

Voice Recognition in Noisy Environment Using Array of Microphone

Thesis submitted in partial fulfilment of the requirements for the
Degree of

Bachelor of Technology

in

Electronics and Instrumentation Engineering

By

Mayank Raj (111EI0256)



Department of Electronics and Communication Engineering

**NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA
769008, INDIA (2011-2015)**

Voice Recognition in Noisy Environment Using Array of Microphone

Thesis submitted in partial fulfilment of the requirements for the
Degree of

Bachelor of Technology

in

Electronics and Instrumentation Engineering

By

Mayank Raj (111EI0256)

Under Guidance of

Professor Lakshi Prosad Roy



Department of Electronics and Communication Engineering

**NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA
769008, INDIA (2011-2015)**

Declaration

I hereby declare that this thesis is my own work and effort. Throughout this documentation wherever contributions of others are involved, every endeavour was made to acknowledge this clearly with due reference to literature. This work is being submitted for meeting the partial fulfilment for the degree of Bachelor of Technology in Electronics and Instrumentation at **National Institute of Technology Rourkela** for the academic session **2011 – 2015**.

Mayank Raj
(111EI0256)



NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA 769008, INDIA

Certificate of Approval

This is to certify that the thesis entitled “Voice Recognition in Noisy Environment using Array of Microphone” submitted by Mayank Raj in partial fulfilment of the requirements for the award of Bachelor of Technology Degree in Electronics & Instrumentation Engineering at National Institute of Technology Rourkela is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

.....
Prof. Lokshi Prosad Roy
Assistant Professor
Dept. of Electronics and communication
National Institute of Technology, Rourkela
Rourkela, 769008

Date:

Acknowledgement

I have put lots of efforts in this project. But it would not have been possible without the kind support and help of many individuals and the department. I would like to thank all of them sincerely.

I am highly obliged to Department of Electronics and Communication for providing necessary information and guidance regarding the project and also for their support in completing the project.

I would like to express my gratitude and special thanks to my project guide Professor Lakshi Prosad Roy, Assistant Professor, Department of Electronics and Communications, NIT Rourkela, for all the cooperation and time and guidance. I would also like to thank my head of the department Prof. Kamalakanta Mahapatra for giving me such attention and time.

I would also like to thank to the people who helped me in creating my database for the project by recording their voices.

Last but not the least I would like to thank my parents and National Institute of Technology Rourkela for providing me this great opportunity.

Mayank Raj

A B S T R A C T

The performance of voice recognition reduces significantly in noisy environments, where the voice signals are distorted severely by addition of noise signal and reverberations. In such environments we can use array of microphone and use beamforming techniques to reduce the effect of noise signals. Presently, microphone-array-based voice recognition is done in two independent stages: first beamforming by array processing and then sending it for recognition. To reduce the effect of noise that is to reduce the distortion in voice waveform array processing algorithm is designed to enhance the signal before feature extraction and recognition. In Beamforming technique an array of sensors, in our case sensors are microphones, is used so that maximum reception can be achieved in a desired specified direction that is in the presence of noise, by the use of estimation of direction algorithm while signals from undesired direction are rejected though they are of same frequency. This is done by using delay and sum method in which the outputs from an array of microphones are delayed by some time so when they are added together, a particular part of the sound field is amplified over other undesired or interfering sources. Then the focussed voice wave is sent to voice recognition algorithm. Correlation algorithm is used for the voice recognition. The algorithm is based on the fact correlation graph between same signal is symmetric and value of correlation is maximum. The system development for this voice recognizer will be done using MATLAB for this project. Using MATLAB a GUI is created which has different function buttons to perform different tasks.

C o n t e n t s

Declaration	i
Certificate	ii
Acknowledgement	iii
Abstract	iv
Contents	v
List of figures	vii

1	Introduction	
	1.1 Beamforming	1
	1.2 Direction of arrival (DOA)	2
	1.3 Voice Recognition	3
	1.4 Problem Statement	4
	1.5 Methodology	4
2	Beam Forming	
	2.1 Non Blind Algorithms	5
	2.2 Blind Beamforming Algorithms	6
	2.3 Delay and Sum Method	7
	2.3.1 Microphone Array Design	8
	2.3.2 Fundamentals of Delay-Sum Beamforming	8
3	Direction of Arrival (DOA)	
	3.1 Introduction	11
	3.2 General Approach	12
	3.3 Trigonometric solutions	12
	3.4 Electrical System	15
	3.5 Least-Mean Square Algorithm (LMS)	15
	3.6 Application to System	16
	3.7 Delay calculation	16
	3.8 Flow Chart of Whole Process	18

4	Voice Recognition	
	4.1 Introduction and overview	19
	4.2 Theory	19
	4.2.1 The DC Level and Sampling Theory	20
	4.2.2 Time Domain to Frequency Domain: DFT and FFT	21
	4.2.3 Frequency Analysis in MATLAB of Speech Recognition	21
	4.2.3.1 Spectrum Normalization	21
	4.3 The Cross Correlation Algorithm	22
5	Simulation and Results	
	5.1 Direction of Arrival (DOA)	24
	5.2 Voice Recognition	25
6	Matlab Codes	33
7	Conclusion	39
	References	40

List of Figures

Fig 1	Array of mics
Fig 2	DOA
Fig 3	Description of Physical Setup
Fig 4	Trigonometric Approach
Fig 5	Diagram of Electrical Setup
Fig 6	LMS Algorithm
Fig 7	The Simple Figure About Sampling the Analog Signal.
Fig 8	Absolute Values of the FFT Spectrum without Normalization
Fig 9	Absolute Values of the FFT Spectrum with Normalization
Fig 10	Examples of Symmetrical and Unsymmetrical Correlation Graph.
Fig 11	Output of Direction of Arrival
Fig 12	Spectrums for 2 Voices
Fig 13	Figure Showing the Symmetry between the same Voices of Mayank
Fig 14	Image of GUI Created by MATLAB
Fig 15	The Output

Chapter 1

Introduction

1.1 Beamforming

In last 20 years the use of beam forming has gained importance. Beamforming has been used in radar, sonar and many more devices. Beamforming term is derived from the fact that earlier spatial filters were designed to form pencil beam from the radiation from specific location and attenuate signals from other signals. It is difficult for an array of mic system to scan and localize a single person's voice in a noisy environment because the mics detects other conversations and background noises in real time. Microphones detect the voice of the person speaking plus any additional side conversations or noises. The clarity of the speaking person's voice can be largely improved by use of a microphone system that implements a focused beam that magnifies the speaker's voice and neglects the unwanted or interfering noises present in the vicinity. This is the basis of Acoustic Beamformer [10].

Beamforming is a technique in which an array of mics or receiver is used in such a way so that in the presence of noise also, maximum reception in a specified direction takes place by estimating the arrival of signal from a desired direction while rejecting signals of the same frequency from other or undesired directions. It basically uses the concept that, the same frequency channel is occupied by the signals coming from different transmitters, but they still arrive from different directions. This spatial separation is used to segregate the wanted signal from the unwanted signals that are noises which interfere and distort the desired signal. In beamforming technique the optimum weights are recursively calculated by the use of complex algorithms which is based upon different criteria [8].

Depending on application sources may be classified into narrow band and broad band, in far field or near field. The beamforming technique in which we know the positions of array of mics is well documented. Different algorithms are used to know the structure of steering matrix to get the information regarding direction of arrival. Beamforming for broadband sources is an extension of narrow band beamforming in frequency domain by using sub band filter or focussing matrix techniques [1].

When we don't know the position and response of the sensors then this type of beamforming is known as blind beamforming. In this technique narrowband sources with some known characteristics is used. The features which are used are:

- Cyclostationarity property
- Spectral Self coherence or finite alphabet property of signal
- Statistical difference between desired and undesired signal
- The constant modulus characteristics of FM/PM.

Beamforming is generally achieved by phasing the feed to each element of an array so that signals received or transmitted from all elements will be in phase in one direction. Adjustment of phases (inter element phase) and amplitudes is done to optimize the received signal.

The use of low end of the spectrum has increased these days so, higher frequency band where more spectrums are available are explored. Multipath fading and cross interference became more serious issues because of the use of higher frequencies, higher data rate and higher user density which results in the degradation of bit error rate (BER). To cope up with these problems array of smart antenna with beamforming is used which also helps to achieve higher communication capacity, and are very effective in suppressing the interference and multipath signals [1].

1.2 Direction of Arrival (DOA)

The need for Direction-of-Arrival estimation has many applications in engineering including wireless communications, radar, radio astronomy, and sonar, navigation, tracking of various objects, rescue and other emergency assistance devices. In its modern version, DOA estimation is considered as part of the more general field of array processing and studied as a part of array processing. Earlier works in this field focussed on finding the direction of radio that is electromagnetic waves coming from one or more antennas are estimated [11].

The science of determining the direction and distance of a source with the only help of its sounds is known as Sound location. If we get these two parameters then we can get an accurate location of a speaker which is important for a certain number of applications. Nevertheless, calculating the distance is not always important since most of these applications is to make estimation of direction more efficient. This allows to design less-complex systems, without compromising a good performance. For example on videoconference, the system does not need to calculate the distance in which the source is emitting sounds. With the knowledge of the direction exclusively the camera can focus the speakers. Audio surveillance systems is another

example that requires only the calculation of the direction. Such systems, used for intrusion detection or gunfire location, they determine the direction of source but they do not give importance to the distance of source.

Adaptive signal processing sensor arrays, which is also known as smart antennas, have been adopted widely in third-generation (3G) mobile systems because they can locate mobile users with the use of DOA estimation techniques. Adaptive antenna arrays also enhances the performance of cellular systems by providing robustness against fading channels and reduced collateral interference. [Direction] With the help of signal processing aspects of smart antenna systems many efficient algorithms for Direction-of-Arrival (DOA) estimation and adaptive beam forming has been developed. The recent trends of adaptive beam forming drive the development of digital beam forming systems [5].

1.3 Voice Recognition

Voice recognition is a topic that very useful in many applications and environments in our daily life. Generally, voice recognizer is a machine which scans the input voice and identifies the speaker after comparing voices from database. Voice recognition performance reduces significantly in noisy environments, where the voice signals can be severely disturbed by addition of noise signal and reverberations. To avoid distortion in such environments we can use array of mics and use beamforming techniques to reduce the effect of noise signals. Presently, microphone-array-based voice recognition is done in two independent stages: first array processing and then recognition. Array processing algorithms is designed for signal enhancement and are applied in to reduce the distortion in the voice signal before feature extraction and recognition. Recently many approaches have been developed to solve the problem of speech recognition, using feature-normalization algorithms, microphone arrays, representations based on human hearing, and other approaches. This approach is carried out to improve the quality of the speech waveform that will necessarily result in improved recognition performance. However, voice recognition systems are statistical pattern classifiers that will not process the waveform as a whole but it will process the features which are derived from the speech waveform.

1.4 Problem Statement

In present era the scope of voice commands and recognition has increased for example laptops and systems will open by recognizing user's voice and then it would allow to perform the latter tasks and many more examples but there is a particular situation in the workplace where microphones pick up the voice of the person speaking as well as any additional side conversations or noises which would make the voice recognition job tough for any systems.

1.5 Methodology

The clarity of the speaking person's voice can be greatly improved using a microphone system that implements a focused beam that magnifies the speaker's voice over unwanted or interfering sound sources and in addition could detect the direction of arrival of the speaker's voice. For this purpose an acoustic beam former can be used. As the distance between source of wave and receiver, in our case microphone increases the accuracy to recognise the voice decreases because of noises and reverberations, hence array of mics is used instead of single mic that will compensate the distortion then after that the focussed waveform will be sent for feature extraction and recognition.

Chapter 2

Beamforming

2.1 Non Blind Algorithms [8]

We want to study means in which certain specific characteristics of the received signal which are incident upon the array of mics (in addition to the spatial separation among users in the environment) can be used to direct beams in directions of wanted users and nulls in directions of interferers. In particular, the Mean Square Error (MSE) criterion of a particular weight vector will be minimized by using statistical expectations, time averages and instantaneous estimates. As well, restoration of the distorted constant modulus of the array output envelope due to noise in the environment is also carried out. Finally, concept of the spreading sequences of a CDMA mobile environment will be utilized to improve the performance of algorithms exploiting the two criteria discussed above. Each of the characteristics which are described above corresponds to adaptive algorithms which can be classified into two categories:

- 1.) Non-Blind Adaptive algorithms
- 2.) Blind Adaptive Algorithms.

When there is need of statistical knowledge of the transmitted signal and the position of sensors in order to converge to a weight solution then this type of algorithm is known as Non-blind adaptive algorithms. This is typically done by using a pilot training sequence which is sent over the channel to the receiver which helps in identifying the desired user. On the other hand, blind adaptive algorithms do not need any training, hence the term “blind”. They restore a particular type of trait of the transmitted signal so that it can be separated from other users in the surrounding environment.

Non Blind Adaptive Algorithms

As we know that non-blind adaptive algorithms requires a training sequence which extracts a desired user from the environment. This feature is itself undesirable because during the transmission of the training sequence, there is no communication in the channel. As a result the spectral efficiency of any communication system reduces dramatically. Additionally, it can

be very difficult to understand the statistics of the channel in order to characterize a reasonable estimate of needed to accurately adapt to a desired user.

Keeping these things in mind following are algorithms for non-blind beamforming.

1. Weiner Optimum Solution.
2. Sample Matrix Inversion (SMI).
3. Least Mean Square (LMS).
4. Recursive Least Squares.

2.2 Blind Beamforming Algorithms [1]

When the information regarding sensor placement and response is partially or totally lacking then this type of beamforming is referred to as blind beamforming. The narrowband sources are used for this technique because narrowband source have some properties that can be used for this process.

These properties are

1. Cyclo-stationary property that means signal has statistical properties that vary cyclically with the time.
2. Spectral self-coherence
3. The constant modulus characteristics of frequency modulation or phase modulation signals.
4. The statistical difference between desired and undesired source.

Blind adaptive algorithms do not need a training sequence in order to determine the required complex weight vector. They attempt to restore some type of property to the received signal for estimation. A common property between polar NRZ waveforms and DS-SS signals is the constant modulus of received signals.

Some of the blind beamforming techniques are:

1. Constant Modulus Algorithm (CMA)
2. Steepest Descent Decision Directed Algorithm (SD-DD)
3. Least Square Constant Modulus Algorithm (LS-CMA)
4. Recursive Least Squares Constant Modulus Algorithm (RLS-CMA)

The method used in this project for beamforming is **Delay and Sum Method**.

2.3 Delay and Sum Method [12]

We can measure the gap between the moment sound strikes the first microphone and moment when sound strikes the second microphone if we have an array of microphones and sufficient signal-processing capability. This time gap is known as delay. We can treat the edge of propagation as a plane if the distance of source is large then we can use simple trigonometry to calculate the delays.

From figure below:

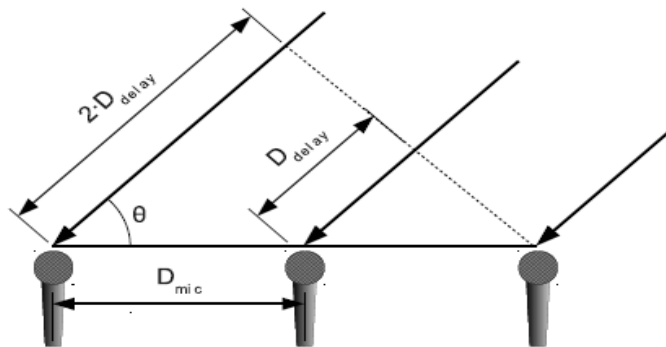


Fig 1 Array of mics.

Assuming a linear array of microphones, m_1 through m_n , each spaced D_{mic} meters apart. Then D_{delay} , the formula to calculate the extra distance the sound has to travel for each successive microphone is:

$$D_{delay} = D_{mic} \cdot \cos \theta$$

The speed of sound is 340.29 m/s

$$T_{delay} = \frac{D_{mic}}{340.29} \cdot \cos \theta$$

To recover the original signal the above delay for each microphone is reversed and summing of inputs is done. If a signal comes from different direction, the delays will be different, and as a result, the individual signals will not line up and will tend to cancel each other when added. This essentially creates a spatial filter, which we can point in any direction by changing the delays.

The direction from which the signal came from can be determined, by sweeping the beam around the room, and recording the total power of the signal received for each beam. The direction in which highest power-signal is detected is the direction from which the signal is coming.

2.3.1 Microphone Array Design

The arrays of microphone can be almost of any shape: linear, circular, rectangular, or even spherical. Beamforming in one dimension is done by one-dimensional array and, for 2-dimensional beamforming additional array dimensions is used. Given the limited number of microphones and amount of time we have, a linear array is the best choice.

Microphone spacing: The desired operating frequency range decides the spacing of the microphones. It is advantageous to use narrower beam width for spatial filtering because unwanted signals from undesired directions are filtered. A narrow beam width is analogous to a narrow transition band for a traditional filter. Lower frequencies will correlate better with delayed versions of themselves than high frequencies, so the lower the frequency, and broader the beam. The length of array decides the delay between the end microphones, so longer the array greater the delay between the end microphones, and will thus reduce the beam width. The highest operating frequency is determined by the spacing between microphones. The spacing between microphones causes a maximum time delay which, together with the sampling frequency, limits the number of unique beams that can be made.

$$N_{\max beam} = 2 \cdot F_s \cdot time_{spacing}$$

$time_{spacing}$ is time taken by sound to travel from one microphone to adjacent.

2.3.2 Fundamentals of Delay-Sum Beamforming

Delay-sum beamforming is a signal processing technique in which the outputs from an array of microphones are delayed by some time so when they are added together, a particular part of the sound field is amplified over other undesired or interfering sources. A linear array has been chosen because it has less processing complexities and is effective over a 180° field.

The figure below illustrates this setup here S is sound acoustic source placed in the field at an angle θ .

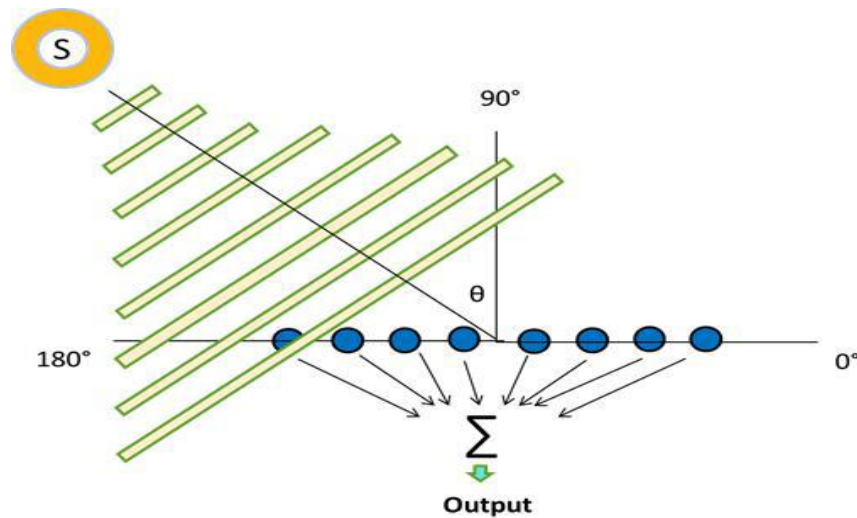


Figure .2 DOA

If the sound waves emitted from source S is observed, it can be noticed that the microphone furthest to the left captures the sound waves first. The next microphones will receive the same signal, but after some time which is the time delay which is because of the additional distance sound waves travels to get to the adjacent microphone. When the outputs of the individual microphones are added, we get:

$$\Sigma output = S(t) + S(t - \Delta d_1) + S(t - \Delta d_2) + \dots \dots S(t - \Delta d_n)$$

$S(t)$ is the wave equation which represents the signal emitted from sound source S . The first microphone has the output $S(t)$, and each subsequent microphone has a time delay Δd_n , where n represents the microphone index (leftmost microphone has $n=0$, and the rightmost microphone has $n=$ (number of microphones $- 1$)). Fourier transform of each term is taken, this output is represented in the frequency domain in form of series of complex valued functions. We get equation in frequency domain:

$$\Sigma output = S(f) + S(f)e^{-j2\pi\Delta d_1} + S(f)e^{-j2\pi\Delta d_2} + \dots \dots S(f)e^{-j2\pi\Delta d_n}$$

From above equation:

It can be seen that if the time delays (Δd_1 to Δd_n) are equal to 0 then there is a maximum magnitude. This is the **principle of delay-sum beamforming**. For example we have to amplify

the received signal from source S at angle θ as shown in figure 1. If we know the parameters of our array, then we can calculate the time delays which is caused by sound waves emitted from a source at angle θ . This calculated delay Δd_{TAn} , in which the term TA denotes these are calculated at our “target angle” θ , and n , or the microphone index, refers to the microphone to have this time delay.

$$\Sigma output = S(f) + S(f)e^{-j2\pi(\Delta d_1 - \Delta d_{TA1})} + S(f)e^{-j2\pi(\Delta d_2 - \Delta d_{TA2})} \dots S(f)e^{-j2\pi(\Delta d_n - \Delta d_{TAn})}$$

If calculation is accurate then:

$$\Delta d_{TAn} = \Delta d_n$$

Expression reduces to

$$\Sigma output = (n + 1)s(t)$$

n = microphone index

TA_n = Target angle

The result is the sound waves heard from source S , it is now expressed in the time domain represented by the wave equation $S(t)$, whose magnitude is multiplied by a factor of $(n+1)$, the number of microphones in array. The signal $S(t)$ in our summated output by multiplication of factor $(n+1)$, it must be amplified considerably more compared to other undesired sound sources that appear in our field in order to *localize* this source. This quality is known as the spatial resolution of our design, which can be more simply characterized as our “beamwidth.”. As seen from above equation when we put $S(t)$ in equation then the θ term comes which is Direction of Arrival

Chapter 3

Direction of Arrival

3.1 Introduction[5]

Sound location is the science in which the direction and distance of a source is determined with the only help of its sounds. These two parameters allow an accurate localization of a speaker which is important for a certain number of applications.

The work of **direction-of-arrival (DOA)** estimation is to estimate the directions of the signals from the desired users as well as the directions of unwanted signals by using the data received by the sensor array. To maximize the radiated power towards desired users the weights of adaptive beamformer is adjusted by using the results of DOA estimation, and radiation nulls are placed in the directions of interference signals. The choice of the DOA estimation algorithm decides a successful design of an adaptive array therefore the DOA estimation algorithm should be highly accurate and robust.

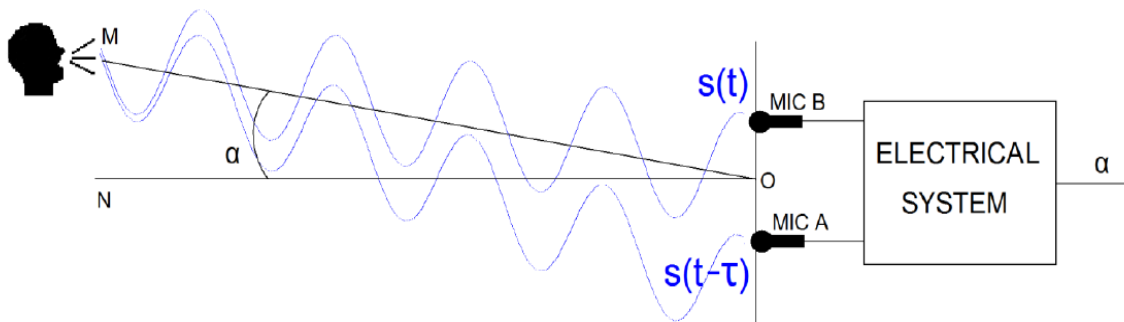


Figure 3 .Description of physical setup

3.2 General Approach[5]

Figure 2 illustrates the entire system, including the physical property of time difference of arrival at two microphones receiving a sound wave coming from a human speaker. The aim of the electrical system is estimation of the real physical angle, α , accurately. A source (M) located in front of two microphones; the aim is to determine the DOA of its sounds. The origin from which the measurements will be performed is fixed. The position of microphones is fixed and is separated by a fixed distance. The middle of the microphones was made the origin. Considering the orthogonal line to the microphone axis at the origin (ON), the angle α is the separation angle between this line and line (OM). From now on, angle α will refer to the term “direction”, where the speaker is located.

Observing the example shown in Figure 1, as the speaker stands nearer to MIC B than MIC A therefore the sound traveling in the air from the speaker to the microphones first reaches MIC B and then to MIC A. τ denotes the time gap between these two moments.

The sound signal can be represented as an analog signal $\mathbf{S}(\mathbf{t})$. Theoretically, the amplitude of the signals captured by both microphones will be equal and will only be delayed a time τ . Hence, considering $\mathbf{S}(\mathbf{t})$ the signal captured by MIC B, it can be affirmed that the signal captured by MIC A would be $\mathbf{S}(\mathbf{t} - \tau)$.

3.3 Trigonometric Solutions

After getting the delay τ between the two signals is, with the help of trigonometric calculations, the angle can be found. The position of the source is represented by a point M with coordinates x and y . This two coordinates are assumed variable and unknown. Two points A and B are also considered with respective coordinates (x_A, y_A) and (x_B, y_B) which corresponding to the positions of the microphones. The distance between them is fixed to d cm. The point of origin (origin) is defined as the middle point between A and B.

The aim is to get the angle which will give the direction of from where speaker speaks. A signal coming from the speaker reaches the point B at time t . In that moment, another point of the same wavefront is in the direction between M and A. This point is B' and as it belongs to the wavefront, the distances BM and B'M are equal. Hence AB' is the distance travelled by the signal during the delay τ . The following figure illustrates the physical setup.

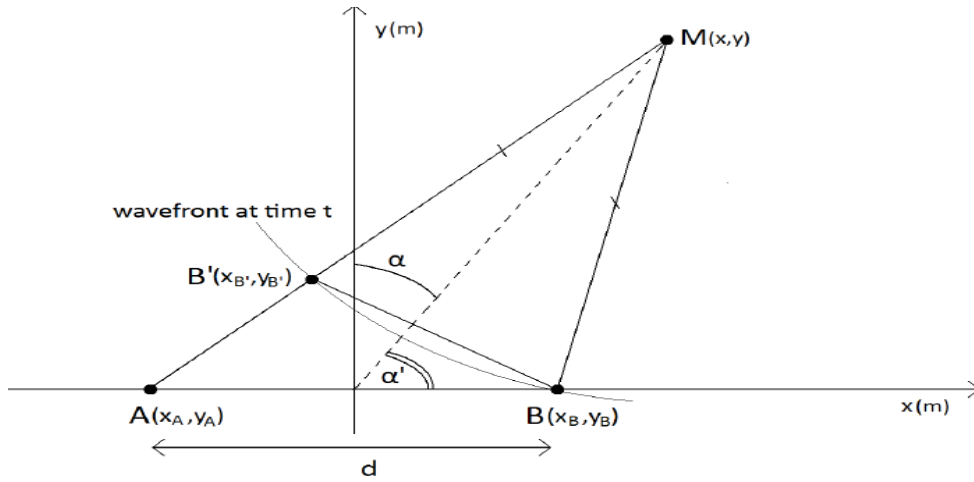


Fig 4 Trigonometric approach

Considering the suppositions exposed above, the following equations can be derived:

$$AB' = AM - B'M$$

Putting $B'M = BM$

$$AB' = AM - BM$$

With

$$AM = \sqrt{(x_A - x)^2 + (y_A - y)^2}$$

$$BM = \sqrt{(x_B - x)^2 + (y_B - y)^2}$$

In order to remove the square roots, the equation (3.3) is squared

$$AB'^2 = (AM - BM)^2 = AM^2 + BM^2 - 2 \cdot AM \cdot BM$$

Since the two microphones have fixed positions the following statements applies:

$$x_A = -x_B$$

$$y_A = y_B = 0$$

This simplifies the equation and after several calculations and term reordering, leads to

$$y = \pm \sqrt{\frac{AB'^2}{4} - x_B^2 + x^2 \left(\frac{4 \cdot x_B^2}{AB'^2} - 1 \right)}$$

In this expression the only variables are y and x . The value is always constant since it represents the position of the microphones, which can be seen as reference points. Moreover even if the direction can vary, the length of AB' remains unchanged. So the equation represents all the possible positions of M , given a certain delay. Considering that the signal travels at the speed of sound c , the distance AB' is:

$$AB' = T \cdot c$$

Now from figure

$$\tan \alpha' = \frac{dy}{dx}$$

From here we get α' then

If $\alpha' \geq 0$

$$\alpha = 90 - \alpha'$$

If $\alpha' < 0$

$$\alpha = -90 - \alpha'$$

3.4 Electrical System

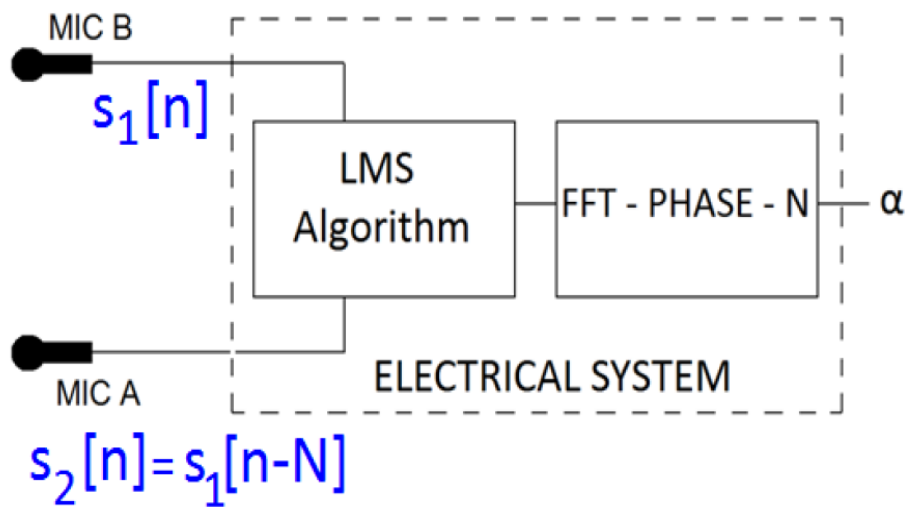


Figure 5 . Diagram of electrical system

To obtain α , the signals are first processed by an algorithm (Least Mean Square) which will provide N .

3.5 Least-Mean Square Algorithm (LMS)

The Least-Mean-Square algorithm (LMS) is used in adaptive filters to calculate the filter coefficients which helps in getting the minimum expected value of the error signal. Error is the difference between the desired signal and the output signal. LMS belongs to the family of stochastic gradient algorithms, i.e. on the basis of the error in the present moment only the filter is adapted. It does not require correlation function calculation nor does it require matrix inversions, so it is relatively simple.

Consider two signals $x[n]$ and $d[n]$, and then consider the filter $h[n]$ such that:

$$d[n] = x[n] * w[n]$$

Where $*$ is convolution operator. Applying LMS algorithm to $x[n]$ and $d[n]$ will theoretically give $w[n]$ as an output. As shown in the picture below, LMS algorithm has two inputs, $x[n]$ and $d[n]$, and three outputs, $y[n]$, $e[n]$ and $w[n]$.

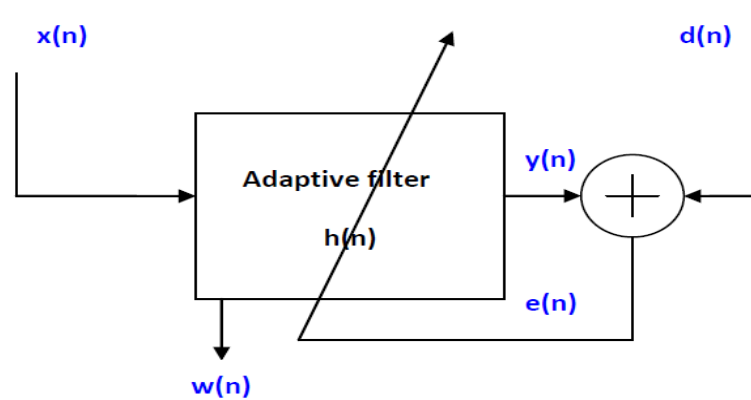


Figure 6 LMS Algorithm

3.6 Application to System

The delay existing between the two captured signals can be expressed on discrete domain as a value N . Besides, according to the expression the function allowing the transformation from to can be represented as a Dirac Delta centred in N .

Comparing:

$$x[n] = s_2[n]$$

$$d[n] = s_1[n]$$

then

$$h[n] = \delta[n - N]$$

$N = \text{delay}$

3.7 Delay Calculation

There are three steps for delay calculation: First the Fast Fourier Transform is obtained, then its phase and finally the delay N . The choice of this three step method was due to its simplicity and good performance. The FFT is an efficient algorithm used to calculate the Discrete Fourier Transform (DFT), so it will help to switch from time domain to frequency domain. Furthermore, it is a tool that MatLab can run very efficiently.

Its DFT is defined as

$$f_j = \sum_{k=0}^{n-1} x_k e^{\frac{-2\pi i}{n}jk} \quad j = 0, \dots, n-1$$

Considering an ideal situation, the filter would be a pure Dirac delta delayed N' samples.

$$h[n] = \delta[n - N']$$

This means

$$h[n] = \{0, 0, \dots, 1, \dots, 0\}$$

So its n -point DFT would be

$$H_j = 1 \cdot e^{\frac{-2\pi i}{n}jN'}$$

This transform has two main components: modulus and phase. Since the desired information is the position of the delta (N'), only the phase will be useful. So the next step consists on calculating the phase, which is really simple with MatLab commands. Calling Ω the phase

$$\Omega = \frac{-2\pi}{n}jN'.$$

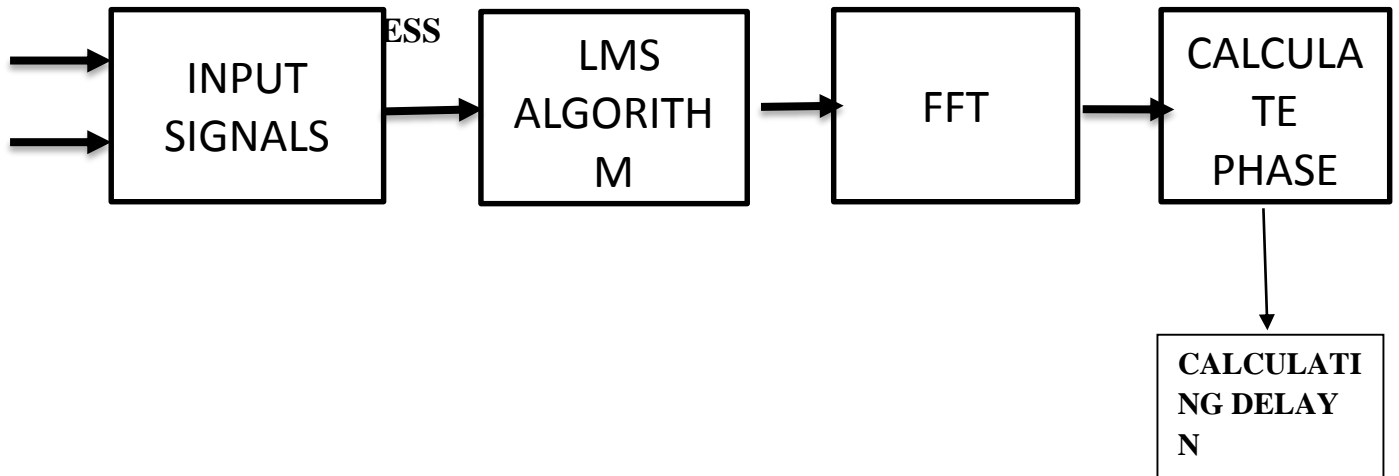
The only variable is j , which is the index of the FFT. So the phase is linear and depends directly on the delay N' . With the derivate, the variable j disappears and the slope S is obtained.

$$S = \frac{-2\pi}{n}N'$$

$$N' = \frac{-S \cdot n}{2\pi}$$

$$\tau = \frac{N}{f_s}$$

3.8 Flow Chart of Whole Process



Chapter 4

V o i c e R e c o g n i t i o n

4.1 Introduction and Overview

In this project a system is designed based on information and data which we get from shape of the cross-correlation plots. The simulation of the programmed system in MATLAB is accomplished by using the mic to record the voice of speaker. After compiling and running the code in MATLAB, it will ask user to record the voice three times. The first and second voices are different are our reference signal for comparing. The third recorded voice is same as one of the two words already stored. After recording voices, these voices will be our signal's information which will be sampled and stored in MATLAB. Now MATLAB will give judgement that what was the word which was recorded the third time after comparing with the first two reference words on the basis of the algorithm used that is correlation algorithm.

For visual clarity their spectrum was plotted. Then Correlation was taken between the test signal and the reference signals and the correlation values were plotted. It was observed that the graph of correlation between the same signals was symmetric since there is no time shift in same signals. Using this concept first a program for limited number of people was developed and tested then after testing results were found positive

The main goal for this project is to create a simple GUI using MATLAB to perform the voice signal analysis for the voice recognition purpose. The GUI will consist of several functions and buttons of operation regarding to the speech recognition analysis. User has to just select one of the buttons to perform certain job. The GUI will display the desired result according to what its task is and have the ability to save or print the results.

4.2 Theory[2]

In this section some definitions and concepts are introduced which are necessary for understanding and which are used in the development of algorithm for voice recognition.

4.2.1 The DC Level and Sampling Theory[2]

When the analysis of signal processing is done then the DC level's information for the target signal if not applied to real analog circuit then it is not that useful. The DC level is not that useful when we analyse the signals in frequency domain. When the target signal is aggregated in the lower frequency band then the magnitude of DC level will interfere with analysis. There is no change in variance and mean value of the signal, as the time changing in WSS condition for the stochastic process. Therefore by deducting the mean value of recorded signals this effect is reduced, hence the zero frequency components for DC level in frequency spectrum is removed.

Since voice is recorded from the microphone, it records the person's analog voice hence voice signal will directly decide the efficiency of voice recognition so the sampling frequency is also a decisive factor.

$$x(t) = A \cos(2\pi ft)$$

This analog signal can not be directly applied to the computer. It needs sampling that is it is converted from $x(t)$ to $x(n)$. The following figure shows the sampling of a signal.

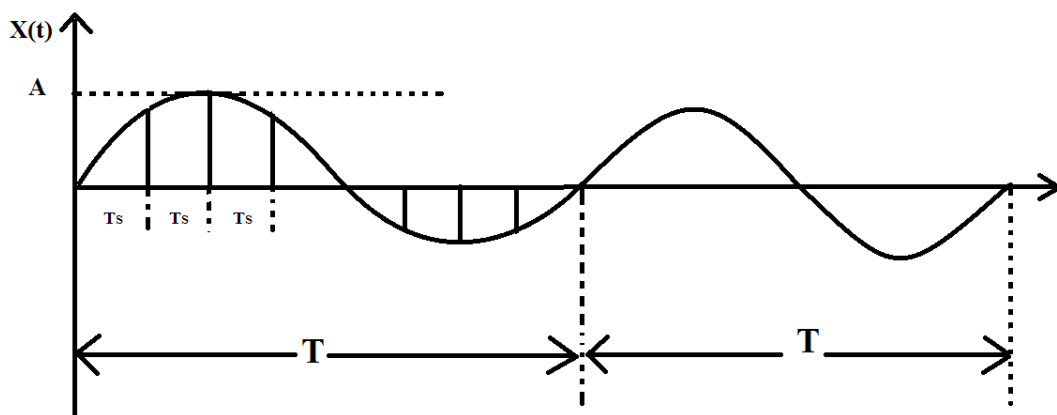


Fig 7 the simple figure about sampling the analog signal

4.2.2 Conversion from Time Domain to Frequency Domain: DFT and FFT[2]

4.2.2.1 DFT

The full form of DFT is *Discrete Fourier Transform*. So the DFT is just a type of Fourier Transform for the discrete-time $x(n)$ instead of the continuous analog signal $x(t)$. The Fourier Transform equation is as follow:

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n) e^{j\omega n}$$

As we can see from the above equation that the variable n is converted to ω that is the signal is transformed from time domain to frequency domain.

4.2.2.2 FFT

The full form of FFT is *Fast Fourier Transform*. FFT is same as DFT that is it also converts the discrete time signals from time domain to frequency domain. The difference between FFT and DFT is that FFT is faster than DFT and it is more efficient than DFT on computation. So FFT is used in our algorithm.

4.2.3 Analysis of Frequency in MATLAB of Speech Recognition

4.2.3.1 Spectrum Normalization

When comparing two voice signals it is hard or irrelevant to compare the spectrum in different measurement unit so using normalization the measurement standard is made same. The normalization reduces the error while comparing the spectrums.

Linear normalization is expressed as below:

$$y = (x - \text{MinValue}) / (\text{MaxValue} - \text{MinValue})$$

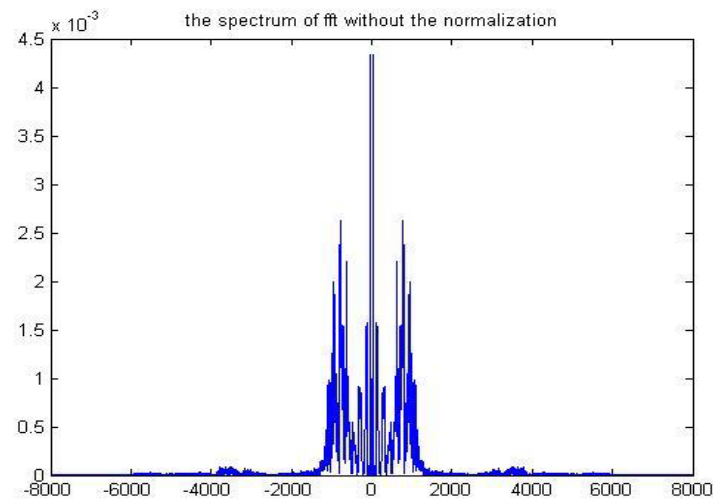


Fig 8 Absolute values of the FFT spectrum without normalization

The above spectrum is normalized and plotted.

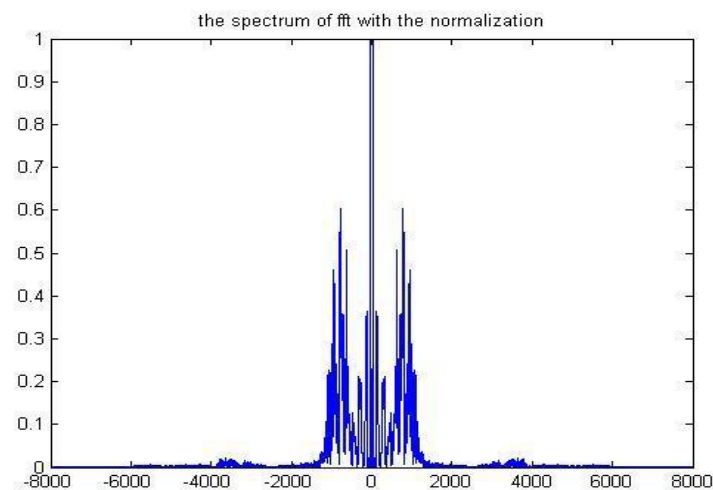


Fig 9 Absolute values of the FFT spectrum with normalization

The interval of $|X(\omega)|$ is changed to $[0,1]$ and no other information is changed.

4.3 The Cross Correlation Algorithm

For the same speaker there exists a different frequency band for the same word due to different vibrations of vocal cord hence spectrums are also different. Similarly the spectrums for different speakers are also different. These are the bases of this thesis for the speech recognition.

The cross correlation is an important tool to estimate the shift parameter that is frequency shift.

The equation of the cross-correlation for two signals is as below

$$r_{xy} = r(m) = \sum_{n=-\infty}^{\infty} x(n) y(n + m), m = 0,1,2..$$

STEP1: One of the two signals $x(n)$ is fixed and other signal $y(n)$ is shifted left or right with some time units.

STEP2: The value of $x(n)$ is multiplied with the shifted signal $y(n+m)$ position by position.

STEP3: Summation is taken of all the multiplied results for $x(n) \cdot y(n+m)$.

Important results which can be derived from cross correlation are:

1. The correlation is maximum if two signals have no time shift.
2. The correlation is symmetrical if two signals have no time shift.

Now when two voices are recorded and if they are voice of same person then their spectrum is same. When cross correlation of same spectrums is taken and graph is plotted then according to cross correlation algorithm the graph should be approximately symmetric and value should be maximum. Hence viewing these symmetric properties the person is recognized. The values are stored in an array and then the maximum value is taken out. The maximum value gives the desired person.

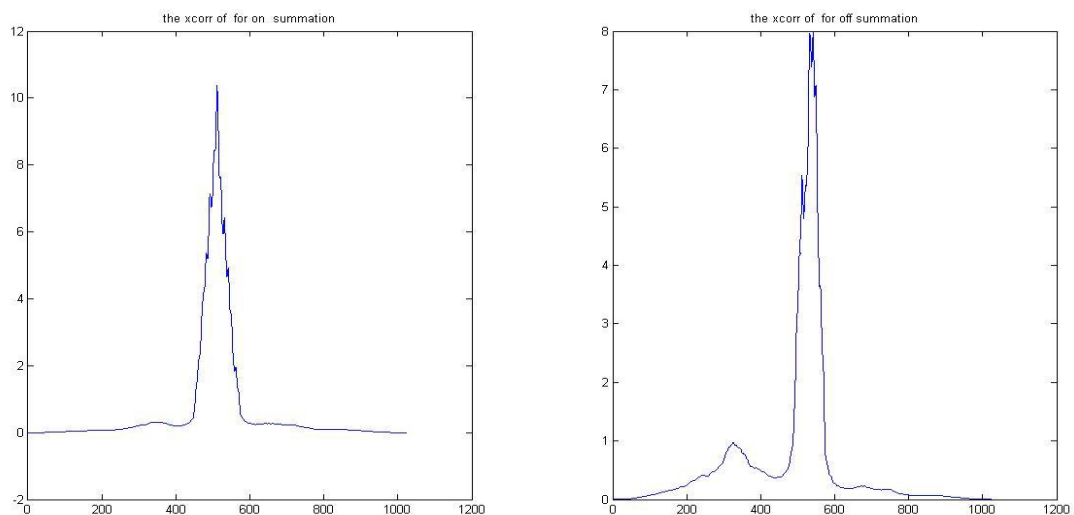


Fig 10. Examples of symmetrical and unsymmetrical correlation graph.

Spectrogram function in MATLAB has been used for plotting spectrums.

Chapter 5

Simulations and Results

5.1 Direction of Arrival (DOA)

The MATLAB code for direction of arrival was run. The code asks the user to record his or her voice using microphone (in our case microphone of laptop is used). Then the code calculates the angle α as shown in figure below.

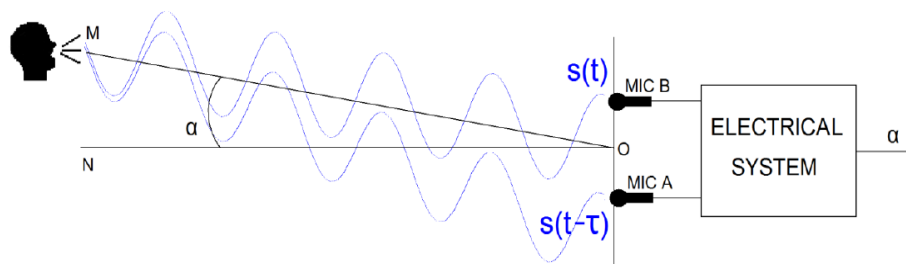


Fig 11. Figure showing the angle of arrival

The following output was obtained

B	<1x50 double>	-0.0189	0.0261
DESP	20	20	20
H	<1x128 complex dou...	0.0022...	-0.051...
H1	<1x50 double>	0	0.0261
I	<1x50 double>	1	50
K	-0.9030	-0.9030	-0.9030
M	<1x127 double>	-2.4728	1.9659
M1	0.3518	0.3518	0.3518
M2	0.3026	0.3026	0.3026
M3	0.3518	0.3518	0.3518
N	-1.6051	-1.6051	-1.6051
Nprime	18.3949	18.3949	18.3949
P	50	50	50
alpha	<1x128 double>	-3.1173	3.1416
angleGRAD1	7.1085	7.1085	7.1085
ans	[606171,2]	2	606171
c	340	340	340
d	<1x22050 double>	-1.3544	1.7200
d1	<1x22070 double>	-1.3544	1.7200
d_micro	0.1000	0.1000	0.1000
e	<1x22050 double>	-1.2829	0.9369
fs	44100	44100	44100
h	<1x50 double>	-0.0669	0.1380
h0	<1x50 double>	0	0
h1	<1x50 double>	-0.0189	0.0261
hDESP	<1x21 double>	0	1
hootie	<425958x2 double>	<Too ...	<Too ...
k	595350	595350	595350
IM	129	129	129
left	<606171x1 double>	<Too ...	<Too ...
lengData	22050	22050	22050

Fig 12 Output of Direction of arrival

The above figure is screenshot of MATLAB output window. Angle Grad1 is the value of α that is direction of arrival. This is just one example showing the results. Many values were calculated using the same code and were tested.

5.2 Voice Recognition

Now the MATLAB code for voice recognition was run. The code prompted to record different voices. For checking purpose initially two voices were recorded and spectrums were plotted to see the similarity between same voice and difference between different voices. Following were the plots for MAYANK and HARSHIT.

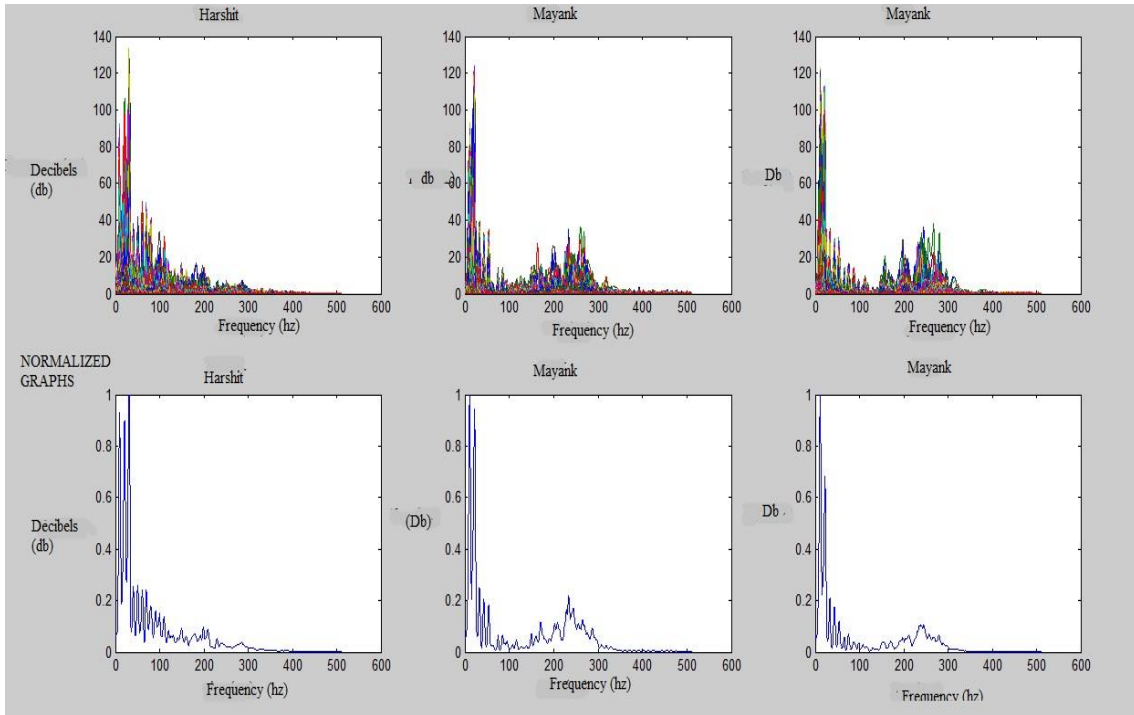


Fig 13. Spectrums for two voices

The second row of graph is normalized graphs. It can be observed clearly the similarity between the same voice and difference in spectrum between two different voices. Similarly many voices can be recorded and checked.

Now after the spectrum plotting the correlation was taken and plotted. The graph correlation between the same voices was symmetrical as observed in the plot below.

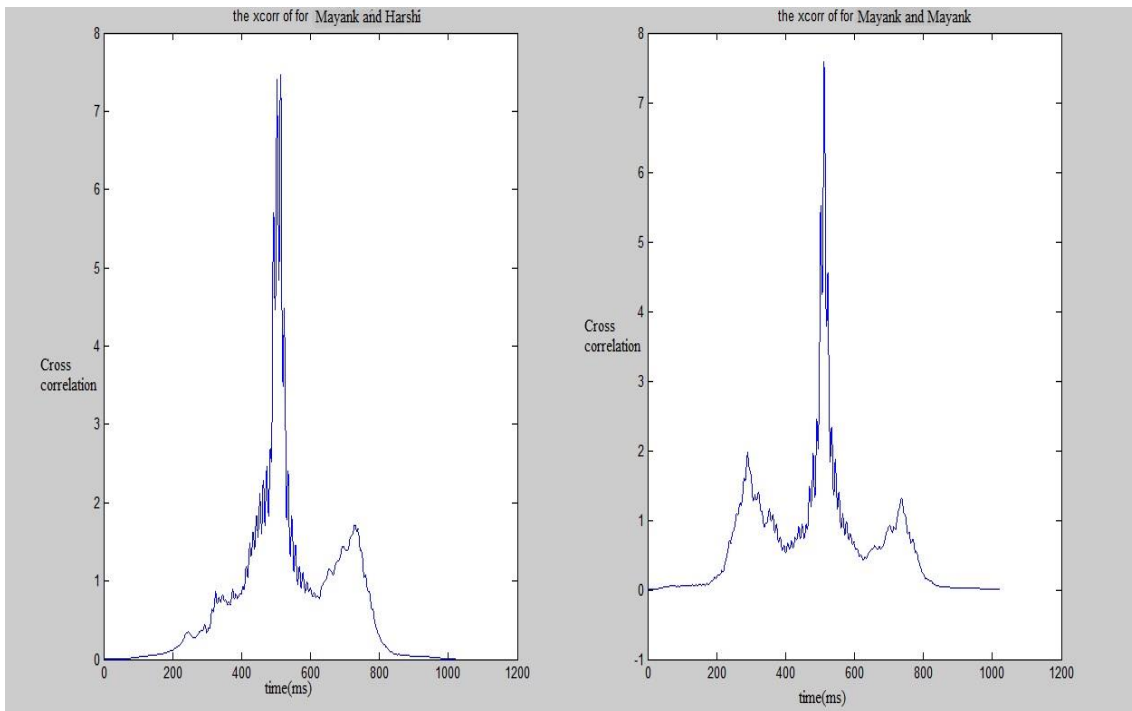


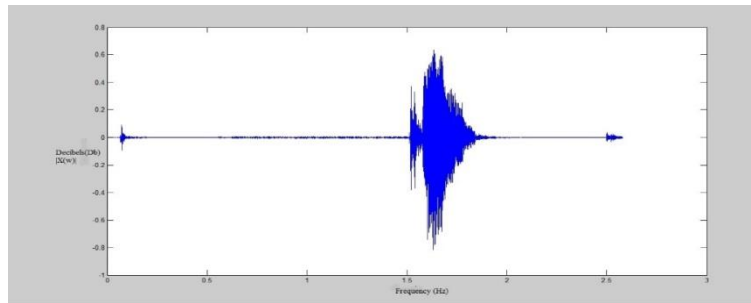
Fig 14. Figure showing the symmetry between the same voices of Mayank

As observed the second graph is approximately symmetrical since it represents the correlation between same voices of Mayank hence the theory is correct. So we proceed in this direction and continue this for more number of people.

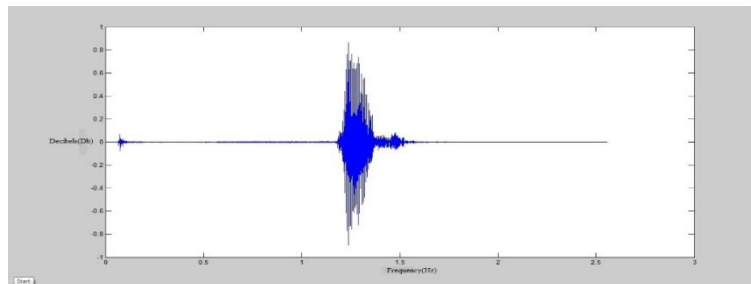
Now the voices of five people are pre-recorded. These recordings act as reference signals. For testing a database of 5 voices is created and their spectrum and cross correlation is plotted and voice is recognized.

Below are the plots. The first graph will represent the test signal that is the voice we want to recognize and second graph is graph of spectrum of voices in database and third represents the correlation graph. Our test signal is of Harshit.

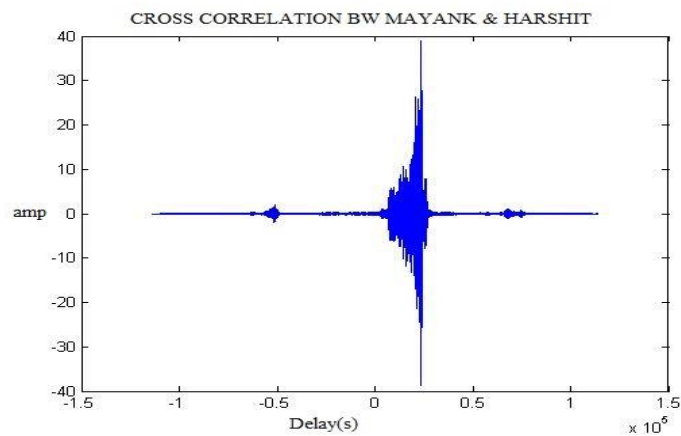
Comparison of spectrums and plots of correlation



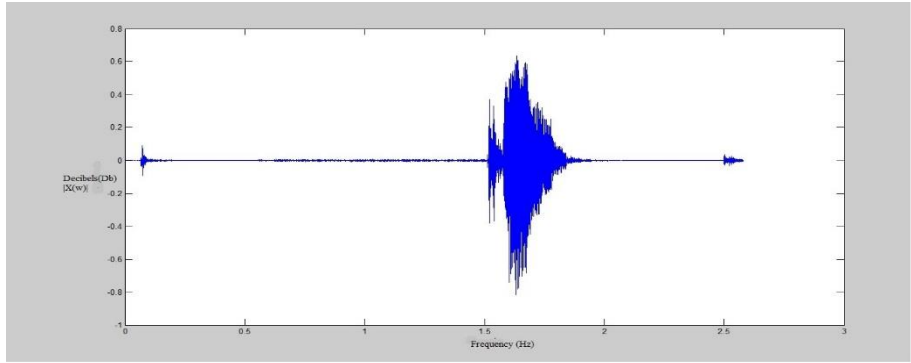
Test signal



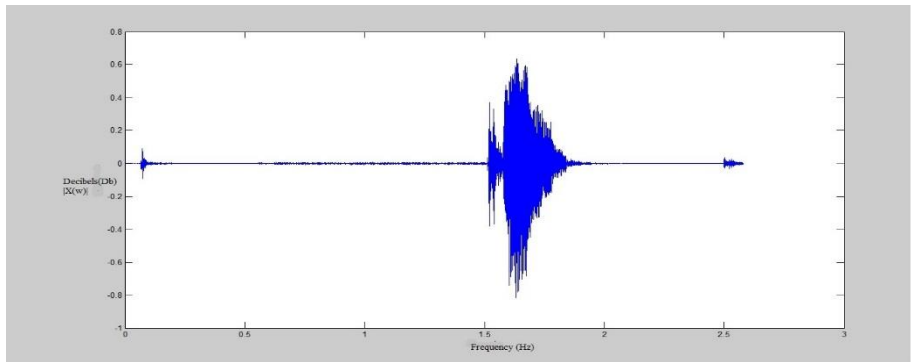
Mayank



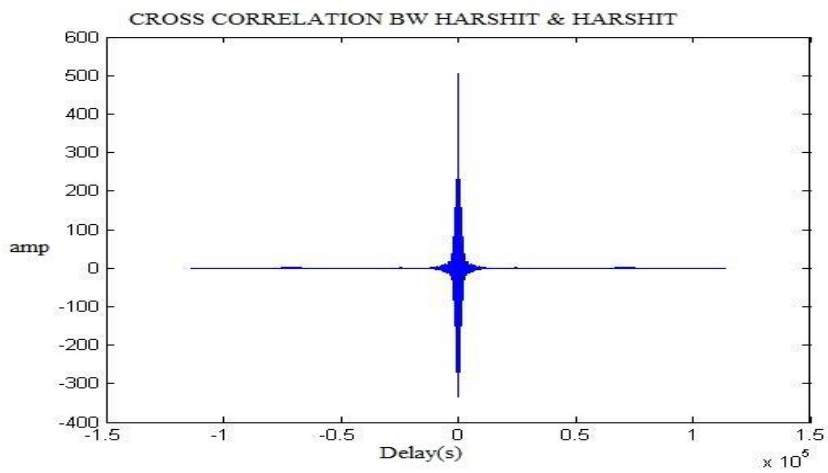
The first two graphs are not identical hence we can observe the value of correlation is not very large hence the test signal is not of Mayank.



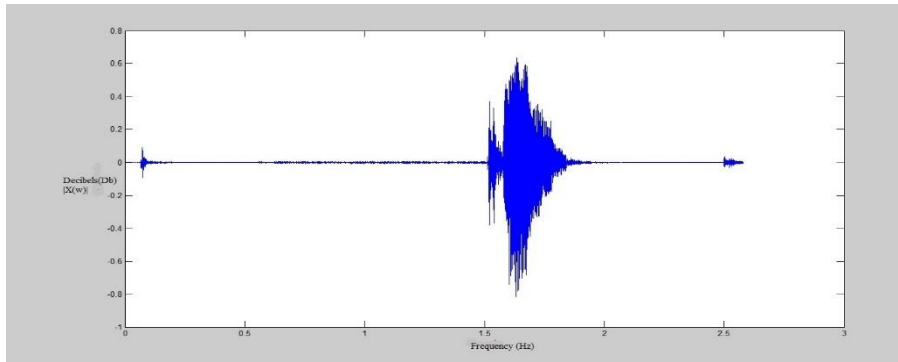
Test Signal



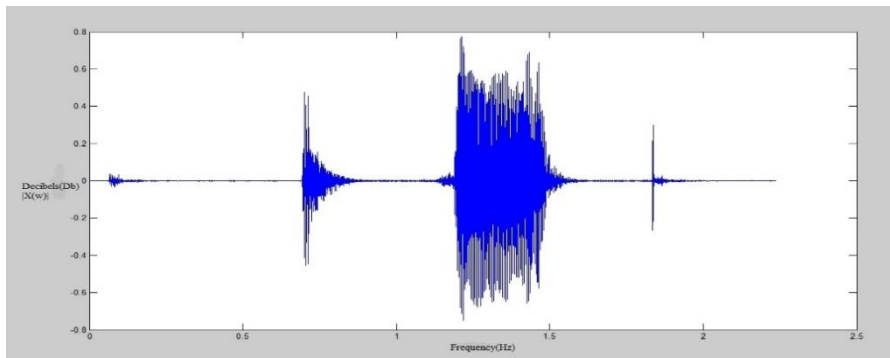
Harshit



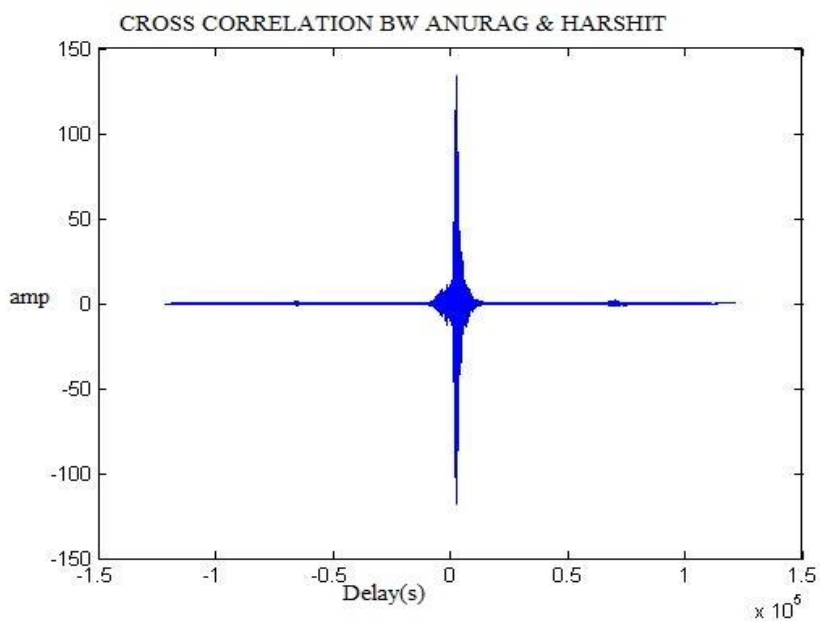
The first two graphs are identical hence we can observe that the value of correlation is very large and probably the maximum hence the test signal is of Harshit.



Test Signal



Anurag



The first two graphs are not identical hence we can observe the value of correlation is not very large hence the test signal is not of Anurag.

After all this recognition algorithm was tested then a GUI was created using MATAB. A dynamic database was created that is user can add any number of voice signals in the database. The GUI has different function buttons which performs different functions. The following figure shows the GUI and its use in recognition of a voice signal. In the dynamic database different voices were recorded and different recognizing ID were given for each person for example

ID 1: Mayank

ID 2: Harshit

And so on.

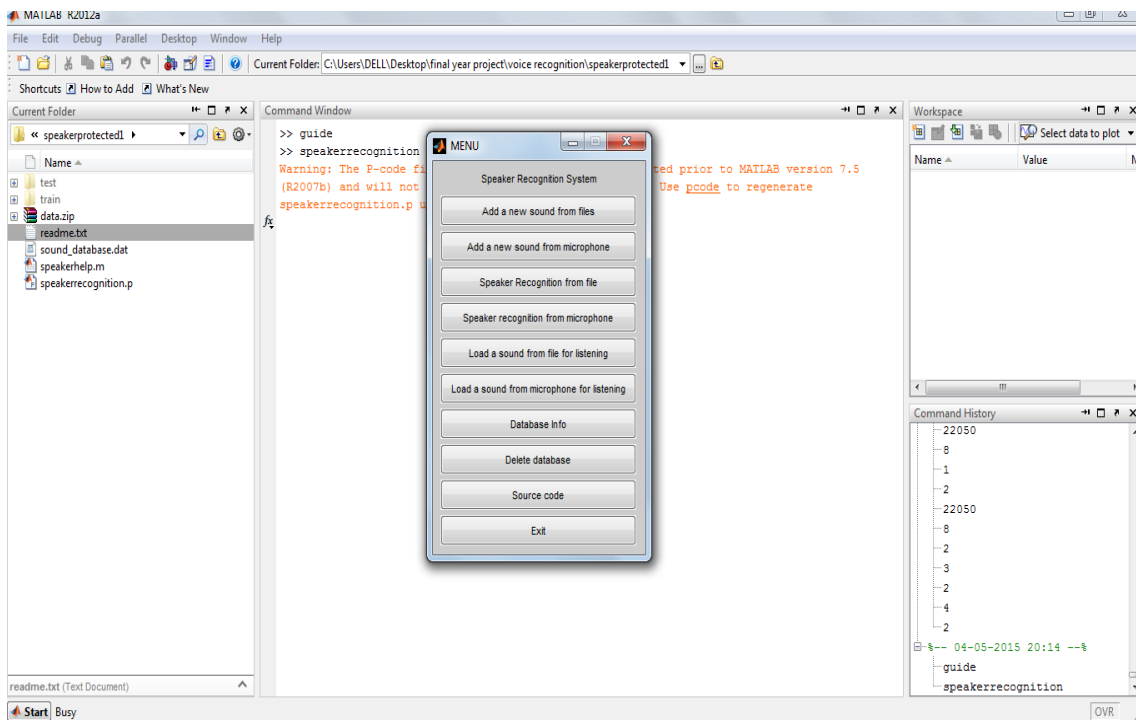


Fig 14 The image of GUI created using MATLAB

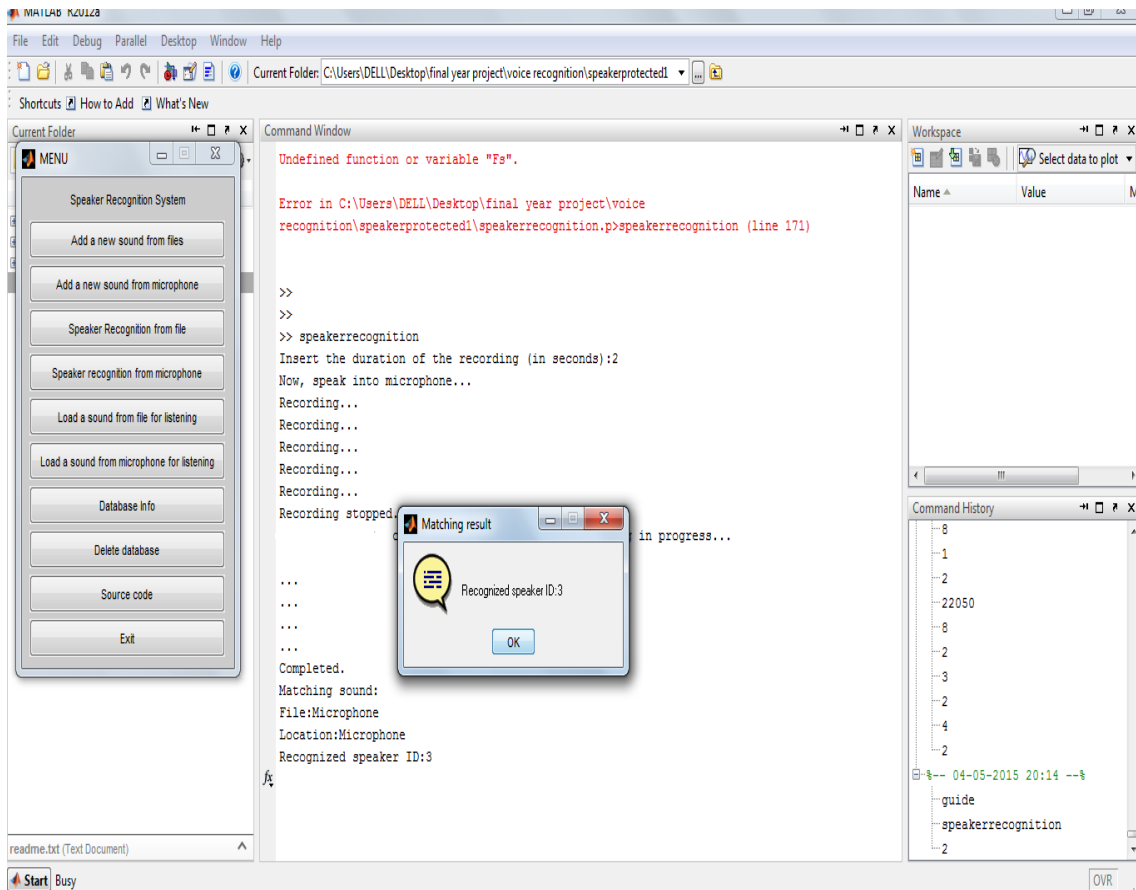


Fig 15. the Output

Finally the GUI created is used to record many voices are in the database and checked for the voice recognition.

Chapter 6

Matlab Codes

THE MAIN CODE FOR BEAMFORMING AND DOA

```
File Edit Text Go Cell Tools Debug Desktop Window Help
1 fs=44100; % Sampling frequency (Hz)
2 d_micro=0.1; % Distance between microphones (m)
3 c=340; % Speed of sound (m/s)
4 muestrasMAX=ceil(d_micro*fs/c); % Maximum number of samples Nmax
5 DESP=ceil(muestrasMAX*1.5); % Delay we insert in the micro 2
6 % We leave 50% of margin of error.
7 lengData=44100*0.5;
8 signal1=wavread('hootie.wav');
9 [road,fs]=wavread('road.wav');
10 size(road);
11 left=road(:,1);
12 right=road(:,2);
13 time=(1/44100)*length(left);
14 t=linspace(0,time,length(left)); %plotting road signal
15 plot(t,left)
16 xlabel('time (sec)');
17 ylabel('relative signal strength') %plotting
18 for k=lengData:lengData:length(signal1)
19 tic;
20 %plot(tic,signal);
21 signal=signal1(1,k-(lengData-1):k); % Signal in MIC B
22 d=signal1(2,k-(lengData-1):k); % Signal in MIC A
23 M1=max(abs(signal)); % Maximum of channel 1
24 M2=max(abs(d)); % Maximum of channel 2
25 M3=max(M1,M2); % Normalization value
26 signal=signal/M3*2; % Normalizing
27 d=d/M3*2;
28 hDESP=[zeros(1,DESP) 1]; % Filter to delay the signal DESP samples.
29 d1=conv(hDESP,d);
30 F=50; % Parameters of the algorithm
31 mu=0.0117;
32 h0=zeros(1,F); h0(1)=0; % Initializing the adaptive filter

speechrecognition123sir.m* Untitled*

File Edit Text Go Cell Tools Debug Desktop Window Help
35 h1(DESP+muestrasMAX+2:length(h1))=0;
36 h1(DESP+1)=h1(DESP+1)/2;
37 [B,I]=sort(h1,'descend');
38 H1=[zeros(1,I(1)-3),h(I(1)-2:I(1)+2),zeros(1,length(h)-(I(1)+2))];
39 lh=128; % Length of the FFT
40 H=fft(h1,lh); % FFT of the filter h(n)
41 alpha=angle(fftshift(H)); % Obtaining the phase
42 q=unwrap(angle(fftshift(H)));
43 M=diff(q); % Obtaining the slope of the phase
44 lM=length(M)+2; % The slope M1 is not a unique value,
45 p1=floor(lM/2-4); % it's an array. So we calculate the
46 p2=ceil(lM/2+4); % average of the values, K.
47 K=mean(M(p1:p2));
48 Nprime=(-K*lh/(2*pi));
49 if Nprime<0 % Two possible cases: negative or positive
50 N=Nprime+lh;
51 N=N-DESP;
52 else
53 N=Nprime;
54 N=N-DESP;
55 end
56
57 % CALLING THE FUNCTION WHICH RETURNS THE ANGLE
58 angleGRAD1=get_angle(N,fs,d_micro);
59 if isreal(angleGRAD1)==1 % Security measures in case
60 %angleGRAD % the number is complex
61 else
62 angleGRAD1=real(angleGRAD1);
63 end
64 timeElapsed=toc; % Time is kept in variable timeElapsed
65 %timeElapsed;
66 end

speechrecognition123sir.m* Untitled*

script Ln 69 Col 1 OVR
```

FUNCTION DEFINITION TO RETURN ANGLE BETWEEN SIGNALS AND TO FIND COORDINATES OF SPEAKER

```

1 FUNCTION DEFINITION TO RETURN ANGLE BETWEEN SIGNALS
2 function [angle]= get_angle(N,fs,d_micro)
3 if N~=0,
4 j=0.1; % Steps
5 x=-20:j:20; % x axis
6 [y1]=hiper(N,x,-d_micro/2,fs); % Calling function hiper
7 x1=round(length(x)/4);x2=round(length(x)/8);
8 pendiente=(y1(x1)-y1(x2))/(j*(x1-x2)); % Slope
9 if N>0
10 angulorad=atan(pendiente);
11 angulo1=angulorad*180/pi;
12 angle=-90-angulo1;
13 else
14 angulorad=atan(-pendiente);
15 angulo1=angulorad*180/pi;
16 angle=90-angulo1;
17 end
18 else
19 angle=0;
20 end
21 end
22
23 TO KNOW THE COORDINATES OF SPEAKER
24 function [y1]= hiper(muestras,x,xA,fs)
25 c=340; % speed of sound (m/sec)
26 pot=2*ones(1,length(x));
27 dist=muestras*c/fs; % distance to B prime
28 y1=sqrt(dist^2/4-xA^2+(4*xA^2/dist^2-1)*x.^pot); % formula with following
29 % requisitions:
30 % xA=-xB; yA=yB=0
31 end
32
speechrecognition123sir.m * Untitled* * Untitled2* *
script Ln 33 Col 1 OVR

```

FUNCTION DEFINITION FOR CALCULATING COEFFICIENT OF FILTER h (n)

```

1 FUNCTION DEFINITION FOR CALCULATING COEFFICIENT OF FILTER h (n)
2
3 function [h,y,e]=f_adap(x,d,h0,mu)
4 h=h0; P=length(h);
5 N=length(x);
6 y=zeros(1,N); e=y; % Reserve space for y[] y e[]
7 rP=0:-1:-P+1;
8 for k=P:N,
9 xx=x(k+rP); % Last P inputs x[k], x[k-1], ... x[k-P]
10 y(k)=xx*h'; % Filter output: x*h Convolution
11 e(k)=d(k)-y(k); % Error
12 h=h+mu*e(k)*xx; % We update the filter coefficients.
13 end
14 end
15

```

MATLAB CODE FOR VOICE RECOGNITION AND SPECTRUM PLOTTING

```

1 clear;
2 on=0;
3 off=0;
4 fs=16000; %sampling frequency, in 1 second take 16000 samples
5 test=0;
6 duration=2; %recording time
7 fprintf('Press any key to start %g seconds of recording.on.\n',duration);
8 pause;
9 fprintf('Recording...\n');
10 r=wavrecord(2*fs,fs);
11 r=r-mean(r);
12 fprintf('Press any key to start %g seconds of recording. off.\n',duration);
13 pause;
14 fprintf('Recording...\n');
15 y=wavrecord(2*fs,fs);
16 fprintf('Press any key to start %g seconds of recording. voice.\n',duration);
17 y=y-mean(y);
18 pause;
19 fprintf('Recording...\n');
20 voice=wavrecord(2*fs,fs);
21 voice=voice-mean(voice);
22 fprintf('Finished recording.\n');
23 nfft = min(1023,length(z)); %define our the length of STFT
24 % define window length and the overlap to decide the window step
25 s=specgram(z, nfft, fs, hanning(511),380);
26 s2=specgram(y, nfft, fs, hanning(511),380);
27 s3=specgram(voice, nfft, fs, hanning(511),380);
28 % PART 1...frequency information
29 % FREQUENCY SPECTRUM
30 absolute=transpose(abs(s)); %take abs make it real and ease to plot
31 absolute2=transpose(abs(s2));
32 absolute3=transpose(abs(s3));
33 a4=sum(absolute); % get time-frequency related spectrum
34 a5=sum(absolute2);
35 a6=sum(absolute3);
36 %normalize
37 a4_norm=(a4-min(a4))/(max(a4)-min(a4));
38 a5_norm=(a5-min(a5))/(max(a5)-min(a5));
39 a6_norm=(a6-min(a6))/(max(a6)-min(a6));
40 %transpose row to columne vector
41 F4=transpose(a4_norm);
42 F5=transpose(a5_norm);
43 F6=transpose(a6_norm);
44 %FREQUENCY SPECTRUM :compare
45 %Crosscorrelation
46 [x3,lag3]=xcorr(F6,F4);
47 [mx3,indice3]=max(x3);
48 frequency_on_shift=lag3(indice3);
49 [x4,lag4]=xcorr(F6,F5);
50 [mx4,indice4]=max(x4);
51 frequency_off_shift=lag4(indice4);
52 % frequency domain spectrum plot
53 subplot(2,3,1)
54 plot(abs(s))
55 subplot(2,3,2)
56 plot(abs(s2))
57 subplot(2,3,3)
58 plot(abs(s3))
59 subplot(2,3,4)
60 plot(F4);
61 subplot(2,3,5)
62 plot(F5);
63 subplot(2,3,6)
64 plot(F6);
65 figure(2)
66 subplot(2,3,1)

```

```

File Edit Text Go Cell Tools Debug Desktop Window Help
- 1.0 + + 1.1 x
66 subplot(1,2,1)
67 plot(x3)
68 title('the xcorr of for on summation');
69 subplot(1,2,2)
70 plot(x4)
71 title('the xcorr of for off summation');
72
73 %checkmig frequency shift which is greater than 2
74 if abs(abs(frequency_on_shift)-abs(frequency_off_shift))>=2
75 if abs(frequency_on_shift)>abs(frequency_off_shift)
76 off=off+1;
77 test=0;
78 else
79 on=on+1;
80 test=1;
81 end
82 %frequency_off_shift test
83 if on>off
84 display('answer is on')
85 else
86 display('answer is off')
87 end
88 else
89 if indice3<length(x3)/2
90 q1=(indice3-1);
91 p1=indice3+length(q):-1:(indice3+1);
92 length(p1);
93 length(q1);
94 x3_left=x3(q1);
95 min(x3_left);
96 x3_right=x3(p1);
97 min(x3_right);
98 error1=mean((abs(x3_right-x3_left)).^2);
99 else
100 q1=frequency_on_shift*2:(indice3-1);
101 p1=length(x3):-1:(indice3+1);
102 length(q1);
103 length(p1);
104 x3_left=x3(q1);
105 x3_right=x3(p1);
106 error1=mean((abs(x3_right-x3_left)).^2);
107 end
108 if indice4<length(x4)/2
109 q2=1:(indice4-1);
110 p2=indice4+length(q2):-1:(indice4+1);
111 x4_left=x4(q2);
112 length(q2);
113 length(p2);
114 min(x4_left);
115 x4_right=x4(p2);
116 min(x4_right);
117 error2=mean((abs(x4_right-x4_left)).^2);
118 else
119 q2=1+frequency_off_shift*2:(indice4-1);
120 p2=length(x4):-1:(indice4+1);
121 length(q2);
122 length(p2);
123 x4_left=x4(q2);
124 x4_right=x4(p2);
125 error2=mean((abs(x4_right-x4_left)).^2);
126 end
127 if @Error1>error2
128 off=off+1;
129 test=0;
130 else
131 on=on+1;
132 test=1;
133 end

```

3. Instance of "Error1" found

print 1 in 0.0 Col 1 FVB

MATLAB CODE FOR CORRELATION ALGORITHM

```

Editor - C:\Users\DELL\Desktop\final year project\voice recognition\speech recogn\speechrecognition123.m
File Edit Text Go Cell Tools Debug Desktop Window Help
- 10 + + 11 x
1 function speechrecognition(filename)
2 %Speech Recognition Using Correlation Method
3 %Write Following Command On Command Window
4 %speechrecognition('test.wav')
5 voice=wavread(filename);
6 x=voice;
7 x=x';
8 x=x(1,:);
9 x=x';
10 y1=wavread('one.wav');
11 y1=y1';
12 y1=y1(1,:);
13 y1=y1';
14 z1=xcorr(x,y1);
15 m1=max(z1);
16 l1=length(z1);
17 t1=-((l1-1)/2):1:(l1-1)/2;
18 t1=t1';
19 %subplot(3,2,1);
20 plot(t1,z1);
21 y2=wavread('two.wav');
22 y2=y2';
23 y2=y2(1,:);
24 y2=y2';
25 z2=xcorr(x,y2);
26 m2=max(z2);
27 l2=length(z2);
28 t2=-((l2-1)/2):1:(l2-1)/2;
29 t2=t2';
30 %subplot(3,2,2);
31 figure
32 plot(t2,z2);
33 y3=wavread('three.wav');
34 y3=y3';

```

```

34 y3=y3';
35 y3=y3(1,:);
36 y3=y3';
37 z3=xcorr(x,y3);
38 m3=max(z3);
39 l3=length(z3);
40 t3=-((l3-1)/2):1:(l3-1)/2;
41 t3=t3';
42 %subplot(3,2,3);
43 figure
44 plot(t3,z3);
45 y4=wavread('four.wav');
46 y4=y4';
47 y4=y4(1,:);
48 y4=y4';
49 z4=xcorr(x,y4);
50 m4=max(z4);
51 l4=length(z4);
52 t4=-((l4-1)/2):1:(l4-1)/2;
53 t4=t4';
54 %subplot(3,2,4);
55 figure
56 plot(t4,z4);
57 y5=wavread('five.wav');
58 y5=y5';
59 y5=y5(1,:);
60 y5=y5';
61 z5=xcorr(x,y5);
62 m5=max(z5);
63 l5=length(z5);
64 t5=-((l5-1)/2):1:(l5-1)/2;
65 t5=t5';
66 %subplot(3,2,5);

```

```

66 %subplot(3,2,5);
67 figure
68 plot(t5,z5);
69 m6=300;
70 a=[m1 m2 m3 m4 m5 m6];
71 m=max(a);
72 h=wavread('allow.wav');
73 if m<=m1
74
75     fprintf('mayank');
76 elseif m<=m2
77
78     fprintf('harshit');
79 elseif m<=m3
80
81     fprintf('anurag');
82 elseif m<=m4
83
84     fprintf('abhishek');
85 elseif m<=m5
86
87     fprintf('akash');
88 else soundsc(wavread('denied.wav'),50000)
89
90 end

```


Conclusion

It was observed that the voice signal is easily disturbed and distorted by the noises present in the surroundings hence it won't be efficient to send the voice signal for feature extraction and recognition. Prior to that a system is designed for reduction of noise signals using beamforming technique. An array of microphone was used with delay and sum method for focussing voice of the desired person. The direction of arrival was calculated and then with help of this the desired voice signal was sent for recognition algorithm. For recognition the correlation algorithm was used. First for testing two voice signals were recorded, their spectrum was plotted and their correlation graph was plotted. After observing the positive result same thing was done for five voice signals. After this a GUI was created using MATLAB code which had function buttons for different functions. A dynamic database was created with the help of this GUI in which the voices of different people can be recorded as many times one needs. Then the GUI was tested by a test signal. It was observed that using beamforming technique increased the efficiency of voice recognition.

For this project the microphone of laptop was used. For future scope the same codes can be used for array of microphones externally.

References

- [1] Kung Yao ,Chris W Reed Daching Chen and Flavio Lorenzelli “Blind Beam forming on random array “1998.
- [2] Ting Xiao Yang, "The Algorithm of Speech recognition & Programming and simulating in MATLAB”
- [3] Aseem Saxena, Amit Kumar Sinha, Shashank Chakrawarti, Surabhi Charu,”Speech Recognition in Matlab”.
- [4] Scott, James; Dragovic, Boris. “Audio location: Accurate Low-Cost Location Sensing”. Intel research Cambridge. Proceedings of the Third International Conference on Pervasive Computing. 2005.
- [5] Carlos Fernández Scola ,María Dolores Bolaños Ortega “Direction of arrival estimation – A two microphones approach.
- [6] Michael L. Seltzer “Microphone Array Processing For Robust Speech Recognition”2001
- [7] Agee, B.G. “The Least Squares CMA: A New Technique for Rapid Correction of Constant Modulus Signals”. Proceedings of the IEEE ICASSP. pgs 19.2.1-19.2.4.
- [8] Debashish Panigrahi, Abhinav Garg & Ravi. S Verma,” A STUDY OF BEAMFORMING TECHNIQUES AND THEIR BLIND APPROACH”
- [9] Pedro J.Merono, “ Speech recognition in Noisy environment”.
- [10] James A Danis, EE, Nicholas J Driscoll, EE, Rebecca J McFarland, CSE, and John M Shattuck, EE, “ The Acoustic Beamformer”.
- [11] Sai Suhas Balabadratruni,” Performance estimation of Direction of arrival estimation using MATLAB”.
- [12] Steven Bell, Nathan West,” Acoustic Beamforming using a TDS3230 DSK”.