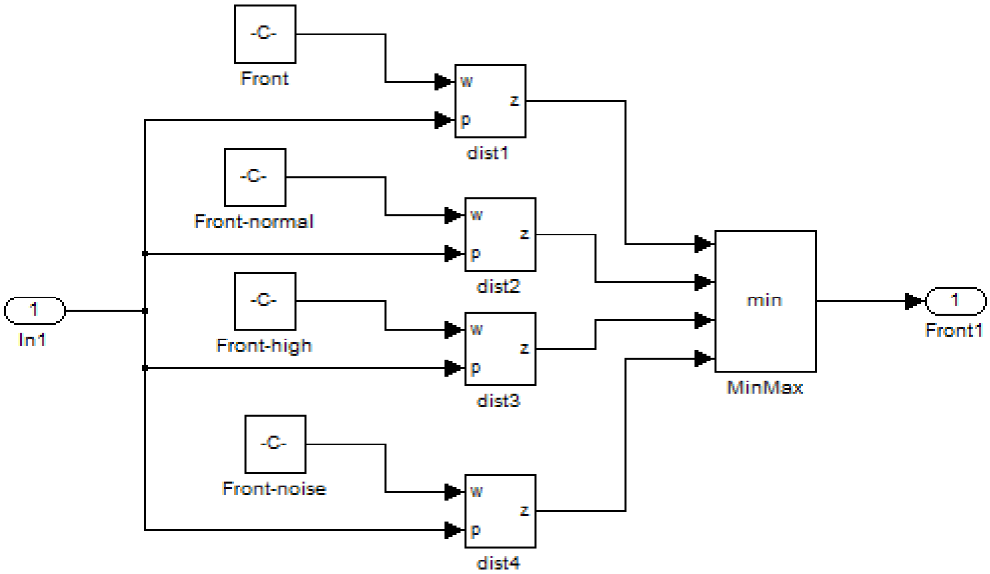


Figure 4.22: Front word in its four cases

Any entered word, our system will extract its feature and compare it with the stored features for five words, if the word spoken matched any stored word, a pulse will be generate. So a five different pulses will be generate depending on which match occurred.

To make this comparison between the entered word and stored word features we take Euclidean distance between the entered word features that generate from feature extraction subsystem and all word features stored in a constant blocks, then the result of Euclidean distance enter to MinMax block which out the min distance. This value determine the path of the system, if it greater than a threshold which represent the minimum distance between similar words or the current word matched with zero state which means user saying nothing then the" if action subsystem" will run which don't generate any signal and that mean the current word is not any word from words specified previously. But if the distance smaller than a threshold, the "else action subsystem" will run and generate a specific pulses with a specific amplitude for each word.

By subtract the minimum distance from all distances we can know this minimum distance comes from which word, as min –min=0 and the branch gives that result, specify the word saying .The following figure (4.23) shows "else port action" that generate pulses if match action occurs, this subsystem located within " else action subsystem" shown in figure (4.21).

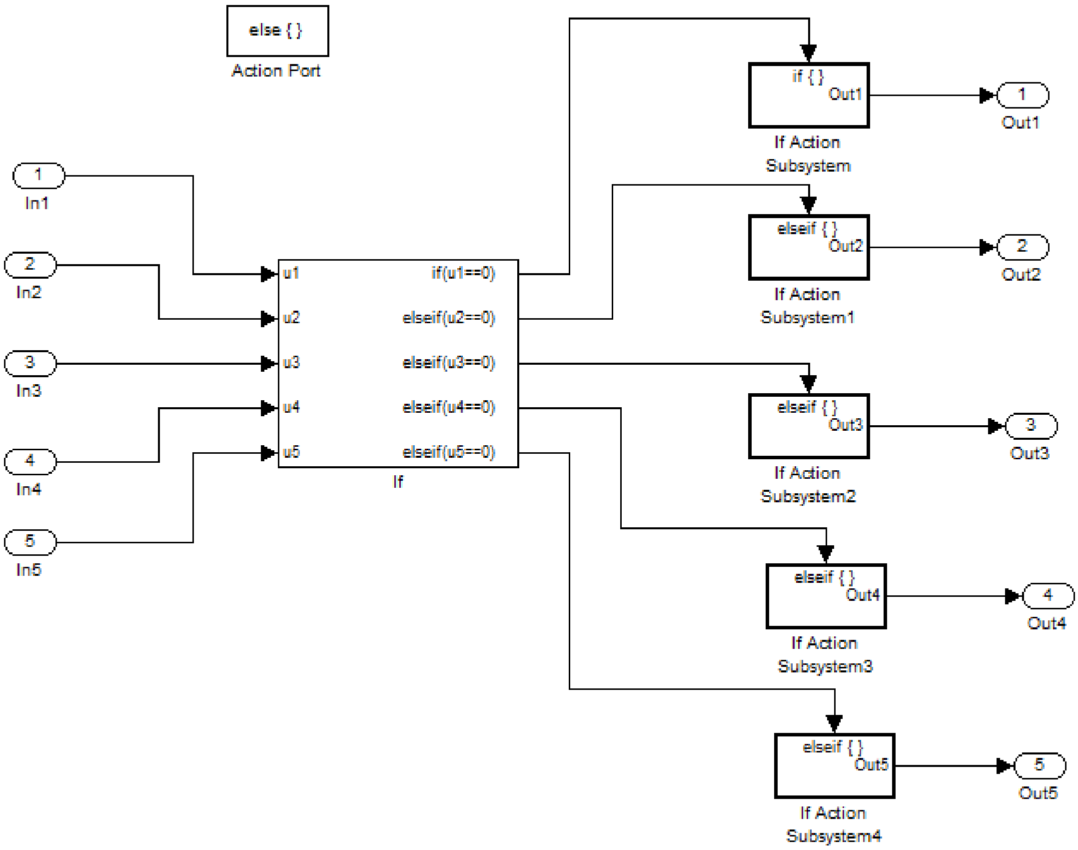


Figure 4.23 Else port action

Within each "If-else action subsystem" shown in figure (4.23) constant value that gives a pulse with specific amplitude for each word command.

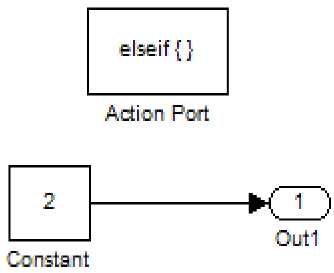


Figure 4.24: Generate pulse

A pulse will be generate as a final result, these pulses different in amplitude depending on which word give a match, this pulses given to the controller to control wheelchair.

By agreement with biomedical grope we specify pulses amplitude as following:

- Front command ⟶ pulse amplitude = 2V.
- Behind command ⟶ pulse amplitude = 1V.
- Right command ⟶ pulse amplitude = 4V.
- Left command ⟶ pulse amplitude = 3V.
- stop command ⟶ pulse amplitude = 5V.

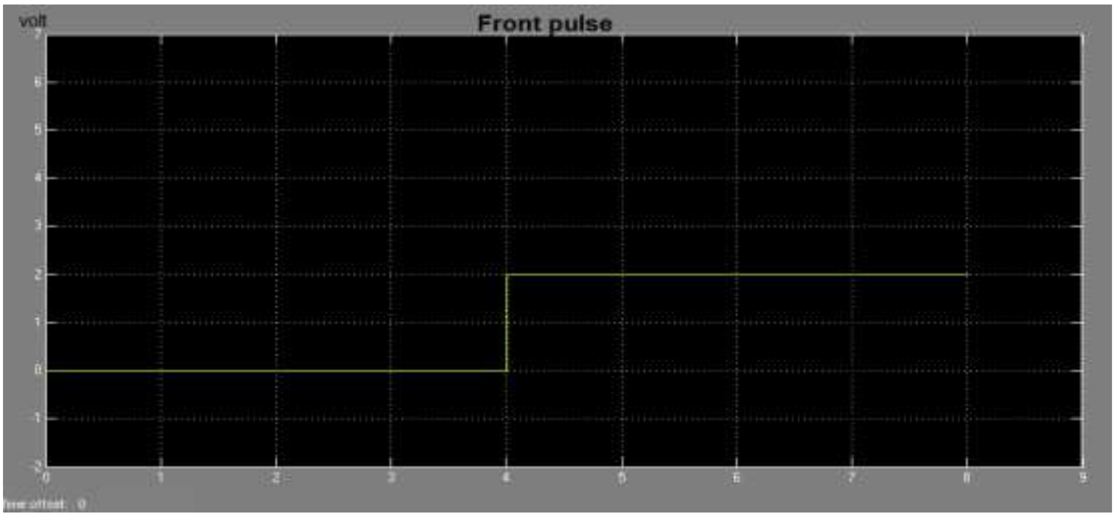


Figure 4.25: Generated pulse for Front word with 2V amplitude.

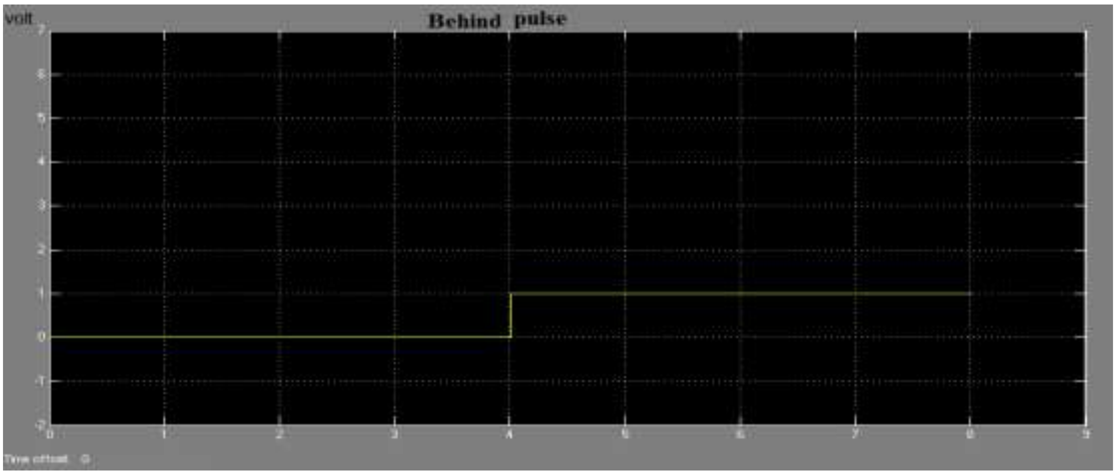


Figure 4.26: Generated pulse for Behind word with 1V amplitude.

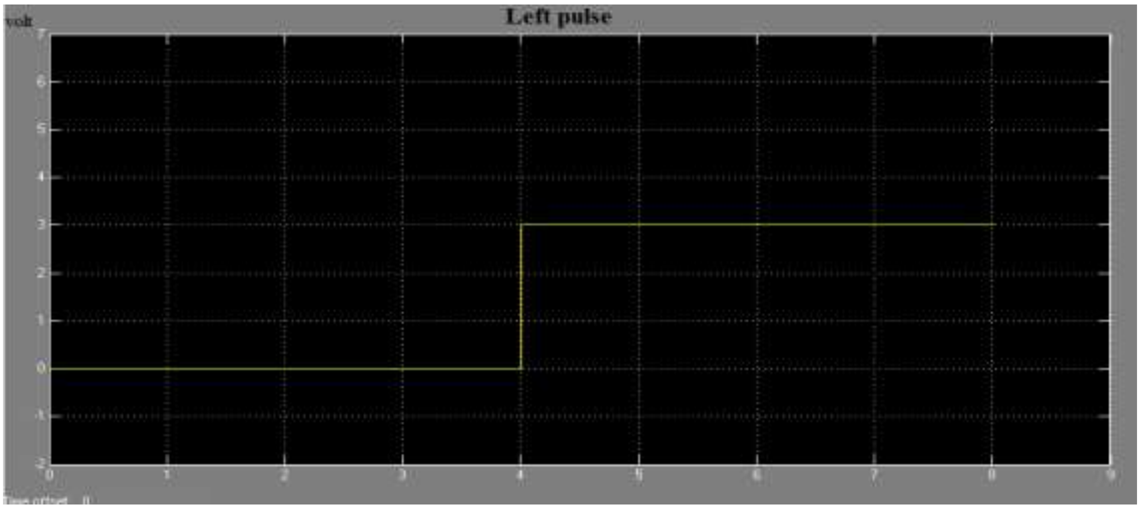


Figure 4.27: Generated pulse for Left word with 3V amplitude.

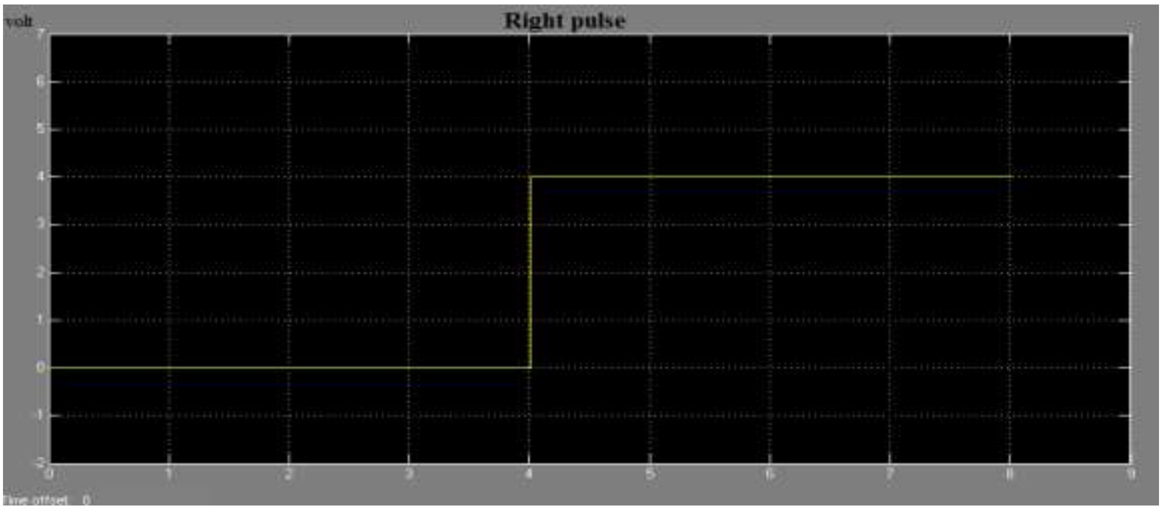


Figure 4.28: Generated pulse for Right word with 4V amplitude.

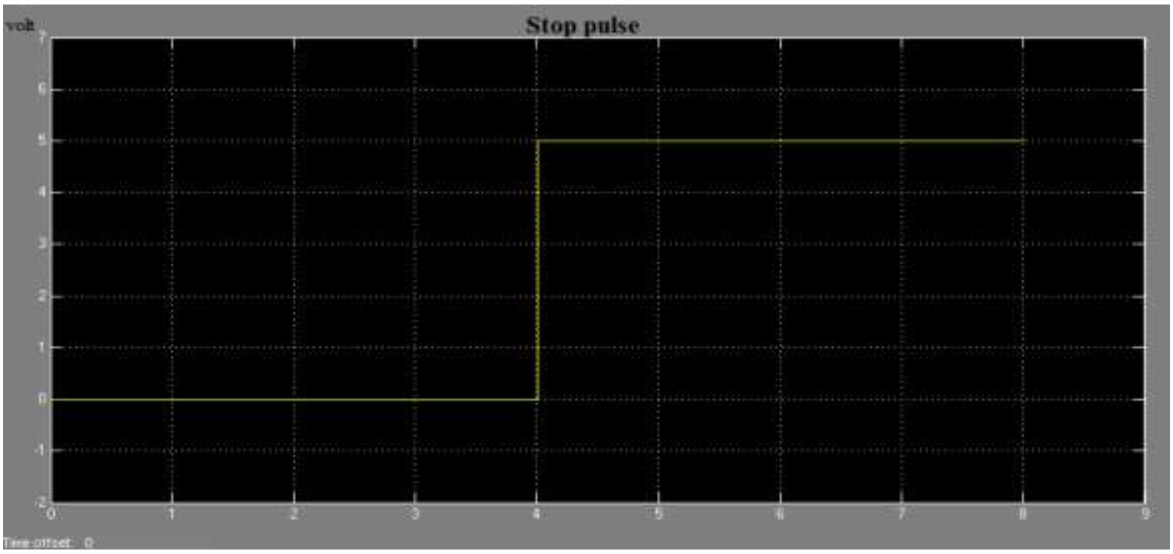


Figure 4.29: Generated pulse for Stop word with 5V amplitude.

4.4 Ultrasonic Sensor Implementation

LV-MaxSonar-EZ0 sensor implemented in this project as protection stage, that means if the sensors detect an object or obstacle in specific range from a wheelchair the microcontroller have to understand that action and must stop movement or change direction of motion, even if voice command found.

There are two sensors implemented to avoid obstacles that depend on the nature of the obstacle strain or slop as explain in chapter 3. The following circuit shows the interfacing between ultrasonic sensors and PIC microcontroller.

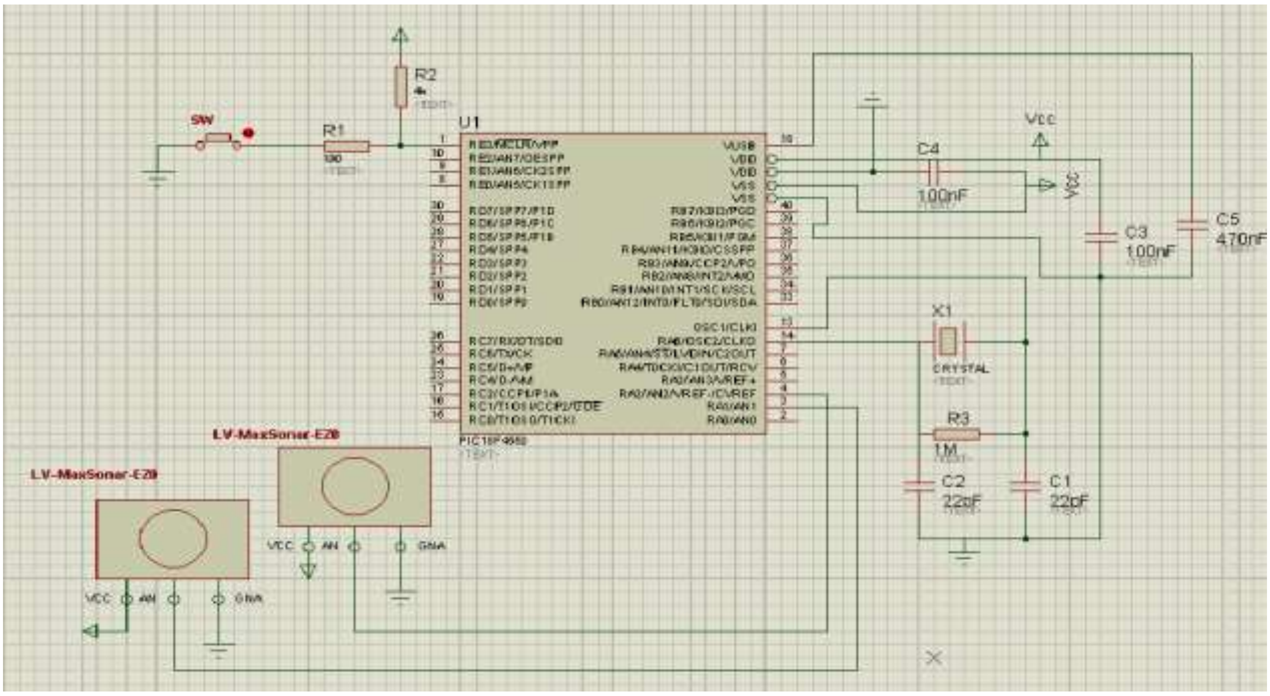


Figure 4.30: interfacing between ultrasonic sensors and PIC microcontroller

4.5 Power Up System

DSK bored and PIC microcontroller need 5 V DC power supply to operate, but the buttery voltage that provide power to motors is 12 V DC. So that, we use voltage regulator to get 5v from 12v buttery to running DSK kit and PIC ,that's called 7805 voltage regulator . The pins of this regulator and how there are connect shown in the figure (4.31) below.

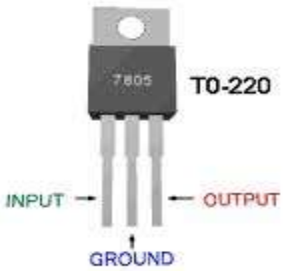


Figure 4.31: 7805 Voltage Regulator Pins

The following circuit is for the 7805 voltage regulator

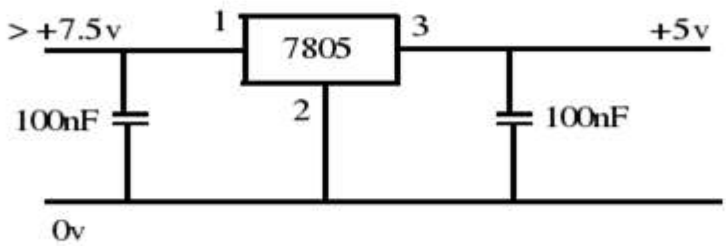


Figure 4.32:7805 Regulator Circuit

5

Chapter Five

System Testing

5.1 Introduction

5.2 DSK Board Testing

5.3 Ultrasonic Sensor Testing

5.4 Voice Command Matching

5.5 MATLAP Simulink Blocks Testing

5.6 Results

5.1 Introduction

After system implementation, the system must be tested to ensure that the system work properly or not, of course for an effective testing, the system must be divided to many subsystem, and checked after each subsystem to make sure that each subsystem achieve its main function and verify the location of error if its exist.

The testing can be done by seeing the output after each stage, if it's as required then passes to the next stage and so on; in other word, testing the device if its works or not, these treatment causes easily and fastest way to find the error, this process reduce times and effort.

In this chapter, we test each stage, to get the final result for the project, such that testing the DSK board connection, every subsystem in SIMULINK MATLAB, voice command matching and the ultrasonic devices to ensure that every stage was designed and implemented in a correct way.

5.2 DSK Board Testing

5.2.1 Software Testing

After power up DSK and watch LEDs, we use DSK diagnostic utility to test DSK functionality, at the beginning diagnostic status will be "Idle" as shown in the figure (5.1) below.

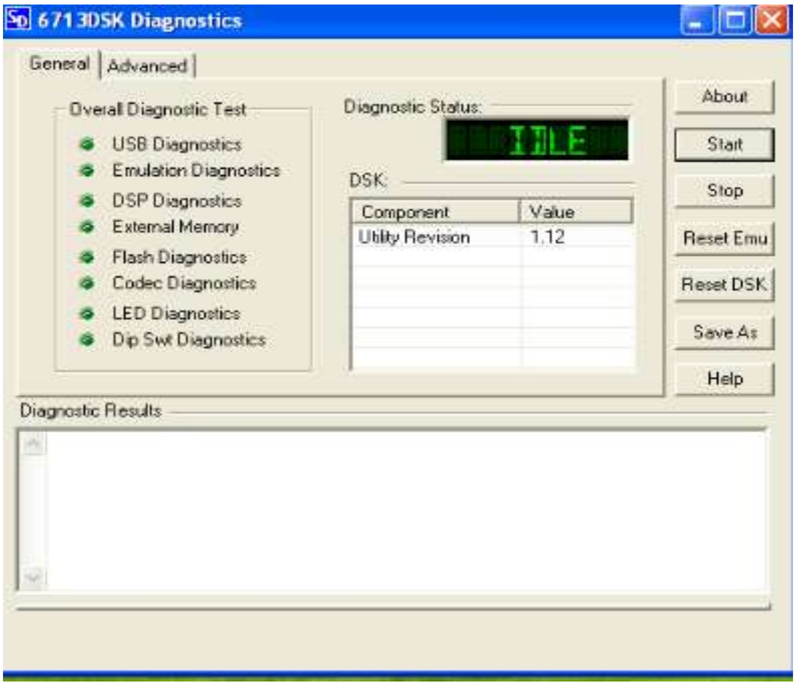


Figure 5.1: 6713 DSK diagnostic status is idle

If any error happens while testing operation, then the diagnostic status gives "Fail" like figure (5.2) below.

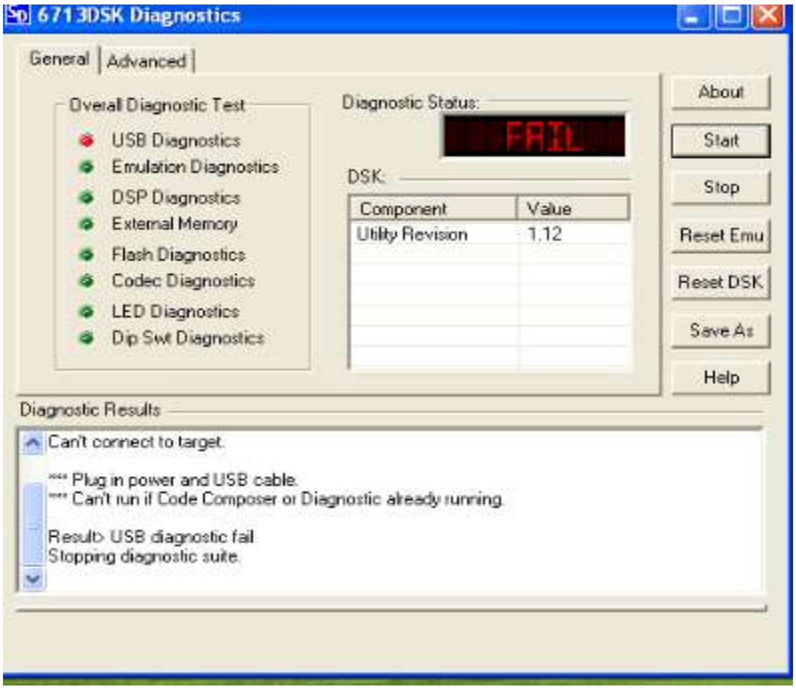


Figure 5.2: 6713 DSK diagnostic status is Fail

If the target (DSK) is not connected as above case in figure(5.2), a message under screen will be appear as in figure(5.3) ,but we can avoid this message by choose 'connect' from 'Debug' menu in code composer studio.

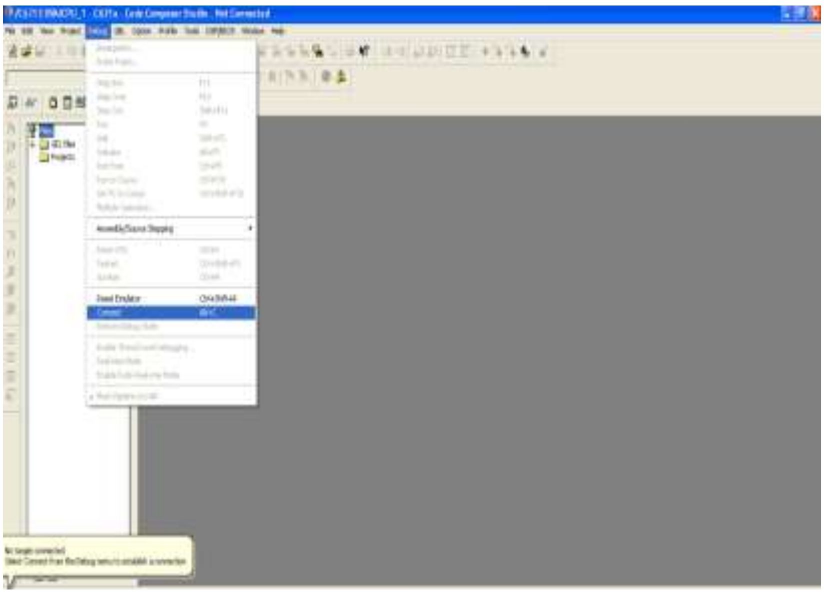


Figure 5.3: No target connected status for C6713 DSK Board

After correct the connection and complete overall diagnostic test ,such that the USB diagnostic, external memory…..etc, then the diagnostic status become "Pass", which mean every testing done correctly, and that shown in figure(5.4) below.

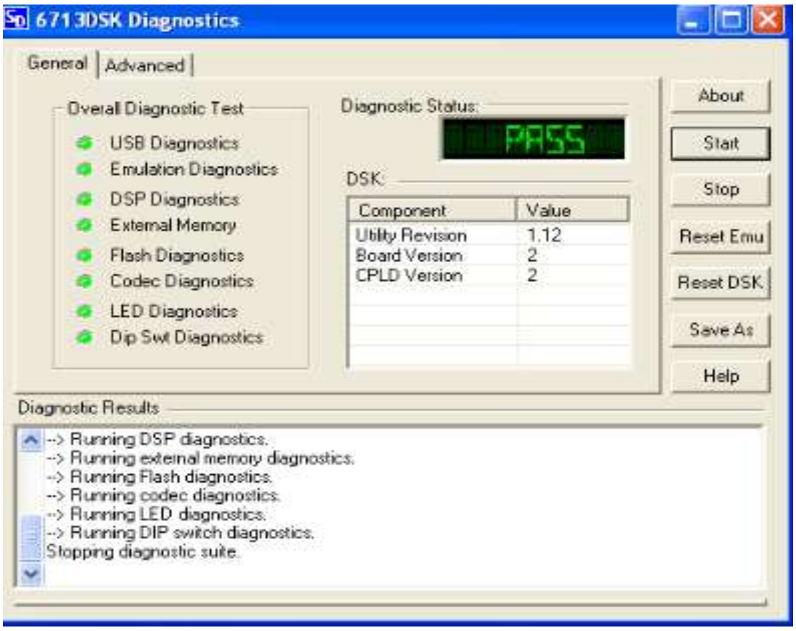


Figure5.4: C6713 DSK diagnostic status is Pass

After download the blocks simulink on the DSK Board, then screen in figure (5.5) will be appear, that’s denotes the download operation was finish without errors.

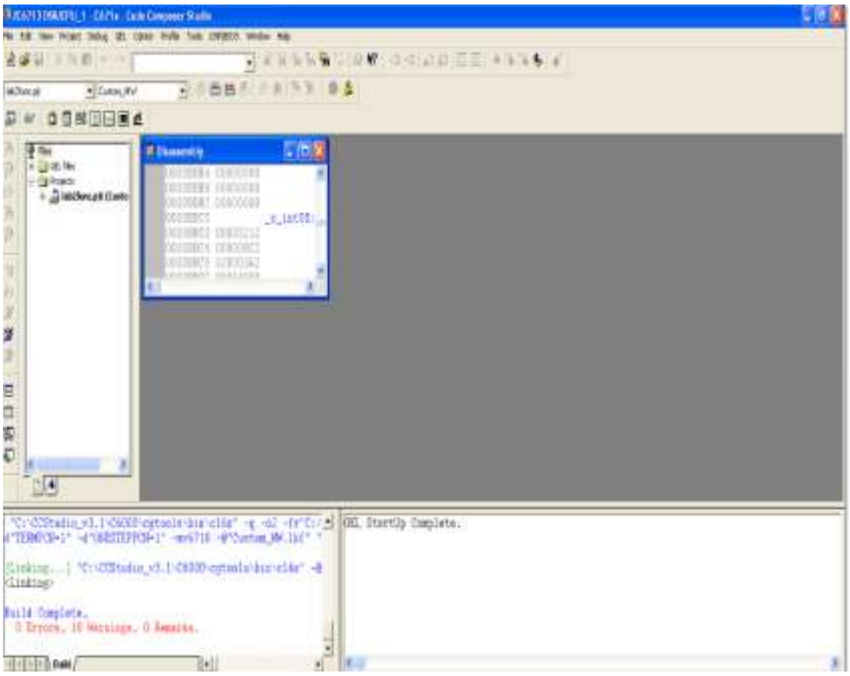


Figure5.5: Finishing downloads without errors

5.2.2 Hardware Testing

Here, to perform the testing operation on the DSK Board, it must operate properly with all peripheral devices, such that microphone, oscilloscope……..etc, and that exactly shown in the figure (5.6).

Figure5.6: DSK connection with peripheral device

5.3 Ultrasonic Sensor Testing

To ensure that, the sensor which we used is operate properly, we must check it according to its operation principles, first testing was to see the maximum output voltage that can be produced, that’s happened when there's no object along the whole range of the sensor, that’s equal (R=6.45m=254 inches).

According to its basis that every 1 inch distance that the object located away from the sensor gives ($V=V_{CC}/512=9.8mv$) if V_{CC} =5V, maximum output voltage resulted around (V=2.5V), which is exact value from datasheet, as it shown in the figure (5.7) next.

Figure 5.7: Maximum output voltage due to the maximum range of the sensor

The incoming testing for these sensors is to verify the relation between the distance that the object located away from the sensor, and the amount of voltage that can be produced.

We can classify it for two cases as the following

***Case 1*:** when there's an object located near from the sensor, it's give a small value of voltage that proportional to the small distance, as it shown in figure (5.8) below.

Figure5.8: Output voltage when object located a small distance from the senor.
Distance =40.7cm

***Case 2*:**When the object located at distance more than distance in case1, this give high output voltage compared to case 1 as it in figure (5.9) shown below.

Figure5.9: Output voltage when object located along distance from the sensor. Distance =391.4cm

After these testing of the output voltage relates with distance, we must check the beam of the ultrasonic sensor, the sensitivity of the sensor from the surrounding area is the most problem we face it.

When power up the sensor, it will sends a beam which reach to around 6.45m the shape of this beam is shown in figure (5.10), because of this shape the sensor may detect another objects which is not the target object, In order to avoid this problem, the sensor mounted on overhead position and isolated from other components in this project, so it sends beams towards the target object.

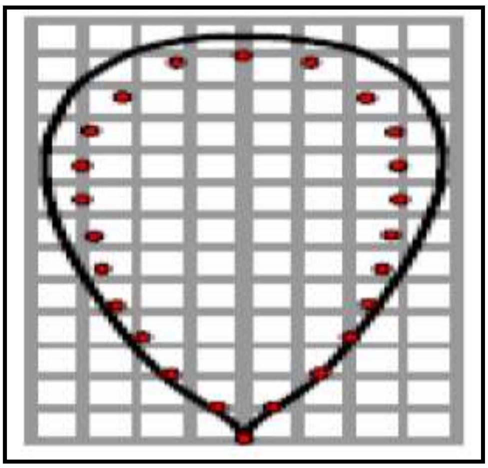


Figure5.10: Beam shape of the sensor

5.4 Voice Command Matching

The voice command was stored in DSK board before, and know a certain matching must be occur between the stored voice command and the current voice command, that's means when the speaker says a command this command will compared with the stored voice command which stored before, after this comparison a matching must be occur to give the correct movement for the wheelchair.

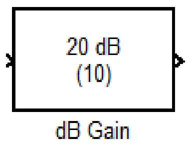
That's mean the entered voice command compared to the whole voice commands was stored before, and the matching present when the entered command has a high similarity with one of the stored command. At that time the two commands will be the same.

5.5 MATLAP Simulink Blocks Testing

In this section we will test all subsystems used in building MFCC system, each subsystem build to provide a specific function. The previous chapter had shown how we build each block and adjust its parameters to match what the system need.

This section contain a brief description for testing every subsystem, and show if it match its main function or not. We will use stop command for system stages testing.

1. dB Gain



As we explained in chapter four speech signals has week amplitude, so we need to gain it ten times at least to provide it against noise and keep its shape and features.

As we note from figures below the amplitude range before dB gain is -1 to1 but after dB gain the amplitude becomes at range -10 to 10 and that's what we hope from this block that's mean the result is correct.

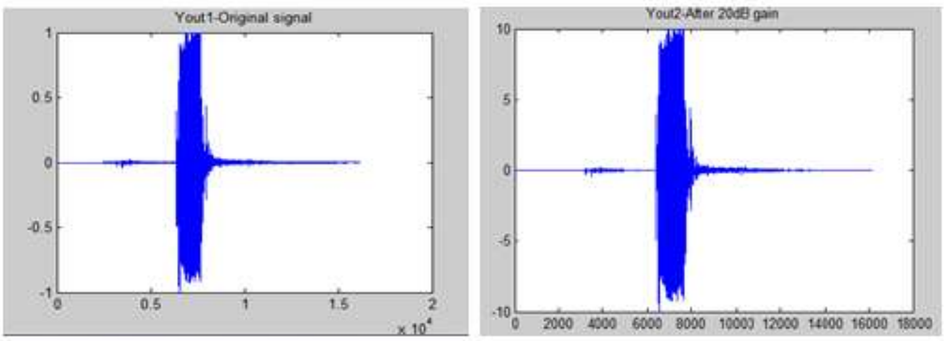
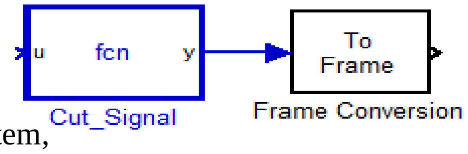


Figure5.11: Signal before and after dB gain

2. Removing silence from speech signal



This stage is very important in any voice recognition system, because it removes silence from the beginning and end of command spoken and take the net word only.

This embedded matlab function accepts command spoken as an input and process it to remove silence from it, and shift the net word to the beginning as shown in the following figure.

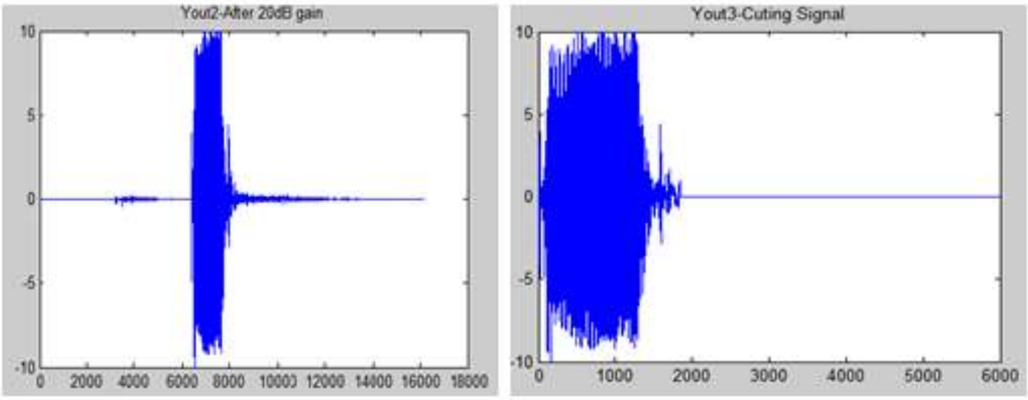


Figure 5.12 Signal before and after silence removing stage

3. Low pass filter

A Digital filter design block build to accept all low frequencies up to 4 KHz and to reject other high frequencies.

Figures below have shown the frequency response for command word before and after entering low pass filter.

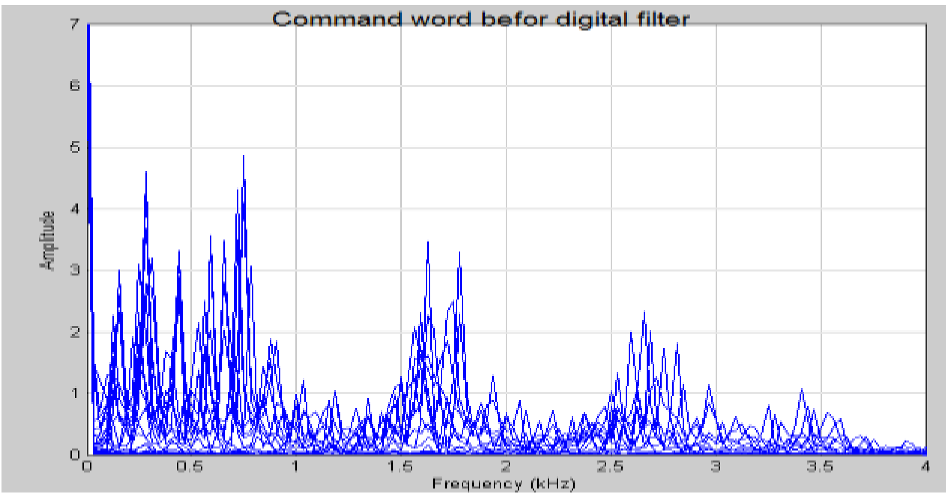


Figure5.13: Frequency response without filtering

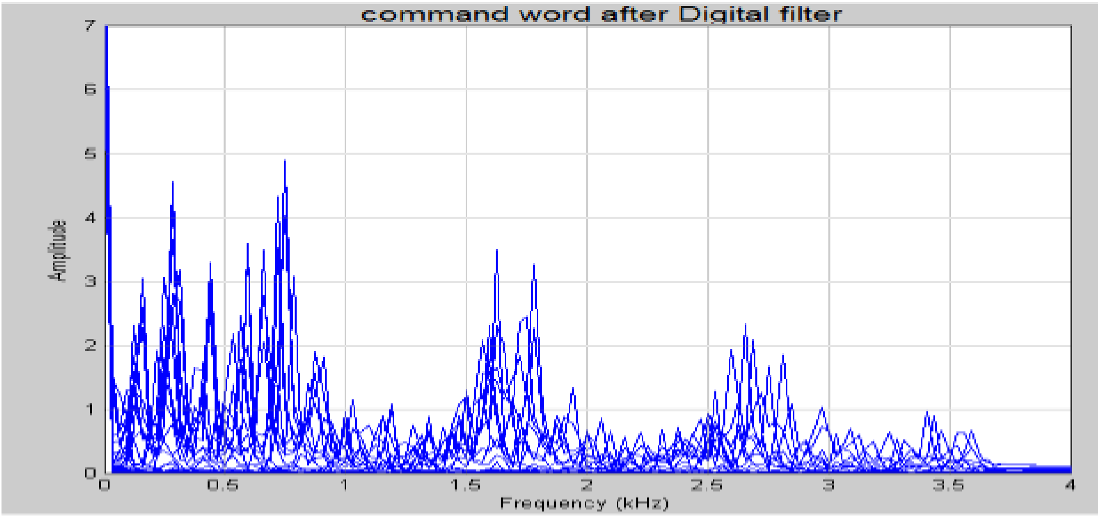
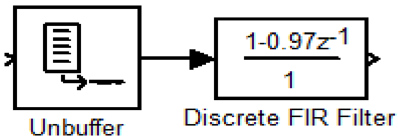


Figure5.14: Frequency response after filtering

From the figures we can notice that; the filter pass all frequencies below 4KHz, its F_{pass}=3.5KHz,Fstop=3.9KHz which match its main function and what this block design for as in design chapter 4.3.1.

4. Pre-emphasis



Pre-emphasis subsystem used to boost the energy of high frequencies, the figure below shows that, this subsystem increase the energy for high frequencies which means that this subsystem do it's major process.

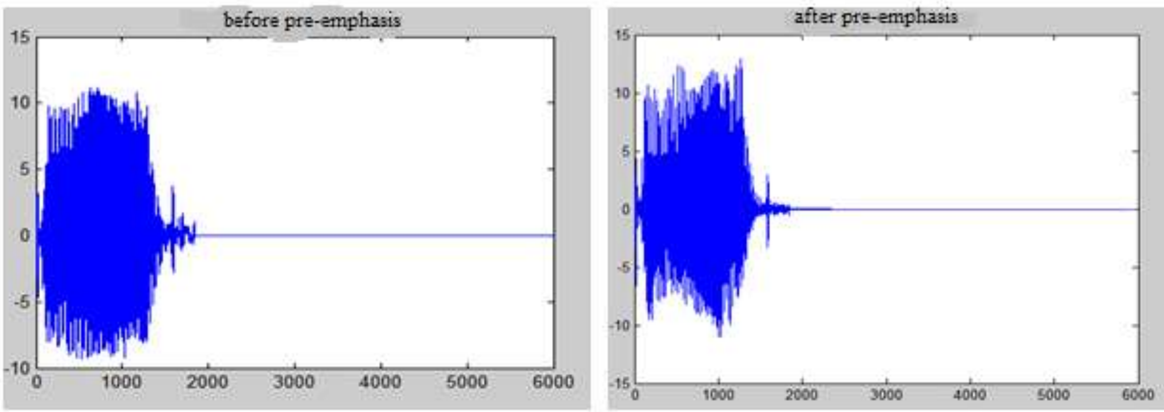
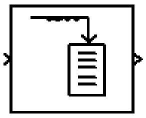


Figure5.15: Signal before and after pre-emphasis

The first one show signal before pre-emphasis, the second one show signal shape after pre-emphasis we can easily notice that the amplitude was increased, this increment in high frequencies more than in low frequencies.

If the pre-emphasis efect is not clear in previous figure (5.15), it sure clear in table beside. This table contain 10 samples taken from signal before and after pre- emphasis, as we note each sample after pre- emphasis block equals the sample before pre- emphasis minus 0.97 of previous one and that's what pre- emphasis stage do as it's equation.

Pre-emphasis	
Before	After
1.313582542	1.313582542
2.930427869	1.656252803
0.372497721	-2.470017311
-0.081709328	-0.443032118
2.554866488	2.634124536
-0.946922369	-3.425142862
-3.180866975	-2.262352278
-2.072831371	1.012609595
0.444684274	2.455330704
1.472147333	1.040803587



Buffer200

5. Frame blocking subsystem

Main function of this subsystem is dividing the input signal into frames of 200 point length, the overlap between two successive frames 80 point.

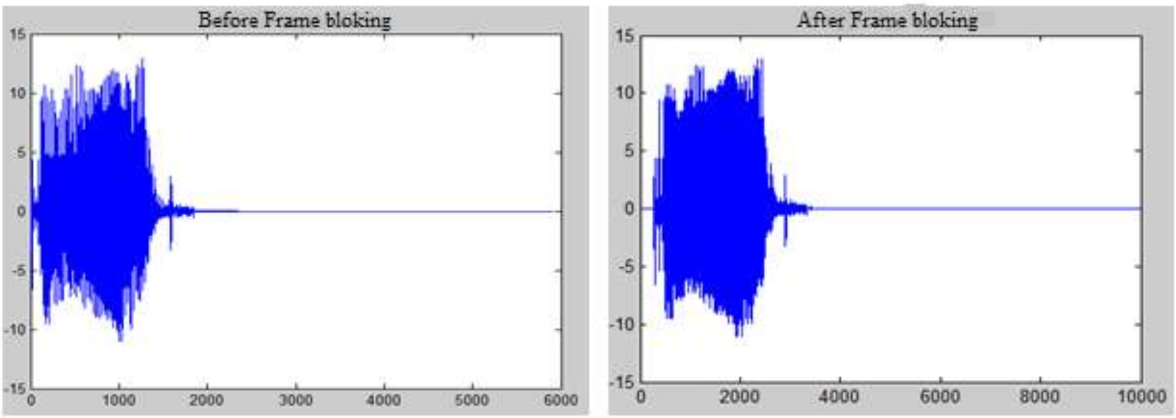


Figure5.16: Signal before and after frame blocking

We can notice from the previous figures the signal before entering frame blocking block has a discontinuity, but after framing it, these discontinuity almost disappear. Also number of frames increases along the signal.

We can conclude that, frame blocking subsystem increases the number of frames and save a copy for frame end which made processing on signal more efficient and no losses in data.

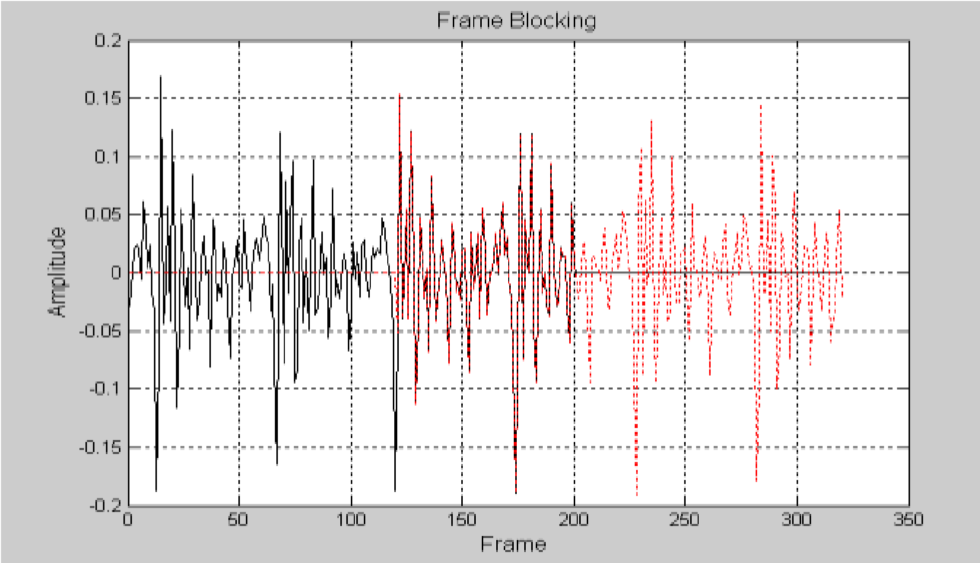
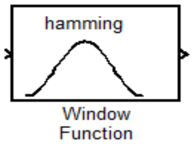


Figure 5.17: Two successive frames

This figure shows two successive frames, each frame consist of 200 point and overlap with each other in 80 point, black color represent the first frame, red color represent the second frame. That's means; there is a real overlap between frames.

6. Windowing Block



Multiplying the signal with hamming window minimize the discontinuities at the beginning and the end of each frame.

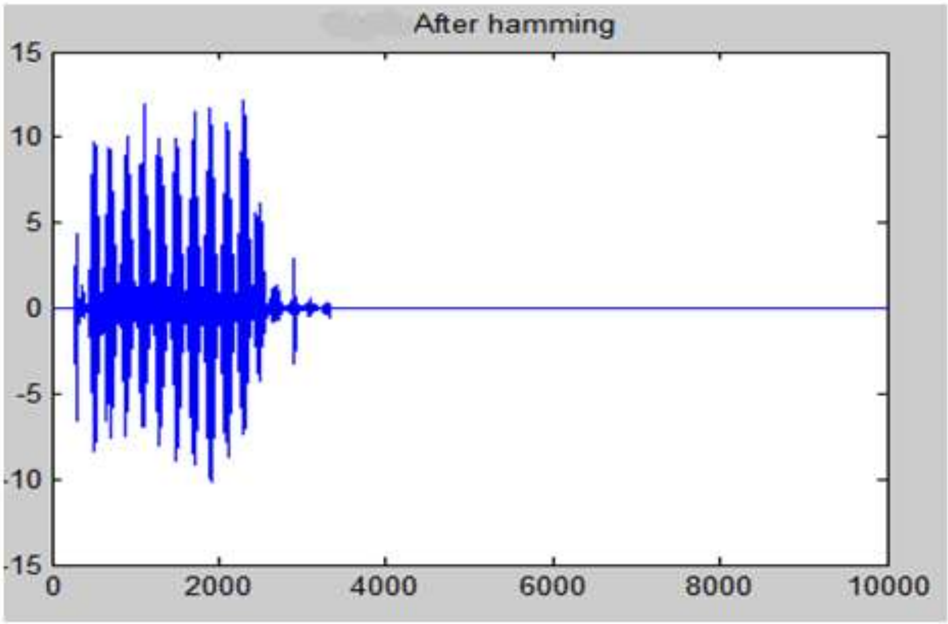


Figure 5.18: Signal after hamming

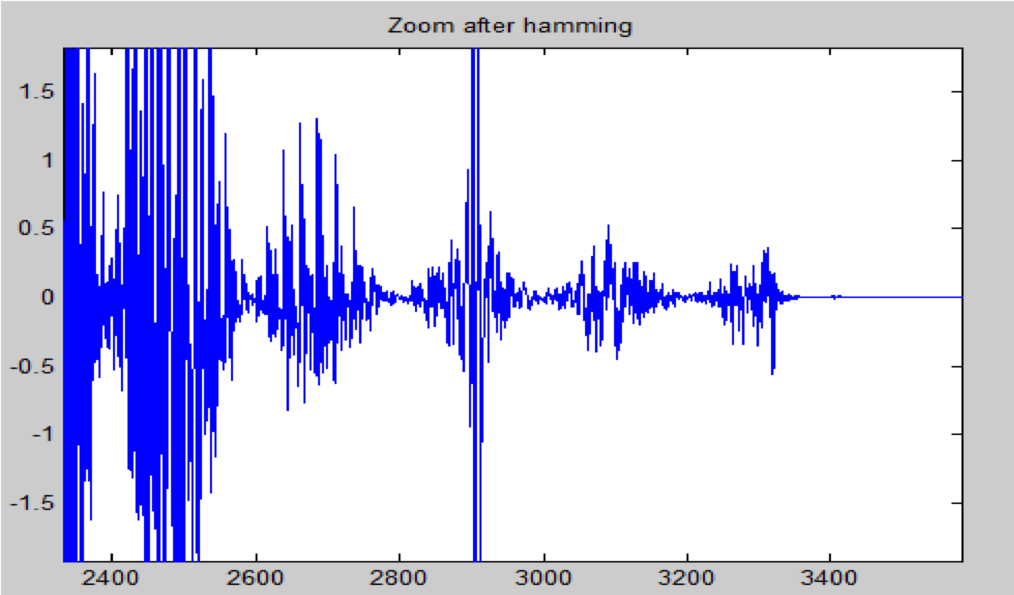
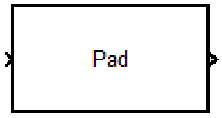


Figure 5.19: Zoom after hamming

We can notice from the figure, window minimize the signal distortion, by made the beginning and the end of each frame approximate to zero.

7. Zeros padding



All previous processing steps were in time domain, now we want to move to frequency domain but to do that the length of frame must be power of two, and as we know the length of current frame is 200 point, so we will pad each frame by 56 point with zero value at the end to make frame length $2^{\wedge 8}$=256 point as shown in the following figure.

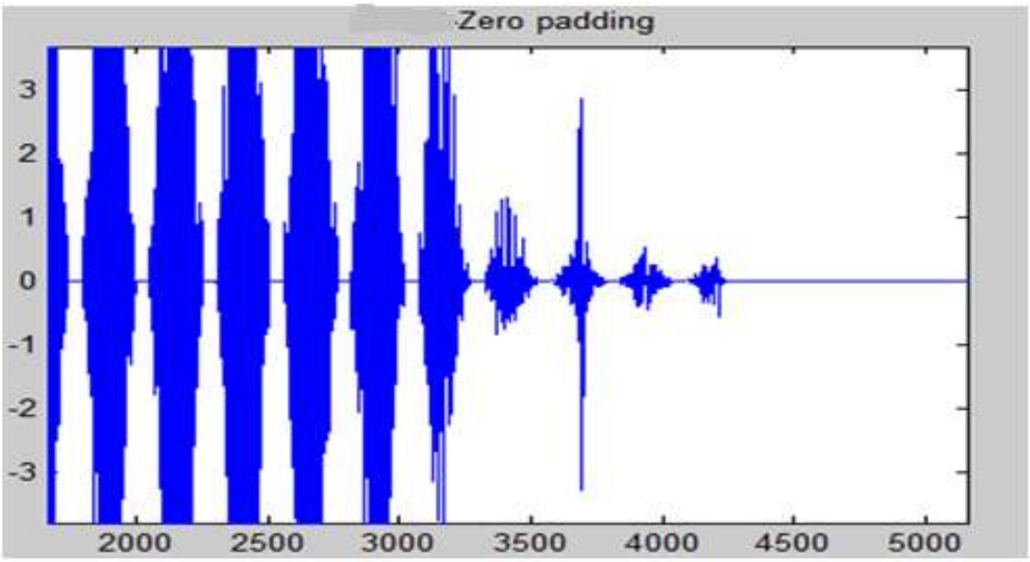
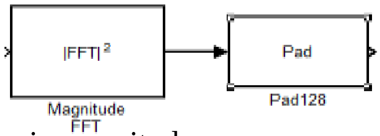


Figure 5.20: Zoom after zeros padding

8. Discrete Fourier Transform and Spectral Magnitudes square subsystem



DFT transfer the samples from time domain to frequency domain, here we take magnitude square because in the next steps we will take the logarithm of mel filter energies and as we know the energy is magnitude square so that we take magnitude square here.

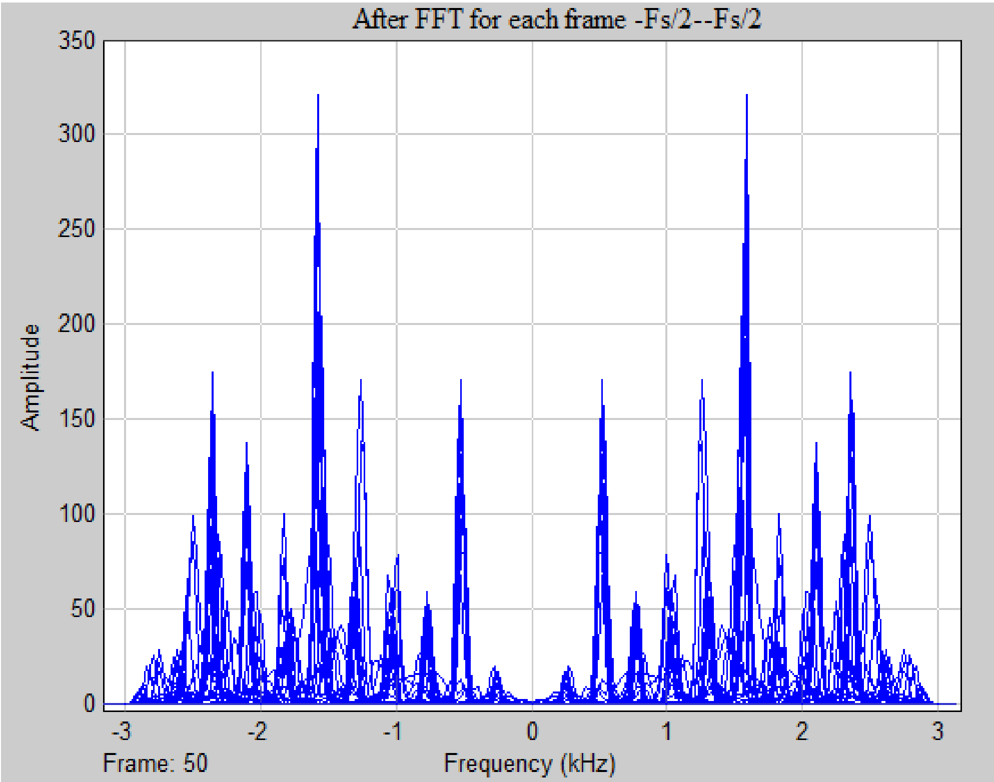
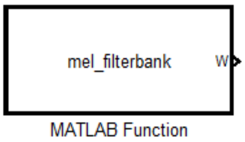


Figure 5.21: Spectral magnitudes square

Figure(5.21) shows that signal magnitude increase, so this subsystem do its main function in good way and as we note frame repeat itself after 128 point that means the first 128 point is the same of the last 128 point in each frame. So that we use the pad block to cut the second half of each frame by specify frame length 128 from pad block properties.

9. Mel Frequency Filter Bank Subsystem



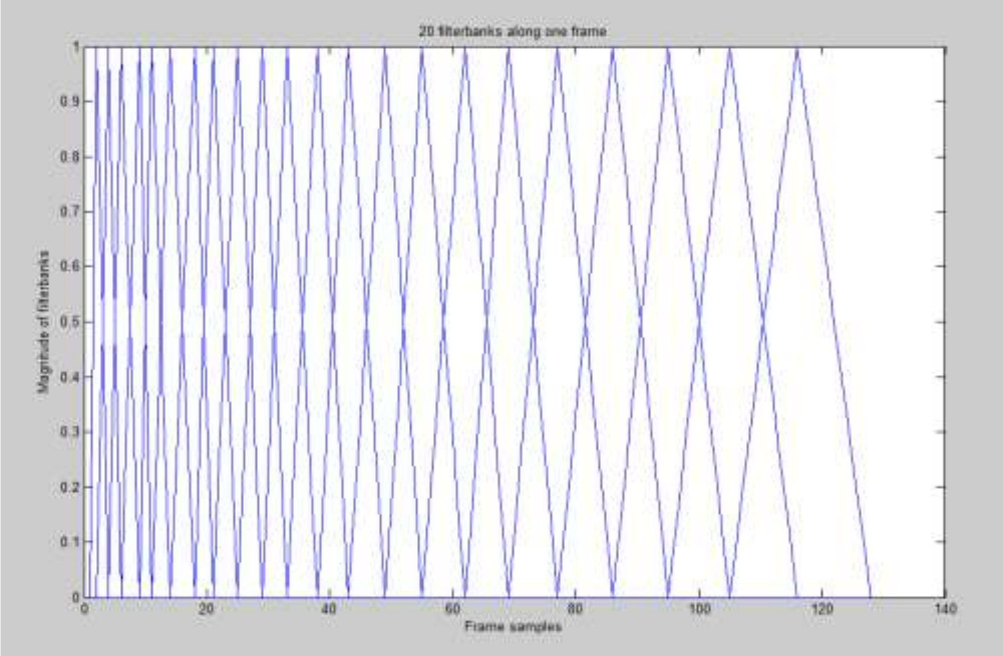


Figure 5.22: Mel filter banks

This figure shown Mel Frequency Filter Bank, main point in mel filters that the number of filters in low frequencies more than in high frequencies, there is an overlap in these filters, each filter starts and end at the center of the adjacent filter, filter take a triangle shape.

By multiplying each FFT frame with length of 128 point with these filters we will have a 20 point from each frame as shown in the following figure.

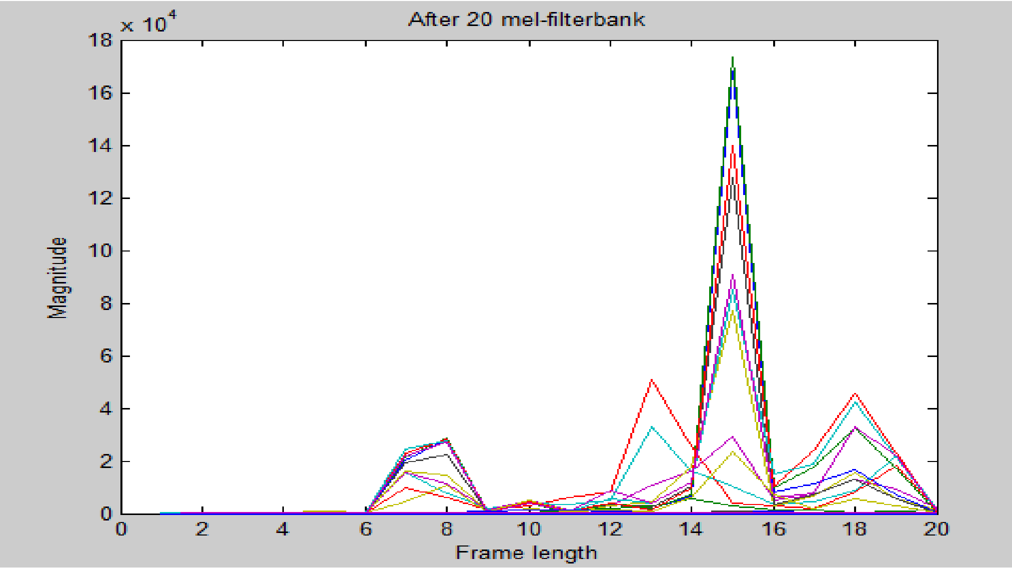
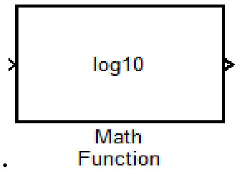


Figure 5.23: Signal after mel filter banks

10. Logarithm of filter energies subsystem



If we look back to figure (5.23) we note the high values of amplitude. To compress these values, we take the logarithm of it. The following figure shows that clearly.

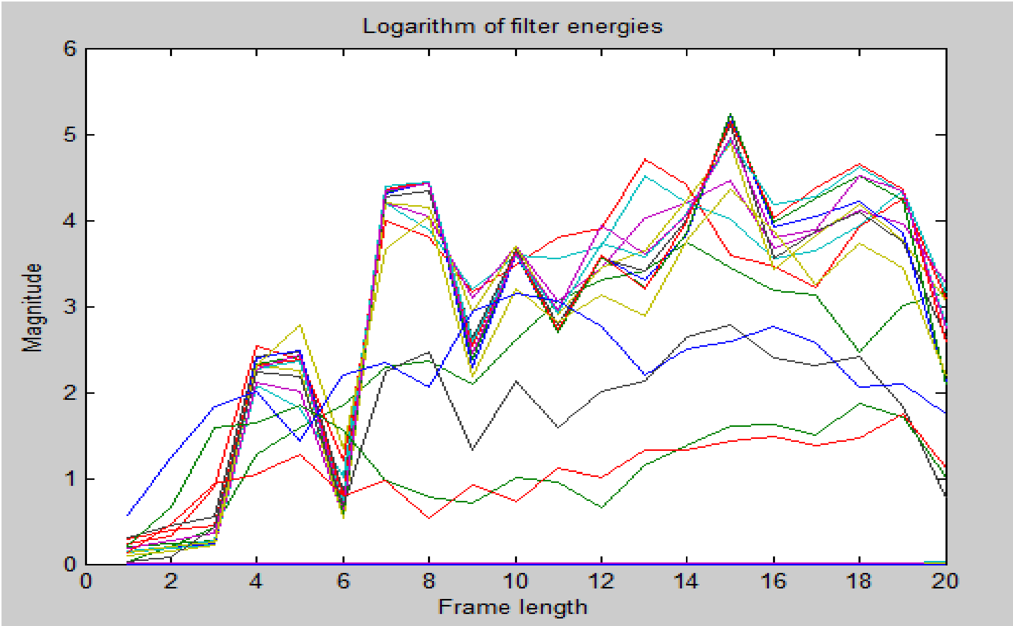
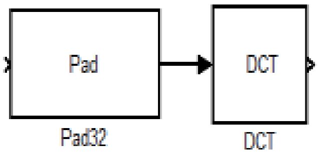


Figure 5.24: Logarithm of filter energy

11. Discreet Cosine Transform subsystem



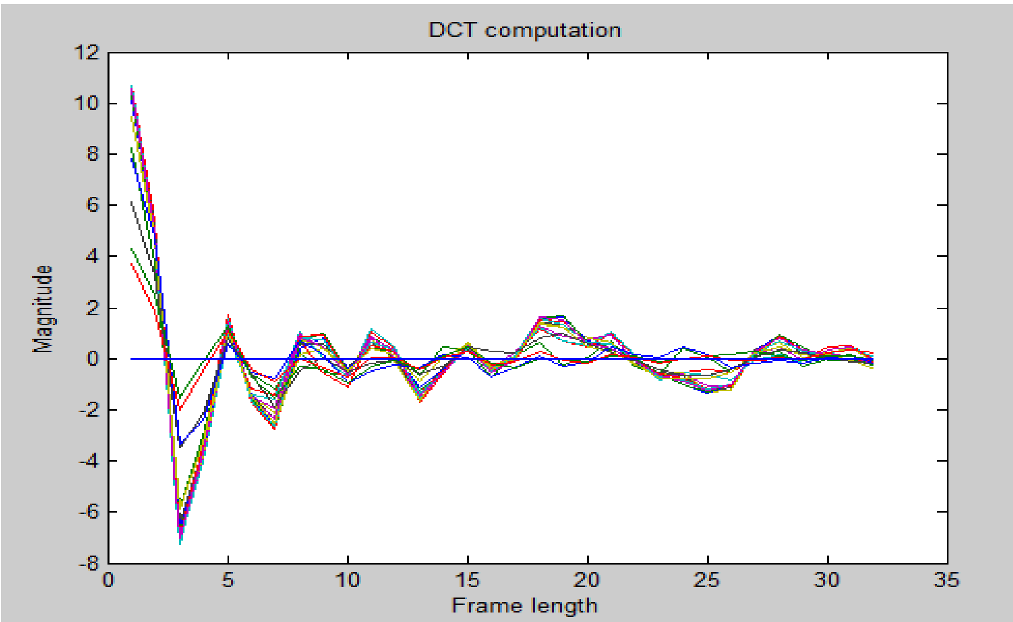
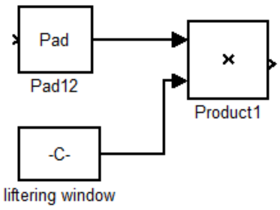


Figure 5.25: Signal after DCT

From figure (5.25) we can conclude that DCT compress and gathers most of information in the signal to its lower order.

As DCT block in matlab accept only frames with length two power something we padded original frames - with length 20 point- zeros to reach $2^{\wedge 5}$=32 point frame length, these zeros will not effect in result because we take the first 12 point only from each frame.



12. Cepstral weighting subsystem

The final step is cepstral weighting that means multiply each frame with sinusoidal window as shown in figure (5.26) below.

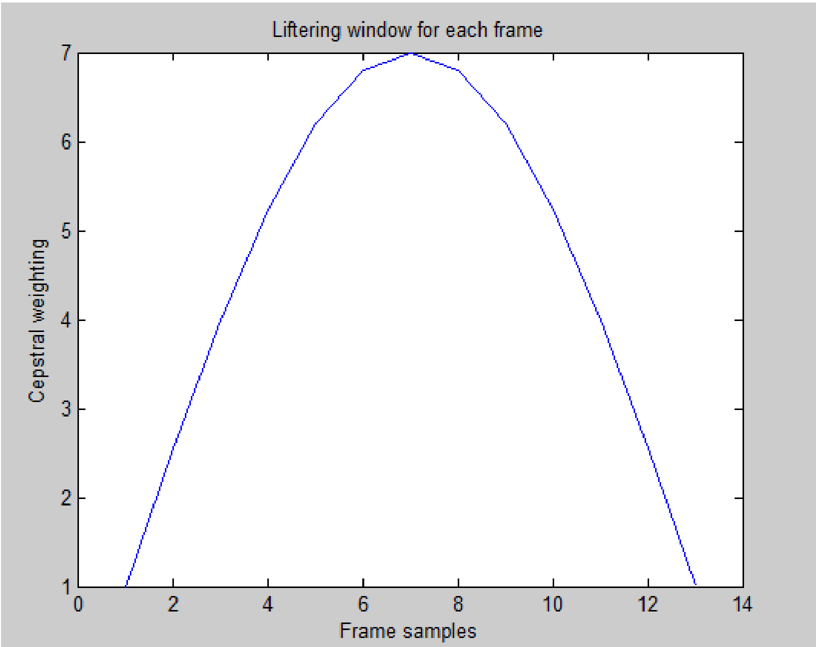


Figure 5.26: Liftering window

From figure (5.25) we can notice that Signal before liftering window has low amplitude, after liftering the amplitude will increase and minimize the sensitivities by lessening the higher and lower cepstral coefficient and that is the aim of cepstral weighting stage.

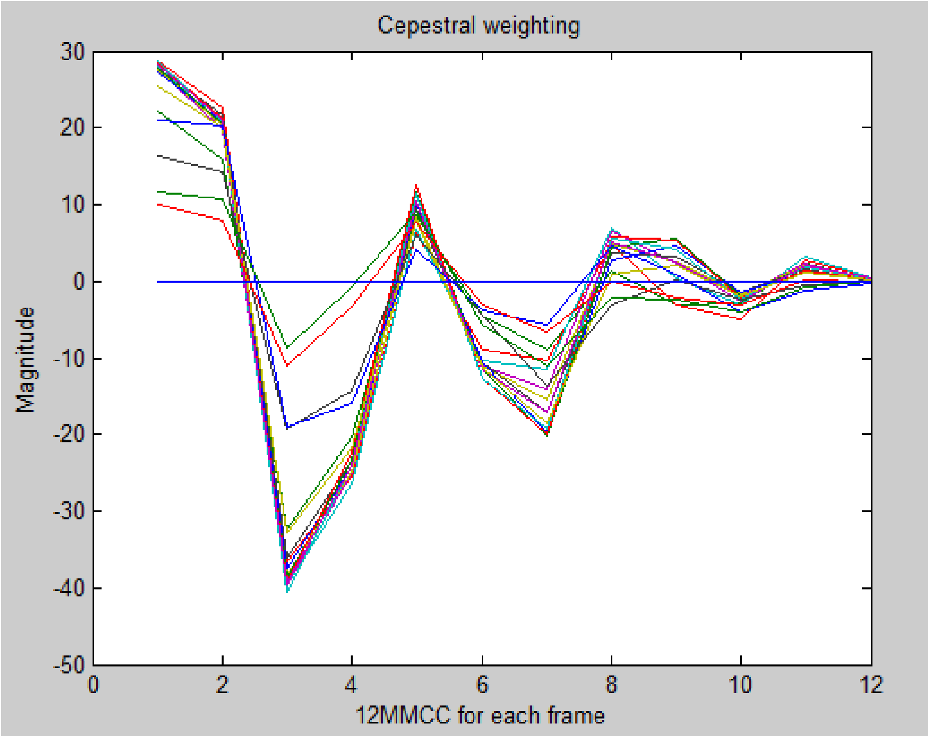


Figure 5.27: Signal after cepstral weighting

From figure (5.27) we note clearly how the amplitude increase after cepstral weighting, and the length of each frame is 12 point which represent MFCC, finally each color in previous figure represent 12 MFCC comes from original frames with 200 point length after all previous processing steps.

13. pattern Recognition subsystem

After testing all stages in MFCC subsystem and verify do its task correctly, we have to test pattern recognition subsystem by saying any command and see the decision is correct or not.

Let us do two experiments for example and see the results from this subsystem, to be clear to you how this subsystem works.

The first trial is saying front command -امام- the following figure shows the results from this subsystem.

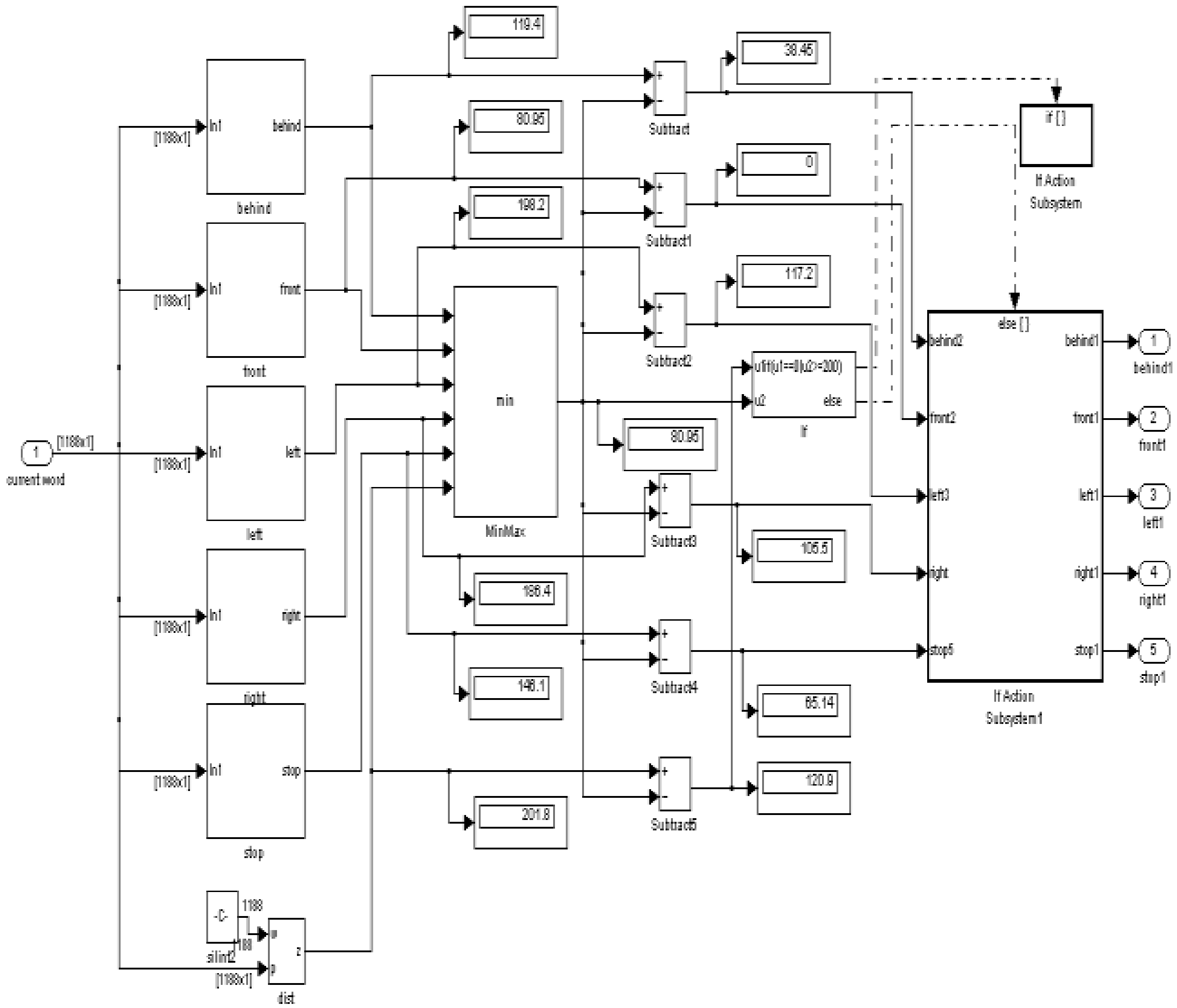


Figure 5.28: Distance results when spoken command is front

As we see from previous figure, the minimum distance is 80.95 and it came from front subsystem which means the decision true.

The second trial is saying nothing or silence state the following figure shows the results from this subsystem.

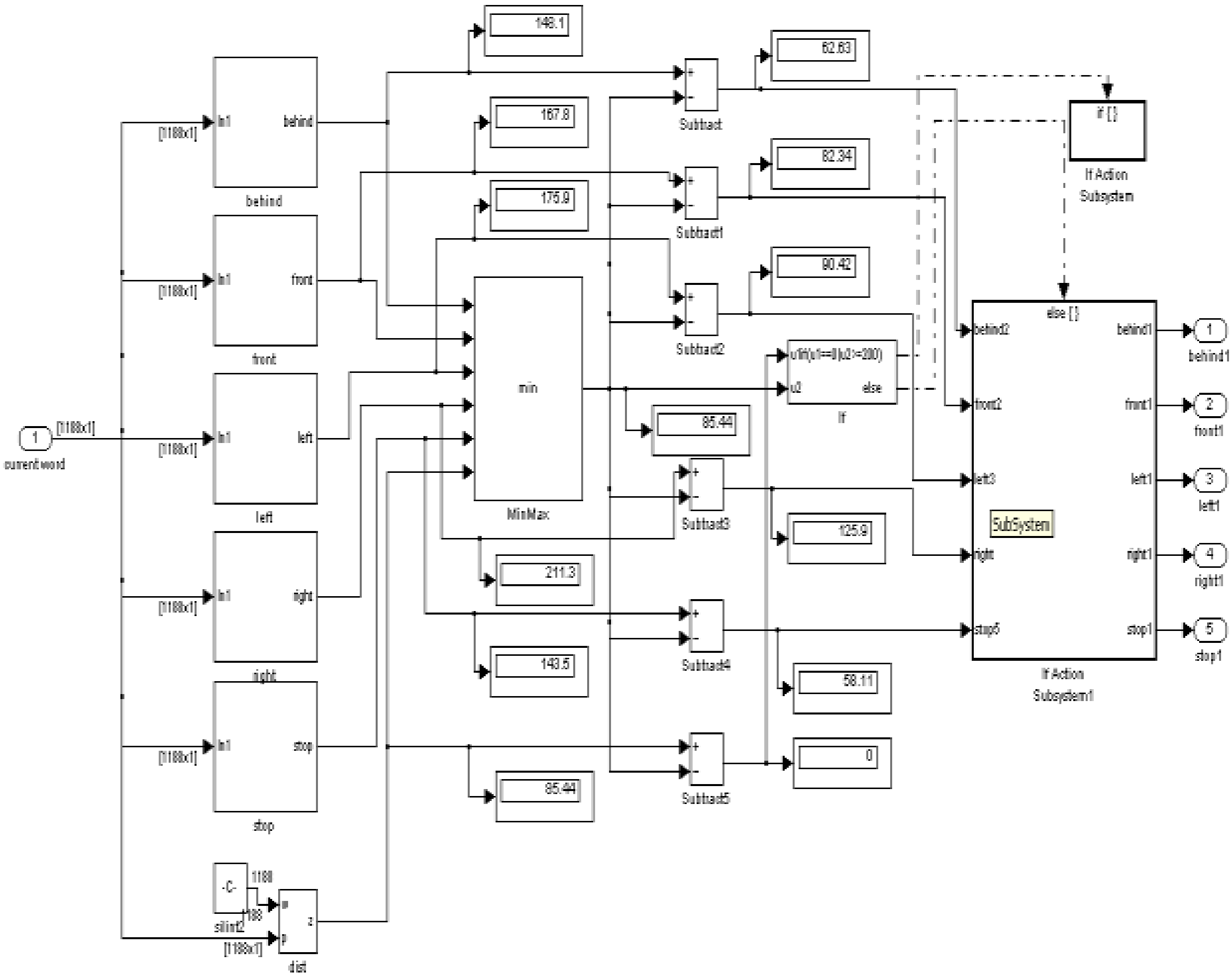


Figure 5.29: Distance results when saying nothing or silence state

As we see from figure (5.29) the input command matched with zero state and produced minimum distance, that means the results true and this subsystem works correctly.

5.6 Results

The following table shows each stage if it match its main function or not :

Name of Subsystem	PASS	FAIL
1. Low pass filter	✓	
2. Pre-emphasis	✓	
3. Frame Blocking	✓	
4. Windowing	✓	
5. DFT & Spectral Magnitudes	✓	
6. Pad block	✓	
7. Mel Frequency Filter Bank	✓	
8. ADD Block	✓	
9. Logarithm of filter energies	✓	
10. DCT	✓	
11. Cepstral weighting	✓	
12. pattern Recognition	✓	
13. Sensors testing	✓	

Table 5.1: Results of testing each subsystem in whole project

The following table shows some trials for matching; we tested recognition system 20 times for each command and calculated the percentage of success trials.

# of trials	word	# of success	# of failed	Percent of success
20	front	18	2	90%
20	behind	13	7	65%
20	left	17	3	85%
20	right	16	4	80%
20	stop	14	6	70%
Total=100		**78%**	**22%**	

Table5.2: Trials to find matching percentage

We note from last table the percentage of success of two commands behind and stop was low, the reason of that is the similarity between these commands in spoke and short length of it, which means recognition these commands as zero state in many trials.

6

Chapter six

Future Work

6.1 Introduction

The project that has been done was a step for developing the idea of voice recognition. Also the project was a good step in developing voice recognition on MATLAB Program. Meanwhile we have some recommendations and suggestions for the future work. The following section will discusses them.

6.2 System Achievements

Almost all the goals of our system have been achieved. In this point the main achievements of the system are discussed and the ways of achieving it. We build system of voice recognition on MATLAB SIMULING program that have five words. Also the project has sensing unit that controls the movement of wheelchair.

We can add more words to our system to be more compatible to our needs in practical life; also we can connect more than device such as sensor to enhance protection process in our project.

6.3 Real Learning Outcomes

After the implementation of the project we have an expert in the following points:

- How we can invest what we learn in our studying “using DSP to build voice recognition system".
- learn how to program in MATLAB SIMULINK big system
- Learn how to use and program 18F4550 microcontroller.
- Faces many problems with MATLAB SIMULINK Blocks and learn how to solve it.

6.4 Recommendations

After our work on this project and after facing many problems during the implementation, we as a project team, see that the following points may be a good improvement for this project in order to make it more sense and more reliable:

- Increase number of words in recognition system to make it more practical in daily usage.
- Increase number of sensing unit.
- Add another sensor type.
- Enhance on the system to be more efficient to noise area.

6.5 Future Steps

Here, there's some points that's relates to our project, that formed development sides that can be taken over our project.

- We are using the Mell Frequency Cepestral Coefficient (MFCC) method to get feature extraction for the voice command , but there's another method to do that ;that's Linear Predictive Coding (LPC) , Perceptual Linear Prediction (PLP) are be used to do that.

- Another way to do what we do , like replacing the TMS320C6713DSP Kit with Field-Programmable Gate Array (FPGA) or other Application-Specific ICs (ASIC).

- Anyone can use our system to control many operations, such that housing controlling i.e.

- There's an ability to do a lot of experiments on DSK board in digital signal processing domain, using our work (matlab simulink block) that what we are doing.

- The system can be develops by entering many modification on it, that leads to better controlling by Wheelchair, such that the speed of the Wheelchair.

References

[1] http://www.answers.com/topic/speech-recognition .

[2] Markus Forsberg, Why is Speech Recognition Difficult? , February 24, 2003 ,page(2-7) available at http://www.speech.kth.se/~rolf/gslt_papers/MarkusForsberg.pdf

[3] http://en.Wikipedia.org./wiki/sound.

[4] http:// www.jhu.edu/virtlab/ray/a coustic.htm.

[5] http://en.Wikipedia.org/wiki/noise

[6] Rappaport , wireless communication,2002.

[7] http://en.wikipedia.org/wiki/Time_domain.

[8] John G.Proakis , Dimitris G . Manolakis , Digital Signal Processing principles, Algorithms , And Applications , 3'th Edition

[9] http://en.wikipedia.org/wiki/Digital_signal_processing

[10] John G.Proakis , John R. Deller, John H.L. Hansen , Discrete Time Processing of Speech Signals , Institute of Electrical and Electronics Engineers, 2000 ,2'nd Edition

[11] Meng Yuan, Speech recognition on DSP: algorithm optimization and performance analysis, Master Degree , The Chinese University of Hong Kong, , July 2004 , available at http://www.ee.cuhk.edu.hk/~myuan/Thesis.pdf

[12] http://en.wikipedia.org/wiki/Euclidean_distance

[13] http://www.cavsi.com/questionsanswers/what-is-microphone

[14] http://artsites.ucsc.edu/EMS/music/tech_background/te-20/teces_20.html

[15] http://media.wiley.com/product_data/excerpt/61/04701386/0470138661.pdf

[16] http://www.echelonembedded.com/dsphwlab/files/ChassaingBook_Chap1.pdf

[17] Rıfat Benveniste, Beril Sırmaçek, Cem Ünsalan, A QUICK START TO TEXAS INSTRUMENTS TMS 320C6713 DSK,2010 , available at http://www.ti.com/ww/eu/university/Yeditepe_C6713_DSK_LAB_MANUAL_and_4.pdf

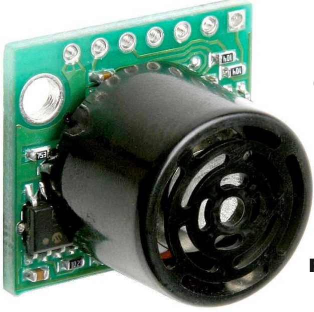
[18] REAL-TIME DSP LABORATORY1: An Introduction to the TMS320C6713 DSK available at http://www.engr.colostate.edu/ECE423/lab01/Lab_Notes/Lab_Notes_1.pdf

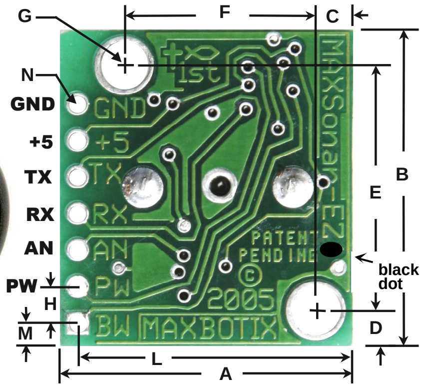
[19] http://www.maxbotix.com/documents/MB1000_Datasheet.pdf.

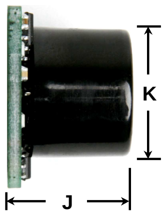
[20] http://en.wikipedia.org/wiki/Microcontroller .

LV-MaxSonar®-EZ0™ High Performance Sonar Range Finder

With 2.5V - 5.5V power the LV-MaxSonar®-EZ0™ provides very short to long-range detection and ranging, in an incredibly small package. The LV-MaxSonar®-EZ0™ detects objects from 0-inches to 254-inches (6.45-meters) and provides sonar range information from 6-inches out to 254-inches with 1-inch resolution. Objects from 0-inches to 6-inches range as 6-inches. The interface output formats included are pulse width output, analog voltage output, and serial digital output.







approximately actual size

A	0.785"	19.9 mm
B	0.870"	22.1 mm
C	0.100"	2.54 mm
D	0.100"	2.54 mm
E	0.670"	17.0 mm
F	0.510"	12.6 mm
G	0.124" dia.	3.1 mm dia.

H	0.100"	2.54 mm
J	0.610"	15.5 mm
K	0.645"	16.4 mm
L	0.735"	18.7 mm
M	0.065"	1.7 mm
N	0.038" dia.	1.0 mm dia.
weight, 4.3 grams		

values are nominal

Features

- Continuously variable gain for beam control and side lobe suppression
- Object detection includes zero range objects
- 2.5V to 5.5V supply with 2mA typical current draw
- Readings can occur up to every 50mS, (20-Hz rate)
- Free run operation can continually measure and output range information
- Triggered operation provides the range reading as desired
- All interfaces are active simultaneously
- Serial, 0 to Vcc, 9600Baud, 81N
- Analog, (Vcc/512) / inch
- Pulse width, (147uS/inch)
- Learns ringdown pattern when commanded to start ranging
- Designed for protected indoor environments
- Sensor operates at 42KHz
- High output square wave sensor drive (double Vcc)

Benefits

- Very low cost sonar ranger
- Reliable and stable range data
- Sensor dead zone virtually gone
- Lowest power ranger
- Quality beam characteristics
- Mounting holes provided on the circuit board
- Very low power ranger, excellent for multiple sensor or battery based systems
- Can be triggered externally or internally
- Sensor reports the range reading directly, frees up user processor
- Fast measurement cycle
- User can choose any of the three sensor outputs

Beam Characteristics

The LV-MaxSonar®-EZ0™ has the most sensitivity of the MaxSonar product line, yielding a controlled wide beam with high sensitivity. Sample results for measured beam patterns are shown below on a 12-inch grid. The detection pattern is shown for;

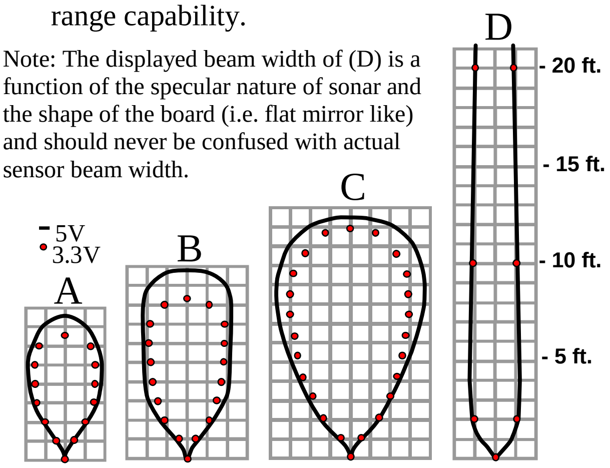
(A) 0.25-inch diameter dowel, note the narrow beam for close small objects,

(B) 1-inch diameter dowel, note the long narrow detection pattern,

(C) 3.25-inch diameter rod, note the long controlled detection pattern,

(D) 11-inch wide board moved left to right with the board parallel to the front sensor face and the sensor stationary. This shows the sensor's range capability.

Note: The displayed beam width of (D) is a function of the specular nature of sonar and the shape of the board (i.e. flat mirror like) and should never be confused with actual sensor beam width.



beam characteristics are approximate

LV-MaxSonar®-EZ0™ Pin Out

GND – Return for the DC power supply. GND (& Vcc) must be ripple and noise free for best operation.

+5V – Vcc – Operates on 2.5V - 5.5V. Recommended current capability of 3mA for 5V, and 2mA for 3V.

TX – When the *BW is open or held low, the TX output delivers asynchronous serial with an RS232 format, except voltages are 0-Vcc. The output is an ASCII capital "R", followed by three ASCII character digits representing the range in inches up to a maximum of 255, followed by a carriage return (ASCII 13). The baud rate is 9600, 8 bits, no parity, with one stop bit. Although the voltage of 0-Vcc is outside the RS232 standard, most RS232 devices have sufficient margin to read 0-Vcc serial data. If standard voltage level RS232 is desired, invert, and connect an RS232 converter such as a MAX232. When BW pin is held high the TX output sends a single pulse, suitable for low noise chaining. (no serial data).

RX – This pin is internally pulled high. The EZ0™ will continually measure range and output if RX data is left unconnected or held high. If held low the EZ0™ will stop ranging. Bring high for 20uS or more to command a range reading.

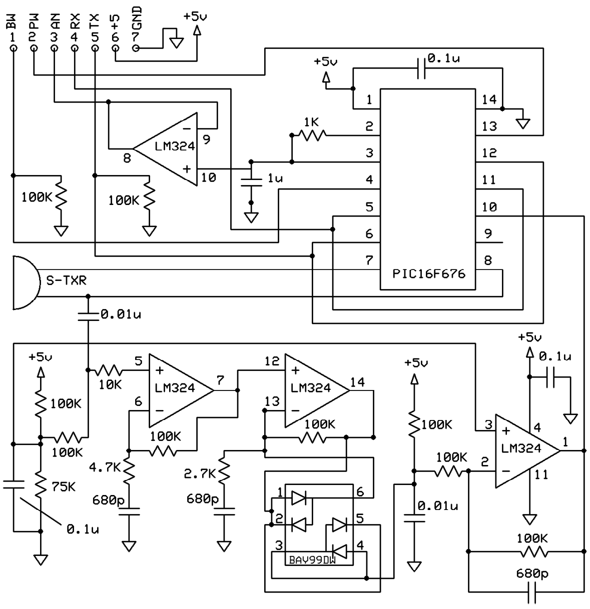
AN – Outputs analog voltage with a scaling factor of (Vcc/512) per inch. A supply of 5V yields ~9.8mV/in. and 3.3V yields ~6.4mV/in. The output is buffered and corresponds to the most recent range data.

PW – This pin outputs a pulse width representation of range. The distance can be calculated using the scale factor of 147uS per inch.

BW – *Leave open or hold low for serial output on the TX output. When BW pin is held high the TX output sends a pulse (instead of serial data), suitable for low noise chaining.

LV-MaxSonar®-EZ0™ Circuit

The LV-MaxSonar®-EZ0™ sensor functions using active components consisting of an LM324, a diode array, a PIC16F676, together with a variety of passive components.



LV-MaxSonar®-EZ0™ Timing Description

250mS after power-up, the LV-MaxSonar®-EZ0™ is ready to accept the RX command. If the RX pin is left open or held high, the sensor will first run a calibration cycle (49mS), and then it will take a range reading (49mS). After the power up delay, the first reading will take an additional ~100mS. Subsequent readings will take 49mS. The LV-MaxSonar®-EZ0™ checks the RX pin at the end of every cycle. Range data can be acquired once every 49mS.

Each 49mS period starts by the RX being high or open, after which the LV-MaxSonar®-EZ0™ sends thirteen 42KHz waves, after which the pulse width pin (PW) is set high. When a target is detected the PW pin is pulled low. The PW pin is high for up to 37.5mS if no target is detected. The remainder of the 49mS time (less 4.7mS) is spent adjusting the analog voltage to the correct level. When a long distance is measured immediately after a short distance reading, the analog voltage may not reach the exact level within one read cycle. During the last 4.7mS, the serial data is sent. The LV-MaxSonar®-EZ0™ timing is factory calibrated to one percent at five volts, and in use is better than two percent. In addition, operation at 3.3V typically causes the objects range, to be reported, one to two percent further than actual.

LV-MaxSonar®-EZ0™ General Power-Up Instruction

Each time after the LV-MaxSonar®-EZ0™ is powered up, it will calibrate during its first read cycle. The sensor uses this stored information to range a close object. It is important that objects not be close to the sensor during this calibration cycle. The best sensitivity is obtained when it is clear for fourteen inches, but good results are common when clear for at least seven inches. If an object is too close during the calibration cycle, the sensor may then ignore objects at that distance.

The LV-MaxSonar®-EZ0™ does not use the calibration data to temperature compensate for range, but instead to compensate for the sensor ringdown pattern. If the temperature, humidity, or applied voltage changes during operation, the sensor may require recalibration to reacquire the ringdown pattern. Unless recalibrated, if the temperature increases, the sensor is more likely to have false close readings. If the temperature decreases, the sensor is more likely to have reduced up close sensitivity. To recalibrate the LV-MaxSonar®-EZ0™, cycle power, then command a read cycle.

Product / specifications subject to change without notice. For more info visit www.maxbotix.com/MaxSonar-EZ1_FAQ

Email: info@maxbotix.com
Web: www.maxbotix.com