****

**An\_Najah National University**

**FACILITY OF ENGINEERING**

**Department of Electrical Engineering**

**SPEECH RECOGNITION CONTROL**

Graduation project submitted in partial fulfillment of the requirements for the Degree of B.S.C in Electrical Engineering.

Supervisor:

**Dr. Raid Jaber**.

Students:

**Ala’a Areef Khader**

**Banan Basem Soudi**

اهــــــــــــــداء

إلى المُعلِّمِ الأوَّلْ ، وسيِّدِ الخلقِ وأطهرِهِم ، مَن أنارَ الله بهِ ظَلامَ الدُّنيا ، وغيَّرَ على يدَيهِ عُقولَ الجهل إلى عقولٍ مُبصِرةٍ بنور الله ، إلى نبيِّنا محمّد صلى اللهُ عليه وسلَّم .

إلى مَن علَّمَتنا كيفَ نصنَعُ مِن جُرحِنا نَصراً ،

ومِن موتِنا عيشاً ،

ومِن قيدِنا شرَفاً ،

وَ مَن تمادَينا لأجلِها على كُل اللاهِثينَ عبثاً لتجهيلِنا ، إلى مَن أتقنَا بها كَيفَ نكونُ رجالاً نحمي وجهَ الحق الغائِب ، وقاتَلنا رغمَ الألَم ، كَي تُفاخِرَ بنا كما فاخَرنا بها مُنذُ وُلِدنا ، وأقسَمنا باللهِ أن نبني ما استطعنا فيها ، كي تبقى عصيَّة على الغرباء ، إلى فلسطين الحبيبة .

إلى أولئكَ الرجال ، في زمن عزَّ فيهِ الرجال ،والمُبصِرينَ في زمَن الظلامِ الدَّامس ، والراسخينَ كشجر هذهِ الأرض ، إلى مَن أقدَموا على دفعِ الثَّمن غالِياً ، وسلَّمونا أمانةٌ نفخرُ بحَملِها ، إلى أسرى هذا الوطن وشهدائهِ .

إلى هذا الصرح الأبي ،

الذي أدركنا معَهُ كيفَ نتحدّى الماضي ، لنرسُمَ وجهاً رائِعاً للحاضر والمُستقبَل

إلى جامعة النجاح ،

وإدارييها ،

ومعلِّميها الذينَ بهِم رسَمنا طريقَ الأمل ، ووجهَ النجاح .

إلى مَن تخونُنا بحضرتِهِم عِباراتُ الحُب والإجلال والعِرفان ، إلى ذاكَ الوجهِ الذي عرَفنا خِلالَهُ كيفَ نكون ، وكَيفَ نُواجِه ، وَكَيفَ نبني ، وكيفَ نكونُ أهلاً لِكُل ما يُؤمِنونَ بأنَّهُ الأرقى ، والأغلى ، والأرفعُ مكانة ، إلى مَن تحدَّينا كي نصِلَ بقُلوبِهِم لِفرحةٍ لاتليقُ بغيرهِم ، وَغايَةِ كريمة هُم منحونا شرفَ الوُصول إلَيها ، إلى والِدَينا الأكارم .

إلى كُل مَن سانَدَنا كي نصِل ، وكُل مَن وقفَ بجانِبِنا كَي نُكمِلَ ما صَبَونا إلَيه ،

نُهدي عمَلَنا هذا

**ِAbstract**

The main objective in this project is to make insure MATLAB codes that will be formed as a complete program that will record a voice command which specified in the four chosen words (GO,STOP,BACK,LEFT,RIGHT ) and analyzing it then recognize it , then that will send a suitable code into parallel port (LPT1) that will be a command suitable for the spoken word this command will control the robotic car which communicated with computer by the radio waves .

This project consists of three main parts which can be simplified as following:

1. **Software**: that will record the signal and make suitable processing on it.
2. **Wireless communication**: link to transfer the command from the computer to the robotic car.
3. **Hard ware**: This can be summarized in the robotic car.

|  |  |
| --- | --- |
| أهـــــداء | 2 |
| Abstract | 3 |
| Table of Contents | 4 |
| List of Figures | 5 |
| List of Tables | 6 |
| List of Equations | 7 |
| **Chapter one : Introduction.** | 7 |
| 1.1 Introduction | 7 |
| 1.2 History of Speech Recognition | 7 |
| 1.2.1 Early automatic speech recognizers | 8 |
| 1.2.2 Technology drivers since the 1970’s | 8 |
| 1.2.3 Technology directions in the 1980’s and 1990’s | 9 |
| 1.3 Summary | 11 |
| **Chapter Two: Preparations.** |  |
| 2.1 project Goals and specifications | 12 |
| 2.2 Project Roadmap | 13 |
| 2.3 Project Challenges | 14 |
| 2.4 System General Block Diagram | 16 |
| **Chapter three: Speech Acquisition.** | 17 |
| 3.1 **Introduction** | 17 |
| 3.2 Microphone | 17 |
| 3.2.1 How microphone works? | 18 |
| 3.2.2 Microphone types | 18 |
| 3.3 Sound Card | 19 |
| 3.3.1 History of sound cards: | 19 |
| 3.3.2 Components | 20 |
| 3.3.3 Color codes | 21 |
| **Chapter Four: Signal Processing.** | 23 |
| 4.1 Introduction | 23 |
| 4.2 Digital Signal Processing | 23 |
| 4.2.1 Speech Processing | 25 |
| 4.3 Digital Transmission and Storage of Speech | 25 |
| 4.3.1 Speech synthesis systems | 26 |
| 4.3.2 Speaker verification and identification systems | 26 |
| 4.3.3 Speech recognition systems | 26 |
| 4.3.4 Aids-to-the-handicapped | 27 |
| 4.3.5 Enhancement of signal quality | 27 |
| 4.4 Operational Considerations for Limited Vocabulary Applications | 27 |
| 4.4.1 Noise background | 27 |
| 4.4.2 Breath noise | 28 |
| 4.4.3 Word Boundary Detection | 29 |
| 4.4.4 Operator-Originated Babble. | 29 |
| 4.5 Recording the Voice. | 30 |
| 4.5.1 Running the program | 31 |
| 4.5.2 Recording | 31 |
| 4.5.3 Listening to the Waveform | 33 |
| 4.6 Filtering Process | 33 |
| 4.6.1 Types of filter | 34 |
| 4.6.2 FIR digital filter | 35 |
| 4.6.3 Characteristics of FIR digital filters | 36 |
| 4.6.4 Butterworth digital Filters | 36 |
| 4.6.5 Overview | 37 |
| 4.6.6 Filter part assumptions | 38 |
| 4.6.7 Filter Circuit Design | 39 |
| 4.6.8 Comparison with other linear filters | 40 |
| 4.7 Spectral Analysis | 41 |
| 4.7.1 Fast Fourier transform (FFT) | 41 |
| 4.7.2 Fourier Analysis and Signal Filtering | 41 |
| 4.7.3 Signal Smoothing Using Fourier Transforms | 43 |
| 4.7.4 Spectral analysis applications | 44 |
| 4.7.5 Spectral processing system requirement | 44 |
| 4.7.6 Hidden MARKOV Modeling | 45 |
| **Chapter five: Controlling Robotic Car .** | 47 |
| 5.1 Parallel port (LPT1) | 47 |
| 5.2 Radio frequency | 49 |
| 5.2.1 Radio frequency advantage | 49 |
| 5.2.2 Radio frequency disadvantages | 49 |
| 5.2.3 Application of radio wave | 49 |
| 5.2.4 Radio frequency transmitter and receiver circuits | 49 |
| 5.2.4.1 Component RF receiver-transmitter | 49 |
| 5.3 Robotic Car | 53 |
| 5.3.1 Robotic car | 53 |
| 5.3.3 Mechanical design of robotic car | 53 |
| 5.3.4 DC-motor characteristics | 54 |
| 5.3.4.1 DC motor principle | 54 |
| 5.3.4.2 DC motor types | 55 |
|  |  |
| **Chapter Six : MATLAB Software .** | 56 |
| 6.1 Introduction | 56 |
| 6.2 MATLAB Software | 56 |
| 6.2.1 The MATLAB Mathematical Function Library | 57 |
| 6.2.2 The MATLAB Language | 57 |
| 6.2.3 Graphics | 57 |
| 6.3 Speech Signal Processing Using MATLAB | 58 |
| 6.3.1 Recording the signals | 58 |
| 6.3.2 Filtering | 59 |
| 6.3.3 Normalizing | 60 |
| 6.3.4 Frequency Domain Analysis (Spectral Analysis) | 61 |
| 6.4 Fingerprint Comparison | 62 |
| 6.4.1 Creating Signals Fingerprints | 62 |
| 6.4.2 Fingerprint Comparison | 64 |
| 6.5 Resultant Recognized Matrix Applications | 64 |
| 6.6Conclusion | 65 |
| **Chapter Seven : Conclusion.** | 66 |
| Appendix | 69 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**chapter ONE**

**Introduction**

**1.1 Introduction**

The challenge to the researchers is to create software that works with imperfect, but historically invaluable, audio tracks. Commercial voice recognition software designed for personal computer users works well if the speaker talks clearly in English, or another common West European language. The personal accounts have been given in many different East European languages. Further challenges are presented by strong accents and the emotion shown in many recordings.

The researchers do not aim to design a system that can transcribe the recordings word-for-word, but a searchable database that links different testimonies to key events and places. "We want to build a speech recognition system that is good enough to recognise some of words," Although the largest strides in the development of voice recognition technology have occurred in the past two decades, this technology really began with Alexander Graham Bell's inventions in the 1870s. By discovering how to convert air pressure waves (sound) into electrical impulses, he began the process of uncovering the scientific and mathematical basis of understanding speech.

In the 1950s, Bell Laboratories developed the first effective speech recognizer for numbers. In the 1970s, the ARPA Speech Understanding Research project developed the technology further - in particular by recognizing that the objective of automatic speech recognition is the understanding of speech not merely the recognition of words.

By the 1980s, two distinct types of commercial products were available. The first offered speaker-independent recognition of small vocabularies. It was most useful for telephone transaction processing. The second, offered by Kurzweil Applied Intelligence, Dragon Systems, and IBM, focused on the development of large-vocabulary voice recognition systems so that text documents could be created by voice dictation.

Over the past two decades, voice recognition technology has developed to the point of real-time, continuous speech systems that augment command, security, and content creation tasks with exceptionally high accuracy.

**1.2 History of Speech Recognition**

Designing a machine that mimics human behavior, particularly the capability of speaking naturally and responding properly to spoken language, has intrigued engineers and scientists for centuries. Since the 1930s, when Homer Dudley of Bell Laboratories proposed a system model for speech analysis and synthesis , the problem of automatic speech recognition has been approached progressively, from a simple machine that responds to a small set of sounds to a sophisticated system that responds to fluently spoken natural language and takes into account the varying statistics of the language in which the speech is produced. Based on major advances in statistical modeling of speech in the 1980s, automatic speech recognition systems today find widespread application in tasks that require a human-machine interface, such as automatic call processing in the telephone network and query-based information systems that do things like provide updated travel information, stock price quotations, weather reports, etc. In this article, we review some major highlights in the research and development of automatic speech recognition during the last few decades so as to provide a technological perspective and an appreciation of the fundamental progress that has been made in this important area of information and communication technology.

**1.2.1 Early automatic speech recognizers**

Early attempts to design systems for automatic speech recognition were mostly guided by the theory of acoustic-phonetics, which describes the phonetic elements of speech (the basic sounds of the language) and tries to explain how they are acoustically realized in a spoken utterance. These elements include the phonemes and the corresponding place and manner of articulation used to produce the sound in various phonetic contexts. For example, in order to produce a steady vowel sound, the vocal cords need to vibrate (to excite the vocal tract), and the air that propagates through the vocal tract results in sound with natural modes of resonance similar to what occurs in an acoustic tube. These natural modes of resonance, called the formants or formant frequencies, are manifested as major regions of energy concentration in the speech power spectrum. In 1952, Davis, Biddulph, and Balashek of Bell Laboratories built a system for isolated digit recognition for a single speaker [1], using the formant frequencies measured (or estimated) during vowel regions of each digit.

**1.2.2 Technology drivers since the 1970’s**

In the late 1960’s, Atal and Itakura independently formulated the fundamental concepts of Linear Predictive Coding (LPC, which greatly simplified the estimation of the vocal tract response from speech waveforms. By the mid 1970’s, the basic ideas of applying fundamental pattern recognition technology to speech recognition, based on LPC methods, were proposed by Itakura, Rabiner and Levinson and others.

Other systems developed under DARPA’s SUR program included CMU’s Hearsay(-II) and BBN’s HWIM , Neither Hearsay-II nor HWIM (Hear What I Mean) met the DARPA program’s performance goal at its conclusion in 1976. However, the approach proposed by Hearsay-II of using parallel asynchronous processes that simulate the component knowledge sources in a speech system was a pioneering concept. The Hearsay-II system extended sound identity analysis (to higher level hypotheses) given the detection of a certain type of (lower level) information or evidence, which was provided to a global “blackboard” where knowledge from parallel sources was integrated to produce the next level of hypothesis. BBN’s HWIM system, on the other hand, was known for its interesting ideas including a lexical decoding network incorporating sophisticated phonological rules (aimed at phoneme recognition accuracy), its handling of segmentation ambiguity by a lattice of alternative hypotheses, and the concept of word verification at the parametric level. Another system worth noting of the time was the DRAGON system by Jim Baker, who moved to Massachusetts to start a company with the same name in the early 1980s.

**1.2.3 Technology directions in the 1980’s and 1990’s**

Speech recognition research in the 1980’s was characterized by a shift in methodology from the more intuitive template-based approach (a straightforward pattern recognition paradigm) towards a more rigorous statistical modeling framework. Although the basic idea of the hidden Markov model (HMM) was known and understood early on in a few laboratories (e.g., IBM and the Institute for Defense Analyses (IDA)), the methodology was not complete until the mid 1980’s and it wasn’t until after widespread publication of the theory that the hidden Markov model became the preferred method for speech recognition. The popularity and use of the HMM as the main foundation for automatic speech recognition and understanding systems has remained constant over the past two decades, especially because of the steady stream of improvements and refinements of the technology.

The hidden Markov model, which is a doubly stochastic process, models the intrinsic variability of the speech signal (and the resulting spectral features) as well as the structure of spoken language in an integrated and consistent statistical modeling framework. As is well known, a realistic speech signal is inherently highly variable (due to variations in pronunciation and accent, as well as environmental factors such as reverberation and noise). When people speak the same word, the acoustic signals are not identical (in fact they may even be remarkably different), even though the underlying linguistic structure, in terms of the pronunciation, syntax and grammar, may remain the same. The formalism of the HMM is a probability measure that uses a Markov chain to represent the linguistic structure and a set of probability distributions to account for the variability in the acoustic realization of the sounds in the utterance. Given a set of known (text-labeled) utterances, representing a sufficient collection of the variations of the words of interest (called a training set), one can use an efficient estimation method, called the Baum-Welch algorithm , to obtain the “best” set of parameters that define the corresponding model or models. The estimation of the parameters that define the model is equivalent to training and learning. The resulting model is then used to provide an indication of the likelihood (probability) that an unknown utterance is indeed a realization of the word (or words) represented by the model. The probability measure represented by the hidden Markov model is an essential component of a speech recognition system that follows the statistical pattern recognition approach, and has its root in Bayes’ decision theory. The HMM methodology represented a major step forward from the simple pattern recognition and acoustic-phonetic methods used earlier in automatic speech recognition systems.

Another technology that was (re)introduced in the late 1980’s was the idea of artificial neural networks (ANN). Neural networks were first introduced in the 1950’s, but failed to produce notable results initially , The advent, in the 1980’s, of a parallel distributed processing (PDP) model, which was a dense interconnection of simple computational elements, and a corresponding “training” method, called error back-propagation, and revived interest around the old idea of mimicking the human neural processing mechanism. A particular form of PDP, the In1990’s great progress was made in the development of software tools that enabled many individual research programs all over the world. As systems became more sophisticated (many large vocabulary systems now involve tens of thousands of phone unit models and millions of parameters), a well-structured baseline software system was indispensable for further research and development to incorporate new concepts and algorithms. The system that was made available by the Cambridge University team (led by Steve Young), called the Hidden Markov Model Tool Kit (HTK), was (and remains today as) one of the most widely adopted software tools for automatic speech recognition research.

**1.3 Summary**

In 1960’s we were able to recognize small vocabularies (order of 10-100 words) of isolated words, based on simple acoustic-phonetic properties of speech sounds. The key technologies that were developed during this time frame were filter-bank analyses, simple time normalization methods, and the beginnings of sophisticated dynamic programming methodologies. In the 1970’s we were able to recognize medium vocabularies (order of 100-1000 words) using simple template-based, pattern recognition methods. The key technologies that were developed during this period were the pattern recognition models, the introduction of LPC methods for spectral representation, the pattern clustering methods for speaker-independent recognizers, and the introduction of dynamic programming methods for solving connected word recognition problems. In the 1980’s we started to tackle large vocabulary (1000-unlimited number of words) speech recognition problems based on statistical methods, with a wide range of networks for handling language structures. The key technologies introduced during this period were the hidden Markov model (HMM) and the stochastic language model, which together enabled powerful new methods for handling virtually any continuous speech recognition problem efficiently and with high performance. In 1990’s large vocabulary systems was built with unconstrained language models, and constrained task syntax models for continuous speech recognition and understanding. The key technologies developed during this period were the methods for stochastic language understanding, statistical learning of acoustic and language models, and the introduction of finite state transducer framework)and the methods for their determination and minimization for efficient implementation of large vocabulary speech understanding systems.



**Fig. 1.1: Milestones In Speech Recognition Technology Over The Past 40 Years**.

After nearly five decades of research, speech recognition technologies have finally entered the marketplace, benefiting the users in a variety of ways. Throughout the course of development of such systems, knowledge of speech production and perception was used in establishing the technological foundation for the resulting speech recognizers. Major advances, however, were brought about in the 1960’s and 1970’s via the introduction of advanced speech representations based on LPC analysis and spectral analysis methods, and in the 1980’s through the introduction of rigorous statistical methods based on hidden Markov models. All of this came about because of significant research contributions from academia, private industry and the government. As the technology continues to mature, it is clear that many new applications will emerge and become part of our way of life – thereby taking full advantage of machines that are partially able to mimic human speech capabilities. The challenge of designing a machine that truly functions like an intelligent human is still a major one going forward. Our accomplishments, to date, are only the beginning and it will take many years before a machine can pass the Turing test, namely achieving performance that rivals that of a human.

**CHAPTER TWO**

**PREPARATIONS**

**2.1 project Goals and specifications**

The main objective is to develop speech recognition system that was flexible, but not necessarily speaker-independent (for speaker-independence is very hard thing to achieve). Since the speech recognition system is geared towards the control of the software application, we placed particular importance on accuracy and robustness, envisioning that this system could one day be incorporated into the hands-free control of certain software or hardware applications, The tendency for a system that relied on our own ingenuity and originality, The used speech recognition system in this project is used as a starting point for building a knowledge base and then improvement was applied to this system to made a unique model , Minimum performance specifications are listed below as following :

* Ability to perform word-recognition with in 1 second of when the person has finished speaking (near real time performance ) .
* Be able to run the recognition engine for an indefinite length of time without it running out of memory.

**2.2 Project Roadmap**

Due to what mentioned earlier, the goal was to differentiate between five voice commands (GO, BACK, LEFT, RIGHT, and STOP) we have followed the following procedure to accomplish this goal:

* The samples of voice are taken for the following words (GO, BACK, STOP, LEFT, RIGHT) for the speaker voice.
* The investigation and study are done about the famous speech recognition techniques and methods and the method of Spectrum Analysis is selected to work on it.
* The mentioned method functions are implemented as MATLAB subroutines that doing the necessary calculations on the signal of the voice that can be summarized as following:
* Normalization.
* Filtering process.
* Spectral analysis.

That will be mentioned in Chapter6 in more specified form.

* The resulted statically data coming out of this analysis have guided to decide which technique is the best for differentiating between different words in the best way.
* The second step is to make the cooperation process between the talked word and the database, the following step is to send the data into the parallel port to control the car.

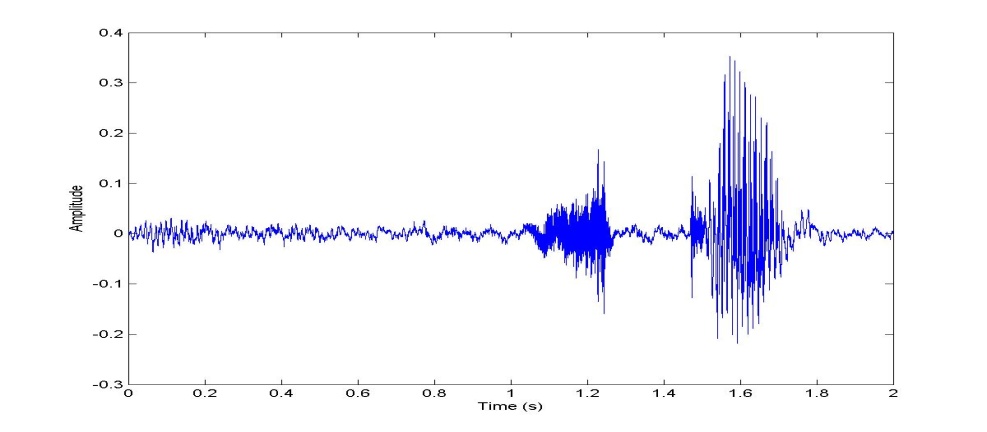
**2.3 Project Challenges**

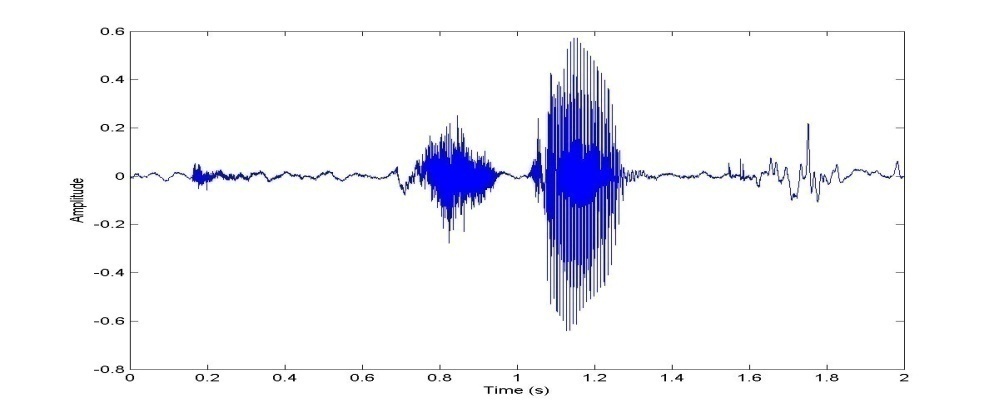
Any sound measured by the microphone is simply a sequence of numbers, the reference word is could be represented also as a sequence of numbers. Speech recognition is a process by which one sequence of numbers is compared to another sequence of numbers in attempt to find the best fit.

In particular an utterance differs from a stored template in 3 ways:

* **Error in magnitude**: such as interference, Noise, and other magnitude distortions corrupt the input signal and can make it sound different from the reference signal.
* **Error in time**: Such as unexpected pauses, unusually fast or slow speaking styles and other changes in speed can randomly shift the position of the input relative to the template.
* **Combination of magnitude and time errors**: Randomly distorter’s signals values and also shifts position randomly in time real speech falls under this category because people never say the same word exactly the same way twice, in addition to whatever background noise might be present in the environment. People can also pause unexpectedly, or say a word faster or slower than expected, or stutter or jump-around, or even be uncooperative. Over a sufficiently long interval an input signal can vary from the ideal and multitude of ways.

The following two graphs are for the same world for two different talker’s shows the last theory clearly.





**Figure 2.1 Two graphs for the recorded word “STOP”**

As shown in figure 2.1 it’s the signal of the word “STOP” which was spoken by the same person, these plots are generally similar. However, there are also differences. Note that the first “STOP” has different values of energy than the second one and its starting point of the “STOP” is later than the second.

Such differences are called intra-speaker differences; the same person can utter the same word in slightly different ways each time. The person can Pause, speak faster and slower or emphasize certain syllables. A recognition system needs to be robust enough to understand these different punctuations of the same word are not entirely different words, but simply different examples.

Matching words across different speakers is an even more challenging task, whereas differences between words spoken by the same person are relatively small, inter speaker differences are huge to discuss now.

**2.4 System General Block Diagram**

This section introduces system block diagram that was followed to design our system, followed by a brief description of each block, further description will be available in the coming chapters.

**Compare with data base**

**Input Voice**

**Car receiver**

**Parallel port**

**To the transmitter**

**Signal processing**

**Sound card**

**microphone**

**Figure 2.2: system level Block Diagram**

The following steps are simply the steps of the project from 1-5:

* The user speaks to the microphone which receives the voice and recording it in the form of wave sound the microphone properties and information are shown in the chapter 3
* The voice that recorded in wav form is saved in MATLAB program in special program and some processing on it must be taken place as shown in following chapter 4 (filtering , Normalizing and spectral analysis ) .
* The produced voice after this step is ready to make compare process with the data base to recognize the spoken word.
* The MATLAB program transfer the word into decimal code that shown in other chapters and transmitting it in the parallel port as shown in Chapter 6.
* The car receives the signal into special radio wav and applies the command which specified in (STOP, GO , BACK , LEFT , RIGHT ).

**Chapter three**

**Speech acquisition**

**3.1 Introduction**

Human hearing system is capable of capturing noise over a very wide frequency spectrum, from 20 Hz on the low frequency end to upwards of 20,000 Hz on the high frequency end. The human voice, however, does not have this kind of range. Typical frequencies for the human voice are on the order of 100 Hz to 2,000 Hz.

According to Nyquist Theory, the sampling rate should be twice as fast as the highest frequency of the signal, to ensure that there are at least two samples taken per signal period. Thus, the sampling rate of the program would have to be no less than 4,000 samples per second.

The project was simplified in two ways or layers:

* Software layer: Consists of MATLAB program and signal processing and controlling the LPT port.
* Hardware layer: Consists of the Microphone, soundcard parallel port, transmitter and receiver of the car and the robotic car.

**3.2 Microphone**

All microphones are converts the sound energy into electrical energy, but there are many different ways to do the conversion of the energy, that effect the process of the project greatly.

The used microphone in this project is known as desktop microphone they are two or three different types of microphones such as desktop, headset, lab top microphones. Each one differs than other in the process of working as mentioned previously.

Here we must mention the efficiency of the conversion process because of its impotency because the amount of the acoustic energy produced by voices is too small. to the uninitiated , the range of available makes and models of microphones can seem daunting , there are so many types , sizes and shapes , polar pattern , frequency and response ; understanding these is the most important step in choosing a microphone for an application .

**3.2.1 How microphone works?**

Microphones are a type of transducers – a device which converts the energy from to another. Microphones converts acoustical energy (sound waves) into electrical energy (Audio signals), Different types of microphones have different ways of converting energy but they all share one thing in common: The diaphragm. This is a thin piece of material (such as paper m Plastic or Aluminum) which vibrates when it’s struck by sound waves.

When the sound wave reaches the diaphragm that causes it to vibrate this vibrations are transmitted to other components in the microphone, and these vibrations are converted into electrical current which becomes the audio signal.

**3.2.2 Microphone types**

There are essentially two types of microphones:

1. **Dynamic**

Dynamic microphones are versatile and ideal for general-purpose use; they use a simple design with few moving parts. They are relatively sturdy and resilient to rough handling. They are also better suited to handling high volume levels, such as from certain musical instruments or amplifiers. They have no internal amplifier and do not require batteries or external power.

When the magnet moves near a coil of wire an electrical current is generated in the wire, this principle is the method of working in the dynamic microphones that uses the wire coil and magnet to create the audio signal.

1. **Condenser**

Condenser means capacitor, an electrical component which stores energy in the form of the electro static field. The term condenser is actually obsolete but has stuck as the name for this type of microphones, which uses a capacitor to convert acoustical energy into electrical energy.

Condenser microphones need external power supply to work from battery or other external sources, the resulting audio signal is stronger signal than that from the dynamic microphones , conductors also tend to be more sensitive and responsive than dynamics , they are not ideal for high-volume work , as their sensitivity makes them prone to distort .

A capacitor has two plates with a voltage between them , in the condenser microphone one of these plates is made from very light material and acts as a diaphragm , the diaphragm vibrates when struck by a sound waves , changing the distance between the two plates and therefore changing the capacitance, specifically , when the plates are closer together , capacitance increases and a charge current occurs , when the plates are further apart , capacitance decreases and a discharge current occurs .

**3.3 Sound Card**

Sound card allows you to connect a microphone to the computer and record your sound files,

When sound is recorded through the microphone the changes in air pressure cause the microphone’s diaphragm to move in similar way to that of the eardrum, these minute movements are then converted into changes in voltage.

Essentially, all sound cards produce sound in this way, only in reverse. They create or playback, sound waves. The changes in voltage are then amplified, causing the loudspeaker to vibrate, these vibrations because changes in air pressure which are further interpreted as sound.

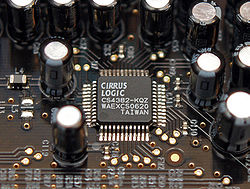
**3.3.1 History of sound cards:**

Sound cards for computers compatible with the [IBM PC](http://en.wikipedia.org/wiki/IBM_PC) were very uncommon until 1988, which left the single internal [PC speaker](http://en.wikipedia.org/wiki/PC_speaker) as the only way early PC software could produce sound and music. The speaker hardware was typically limited to [square waves](http://en.wikipedia.org/wiki/Square_wave), which fit the common nickname of "beeper". The resulting sound was generally described as "beeps and bops". Several companies, most notably [Access Software](http://en.wikipedia.org/wiki/Access_Software), developed techniques for digital sound reproduction over the PC speaker; the resulting audio, while baldly functional, suffered from distorted output and low volume, and usually required all other processing to be stopped while sounds were played. Other home computer models of the 1980s included hardware support for digital sound playback, or music synthesis (or both), leaving the IBM PC at a disadvantage to them when it came to multimedia applications such as music composition or gaming.

It is important to note that the initial design and marketing focuses of sound cards for the IBM PC platform were not based on gaming, but rather on specific audio applications such as music composition ([AdLib Personal Music System](http://en.wikipedia.org/wiki/AdLib), [Creative Music System](http://en.wikipedia.org/wiki/Creative_Music_System), [IBM Music Feature Card](http://en.wikipedia.org/w/index.php?title=IBM_Music_Feature_Card&action=edit&redlink=1)) or on speech synthesis (Digispeech *DS201*, [Covox Speech Thing](http://en.wikipedia.org/wiki/Covox_Speech_Thing), Street Electronics *Echo*). Only until Sierra and other game companies became involved in 1988 was there a switch toward gaming.[2]

**3.3.2 Components**

The modern PC sound card contains several hardware systems relating to the production and capture of audio, the two main subsystems being for digital audio captor and reply and music synthesis along with some glue hardware.

**[](http://en.wikipedia.org/wiki/File:CirrusLogicCS4282-AB.jpg)**

**Figure 3.1 : Sound Card Components**

The digital audio section of a sound card consists of matched pair of 16-bit digital-to-analog (DAC) and analogue to digital (ADC) converts and a programmable sample rate generator.

For some years, most PC sound cards have had multiple FM synthesis voices (typically 9 or 16) which were usually used for MIDI music. The full capabilities of advanced cards aren't often completely used; only one (mono) or two ([stereo](http://en.wikipedia.org/wiki/Stereo)) voice(s) and channel(s) are usually dedicated to playback of digital sound samples, and playing back more than one digital sound sample usually requires a software [down mix](http://en.wikipedia.org/wiki/Downmixing) at a fixed sampling rate. Modern low-cost integrated soundcards (i.e., those built into motherboards) such as [audio codec](http://en.wikipedia.org/wiki/Audio_codec) like those meeting the [AC'97](http://en.wikipedia.org/wiki/AC'97) standard and even some budget expansion soundcards still work that way. They may provide more than two sound output channels (typically 5.1 or 7.1 [surround sound](http://en.wikipedia.org/wiki/Surround_sound)), but they usually have no actual hardware polyphony for either sound effects or MIDI reproduction, these tasks are performed entirely in software. This is similar to the way inexpensive [soft modems](http://en.wikipedia.org/wiki/Softmodem) perform modem tasks in software rather than in hardware).

Also, in the early days of [wavetable synthesis](http://en.wikipedia.org/wiki/Wavetable_synthesis), some sound card manufacturers advertised polyphony solely on the MIDI capabilities alone. In this case, the card's output channel is irrelevant (and typically, the card is only capable of two channels of digital sound). Instead, the polyphony measurement solely applies to the amount of MIDI instruments the sound card is capable of producing at one given time.

Today, a sound card providing actual hardware polyphony, regardless of the number of output channels, is typically referred to as a "hardware audio accelerator", although actual voice polyphony is not the sole (or even a necessary) prerequisite, with other aspects such as hardware acceleration of 3D sound, [positional audio](http://en.wikipedia.org/w/index.php?title=Positional_audio&action=edit&redlink=1) and real-time DSP effects being more important.

Since digital sound playback has become available and provided better performance than synthesis, modern soundcards with hardware polyphony don't actually use DACs with as many channels as voices, but rather perform voice mixing and effects processing in hardware (eventually performing digital filtering and conversions to and from the frequency domain for applying certain effects) inside a dedicated DSP. The final playback stage is performed by an external (in reference to the DSP chip(s)) DAC with significantly fewer channels than voices.

**3.3.3 Color codes**

Connectors on the sound cards are color coded as per the [PC System Design Guide](http://en.wikipedia.org/wiki/PC_System_Design_Guide). They will also have symbols with arrows, holes and sound waves that are associated with each jack position; the meaning of each is given below:

**Table 1: Color code for sound card output connectors**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Color | | Function | Connector | symbol |
|  | [Pink](http://en.wikipedia.org/wiki/Pink) | Analog [microphone](http://en.wikipedia.org/wiki/Microphone) audio input. | 3.5 mm [TRS](http://en.wikipedia.org/wiki/TRS_connector) | A microphone |
|  | Light [blue](http://en.wikipedia.org/wiki/Blue) | Analog [line level](http://en.wikipedia.org/wiki/Line_level) audio input. | 3.5 mm [TRS](http://en.wikipedia.org/wiki/TRS_connector) | An arrow going into a circle |
|  | [Lime green](http://en.wikipedia.org/wiki/Lime_green) | Analog line level audio output for the main stereo signal (front speakers or headphones). | 3.5 mm [TRS](http://en.wikipedia.org/wiki/TRS_connector) | Arrow going out one side of a circle into a wave |
|  | [Brown](http://en.wikipedia.org/wiki/Brown)/Dark | Analog line level audio output for a special panning, 'Right-to-left speaker'. | 3.5 mm [TRS](http://en.wikipedia.org/wiki/TRS_connector) |  |
|  | [Black](http://en.wikipedia.org/wiki/Black) | Analog line level audio output for surround speakers, typically rear stereo. | 3.5 mm [TRS](http://en.wikipedia.org/wiki/TRS_connector) |  |

**chapter four**

**signal processing**

**4.1 Introduction**

The general problem of information manipulation and processing is depicted in Fig.1. In the case of speech signals the human speaker is the information source. The measurement or observation is generally the acoustic waveform.

Signal processing involves first obtaining a representation of the signal based on a given model and then the application of some higher level transformation in order to put the signal into a more convenient form. The last step in the process is the extraction and utilization of the massage information. This step may be performed either by human listeners or automatically by machine. By way of example, a system whose function is to automatically identify a speaker from a given set of speakers might use a time-dependent spectral representation of the speech signal. One possible signal transformation would be to average spectra across an entire sentence, compare the average spectrum to a stored averaged spectrum template for each possible speaker, and then based on a spectral similarity measurement choose the identity of speaker. For this example the “information” in the signal is the identity of the speaker.

Thus, processing of speech signals generally involves two tasks. First, it is a vehicle for obtaining a general representation of a speech signal in either waveform or parametric form. Second, signal processing serves the function of aiding in the process of transforming the signal representation into alternate forms which are less general in nature, but more appropriate to specific applications.

**4.2 Digital Signal Processing**

Digital signal processing is concerned both with obtaining discrete representation of signals, and with the theory, design, and implementation of numerical procedures for processing the discrete representation. The objectives in digital signal processing are identical to those in analog signal processing. Therefore, it is reasonable to ask why digital signal processing techniques should be signaled out for special consideration in the context of speech communication. A number of very good reasons can be cited. First, and probably most important, is the fact that extremely sophisticated signal processing functions can be implemented using digital techniques. The algorithms are intrinsically discrete-time, signal processing system. For the most part, it is not appropriate to view these systems as approximations to analog systems. Indeed in many cases there is no realizable counterpart available with analog implementation.

Digital signal processing techniques were first applied in speech processing problems, as simulations of complex analog systems. The point of view initially was that analog systems could be simulated on a computer to avoid the necessity of building the system in order to experiment with choices of parameters and other design considerations. When digital simulations of analog systems were first applied, the computations required a great deal of time. For example, as much as an hour might have been required to process only a few seconds of speech. In the mid 1960's a revolution in digital signal processing occurred. The major catalysts were the development of faster computers and rapid advances in the theory of digital signal processing techniques. Thus, it became clear that digital signal processing systems had virtues far beyond their ability to simulate analog systems. Indeed the present attitude toward laboratory com­puter implementations of speech processing systems is to view them as exact simulations of a digital system that could be implemented either with special purpose digital hardware or with a dedicated computer system.

In addition to theoretical developments, concomitant developments in the area of digital hardware have led to further strengthening of the advantage of digital processing techniques over analog systems. Digital systems are reliable and very compact. Integrated circuit technology has advanced to a state where extremely complex systems can be implemented on a single chip. Logic speeds are fast enough so that the tremendous number of computations required in many signal processing functions can be implemented in real-time at speech sampling rates.

There are many other reasons for using digital techniques in speech comm­unication systems. For example, if suitable coding is used, speech in digital form can be reliably transmitted over very noisy channels. Also, if the speech signal is in digital form it is identical to data of other forms. Thus a communi­cations network can be used to transmit both speech and data with no need to distinguish between them except in the decoding. Also, with regard to transmission of voice signals requiring security, the digital representation has a distinct advantage over analog systems. For secrecy, the information bits can be scrambled in a manner which can ultimately be unscrambled at the receiver. For these and numerous other reasons digital techniques are being increasingly applied in speech communication problems.

**4.2.1 Speech Processing**

In considering the application of digital signal processing techniques to speech communication problems, it is helpful to focus on three main topics: the representation of speech signals in digital form, the implementation of sophisti­cated processing techniques, and the classes of applications which rely heavily on digital processing.

The representation of speech signals in digital form is, of course, of fun­damental concern. In this regard we are guided by the well-known sampling theorem which states that a band limited signal can be represented by sam­ples taken periodically in time - provided that the samples are taken at a high enough rate. Thus, the process of sampling underlies all of the theory and application of digital speech processing. There are many possibilities for discrete representations of speech signals, these representations can be classified into two broad groups, namely waveform representations and parametric representations. Waveform representations, as the name implies, are concerned with simply preserving the "wave shape" of the analog speech signal through a sampling and quantization process. Parametric representations, on the other hand, are concerned with representing the speech signal as the output of a model for speech production. The first step in obtain­ing a parametric representation is often a digital waveform representation; that is, the speech signal is sampled and quantized and then further processed to obtain the parameters of the model for speech production. The parameters of this model are conveniently classified as either excitation parameters (i.e., related to the source of speech sounds) or vocal tract response parameters (i.e., related to the individual speech sounds).

**4.3 Digital Transmission and Storage of Speech**

One of the earliest and most important applications of speech processing was the vocoder or voice coder, invented by Homer Dudley in the 1930's. The purpose of the vocoder was to reduce the bandwidth required to transmit the speech signal. The need to conserve bandwidth remains, in many situa­tions, in spite of the increased bandwidth provided by satellite, microwave, and optical communications systems. Furthermore, a need has arisen for systems which digitize speech at as Iowa bit rate as possible, consistent with low termi­nal cost for future applications in the all-digital telephone plant. Also, the pos­sibility of extremely sophisticated encryption of the speech signal is sufficient motivation for the use of digital transmission in many applications.

**4.3.1 Speech synthesis systems**

Much of the interest in speech synthesis systems is stimulated by the need for economical digital storage of speech for computer voice response sys­tems. A computer voice response system is basically an all-digital, automatic information service which can be queried by a person from a key­board or terminal, and which responds with the desired information by voice. Since an ordinary Touch-Tone® telephone can be the keyboard for such a sys­tem, the capabilities of such automatic information services can be made universally available over the switched telephone facilities without the need for any additional specialized equipment. Speech synthesis systems also play a fundamental role in learning about the process of human speech production.

**4.3.2 Speaker verification and identification systems**

The techniques of speaker verification and identification involve the authentication or identification of a speaker from a large ensemble of possible speakers. A speaker verification system must decide if a speaker is the person he claims to be. Such a system is potentially applicable to situations requiring control of access to information or restricted areas and to various kinds of automated credit transactions. A speaker identification system must decide which speaker among an ensemble of speakers produced a given speech utter­ance. Such systems have potential forensic applications.

**4.3.3 Speech recognition systems**

Speech recognition is, in its most general form, a conversion from an acoustic waveform to a written equivalent of the message information. The nature of the speech recognition problem is heavily dependent upon the con­straints placed on speaker, speaking situation and message context. The potential applications of speech recognition systems are many and varied; e.g. a voice operated typewriter and voice communication with computers. Also, a speech recognizing system combined with a speech synthesizing system comprises the ultimate low bit rate communication system.

**4.3.4 Aids-to-the-handicapped**

This application concerns processing of a speech signal to make the infor­mation available in a form which is better matched to a handicapped person than is normally available. For example variable rate playback of prerecorded tapes provides an opportunity for a blind "reader" to proceed at any desired pace through given speech material. Also a variety of signal processing techniques have been applied to design sensory aids and visual displays of speech informa­tion as aids in teaching deaf persons to speak.

**4.3.5 Enhancement of signal quality**

In many situations, speech signals are degraded in ways that limit their effectiveness for communication. In such cases digital signal processing tech­niques can be applied to improve the speech quality. Examples include such applications as the removal of reverberation (or echoes) from speech, or the removal of noise from speech, or the restoration of speech recorded in a helium-oxygen mixture as used by drivers.

**4.4 Operational Considerations for Limited Vocabulary Applications**

**4.4.1 Noise background**

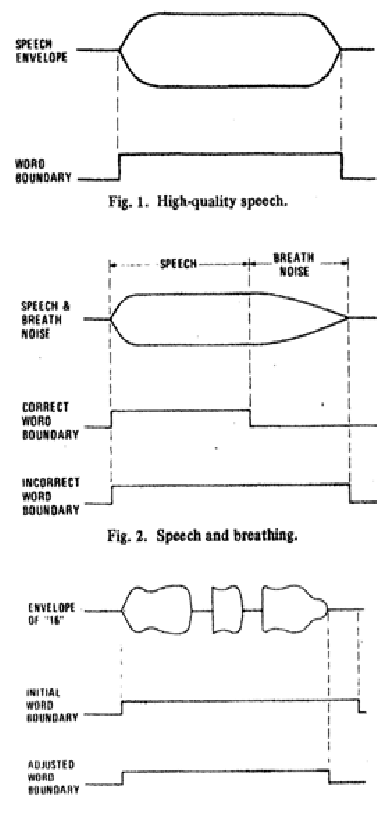
It is important to consider some of the physical aspects of limited vocabulary speech recognitions systems in intended for operational use. One of the first to be considered is interfacing acoustical signals and noise background. If a system is to be used for high quality wide range microphones, it will naturally pick up other sounds from within the immediate vicinity of individual attempting to use the speech recognition system. There are two such solutions to this problem; the first solution is to remove the interfering sound by placing the individuals in an acoustically shielded environment. Should this be possible, noise background can be reduced generally to the point where they are non interfering. However, the restrictions resulting from an acoustic enclosure are such that the mobility of individuals is reduced and can possibly eliminate any ability to perform any other functions. Many applications which are economically justifiable for speech recognition systems involve an individual who will do more than one function at a time. The purpose of speech recognition system is to add to his capabilities or remove some of the overload on his manual or visual operations. Usually this type of an individual cannot be placed in restrictive enclosure.

The second method of removing interfering sound is to eliminate the noise at the microphone itself. Close talking noise-canceling microphones and contact microphones will both achieve a degree of noise cancellation. The contact microphone, however, does not pick up many of the attributes of unvoiced frictional sounds. It is, therefore, a device which can be used only with a very limited capability speech recognizer. The contact microphone can also produce erroneous signals that are result of body movement. Therefore, a close talking noise-cancelling microphone worn on a lightweight headband or mounted in a handset is the optimum compromise between obtaining high-quality speech and reducing noise background.

**4.4.2 Breath noise**

Once it is determined that close talking noise cancelling microphone is to be used for a speech recognizer, a very critical factor must be considered in the system. This factor relates to extraneous signals caused by breath noise. A highly trained speech researcher working in a laboratory will be able to pronounce distinct utterances to an automatic speech recognizer. Unconsciously he will control his breathing such that when he is producing the speech signal it is crisp and well pronounced. He can be lulled into a sense of false achievement until the first time an untrained speaker, having little or no interest in his work, speaks into the system with very poor results. A similar result will occur for an individual who is doing no physical movement whatsoever. This individual can achieve very high recognition accuracies on a particular system. However, once he begins to move around and perform other functions, recognition can deteriorate. The most likely cause for lower recognition accuracy in both cases is breathing noise. A strong tendency exist to exhale at the end of isolated words and to inhale at the beginning. Inhaling produces no significant direct air blast on the close-talking microphone, where as exhaling can produce signal levels in a microphone comparable to speech levels. In a limited vocabulary, isolated word recognition system, the breath noise can be a serious problem.

**4.4.3 Word Boundary Detection**

It has already been mentioned in the discussion on breath noise that variable back-up duration from an initially derived word boundary signal. If a variable back up is not used, a fixed duration back up can be of some value. An initial word boundary signal can be derived from a combination of amplitude of the speech signal overall or amplitude within predetermined spectral bands. This word boundary signal must not, however, be responsive to brief intervocalic pauses caused by the stop consonants and affricatives. Fig.3 illustrates this point for the word “sixteen.” The initial word boundary extends beyond the end of the word by an amount somewhat greater than the duration of the short pauses from the Internal stop consonants. In this case an adjustment to the actual word boundary can be made by a fixed duration back up. The fixed duration back up will more accurately locate the end of the word, although the best results are obtained with variable back up.

**Figure 4.1: Internal stop consonant**

**4.4.4 Operator-Originated Babble.**

It is inevitable that an operator using an ASR system will wish to communicate with his supervisor or other individuals within his area. Regardless of the ease with which an ON/OFF switch can be utilized by an operator, he will occasionally forget to turn the microphone off and will begin to carry on a conversation with another individual. Since the operator will rarely use the words that are in the limited vocabulary the speech recognition system should generally reject the ordinary conversation. It is important in practical applications that a reject capability exists so that inadvertent conversations, sneezes, coughs, throat clearings, etc., do not produce spurious recognition decisions. Both audible and visual alerts can be supplied to the operator indicating that he is talking into a live microphone. This will minimize the number of inadvertent entries that are made into a speech recognition system. Another safeguard to prevent inadvertent message entry to the speech recognition system is to format the data entry sequence as much as possible so that after a block of data has been entered, a verification word is required before the entry is considered to be valid by the speech recognition system.

**4.5 Recording the Voice.**

Before you can do any recording through Record, you will need to connect a microphone . or other sound source to the microphone input on your sound card. The next step is to ensure that your computer is set up to record from the microphone. On a Windows machine, you must select the microphone as the source in the Record Control window (see illustration below). The Record Control window can usually be accessed from a speaker icon in the system tray.



**Figure 4.2: Recording Control**

*Note: Microphone selected as the recording source.*

Windows *Sound Recorder* program can be used to verify that the microphone is configured correctly. If sounds can be recorded using this program, they can also be recorded in MATLAB. If you *can’t* record sounds, there is some problem with the configuration.

****

**Figure 4.3: Windows Sound Recorder in action**

The remainder of this manual will describe the MATLAB Recordprogram—its inner working and functionality.

**4.5.1 Running the program**

The program can be run by typing record at the Matlab prompt or by opening the program in the MATLAB editor and selecting **Run** from the **Debug** menu

**4.5.2 Recording**

Sound recording is initiated through the MATLAB graphical user interface (GUI) by clicking on the record button. The duration of the recording can be adjusted to be anywhere from 1 to 6 seconds. (These are the GUI defaults, but the code can be modified to record for longer durations if desired). Most of the important information in a typical voice waveform is found below a frequency of about 4 kHz. Accordingly, we should sample at a least twice this frequency, or 8 kHz. (Note that all sound cards have a built in pre-filter to limit the effects of aliasing.)Since, there is at least some valuable information above 4 kHz, the Record GUI. Has a default sampling rate of 16 kHz (however, the waveforms portrayed in this document were sampled at 11.025 kHz), Once recorded, the time data is normalized to maximum amplitude of 0.99 and displayed on the upper plot in the GUI window. In addition to the time domain waveform, a spectrogram is computed using MATLAB’S built in spec gram function (part of the signal processing toolbox).

In Figure 4.4 shows an example recording of the sentence, “We were away a year ago” is shown below.



**Figure 4.4: Recording of “We were away a year ago”**

One can examine a region of interest in the waveform using the *Zoom in* button. When Zoom in is clicked, the cursor will change to a cross hair. Clicking the left mouse button and dragging a rectangle around the region of interest in the time domain waveform will select a sub-section of data. In the example below we have zoomed in on the region from about 1 to 1.2 seconds.



**Figure 4.5: ‘Zoomed in’ on the waveform**

As shown in Figure 4.5 the Zoom out button will change the axis back to what it was before Zoom in was used. If you zoom in multiple times, zooming out will return you to the previous axis limits.

**4.5.3 Listening to the Waveform**

The *Play* button uses MATLAB’s sound function to play back (send to the speakers) the waveform that appears in the GUI. If you have zoomed in on a particular section of the waveform, only that portion of the waveform will be sent to the speakers.

***Saving and Loading Data*:**

Save is used to write the waveform to a wave file. If zoomed in on segment of data, only that portion of the waveform will be saved.

Click Load to import any mono wave file into the Record GUI for analysis.

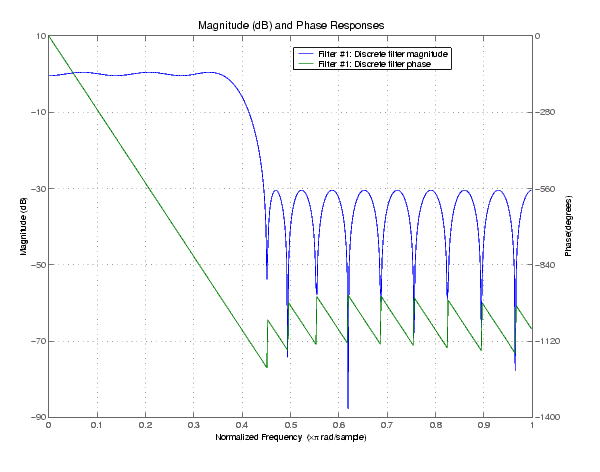
**4.6 Filtering Process**

The primary purpose of digital filtering is to alter the spectral information contained an input signal xk, thus producing an enhanced output signal yk. While this can be accomplished in either the time or frequency domain, much of the early work of signal processing was done in the analog or continues, time domain. While the ultimate goals of digital and analog filtering are the same, the practical aspects vary greatly. In analog filtering we are concerned with active component count and size, termination impendence matching, and lossy reactive elements; but in digital filtering we must consider work length, rounding errors, and in some cases processing delays.

Digital filtering can be performed either off-line using a general purpose computer or in real time via dedicated hardware. Although numerical precision determined by available digital word length must be considered in either instance, precision is typically less of a problem with general purpose computers. For cases where digital processing accuracy is restricted by fixed point, or integer arithmetic, special techniques have been developed for filter design.

**4.6.1 Types of filter**

To facilitate discussion of the various type of filter, three basic terms must first be defined. These terms are illustrated pictorially in the context of the normalized low-pass filter as in figure 4.1. In general, the filter passband is defined as the frequency range over which the spectral power of the input signal is passed to the filter output with approximately unity gain. The input spectral power that lies within the filter stopband is attenuated to a level that effectively eliminates it forms the output signal. The transmission band is the range of the frequencies between the passband and the stopband. In this region, the filter magnitude response typically makes a smooth transition from the passband gain level to that of the stopband as shown in Figure 4.6

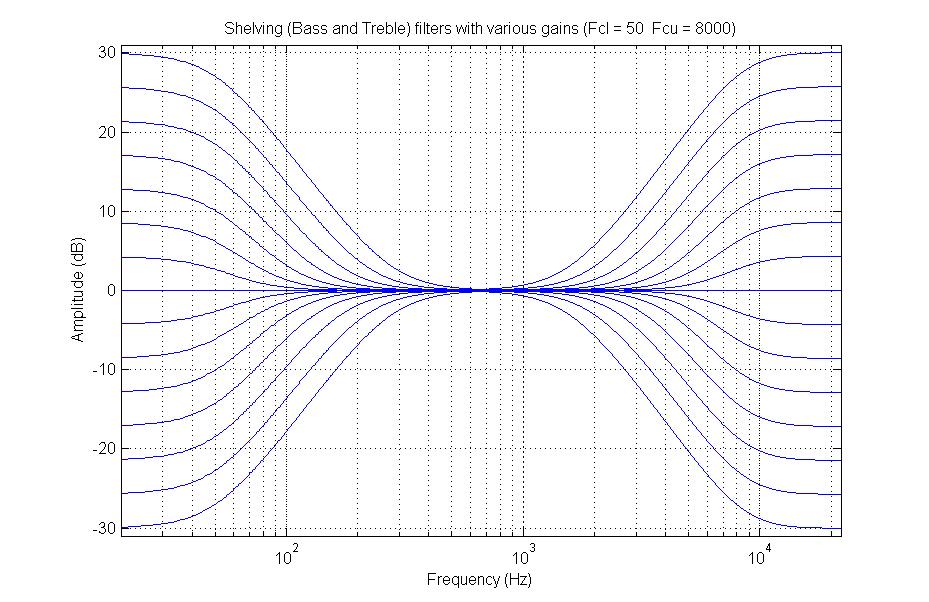


**Figure 4.6: Magnitude of response of normalized low pass filter.**

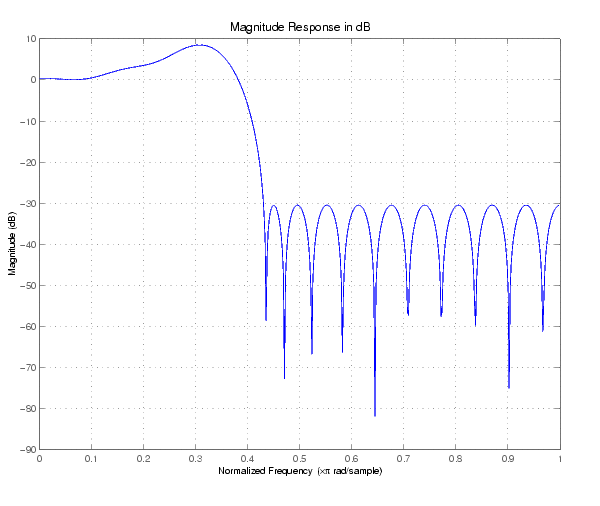
**4.6.2 FIR digital filter**

FIR: It means Finite Impulse Response Filter

We know that consider digital filters whose impulse response is of finite of duration, so these filter are appropriately referred to as finite impulse response **(FIR)** digitals filters. So if the output samples of the system depend only on the present input, and a finite number of past input samples, then the filter has a finite impulse response as shown in Figure 4.7,8.



**Figure 4.7 : Relation between frequency and the amplitude**



**Figure 4.8: FIR Digital Filters**

**4.6.3 Characteristics of FIR digital filters**

Some advantage and disadvantage of **FIR** filters compared to their **IIR** counterparts are as follow:

1. **FIR** filters can be designed with exactly linear phase. Linear phase important for applications where phase distortion due to nonlinear phase can degrade performance, for example, speech processing, data transmission and correlation processing
2. **FIR** filters realized no recursively are inherently stable, that is, the filter impulse response is of finite length and there for bonded.
3. Quantization noise due to finite precision arithmetic can be made negligible for no recursive realizations.
4. Coefficient accuracy problems inherent in sharp cutoff **IIR** filters can be made less severe for realizations of equally sharp **FIR** filter
5. **FIR** filters can be efficiently implemented in multi rate systems.

A disadvantage of FIR filters compared to IIR filters is that an appreciably higher order filter is required to achieve a specified magnitude response, thereby requiring more filter coefficient storage.

**4.6.4 Butterworth digital Filters**

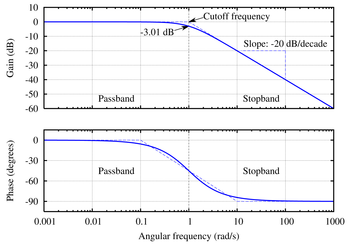
In this project the filter was used the Butterworth filter, The Butterworth filter attempts to be linear and pass the input as close to unity as possible in the **pass band**.

-Eq-4.1

 The **high pass** Butterworth equation is as follows:

-Eq-4.2

The Butterworth filter is one type of [electronic filter](http://en.wikipedia.org/wiki/Electronic_filter) design. It is designed to have a [frequency response](http://en.wikipedia.org/wiki/Frequency_response) which is as flat as mathematically possible in the [passband](http://en.wikipedia.org/wiki/Passband). Another name for them is 'maximally flat magnitude' filters.[3]

[](http://en.wikipedia.org/wiki/File:Butterworth_filter_bode_plot.png)

**Figure 4.9: Shape of Butterworth filter.**

**4.6.5 Overview**

The frequency response of the Butterworth filter is maximally flat (has no [ripples](http://en.wikipedia.org/wiki/Ripple_(filters))) in the passband, and rolls off towards zero in the stopband. When viewed on a logarithmic [Bode plot](http://en.wikipedia.org/wiki/Bode_plot), the response slopes off linearly towards negative infinity. For a first-order filter, the response rolls off at −6 [dB](http://en.wikipedia.org/wiki/Decibel) per [octave](http://en.wikipedia.org/wiki/Octave) (−20 dB per [decade](http://en.wikipedia.org/wiki/Decade)) (all first-order filters, regardless of name, have the same normalized frequency response). For a second-order Butterworth filter, the response decreases at −12 dB per octave, a third-order at −18 dB, and so on. Butterworth filters have a monotonically changing magnitude function with ω. The Butterworth is the only filter that maintains this same shape for higher orders (but with a steeper decline in the stopband) whereas other varieties of filters ([Bessel](http://en.wikipedia.org/wiki/Bessel_filter), [Chebyshev](http://en.wikipedia.org/wiki/Chebyshev_filter), [elliptic](http://en.wikipedia.org/wiki/Elliptic_filter)) have different shapes at higher orders.

Compared with a [Chebyshev](http://en.wikipedia.org/wiki/Chebyshev_filter) Type I/Type II filter or an [elliptic filter](http://en.wikipedia.org/wiki/Elliptic_filter), the Butterworth filter has a slower roll-off, and thus will require a higher order to implement a particular [stopband](http://en.wikipedia.org/wiki/Stopband) specification. However, Butterworth filter will have a more linear phase response in the passband than the Chebyshev Type I/Type II and elliptic filters **(a)**

A simple example of a Butterworth filter is the 3rd order [low-pass](http://en.wikipedia.org/wiki/Low_pass_filter) design shown in the figure on the right, with *C*2 = 4 / 3 farad, *R*4 = 1 ohm, *L*1 = 3 / 2 and *L*3 = 1 / 2 Henry. Taking the [impedance](http://en.wikipedia.org/wiki/Electrical_impedance) of the capacitors *C* to be *1/Cs* and the impedance of the inductors *L* to be *Ls*, where (*s* = σ + *j*ω) is the complex frequency, the circuit equations yields the [transfer function](http://en.wikipedia.org/wiki/Transfer_function) for this device:

H(s)=\frac{V_o(s)}{V_i(s)}=\frac{1}{1+2s+2s^2+s^3} Eq-4.3

In this project the filter was used the Butterworth filter, so the transfer functions for this filter was:

* Fs=44100;
* Wp = [150 8450]/11025 ; Ws = [100 9450]/11025; Rp = 0.8; Rs = 30.8;
* Transfer function:
* Sampling time: 4.5351e-005.
  + 1. **Filter part assumptions**
* **Low Pass Filter**

The filter selected is a unity gain Sullen-Key filter, with a Butterworth response characteristic. Numerous articles and books describe this topology.

* **High Pass Filter**

The filter selected is a unity gain Sullen-Key filter, with a Butterworth response characteristic. Numerous articles and books describe this topology.

* **Wide Band Pass Filter**

This is nothing more than cascaded Sullen-Key high pass and low pass filters. The high pass comes first, so energy from it that stretches to infinite frequency will be low passed.

* **Notch Filter**

This is the Liege Filter topology, set to a Q of 10. The Q can be adjusted independently from the center frequency by changing R1 and R2. Q is related to the center frequency set resistor by the following:

R1 = R2 = 2 \*Q\*R3 Eq-4.4

The liege filter topology has a fixed gain of 1.

The only real possibility of a problem is the common mode range of the bottom amplifier in the single supply case.

* **Band Reject Filter**

This is nothing more than summed Sullen-Key high pass and low pass filters. They cannot be Cascaded, because their responses do not overlap as in the wide band pass filter case.

**4.6.7 Filter Circuit Design**

1. Lowbass filter : As shown in the following figure electrical circuit for the lowbass filter .



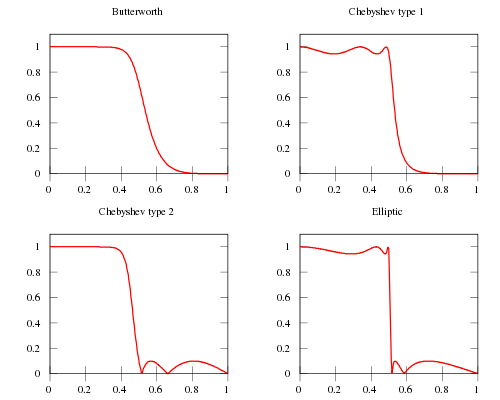
**Figure 4.10 : Low Pass Filter for Supplies**



**Figure 4.11: Low Pass Filter for a Single Supply.**

**4.6.8 Comparison with other linear filters**

Figure 4.12 shows the gain of a discrete-time Butterworth filter next to other common filter types. All of these filters are fifth-order.

[](http://en.wikipedia.org/wiki/File:Electronic_linear_filters.svg)

**Figure 4.12: Comparison for some types filters**

**4.7 Spectral Analysis**

**4.7.1 Fast Fourier transform (FFT)**

Spectral analysis applications often require Dfts in realtime on contiguous stets of input samples. Computation of the DFT: Discrete Fourier Transform of discrete-time signal X (n) is defined as:

Eq-4.5

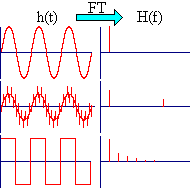
For N input sample points requires N2 complex multiplies and N2-N complex additions For N frequency output points. This assumes that all twiddle factor coefficients require complex multiplications, even those that real or imaginary parts equal to 1 or 0 The FFT is the fast algorithm for efficient implementations of the DFT where the number of the time samples of the input signal N transformed in to N frequency points. The computational requirements of the FFT are expressed as:

FFT CMPS=Eq-4.6

FFT CAPS=Eq-4.7

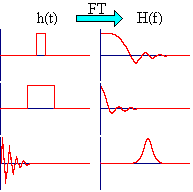
**4.7.2 Fourier Analysis and Signal Filtering**

Non-sinusoidal periodic signals are made up of many discrete sinusoidal frequency components (see applet Fourier Synthesis of Periodic Waveforms). The process of obtaining the spectrum of frequencies H(f) comprising a time-dependent signal h(t) is called Fourier Analysis and it is realized by the so-called Fourier Transform (FT). Typical examples of frequency spectra of some simple periodic signals composed of finite or infinite number of discrete sinusoidal components are shown in the figure below **(b)**



**Figure 4.13: spectral analysis tarnsform**

 However, most electronic signals are not periodic and also have a finite duration. A single square pulse or exponentially decaying sinusoidal signals are typical examples of non-periodic signals, of finite duration. Even these signals are composed of sinusoidal components but not discrete in nature, i.e. the corresponding H(f) is a continuous function of frequency rather than a series of discrete sinusoidal components, as shown in the figure below.



**Figure 4.14: Spectral analysis tarnsform.**

 Note :H(f) can be derived from h(t) by employing the Fourier Integral.

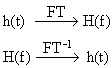
http://www.chem.uoa.gr/Applets/appletfouranal/images/Text_F3.gifEq-4.8

This conversion is known as (forward) Fourier Transform (FT). The inverse Fourier Transform (FT-1) can also be carried out. The relevant expression is:

http://www.chem.uoa.gr/Applets/appletfouranal/images/Text_F4.gif Eq-4.9

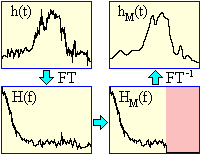
These conversions (for discretely sampled data) are normally done on a digital computer and involve a great number of complex multiplications (N2, for N data points). Special fast algorithms have been developed for accelerating the overall calculation, the most famous of them being the Cooley-Turkey algorithm, known as Fast Fourier Transform (FFT). With FFT the number of complex multiplications is reduced to Nlog2N. The difference between Nlog2N and N2 is immense, i.e. with N=106, it is the difference between 0.1 s and 1.4 hours of CPU time for a 300 MHz processor.

All FT algorithms manipulate and convert data in both directions, i.e. H (f) can be calculated from h(t) and vice versa, or schematically:



**4.7.3 Signal Smoothing Using Fourier Transforms**

Selected parts of the frequency spectrum H (f) can easily be subjected to piecewise mathematical manipulations (attenuated or completely removed). These manipulations result into a modified or "filtered" spectrum HΜ (f). By applying FT-1 to HΜ (f) the modified signal or "filtered" signal hΜ(t) can be obtained. Therefore, signal smoothing can be easily performed with removing completely the frequency components from a certain frequency and up, while the useful (information bearing) low frequency components are retained. The process is depicted schematically below (the pink area represents the range of removed frequencies) as shown in Figure (4.15) .



**Figure 4.15: The shape of Fourier and inverse Fourier transforms**

**4.7.4 Spectral analysis applications**

 The detections of discrete frequency components embedded in broadband spectral noise is encountered in many signal processing applications. The time-domain representations of the composite signal are the summations of individual noise and discrete frequency components. The signal is reprehensive of ocean acoustic signals, which consist of many noise sources and radiated spectral energy from surface ships and submarines, which are generally masked by the higher noise signal components levels. We will perform a DSP system design to determine the processing required for this important application. This approach will be used to implement the discrete frequency (narrowband) spectral analysis detections.

**4.7.5 Spectral processing system requirement**

A stationary input signal consisting of broadband Gaussian noise discrete frequency components that exist between 0 to 10,000 Hz is to be processed in order to detect the discrete frequency components that exist between 62.5 Hz and 1000 Hz.

The interface design and system level requirements specifications are started in step of the design. The specifications are essential to developing a system that meets all signals and no signal processing requirements. Since the signal processor has been specified, and then the principles of operation and the unit level specifications form a basic for the design. Otherwise they would be developed as a part of design process, the interface of each these process system to the processor must be completely described.

* Number of data bits.
* Number of control bits.
* Control protocol.
* Maximum data rate.
* Electrical characteristics.
* Connecter requirements.

**4.7.6 Hidden MARKOV Modeling**

The Hidden Markov Modeling algorithm is a very involved process. This following information represents my most basic understanding of the procedure. In the coming weeks I hope to fully understand every aspect of the process. Hidden Markov Processes are part of a larger group known as statistical models; models in which one tries to Characterize the statistical properties of the signal with the underlying assumption that a Signal can be characterized as a random parametric signal of which the parameters can Be estimated in a precise well defined manner. In order to implement an isolated word Recognition system using HMM the following steps must be taken:

1. For each reference word, a MARKOV model must be built using Parameters that optimize the observations of the word.
2. A calculation of model likelihoods for all possible reference models against the unknown model must be completed using the Viterbi algorithm followed by the selection of the reference with the highest model likelihood value. I too have a very basic understanding of the Viterbi algorithm. In the coming weeks I Wish to gain a better understanding of this process as well. With the Viterbi algorithm, we take a particular HMM, and determine from an observation sequence the most likely sequence of underlying hidden states that might have generated it **(c)**. For example be examining the observation sequence of the s1\_1 test HMM one would determine that thes1 train HMM is most likely the voiceprint that created it, thus returning the highest Likelihood value.

CHAPTER FIVE

CONTROLLING ROBOTIC CAR

The connection of speech control is one of the important hardware components of this project, it’s used for connection between MATLAB and the robotic car , there are many types of connection in engineering field such as , radio frequency (RF) and ultrasonic waves , in this project (RF) connection links are used to perform the data transfer from the computer’s port (LPT1) to the robotic car .

Wireless connection involved in many applications that we handle in our real life, in homes devices, planets equipments and even in communications means.

This chapter will discuss the transmitter-receiver circuits for each link and their advantages and disadvantages robotic car, parallel port characteristics in interfacing hardware by the computer is presented also and discussed too.

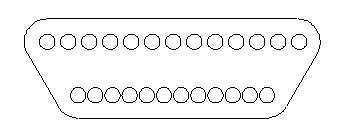
# 5.1 Parallel port (LPT1)

In the CH 6 used MATLAB for analysis voice commands, and print an output on parallel port , the question is what is parallel port ? And what are the differences between parallel and series ports ? and how parallel port will be controlled ?

The parallel port is extensively used in computers, usually to communicate with close devices , this is due to the need of using many cables , at least 9 for software handshaking , The main advantage in using parallel port is that its higher speed about 10 times than serial port , because the data in parallel is send at the same time , in the parallel port the speed of transfer is several hundred kilobytes per second.

The most common connection using the parallel port is done with printer , as long as the printer does not generate data , its only accept them , the parallel port was made unidirectional , only for output . However there was no real need for that in the hardware, today parallel ports are made bidirectional because some devices can take advantage of with the computer communicating.

As shown in Figure 5.1 the 25 pins of the parallel port from 1 into 25.



25

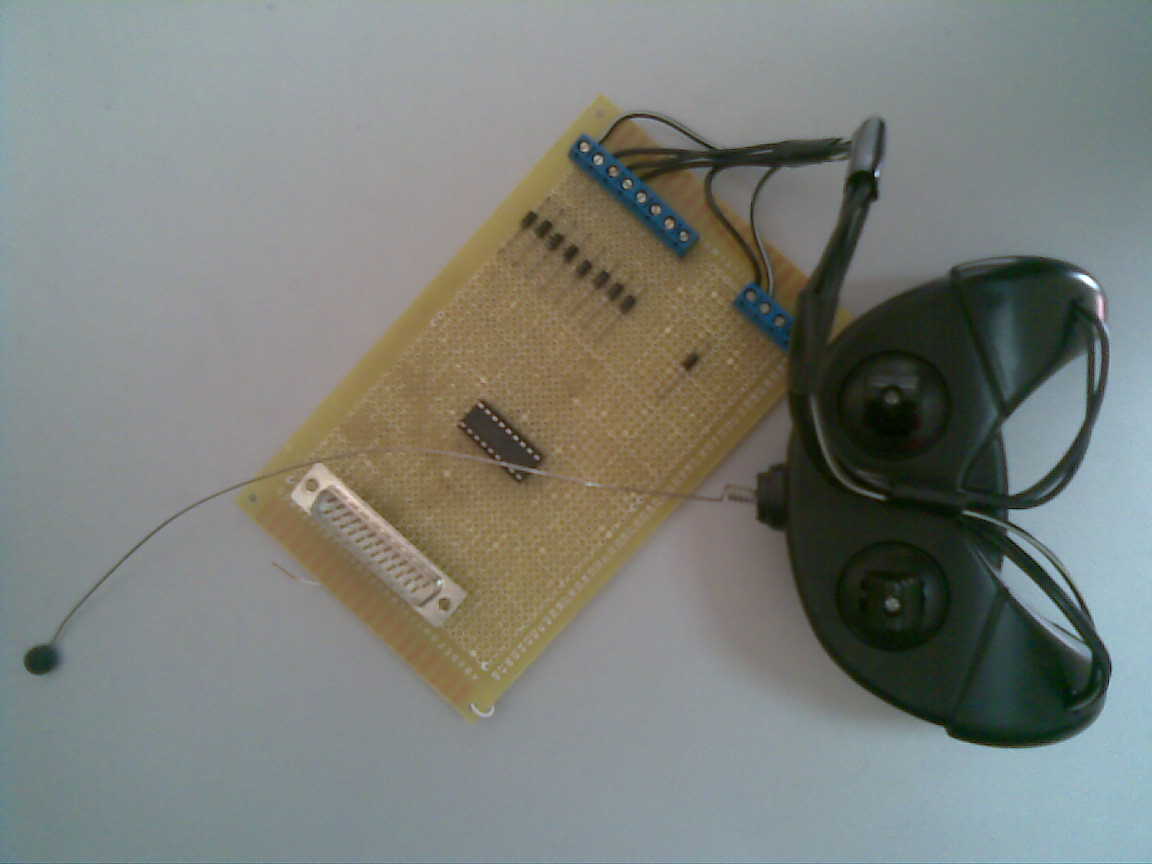
1

**Figure 5.1 The parallel port Pins numbers .**

**Table 2 : Pin layout of the parallel port connection .**

|  |  |
| --- | --- |
| Pin # | Description |
| 1 | Status |
| 2-9 | Data input/output |
| 10 | Input, interrupt generator |
| 11-13 | Input |
| 14 | In/Out put |
| 15 | Input |
| 16-17 | In/out put |
| 18-25 | GND |

The communication is physically controlled by an integrated CCT which works with TTL, levels; therefore it uses 0 volts as logic 0 and 9 volts as logic 1.



**Figure 5.2 Board connected to Parallel Port**

# 5.2 Radio frequency

Radio wave energy and microwave energy are both dielectric heating technologies, where high-frequency electromagnetic radiation generates heat to dry moisture in nonmetallic materials, radio frequency (RF),waves are longer than microwaves , enabling them to penetrate larger objects better than microwave energy .

## 5.2.1 Radio frequency advantage

There are several advantages of the radio wave radiation, such as:

* Not line of sight.
* Not blocked by common materials (can penetrate most solids and pass through walls) .
* Longer ranges.
* Not light sensitive.
* Not a sensitive to environmental conditions such as rain, moisture etc..

## 5.2.2 Radio frequency disadvantages

There several disadvantages for radio wave radiation:

1. Interference: communication devices using similar frequencies – wireless phones, wrist radios, and personal locators can interfere with transmission.
2. Lack of security: easier to eavesdrop on transmission since signals are spared out in space rather than confined to a wire.
3. Higher cost than infrared.
4. Lower speed: Data rate transmission is lower than infrared transmission.

## 5.2.3 Application of radio wave

Signals with radio wave frequency (RF) are used in radio, television , wireless security system , car alarms , remote gate controls , remote sensing and data communication .

## 6.2.4 Radio frequency transmitter and receiver circuits

This subsection will discuss the components of the transmitter and receiver circuits.

## 6.2.4.1 Component RF receiver-transmitter

1- Transmitter module

2- Receiver module

3- SM6136B encoder IC

4- SM1647C Decoder IC

5- Antenna

**1 - Transmitter module**

The module operates on 433.92 MHZ , and offer about 150 meter range in line of sight operating form 2-12 volts , the transmitter module has four pins as following : (DATA,GROUND,POWER,ANTENA) respectively ,look for the following Figure 6.2.

3

2

1

4

**Fig 5.3 RF Transmitter Module**

**2 – Receiver module TX433N**

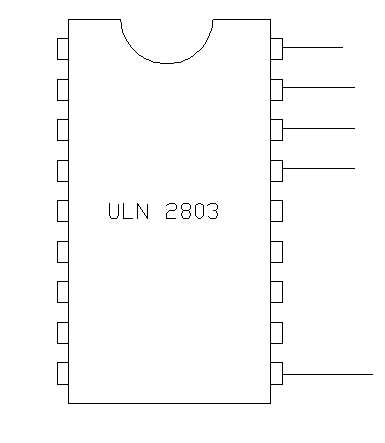
TX433N the receiver works fro 5 to 6.2 volts-DC with both linear and digital output , the typical current consumption is 3.5 mA for 5 V operation voltage , the receiver module has eight pins that given in table 3 .

**Table 3: Receiver pin description**

|  |  |
| --- | --- |
| Description | PIN # |
| Ground | 1,8,6 |
| Power Supply | 4,5 |
| Data | 2,3 |
| Antenna | 7 |

**3- SM6136B Encoder IC**

The encoder are a series of CMOS for remote control system applications, they are capable of encoding information , SM6136B (SM refers to the company that manufacture this IC ) , this IC consists of 8-N address bits and 4-N data bits , each address/data input can be set to one of the two logic states , these pins must either unconnected to 0V as required pin 9 should be connected to GND , and pin 18 is connected to Vdd , Look for the figure 5.4.



SM6136B

18

10

9

1

**Fig 5.4:SM6136B Encoder IC**

**4- SM1647C Decoder IC**

The Decoder IC are a series of CMOS for remote control system applications , for proper operation , a pair of encoder and decoder with the same number of addresses and data format should be chosen , it’s the same distribution of pins to the encoder look for figure 5.4 .

**5- Antenna**

Antenna is just a length of wire its function being to convert an electromagnetic energy into voltage for reception, or to transducer a varying voltage into electromagnetic energy for transmission, an electromagnetic wave consists of changing magnetic and electric fields and travels through space at 300\*106meter/sec, the wavelength of an electromagnetic wave or radio wave can be determined from the following equation

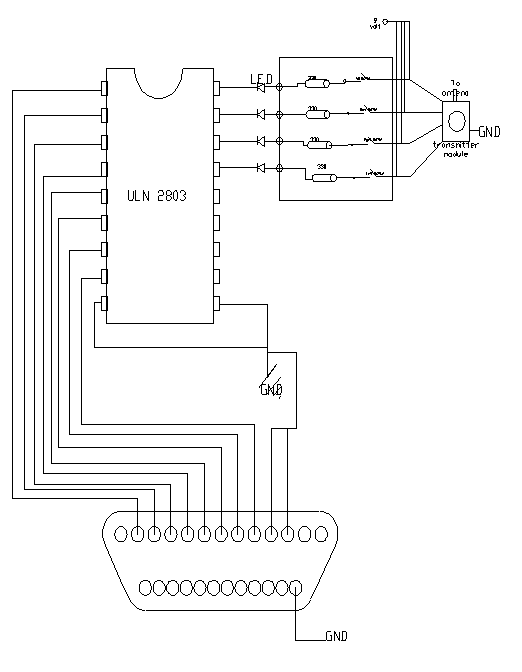
Eq-5.1

Where:

* F= Frequency (MHZ) .
* Wavelength (meters) .

Antenna characteristics differ with the design and operating frequency and there many factors that will affect the performance of the antenna.

The following figure for the parallel port connection and transmitter of RF.



**Figure 5.5 “CCT of parallel port connected to the transmitter**

# 5.3 Robotic Car

In the last decades engineers tries to make and develop machines that did work instead of human, these machines are called robots , these robot can use in industrial works , and medical work also in surgical operations recently they found the robot arm can make the surgeon with more accuracy than human arm , this will push the engineer to develop their robots to be used in this field and other fields .

In these days ,many forms are developed from robots upon the application needed until mobile robotic car born , mobile robots is a new technology that enter this field , and spread in the world widely , in last few years , many mobile robots built and developed in both mechanical and control design , the powerful to the robots comes from that it can reach many points that seem to be impossible for human, scientism and engineers trying to develop mobile robot technology to be smart , cute and lovely machine , maybe in future we can see robots washing dishes ,cooking , cleaning , and might joking and kidding .

## 5.3.1 Robotic car

Robotic car is a robot that has a mobility property, it can GO straight, Turning LEFT or RIGHT or Returning BACK, this robot properties that make it powerful are as following:

1. Easy to install.
2. Robust enough to work in hard conditions.
3. Easy to be programmed.
4. Low cost.

The good thing in the robotic car is that it can reach the points that is seems are impossible and enabling technologies such as path planning localization and obstacle avoidance.

## 5.3.3 Mechanical design of robotic car

The wheeled robot used in this project is a toy car, this car has following outside shape ,presented in Figure 5.6

****

**Figure 5.6 : Outside shape of the car**

The car have four wheels two are connected to a shaft that provides with rotational motion from DC-motor (that will discussed later) ,the rotational motion of the wheel will move the body of the car forward and backward , and other two wheels allows the car to rotate right and left , these two wheels are connected to other DC-motor .

## 5.3.4 DC-motor characteristics

This subsection will discuss the Dc-motor characteristics, and its principle, its type and they way we drive motors and components used for this purpose.

## 5.3.4.1 DC motor principle

In General the motor is the machine that converts the electrical energy into mechanical energy, DC-motor is a motor which operate using a DC power supply source (Battery) , and converts it into mechanical energy .

The motor consists of two main parts, (The rotating part of the machine and it’s called **ROTOR**, and the stationary part that called **STATOR)**, the stationary part of the motor consists of the outer frame of the motor , which provides the physical support and pole piece which provides a main magnetic flux in dc motor , the rotor or called (armature) of the motor consists of steel bar with core built up over hold the armature winding .

The figure bellow shows the principle of working for the DC motor, the magnetic field in the stator generated by north and south poles, and that generate the magnetic flux in the armature by applying a voltage on the commutator , these magnetic field will produce a rotary motion in the rotor , the following figure shows the single loop of armature winding .

To connect the commutator with the power supply voltage we need Brushes ,these brushes are made by carbon or graphite , they have a high conductivity to reduce electrical losses and low coefficient of friction to reduce wears.

## 5.3.4.2 DC motor types

DC motors are classified to five major types, depends on the way that the armature and filed circuit are connected, and its depend on the source that produce the main magnetic flux in stator , these types are as following :

1. Separately excited DC motor.
2. Shunt DC motor.
3. Permanent-magnet DC motor.
4. Series DC motor.

The used motor in this project is the permanent-magnet DC motor, which is a motor whose poles are made of permanent magnets, so there is no need to filed circuit to produce the main flux [9] .

Permanent magnet DC motor offers a number of benefits compared with other types of the DC motor in this application, the main advantage of the Dc motor is the size benefit it’s smaller than other types of Dc motors, and other advantage is the simplicity of the connection.

In other way there are some disadvantages for the Dc-motor, the permanent magnet cannot produce high flux density so this will negatively affect the output torque of the motor but in this application its suitable because no need for the high torque motor .

# CHAPTER SIX

# MATLAB SOFTWARE

# 6.1 Introduction

The Chapter aims to build a simple, complete and representative automatic speech recognition system using MATLAB. MATLAB is a data-manipulation software package that allows data to be analyzed and visualized using existing functions as well as user-designed programs. There are two fundamental phases of the speech recognition process: training and recognition. During the training phase a database of reference “fingerprints” are created in order to be tested against in the recognition phase. A fingerprint represents the most basic, yet unique, features of a particular word. A fingerprint is merely a matrix of numbers in which each number represents the energy or average power that is heard in a particular frequency band during a specific interval.

# 6.2 MATLAB Software

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.

Typical uses include Math and computation Algorithm development Data acquisition Modeling, simulation, and prototyping Data analysis, exploration, and visualization Scientific and engineering graphics Application development, including graphical user interface building MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar no interactive language such as C or FORTRAN. The name MATLAB stands for matrix laboratory.

Today, MATLAB engines incorporate the LAPACK and BLAS libraries, embedding the state of the art in software for matrix computation. MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering, and science. In industry, MATLAB is the tool of choice for high-productivity research, development, and analysis. MATLAB features toolboxes. Very important to most users of MATLAB, toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

The MATLAB system consists of five main parts: Development Environment. This is the set of tools and facilities that help you use MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, a command history, an editor and debugger, and browsers for viewing help, the workspace, files, and the search path.

## 6.2.1 The MATLAB Mathematical Function Library

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix Eigen values, Bessel functions, and fast Fourier transforms.

## 6.2.2 The MATLAB Language

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

## 6.2.3 Graphics

MATLAB has extensive facilities for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It includes high-level functions for two-dimensional and three-dimensional data visualization, image processing, animation, and presentation graphics. It also includes low-level functions that allow you to fully customize the appearance of graphics as well as to build complete graphical user interfaces on your MATLAB applications. The MATLAB Application Program Interface (API). This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

**6.3 Speech Signal Processing Using MATLAB**

MATLAB is a powerful environment to handle signal processing and matrices operations. After recording the speech signal, it will be saved in a string. The first operation is filtering the signal. The second process is normalizing the signal to have all the signals with same amplitude. The last operation is spectral analysis.

## 6.3.1 Recording the signals

In this part five words were recorded in order to be processed to create the fingerprints. The five words are Go, Stop, Back, Left, and Right.

The speech signals were recorded using the MATLAB function ‘wavrecord ( 2\*Fs, Fs, 'double')’, which record the sound in a Windows wav format (.wav). The signals have a sampling frequency (Fs) of 44100 samples per second and double channel (16 bits).

After recording the speech signals, they were saved on the computer using the MATLAB function ‘wavwrite (s, *Fs*, ‘Right’) ‘, and then they were played using the MATLAB command ‘ wavread (' Right’, Fs)’.

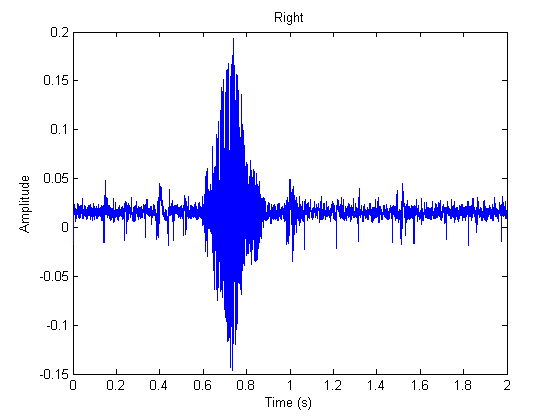
**MATLAB code:**

***Fprintf ('Record Right now')***

***s = wavrecord (2\*Fs, Fs ,'double');***

***wavwrite (s, Fs, 'right')***

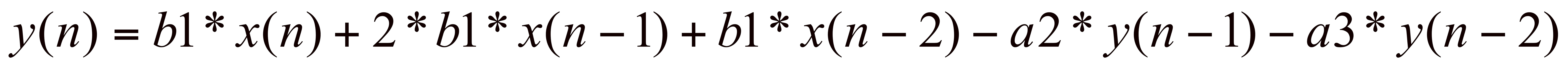
***[s, fs]=wavread ('right');***



## Fig 6.1 Raw signal (Right)

## 6.3.2 Filtering

The Butterworth filter attempts to be linear and pass the input as close to unity as possible in the pass band.

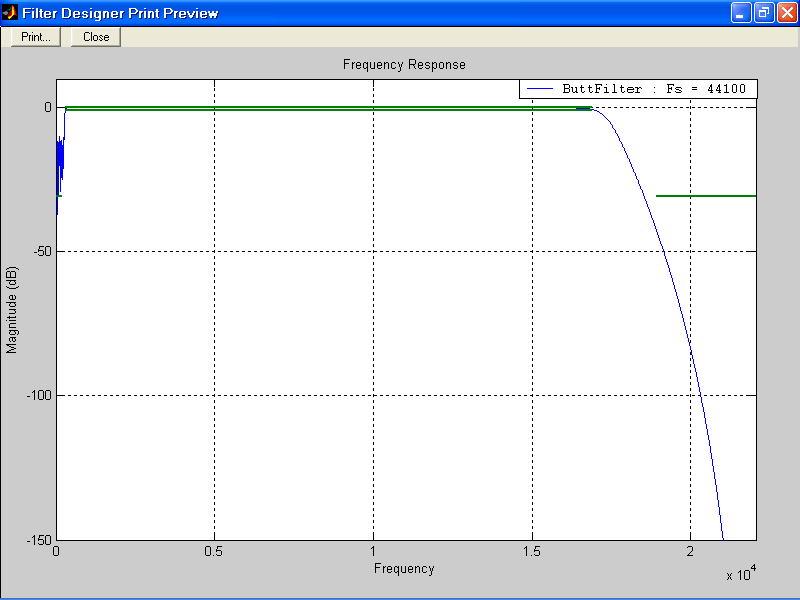
Eq-6.1

The Butterworth filter was of eleventh order. The passing Frequency was from 150 Hz to 8450 Hz and the lower stopping frequency was 100 Hz and the upper one is 9450 Hz. The human voice has typical frequency for the human voice is on the order of 100 Hz to 8000 Hz.

After deciding on the intervals for the digital filter, the MATLAB function *‘[b, a] = butter (n, Wn )’* was written to find the coefficients for this filter.

The transfer function of this Butterworth filter is

Eq-6.2



## Fig. 6.2: Butterworth Filter (Wide Band Pass)

***MATLAB code:***

***Wp = [300 16900]/22050 ;***

***Ws = [200 18900]/22050;***

***Rp = 0.8; Rs = 30.8;***

***[n,Wn] = buttord(Wp,Ws,Rp,Rs);***

***[b,a] = butter(n,Wn);***

***sf=filter(b,a,s);***

## 6.3.3 Normalizing

It means normalization of a reference signal, so that its maximum and minimum values are equal +1 and –1 respectively. Normalizing is important because the resulted signals will have the same amplitude reference. The signals were normalized by dividing the signal values by the maximum absolute value.

***MATLAB code***

***sf = sf / max(abs(s));***



## Figure 6.3: A Processed signal

## 6.3.4 Frequency Domain Analysis (Spectral Analysis)

Signals are converted from time or space domain to the frequency domain usually through the Fourier transform. The Fourier transform converts the signal information to a magnitude and phase component of each frequency. This is done by using the MATLAB function *‘ [B,f] = specgram(sf,Fs,Fs)’*

Often the Fourier transform is converted to the power spectrum, which is the magnitude of each frequency component squared. This is done by multiplying the signal by its conjugate using the MATLAB function *‘*sff=B.\*conj(B)’.

The most common purpose for analysis of signals in the frequency domain is analysis of signal properties. The engineer can study the spectrum to determine which frequencies are present in the input signal and which are missing.

***MATLAB code***

***[B,f] = specgram(sf,Fs,Fs);***

***sff=B.\*conj(B);***

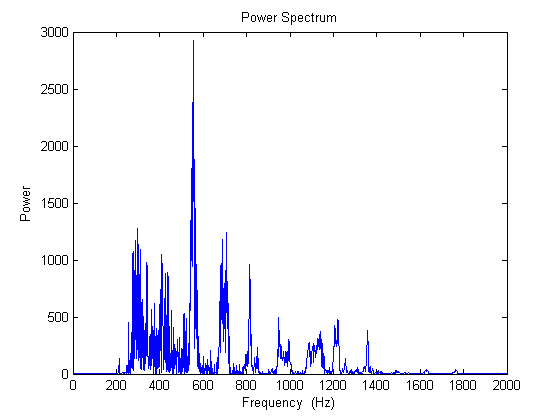


Figure 6.4: Power density spectrum

# 6.4 Fingerprint Comparison

At this point we needed a way to encode the relevant information of the spoken word. The relevant information for each word was encoded in a “fingerprint”. To compare fingerprints we used the Euclidean distance formula between sampled word fingerprint and the stored fingerprints to find correct word.

## 6.4.1 Creating Signals Fingerprints

After converting the signals to frequency domain and then to the power spectrum, the finger print is found by calculating the frequencies that present the input signal. This is done by creating an algorithm that calculated the local peak values for the frequencies, as shown in the next MATLAB code

***MATLAB code***

***for i=101:2500***

***for j=1:100***

***for k=1:100***

***if sff (i-j) < sff (i) & sff (i+k) <sff (i);***

***sff (i) =sff (i);***

***else***

**sff (i) =0;**

**end**

**end**

**end**

**end**

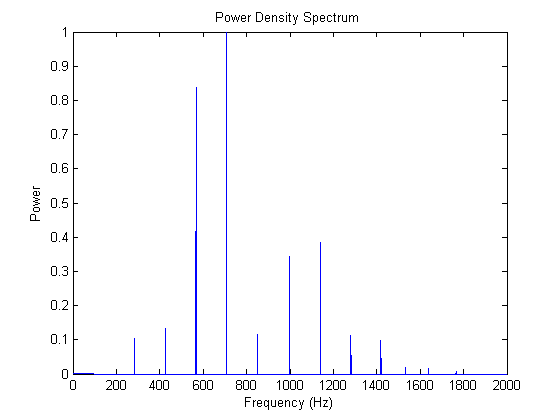
**for i= 1:2500**

**if sff (i) <0.05**

**sff (i) =0;**

**end**

**end**



**Figure 6.5: Power density spectrum (Go’s Fingerprint)**

## 6.4.2 Fingerprint Comparison

Once the fingerprints are created and stored in the dictionary when a word was spoken, it was compared against the dictionary fingerprints. In order to do the comparison, we use Euclidean distance formula by calculating the sum of the absolute value of the difference between each sample finger print a finger print from the dictionary.   The dictionary has multiple words in it and the lookup went through all of them and picked the word with the smallest calculated number.

Euclidean distance formula is:

Eq-6.2

Where:

* Y is the recorded signal, Y = (y1, y2,…, yn )
* Q is the sampled word fingerprint , Q = (q1, q2,…, qn )

***MATLAB Code***

***x1 =norm(y-i1)/3;***

***[s I]=min(x);***

# 6.5 Resultant Recognized Matrix Applications

After MATLAB recognized the intended matrix ‘y’, several operations can be made on it to achieve the main goals of the speech control program.

First of all MATLAB will play the sound command related to the recognized matrix, and then MATLAB will plot the signal in time domain. Another application is printing data via the computer parallel port (LPT1) to control certain hardware connected to the computer.

The following subprogram illustrates the operation of playing, plotting the recognized signal and also printing data through the LPT1 parallel port

***MATLAB Code***

***fprintf('Go\n')***

***wavplay (go,Fs);***

***output=1;***

***plot (t,sf)***

***dio =digitalio('parallel','LPT1');***

***addline (dio,1:3,'out');***

***putvalue ( dio.line(1:4),data);***

## Table 4: Truth table of the speech recognition software LPT1 output

|  |  |  |
| --- | --- | --- |
| **Command** | **Logical output in decimal** | **Logical output in Binary** |
| Go | 1 | 0001 |
| Stop | 0 | 0000 |
| Back | 2 | 0010 |
| Left | 5 | 0101 |
| Right | 9 | 1001 |

# 6.6 Conclusion

MATLAB can easily and effectively be used to construct and test a speech recognition System. With its built in routines, the many procedures which make up a particular Speech recognition algorithm is easily mimicked. A waveform can be sampled, in the time domain, into MATLAB using the wavread command. After a waveform has been stored in a string , the waveform has to be processed to create a fingerprint. A fingerprint represents the basic but unique characteristics of the sound file in the frequency domain. The fingerprint is merely a vector of numbers where each number represents the magnitude of sound that was heard during a particular. This vector is then stored in a database as a reference. The last step is comparing the signals with the stored fingerprints and prenting the recognized signal through the parallel port (LPT1) to control certain hardware ( car toy in this project).

**Chapter seven**

**Conclusion**

The project has not met our expectations fully, as we initially specified that the system would be able to recognize a sentence as a command. But we are more than happy that it is able to recognize a word as the command by more than 70%-80% of the time, depending on the command. There is a training procedure that needs to be implemented, which is an added feature to increase the accuracy of the program. However, the system can still be used without training but with much lower accuracy.

There are two hardest parts in our speech recognition project. One is for filter design, the other is fingerprints analysis. The shortcoming for filter is its frequency spectrum resolution is coarse and can't tell the difference in its band. So we have to select some distinct words as our codes. FFT is a good candidate for filter design and also for fingerprints analysis,.

Another problem is when a tester spoke the same word, even if there is a tiny difference when he spoke, the fingerprint changed a lot. We didn’t solve this problem until now. But we think if we increase frequency resolutions, maybe it will be helpful.

Actually, we have a big problem during the testing. We found the fingerprint of the same word will change a lot even if his pronunciation changes a little. So tried to record the same word for 20 times and get the average of the fingerprints. But we can't calculate their average value directly because their amplitude is quite different. So we use linear regression method i.e., try to normalize the every training sample to equivalent level then get their arithmetic average.

The program was able to recognize five words, but sometimes it would become confused and match the incorrect word if the word that was spoken varied too much from the word stored in the dictionary. As a rough estimate the program recognized the correct word about 70% of the time a valid word was spoken. The program achieved success using some voices, and with sufficient practice a person could say the same word with a small enough variation for the program to recognize the spoken word most of the time. For the general person though the recognition program would have a much lower percentage of success. Also the words in the dictionary are words spoken by only one person. If someone else said the same words it is unlikely the program would recognize the correct word most of the time, if at all.

For safety an testing we made sure the PWM signals sent to the car were as close to neutral as possible, while still letting the move go forward and backward. We did this to prevent the car from going out of control and potentially hurting others. Our project did not use any RF signals and the board we used ran just off of a battery so there were no physical connections to anything involving other people’s projects. Also the only pins switching state were the pins for the PWM, which were mostly covered by wire.

Using humanoid approach was not be able to our applications, and simple statistical was more robust and more accurate. This conclusion will not remain valid if number of voice commands increased; because statistical approaches fail to work find thresholds to separate between values coming from each command.

**References** :

Books ;

1. Rpdman,Rebert “Computer Speech Technology” 1999,Boston Pub .
2. Walter A.tribel ,Avtar “the 8088 and 8086 Microprocessor , interfacing”2000,prentice hall .inc .
3. [Stephen](http://en.wikipedia.org/wiki/Stephen_Butterworth) "Theory of Filter Amplifiers", *Wireless Engineer* (also called Experimental Wireless and the Wireless Engineer), vol. 7, 1930, and pp. 536-541.
4. Robin R.Murrhy , “introduction to Al Robotics”,2000 press Cambridge .
5. Stephen J.Chapman, “Electric Machinery Fundamentals” 1994 4th edition mcgraw-hill.

Websites :

1. [www.wikibidia.com](http://www.wikibidia.com)
2. [www.microchip.com](http://www.microchip.com)
3. [www.Mathworks.com](http://www.Mathworks.com)

**Appendix**

MATLAB SPEECH RECOGNITION SOFTWARE BASIC .

**Database**

% Butterworth Filter Design

fs=44100; %sampling rate

Fs=44100;

Wp = [150 8450]/11025 ; %Pass Frequency

Ws = [100 9450]/11025; %Stop

Rp = 0.8; Rs = 30.8;

[n,Wn] = buttord(Wp,Ws,Rp,Rs);

[b,a] = butter(n,Wn);

% Recording signals for the Five words (GO. Stop, Back, Left and Right)

z=1;

for z=1:5

if z==1;

fprintf('Record Go now')

s = wavrecord(2\*Fs,Fs,'double');

wavwrite(s,Fs,'go')

[s,fs]=wavread('go');

end

if z==2;

fprintf('Record Stop now')

s = wavrecord(2\*Fs,Fs,'double');

wavwrite(s,Fs,'stop')

[s,fs]=wavread('stop');

end

if z==3;

fprintf('Record Back now')

s = wavrecord(2\*Fs,Fs,'double');

wavwrite(s,Fs,'back')

[s,fs]=wavread('back');

end

if z==4;

fprintf('Record Left now')

s = wavrecord(2\*Fs,Fs,'double');

wavwrite(s,Fs,'left')

[s,fs]=wavread('left');

end

if z==5;

fprintf('Record Right now')

s = wavrecord(2\*Fs,Fs,'double');

wavwrite(s,Fs,'right')

[s,fs]=wavread('right');

end

% Filtering the Signals

sf=filter(b,a,s);

sf =sf/max(abs(sf));

wavplay (sf,Fs);

% Spectral Analysis

[B,f] = specgram(sf,Fs,Fs);

sff=B.\*conj(B);

sff(1:10)=0;

sff=sff/max(sff);

% Creating The Fingerprints

for i=101:2500

for j=1:100

for k=1:100

if sff (i-j)< sff (i) & sff(i+k)<sff (i);

sff(i)=sff(i);

else

sff(i)=0;

end

end

end

end

for i= 1:2500

if sff(i)<0.05

sff(i)=0;

end

end

n=sff(1:2000);

[c ns]=sort (n);

ns=flipud (ns);

% The Signals Database

x1=1;

while ns(x1)<2000

x11=x1;

x1=x1+1;

end

qw=ns(1:x11);

if x11>=3

q=ns(1:3);

else

fprintf('Record again')

end

q=sort (q)

if z==1

i1=q;

go=sf;

end

if z==2

i2=q;

st=sf;

end

if z==3

i3=q;

ba=sf;

end

if z==4

i4=q;

le=sf;

end

if z==5

i5=q;

rig=sf;

end

z=z+1;

end

ii =[i1 i2 i3 i4 i5]

**Speech Recognition**

Fs=44100;

fprintf('\n recod now\n')

s = wavrecord(2\*Fs,Fs,'double');

t=0:1/(Fs):1.99999;

% The Butterworth Filter

Wp = [150 8450]/11025 ; Ws = [100 9450]/11025; Rp = 0.8; Rs = 30.8;

[n,Wn] = buttord(Wp,Ws,Rp,Rs);

[b,a] = butter(n,Wn);

sf=filter(b,a,s);

sf =sf/max(abs(sf));

wavplay (s,Fs);

%Spectral Analysis

[B,f] = specgram(sf,Fs,Fs);

sff=B.\*conj(B);

sff(1:10)=0;

sff=sff/max(sff);

for i=101:2500

for j=1:100

for z=1:100

if sff (i-j)< sff (i) & sff(i+z)<sff (i);

sff(i)=sff(i);

else

sff(i)=0;

end

end

end

end

for i= 1:2500

if sff(i)<0.05

sff(i)=0;

end

end

n=sff(1:2000);

%hold on

[c ns]=sort (n);

ns=flipud (ns);

x1=1;

while ns(x1)<2000

x11=x1;

x1=x1+1;

end

qw=ns(1:x11);

if x11>=3

q=ns(1:3);

else

fprintf('recod again')

end

q=sort (q);

y=q

% Finding the Signal

x1 =norm(y-i1)/3;

x2=norm(y-i2)/3;

x3=norm(y-i3)/3;

x4=norm(y-i4)/3;

x5=norm(y-i5)/3;

x=[x1 x2 x3 x4 x5]

[s I]=min(x);

% Recognaize The Word

if I==1

fprintf('Go\n')

wavplay (go,Fs);

output=1;

elseif I==2

fprintf('Stop\n')

wavplay (st,Fs);

output=0;

elseif I==3

fprintf('back\n')

wavplay (ba,Fs);

output=2;

elseif I==4

fprintf('left\n')

wavplay (le,Fs);

output=5;

elseif I==5

fprintf('right\n')

wavplay (rig,Fs);

output=9;

end

plot (t,sf)

xlabel ('time (s)')

ylabel ('Amplitude')