REAL TIME SPEAKER RECOGNITION USING MFCC AND VQ

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

In

Telematics and Signal Processing

By

ARUN RAJSEKHAR. G

20607032

Under the Guidance of

PROF G. S. RATH

Department of Electronics & Communication Engineering

National Institute of Technology

Rourkela – 769008.

2008

National Institute of Technology
Rourkela

CERTIFICATE

This is to certify that the Thesis Report entitled ***"Real time speaker recognition using MFCC and VQ"*** submitted by **Mr.** A**run Rajsekhar (20607032)** in partial fulfillment of the requirements for the award of Master of Technology degree in Electronics and Communication Engineering with specialization in **"Telematics & Signal Processing"** during session 2007-2008 at National Institute Of Technology, Rourkela (Deemed University) and is an authentic work 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. G.S.RATH
Dept. of E.C.E
National Institute of Technology
Rourkela-769008

Date: 29-05-2008

Acknowledgement

First of all, I would like to express my deep sense of respect and gratitude towards my advisor and guide **Prof. G. S. Rath**, who has been the guiding force behind this work. I am greatly indebted to him for his constant encouragement, invaluable advice and for propelling me further in every aspect of my academic life. His presence and optimism have provided an invaluable influence on my career and outlook for the future. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

Next, I want to express my respects to **Prof. G. Panda, Prof. K. K. Mahapatra, Prof. S.K. Patra** and **Dr. S. Meher** for teaching me and also helping me how to learn. They have been great sources of inspiration to me and I thank them from the bottom of my heart.

I would like to thank all faculty members and staff of the Department of Electronics and Communication Engineering, N.I.T. Rourkela for their generous help in various ways for the completion of this thesis.

I would also like to mention the names of **madhu** and **pradeep** for helping me a lot during the thesis period.

I would like to thank all my friends and especially my classmates for all the thoughtful and mind stimulating discussions we had, which prompted us to think beyond the obvious. I've enjoyed their companionship so much during my stay at NIT, Rourkela.

I am especially indebted to my parents for their love, sacrifice, and support. They are my first teachers after I came to this world and have set great examples for me about how to live, study, and work.

Arun Rajsekhar. G

Roll No: 20607032

Dept of ECE, NIT, Rourkela

CONTENTS

LIST OF FIGURES

ABSTRACT

Speaker Recognition is a process of automatically recognizing who is speaking on the basis of the individual information included in speech waves. Speaker Recognition is one of the most useful biometric recognition techniques in this world where insecurity is a major threat. Many organizations like banks, institutions, industries etc are currently using this technology for providing greater security to their vast databases.

Speaker Recognition mainly involves two modules namely feature extraction and feature matching. Feature extraction is the process that extracts a small amount of data from the speaker's voice signal that can later be used to represent that speaker. Feature matching involves the actual procedure to identify the unknown speaker by comparing the extracted features from his/her voice input with the ones that are already stored in our speech database.

In feature extraction we find the Mel Frequency Cepstrum Coefficients, which are based on the known variation of the human ear's critical bandwidths with frequency and these, are vector quantized using LBG algorithm resulting in the speaker specific codebook. In feature matching we find the VQ distortion between the input utterance of an unknown speaker and the codebooks stored in our database. Based on this VQ distortion we decide whether to accept/reject the unknown speaker's identity. The system I implemented in my work is 80% accurate in recognizing the correct speaker.

In second phase we implement on the acoustic of Real Time speaker recognition using mfcc and vq on a TMS320C6713 DSP board. We analyze the workload and identify the most time-consuming operations.

Chapter 1

INTRODUCTION

1.1 INTRODUCTION

Speaker recognition is the process of identifying a person on the basis of speech alone. It is a known fact that speech is a speaker dependent feature that enables us to recognize friends over the phone. During the years ahead, it is hoped that speaker recognition will make it possible to verify the identity of persons accessing systems; allow automated control of services by voice, such as banking transactions; and also control the flow of private and confidential data. While fingerprints and retinal scans are more reliable means of identification, speech can be seen as a non-evasive biometric that can be collected with or without the person’s knowledge or even transmitted over long distances via telephone. Unlike other forms of identification, such as passwords or keys, a person's voice cannot be stolen, forgotten or lost.

Speech is a complicated signal produced as a result of several transformations occurring at several different levels: semantic, linguistic, articulatory, and acoustic. Differences in these transformations appear as differences in the acoustic properties of the speech signal. Speaker-related differences are a result of a combination of anatomical differences inherent in the vocal tract and the learned speaking habits of different individuals. In speaker recognition, all these differences can be used to discriminate between speakers.

Speaker recognition allows for a secure method of authenticating speakers. During the enrollment phase, the speaker recognition system generates a speaker model based on the speaker's characteristics. The testing phase of the system involves making a claim on the identity of an unknown speaker using both the trained models and the characteristics of the given speech. Many speaker recognition systems exist and the following chapter will attempt to classify different types of speaker recognition systems.

1.2 MOTIVATION

Let’s say that we have years of audio data recorded everyday using a portable recording device. From this huge amount of data, I want to find all the audio clips of discussions with a specific person. How can I find them? Another example is that a group of people are having a discussion in a video conferencing room. Can I make the camera automatically focus on a specific person (for example, a group leader) whenever he or she speaks even if the other people are also talking? Speaker identification recognition system, which allows us to find a person based on his or her voice, can give us solutions for these questions.

ASV and ASI are probably the most natural and economical methods for solving the problems of unauthorized use of computer and communications systems and multilevel access control. With the ubiquitous telephone network and microphones bundled with computers, the cost of a speaker recognition system might only be for software. Biometric systems automatically recognize a person by using distinguishing traits (a narrow definition). Speaker recognition is a performance biometric, i.e., you perform a task to be recognized. Your voice, like other biometrics, cannot be forgotten or misplaced, unlike knowledge-based (e.g., password) or possession-based (e.g., key) access control methods. Speaker-recognition systems can be made somewhat robust against noise and channel variations, ordinary human changes (e.g., time-of-day voice changes and minor head colds), and mimicry by humans and tape recorders.

1.3 PREVIOUS WORK

There is considerable speaker-recognition activity in industry, national laboratories, and universities. Among those who have researched and designed several generations of speaker-recognition systems are AT&T (and its derivatives); Bolt, Beranek, and Newman [4]; the Dalle Molle Institute for Perceptual Artificial Intelligence (Switzerland); ITT; Massachusetts Institute of Technology Lincoln Labs; National Tsing Hua University (Taiwan); Nagoya University(Japan); Nippon Telegraph and Telephone (Japan);Rensselaer Polytechnic Institute; Rutgers University; and Texas Instruments (TI) [1]. The majority of ASV research is directed at verification over telephone lines. Sandia National Laboratories, the National Institute of Standards and Technology, and the National Security Agency have conducted evaluations of speaker-recognition systems. It should be noted that it is difficult to make meaningful comparisons between the text-dependent and the generally more difficult text-independent tasks. Text-independent approaches, such as Gish's segmental Gaussian model and Reynolds' Gaussian Mixture Model [5], need to deal with unique problems (e.g., sounds or articulations present in the test material but not in training). It is also difficult to compare between the binary choice verification task and the generally more difficult multiple-choice identification task. The general trend shows accuracy improvements over time with larger tests (enabled by larger data bases), thus increasing confidence in the performance measurements. For high-security applications, these speaker-recognition systems would need to be used in combination with other authenticators (e.g., smart card). The performance of current speaker-recognition systems, however, makes them suitable for many practical applications. There are more than a dozen commercial ASV systems, including those from ITT, Lernout & Hauspie, T-NETIX, Veritel, and Voice Control Systems. Perhaps the largest scale deployment of any biometric to date is Sprint's Voice FONCARD, which uses TI's voice verification engine. Speaker-verification applications include access control, telephone banking, and telephone credit cards. The accounting firm of Ernst and Young estimates that high-tech computer thieves in the United States steal $3–5 billion annually. Automatic speaker-recognition technology could substantially reduce this crime by reducing these fraudulent transactions. As automatic speaker-verification systems gain widespread use, it is imperative to understand the errors made by these systems. There are two types of errors: the false acceptance of an invalid user (FA or Type I) and the false rejection of a valid user (FR or Type II). It takes a pair of subjects to make a false acceptance error: an impostor and a target. Because of this hunter and prey relationship, in this paper, the impostor is referred to as a wolf and the target as a sheep. False acceptance errors are the ultimate concern of high-security speaker-verification applications; however, they can be traded off for false rejection errors. After reviewing the methods of speaker recognition, a simple speaker-recognition system will be presented. A data base of 186 people collected over a three-month period was used in closed-set speaker identification experiments [1]. A speaker-recognition system using methods presented here is practical to implement in software on a modest personal computer. The features and measures use long-term statistics based upon an information-theoretic shape measure between line spectrum pair (LSP) frequency features. This new measure, the *divergence shape*, can be interpreted geometrically as the shape of an information-theoretic measure called divergence. The LSP's were found to be very effective features in this divergence shape measure. The following chapter contains an overview of digital signal acquisition, speech production, speech signal processing, and Mel cepstrum [2].

1.4 THESIS CONTRIBUTION

The design is first tested with MATLAB. A total of eight speech samples from eight different people (eight speakers, labeled S1 to S8) are used to test this project. Each speaker utters the same single digit, *zero*, once in a training session (then also in a testing session). A digit is often used for testing in speaker recognition systems because of its applicability to many security applications. This project was implemented on the C6711 DSK and can be transported to the C6713 DSK. Of the eight speakers, the system identified six correctly (a 75% identification rate). The identification rate can be improved by adding more vectors to the training codeword's. The performance of the system may be improved by using two-dimensional or four dimensional VQ (training header file would be 8 ¥ 20 ¥ 4) or by changing the quantization method to dynamic time wrapping or hidden Markov modeling.

1.5 OUTLINE OF THESIS

The purpose of this introductory section is to present a general framework and motivation for speaker recognition, an overview of the entire paper, and a presentation of previous work in speaker recognition.

Chapter 2 contains different biometric techniques available in present day industry, introduction to speaker recognition, performance measures of a biometric system and classification of automatic speaker recognition system

Chapter 3 contains the different stages of speech feature extraction which are Frame blocking, Windowing, FFT, Mel-frequency wrapping and the cepstrum from the Mel-frequency wrapped spectrum which are the MFCC's of the speaker.

Chapter 4 contains an introduction to Vector Quantization, Linde Buzo and Gray algorithm for VQ, and formation of a speaker specific codebook by using LBG VQ algorithm on the MFCC's obtained in the previous section.

Chapter 5 explains the speech feature matching and calculation of the Euclidean distance between the codebooks of each speaker.

Chapter 6 explains the DSP processor **TMS320C6013** kit.

Chapter 7 contains the results I got and the plots in this chapter clearly explain the distance between Vector Quantized MFCC's of each speaker. Contains the results in DSP kit and each program time calculated.

Then I made a conclusion to my work and the points to possible directions for future work.

Chapter 2

INTRODUCTION TO SPEAKER RECOGNITION

2.1 INTRODUCTION

Speaker identification [6] is one of the two categories of speaker recognition, with speaker verification being the other one. The main difference between the two categories will now be explained. Speaker verification performs a binary decision consisting of determining whether the person speaking is in fact the person he/she claims to be or in other words verifying their identity. Speaker identification performs multiple decisions and consists comparing the voice of the person speaking to a database of reference templates in an attempt to identify the speaker. Speaker identification will be the focus of the research in this case.

Speaker identification further divides into two subcategories, which are text dependent and text-independent speaker identification [10]. Text-dependent speaker identification differs from text-independent because in the aforementioned the identification is performed on a voiced instance of a specific word, whereas in the latter the speaker can say anything. The thesis will consider only the text-dependent speaker identification category.

The field of speaker recognition has been growing in popularity for various applications. Embedding recognition in a product allows a unique level of hands-free and intuitive user interaction. Popular applications include automated dictation and command interfaces. The various phases of the project lead to an in-depth understanding of the theory and implementation issues of speaker recognition, while becoming more involved with the speaker recognition community. Speaker recognition uses the technology of biometrics.

2.2 BIOMETRICS

Biometric techniques based on intrinsic characteristics (such as voice, finger prints, retinal patterns) [17] have an advantage over artifacts for identification (keys, cards, passwords) because biometric attributes cannot be lost or forgotten as these are based on his/her physiological or behavioral characteristics. Biometric techniques are generally believed to offer a reliable method of identification, since all people are physically different to some degree. This does not include any passwords or PIN numbers which are likely to be forgotten or forged. Various types of biometric systems are in vogue.

A biometric system is essentially a pattern recognition system, which makes a personal identification by determining the authenticity of a specific physiological or behavioral characteristics possessed by the user. An important issue in designing a practical system is to determine how an individual is identified. A biometric system can be either an identification system or a verification system. Some of the biometric security systems are:

- **Fingerprints**
- **Eye Patterns**
- **Signature Dynamics**
- **Keystroke Dynamics**
- **Facial Features**
- **Speaker Recognition**

Fingerprints

The stability and uniqueness of the fingerprint are well established. Upon careful examination, it is estimated that the chance of two people, including twins, having the same print is less than one in a billion. Many devices on the market today analyze the position of tiny points called minutiae, the end points and junctions of print ridges. The devices assign locations to the minutiae using x, y and directional variables. Another technique counts the number of ridges between points. Several devices in development claim they will have templates of fewer than 100 bytes depending on the application. Other machines approach the finger as an image-processing problem. The fingerprint requires one of the largest data templates in the biometric field, ranging from several hundred bytes to over 1,000 bytes depending on the approach and security level required; however, compression algorithms enable even large templates fit into small packages.

Eye Patterns

Both the pattern of flecks on the iris and the blood vessel pattern on the back of the eye (retina) provide unique bases for identification. The technique's major advantage over retina scans is that it does not require the user to focus on a target, because the iris pattern is on the eye's surface. In fact, the video image of an eye can be taken from several up to 3 feet away, and the user does not have to interact actively with the device.

Retina scans are performed by directing a low-intensity infrared light through the pupil and to the back part of the eye. The retinal pattern is reflected back to a camera, which captures the unique pattern and represents it using less than 35 bytes of information. Most installations to date have involved high-security access control, including numerous military and bank facilities. Retina scans continue to be one of the best biometric performers on the market with small data template, and quick identity confirmations. The toughest hurdle for the technologies continues to be user resistance.

Signature Dynamics

The key in signature dynamics is to differentiate between the parts of the signature that are habitual and those that vary with almost every signing. Several devices also factor the static image of the signature, and some can capture a static image of the signature for records or reproduction. In fact, static signature capture is becoming quite popular for replacing pen and paper signing in bankcard, PC and delivery service applications. Generally, verification devices use wired pens, sensitive tablets or a combination of both. Devices using wired pens are less expensive and take up less room but are potentially less durable. To date, the financial community has been slow in adopting automated signature verification methods for credit cards and check applications, because they demand very low false rejection rates. Therefore, vendors have turned their attention to computer access and physical security. Anywhere a signature used is already a candidate for automated biometrics.

Keystroke Dynamics

Keystroke dynamics, also called typing rhythms, is one of the most eagerly waited of all biometric technologies in the computer security arena. As the name implies, this method analyzes the way a user types at a terminal by monitoring the keyboard input 1,000 times per second. The analogy is made to the days of telegraph when operators would identify each other by recognizing "the fist of the sender." The modern system has some similarities, most notably which the user does not realize he is being identified unless told. Also, the better the user is at typing, the easier it is to make the identification. The advantages of keystroke dynamics in the computer environment are obvious. Neither enrollment nor verification detracts from the regular workflow, because the user would be entering keystrokes anyway. Since the input device is the existing keyboard, the technology costs less. Keystroke dynamics also can come in the form of a plug-in board, built-in hardware and firmware or software.

Still, technical difficulties abound in making the technology work as promised, and half a dozen efforts at commercial technology have failed. Differences in keyboards, even of the same brand, and communications protocol structures are challenging hurdles for developers.

Facial Features

One of the fastest growing areas of the biometric industry in terms of new development efforts is facial verification and recognition. The appeal of facial recognition is obvious. It is the method most akin to the way that we, as humans identify people and the facial image can be captured from several meters away using today's video equipment. But most developers have had difficulty achieving high levels of performance when database sizes increase into the tens of thousands or higher. Still, interest from government agencies and even the financial sector is high, stimulating the high level of development efforts.

Speaker Recognition

Speaker recognition is the process of automatically recognizing who is speaking on the basis of individual information included in speech waves. It has two sessions. The first one is referred to the enrollment session or training phase while the second one is referred to as the operation session or testing phase. In the training phase, each registered speaker has to provide samples of their speech so that the system can build or train a reference model for that speaker. In case of speaker verification systems, in addition, a speaker-specific threshold is also computed from the training samples. During the testing (operational) phase, the input speech is matched with stored reference model(s) and recognition decision is made This technique makes it possible to use the speaker's voice to verify their identity and control access to services such as voice dialing, banking by telephone, telephone shopping, database access services, information services, voice mail, security control for confidential information areas, and remote access to computers.

Among the above, the most popular biometric system is the speaker recognition system because of its easy implementation and economical hardware.

2.3 STRUCTURE OF THE INDUSTRY

Segmentation of the biometric industry can first be done by technology, then by the vertical markets and finally by applications these technologies serve. Generally, the industry can be segmented, at the very highest level, into two categories: physiological and behavioral biometric technologies. Physiological characteristics, as stated before, include those that do not change dramatically over time. Behavioral biometrics, on the other hand, change over time and some times on a daily basis. The following chart (Chart 1) depicts the easiest way to segment the biometrics industry. In each of the technology segments, the number of companies competing in that segment has been noted.

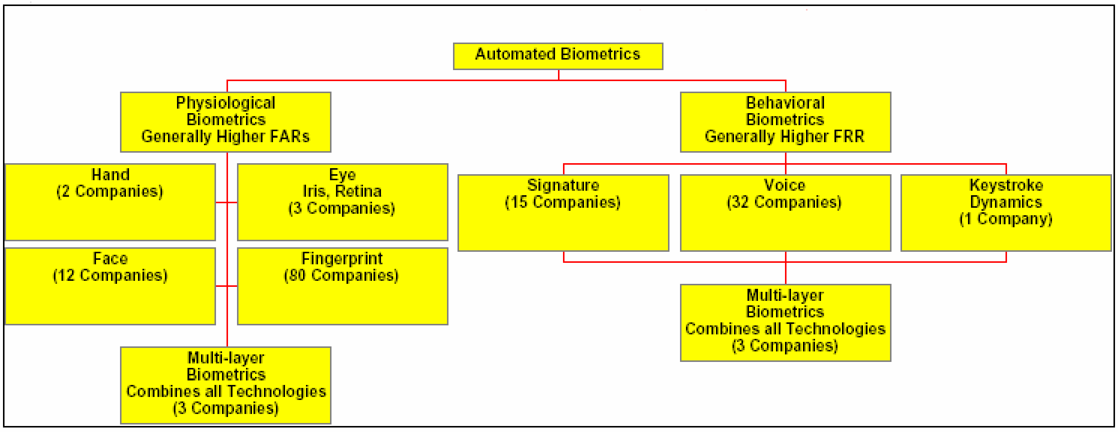


Figure 2.1. Proportionate usage of available biometric techniques in Industry

2.4 PERFORMANCE MEASURES

The most commonly discussed performance measure of a biometric is its Identifying Power. The terms that define ID Power are a slippery pair known as False Rejection Rate (FRR), or Type I Error, and False Acceptance Rate (FAR) [1], or Type II Error. Many machines have a variable threshold to set the desired balance of FAR and FRR. If this tolerance setting is tightened to make it harder for impostors to gain access, it also will become harder for authorized people to gain access (i.e., as FAR goes down, FRR rises). Conversely, if it is very easy for rightful people to gain access, then it will be more likely that an impostor may slip though (i.e., as FRR goes down, FAR rises).

2.5 CLASSIFICATION OF AUTOMATIC SPEAKER RECOGNITION

Speaker recognition is the process of automatically recognizing who is speaking on the basis of individual information included in speech waves. This technique makes it possible to use the speaker's voice to verify their identity and control access to services such as voice dialing, banking by telephone, telephone shopping, database access services, information services, voice mail, security control for confidential information areas, and remote access to computers.

Automatic speaker identification and verification are often considered to be the most natural and economical methods for avoiding unauthorized access to physical locations or computer systems. Thanks to the low cost of microphones and the universal telephone network, the only cost for a speaker recognition system may be the software. The problem of speaker recognition is one that is rooted in the study of the speech signal. A very interesting problem is the analysis of the speech signal, and therein what characteristics make it unique among other signals and what makes one speech signal different from another.

When an individual recognizes the voice of someone familiar, he/she is able to match the speaker's name to his/her voice. This process is called speaker identification, and we do it all the time. Speaker identification exists in the realm of speaker recognition, which encompasses both identification and verification of speakers. Speaker verification is the subject of validating whether or not a user is who he/she claims to be. To have a simple example, verification is *Am I the person whom I claim I am?* Whereas identification is *who am I?*

This section covers the speaker recognition systems (see Fig. 1.1), their differences and how the performances of such systems are accessed. Automatic speaker recognition systems can be divided into two classes depending on their desired function; Automatic Speaker Identification (ASI) classification of

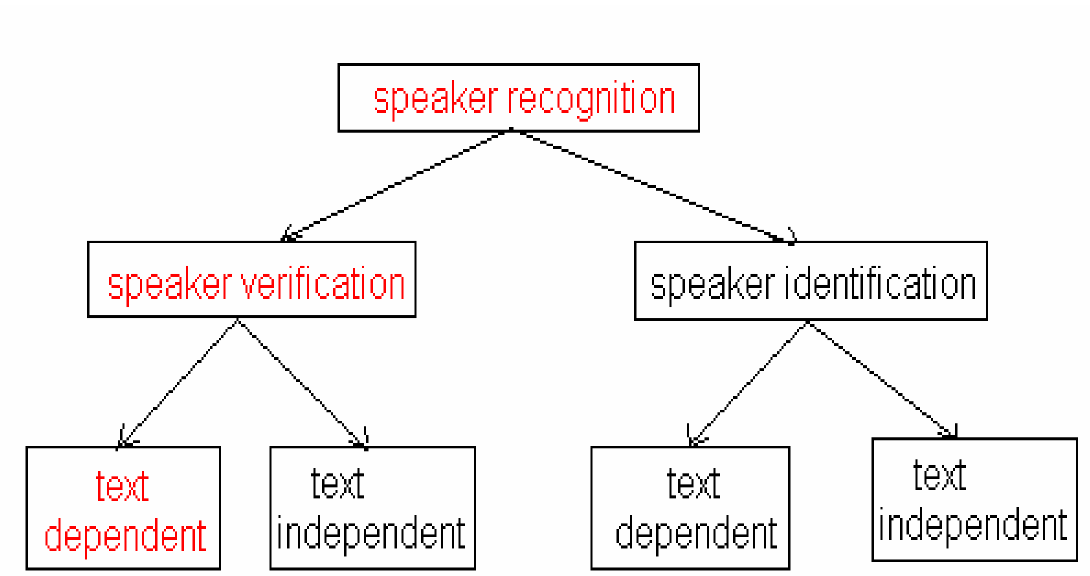


Figure 2.2. The Scope of Speaker Recognition

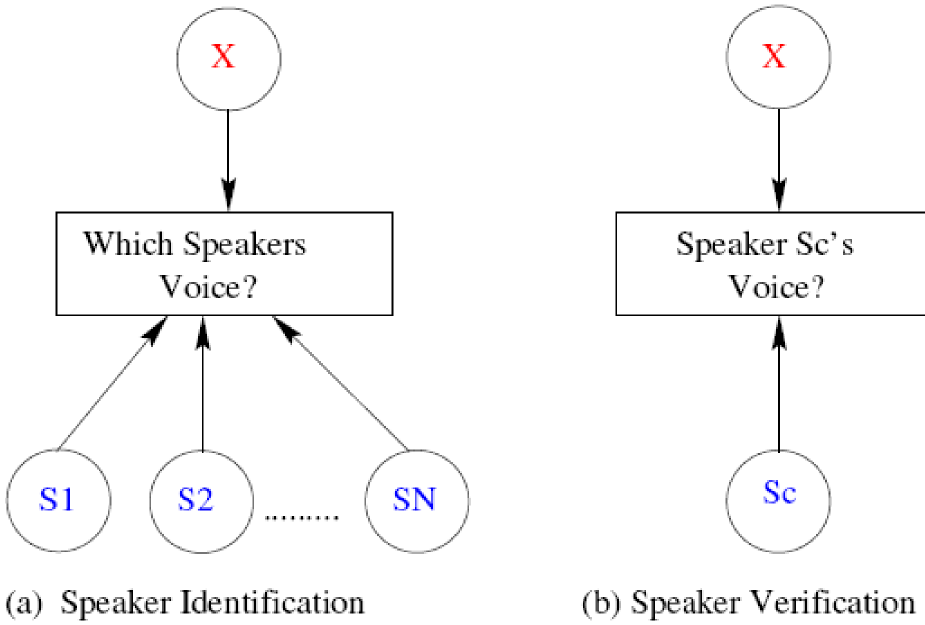
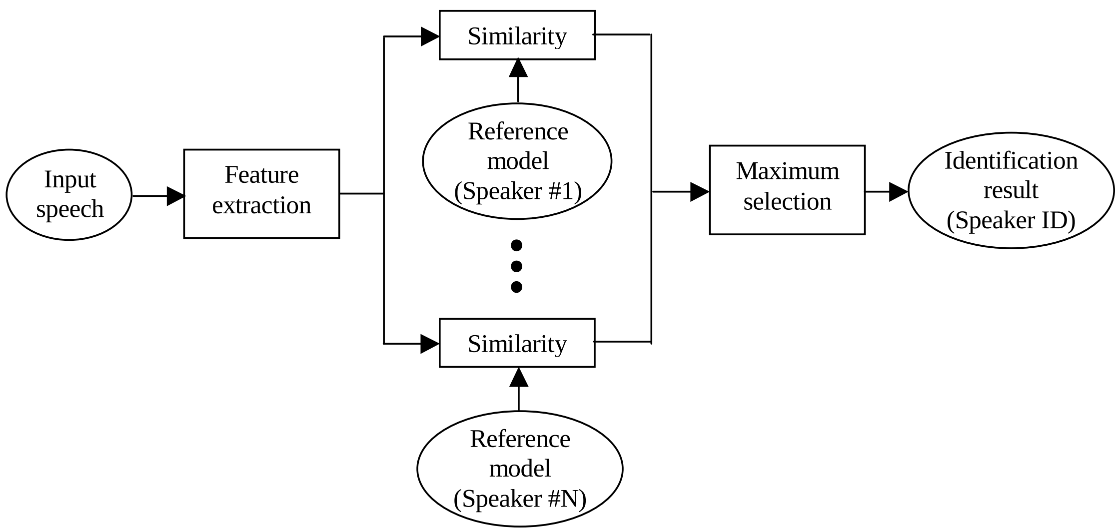
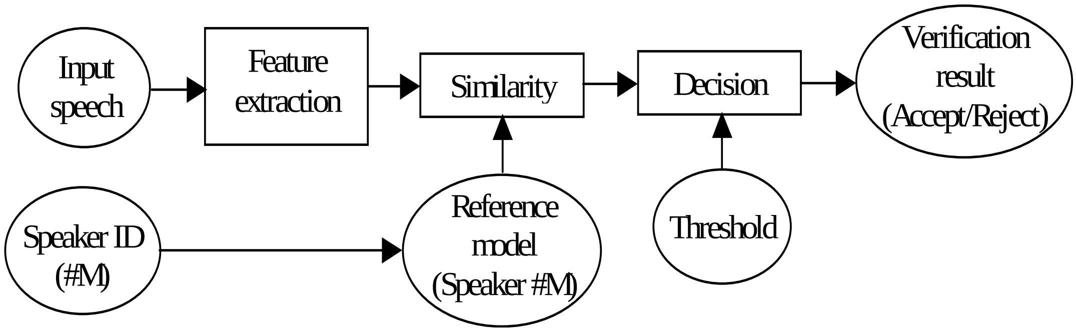


Figure 2.3. Speaker Identification and Speaker Verification.

In this report, I pursue a speaker recognition system, so I will abandon discussion of the other topics.



(a) Speaker identification



(b) Speaker verification

Figure 2.4. Basic structure of speaker recognition systems

The goal of this project is to build a simple, yet complete and representative automatic speaker recognition system. The vocabulary of digit is used very often in testing speaker recognition because of its applicability to many security applications. For example, users have to speak a PIN (Personal Identification Number) in order to gain access to the laboratory door, or users have to speak their credit card number over the telephone line. By checking the voice characteristics of the input utterance using an automatic speaker recognition system similar to the one I will develop, the system is able to add an extra level of security.

Speaker recognition methods can be divided into text independent and text dependent methods. In a text independent system, speaker models capture characteristics of somebody's speech which show up irrespective of what one is saying. This system should be intelligent enough to capture the characteristics of all the words that the speaker can use. On the other hand in a text dependent system, the recognition of the speaker's identity is based on his/her speaking one or more specific phrases like passwords, card numbers, PIN codes etc.

This project involves two modules namely *feature extraction* and *feature matching*. Feature extraction is the process that extracts a small amount of data from the voice signal that can be used to represent each speaker. Feature matching involves that actual procedure to identify the unknown speaker by comparing extracted features from his/her voice input with the ones from a set of known speakers.

Feature extraction involves finding MFCCs of the speech and vector quantizing them to obtain the speaker specific codebook. For this, I use short time spectral analysis, FFT, Windowing, Mel Spaced Filter Banks and convert the speech signal to a parametric representation. i.e. to Mel Frequency Cepstrum Coefficients. These MFCCs are based on the known variation of the human ear’s critical bandwidths with frequency i.e. linear at low frequencies and logarithmic at high frequencies. These are less susceptible to the variation in speaker’s voice. Vector quantization is the process of mapping vectors from a large vector space to a finite number of regions in that space. Each region is called a cluster and can be represented by using its centroid. Centroids of all clusters are combined to form the speaker specific codebook.

In feature matching the input utterance of an unknown speaker is converted into MFCCs and then the total VQ distortion between these MFCCs and the codebooks stored in our database is measured. VQ distortion is the distance from a vector to the closest code word of a codebook. Based on this VQ distortion we decide whether the speaker is a valid person or an impostor. i.e. if the VQ distortion is less than the threshold value then the speaker is a valid person and if it exceeds the threshold value then he is considered as an impostor. This system is at its best roughly 80% accurate in identifying the correct speaker.

2.6 SUMMARY

Explained different biometric techniques available in present day industry, made an introduction to speaker recognition, explained the performance measures of a biometric system and classification of automatic speaker recognition system.

Chapter 3

SPEECH FEATURE EXTRACTION

3.1 INTRODUCTION

The purpose of this module is to convert the speech waveform to some type of parametric representation for further analysis and processing. This is often referred to as the *signal-processing front end*.

The speech signal is a slowly time varying signal. An example of speech signal is shown in Figure 2. When examined over a sufficiently short period of time (between 5 and 100 msec), its characteristics are fairly stationary. However, over longer periods of time (on the order of 1/5 seconds or more) the signal characteristics change to reflect the different speech sounds being spoken. Therefore, *short-time spectral analysis* is the most common way to characterize the speech signal [17].

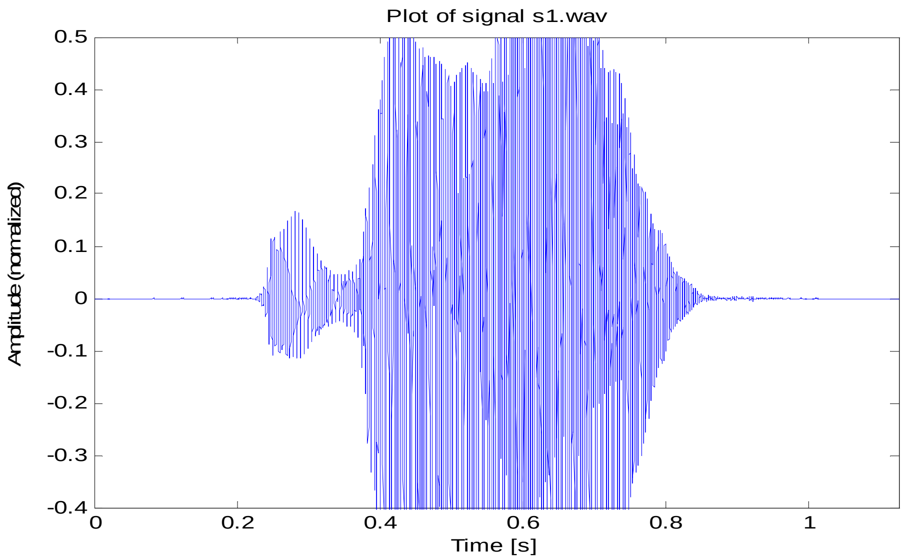


Figure 3.1. An example of speech signal

A wide range of possibilities exist for parametrically representing the speech signal for the speaker recognition task, such as Linear Prediction Coding (LPC), Mel-Frequency Cepstrum Coefficients (MFCC), and others. MFCC is perhaps the best known and most popular, and this is used in this project.

The LPC [10] features were very popular in the early speaker-identification and speaker-verification systems. However, comparison of two LPC feature vectors requires the use of computationally expensive similarity measures such as the Itakura-Saito distance and hence LPC features are unsuitable for use in real-time systems. Furui suggested the use of the

Cepstrum, defined as the inverse Fourier transform of the logarithm of the magnitude spectrum, in speech-recognition applications. The use of the cepstrum allows for the similarity between two cepstral feature vectors to be computed as a simple Euclidean distance. Furthermore, Ata has demonstrated that the cepstrum derived from the MFCC features rather than LPC features results in the best performance in terms of FAR [False Acceptance Ratio] and FRR [False Rejection Ratio] for a speaker recognition system. Consequently, I have decided to use the MFCC derived cepstrum for our speaker recognition system.

MFCCs are based on the known variation of the human ear's critical bandwidths with frequency, filters spaced linearly at low frequencies and logarithmically at high frequencies have been used to capture the phonetically important characteristics of speech. This is expressed in the *mel-frequency* scale, which is linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. Here the mel scale is being used which translates regular frequencies to a scale that is more appropriate for speech, since the human ear perceives sound in a nonlinear manner. This is useful since our whole understanding of speech is through our ears, and so the computer should know about this, too. Feature Extraction is done using MFCC processor

3.2 MEL-FREQUENCY CEPSTRUM COEFFICIENTS PROCESSOR

A block diagram of the structure of an MFCC processor is given in Figure 3.1. The speech input is typically recorded at a sampling rate above 12500 Hz. This sampling frequency was chosen to minimize the effects of *aliasing* [18] in the analog-to-digital conversion.

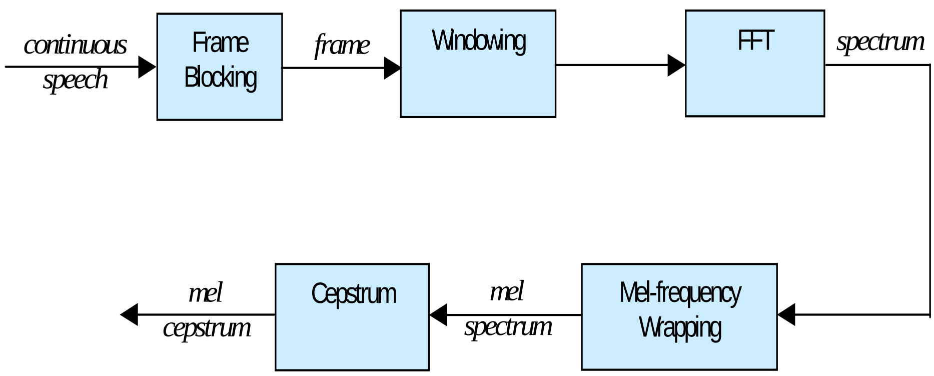


Figure 3.2. Block diagram of the MFCC processor

3.2.1 Frame Blocking

In this step, the continuous speech signal is blocked into frames of *N* samples, with adjacent frames being separated by *M* (*M* < *N*). The first frame consists of the first *N* samples. The second frame begins *M* samples after the first frame, and overlaps it by *N* - *M* samples. Similarly, the third frame begins 2*M* samples after the first frame (or *M* samples after the second frame) and overlaps it by *N* - 2*M* samples. This process continues until all the speech is accounted for within one or more frames [18].

The values for *N* and *M* are taken as *N* = 256 (which is equivalent to ~ 30 msec windowing and facilitate the fast radix-2 FFT) and *M* = 100. Frame blocking of the speech signal is done because when examined over a sufficiently short period of time (between 5 and 100 msec), its characteristics are fairly stationary. However, over long periods of time (on the order of 1/5 seconds or more) the signal characteristic change to reflect the different speech sounds being spoken. Overlapping frames are taken not to have much information loss and to maintain correlation between the adjacent frames.

N value 256 is taken as a compromise between the time resolution and frequency resolution. One can observe these time and frequency resolutions by viewing the corresponding power spectrum of speech files which was shown in the figure 3.2. In each case, frame increment M is taken as N/3.

For N = 128 we have a high resolution of time. Furthermore each frame lasts for a very short period of time. This result shows that the signal for a frame doesn't change its nature. On the other hand, there are only 65 distinct frequencies samples. This means that we have a poor frequency resolution.

For N = 512 we have an excellent frequency resolution (256 different values) but there are lesser frames, meaning that the resolution in time is strongly reduced.

It seems that a value of 256 for N is an acceptable compromise. Furthermore the number of frames is relatively small, which will reduce computing time.

So, finally for N = 256 we have a compromise between the resolution in time and the resolution in frequency.

3.2.2 Windowing

The next step in the processing is to window each individual frame so as to minimize the signal discontinuities at the beginning and end of each frame. The concept here is to minimize the spectral distortion by using the window to taper the signal to zero at the beginning and end of each frame. If we define the window as $w(n),\ 0 \le n \le N-1$, where N is the number of samples in each frame, then the result of windowing is the signal

$$y_l(n) = x_l(n)w(n), \quad 0 \le n \le N-1$$

Typically the *Hamming* window is used, which has the form and plot is given in

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \quad 0 \le n \le N-1$$

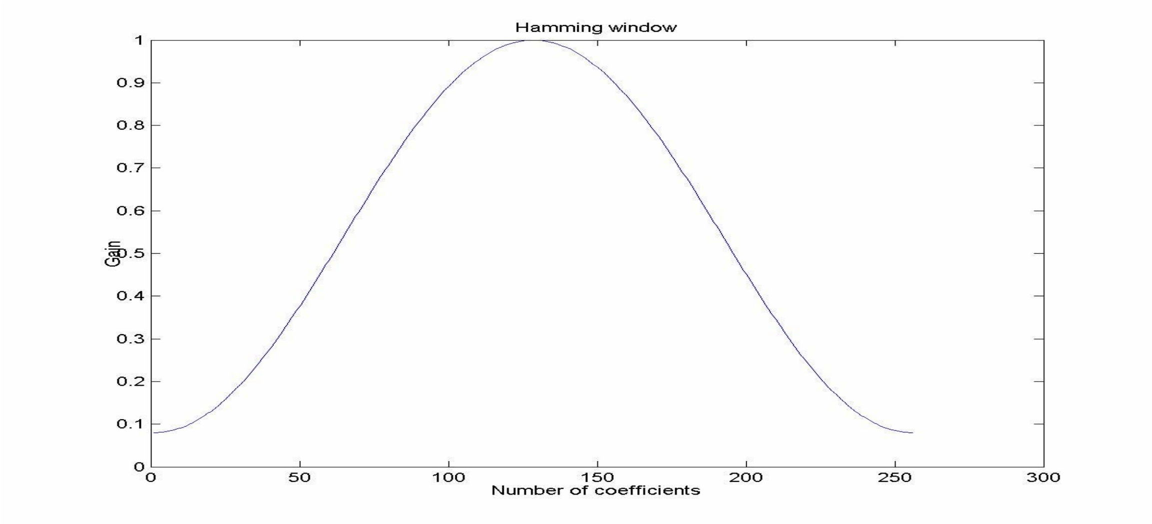


Figure 3.3: Hamming window

3.2.3. Fast Fourier Transform (FFT)

The next processing step is the Fast Fourier Transform, which converts each frame of N samples from the time domain into the frequency domain. These algorithms are popularized by Cooley and Tukey [18] and are based on decomposing and breaking the transform into smaller transforms and combining them to give the total transform. FFT reduces the computation time required to compute a discrete Fourier transform and improves the performance by a factor of 100 or more over direct evaluation of the DFT. FFT reduces the number of complex multiplications from N^2 to $(N/2)\log_2 N$ and it's speed improvement factor is $N^2/(N/2)\ Log_2N$). In other words FFT is a fast algorithm to implement the Discrete Fourier Transform (DFT) which is defined on the set of N samples $\{x_n\}$, as follow:

$$X_n = \sum_{k=0}^{N-1} x_k e^{-2\pi jkn/N}, \qquad n = 0,1,2,\ldots,N-1$$

We use j here to denote the imaginary unit, i.e. $j = \sqrt{-1}$. In general X_n's are complex numbers. The resulting sequence $\{X_n\}$ is interpreted as follows: the zero frequency corresponds to $n = 0$, positive frequencies $0 < f < F_s/2$ correspond to values $1 \le n \le N/2-1$, while negative frequencies $-F_s/2 < f < 0$ correspond to $N/2+1 \le n \le N-1$. Here, F_s denote the sampling frequency.

The result after this step is often referred to as *spectrum* or *periodogram*.

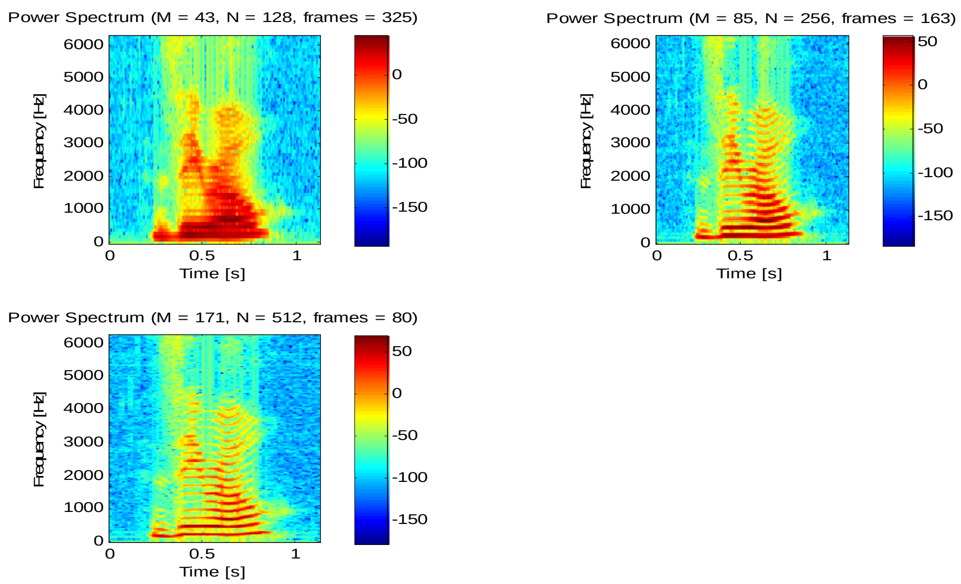


Figure 3.4 Power spectrums of speech files for different M and N values

3.2.4 Mel-frequency Wrapping

As mentioned above, psychophysical studies have shown that human perception of the frequency contents of sounds for speech signals does not follow a linear scale. Thus for each tone with an actual frequency, *f*, measured in Hz, a subjective pitch is measured on a scale called the 'mel' scale. The *mel-frequency* scale is linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. As a reference point, the pitch of a 1 kHz tone, 40 dB above the perceptual hearing threshold, is defined as 1000 mels [1][2]. Therefore we can use the following approximate formula to compute the mels for a given frequency *f* in Hz:

$$mel(f) = 2595 * \log_{10}(1 + f / 700)$$

One approach to simulating the subjective spectrum is to use a filter bank, spaced uniformly on the mel scale (see Figure 4). That filter bank has a triangular bandpass frequency response, and the spacing as well as the bandwidth is determined by a constant mel frequency interval. The modified spectrum of $S(\omega)$ thus consists of the output power of these filters when $S(\omega)$ is the input. The number of mel spectrum coefficients, K, is typically chosen as 20.

This filter bank is applied in the frequency domain, therefore it simply amounts to taking those triangle-shape windows in the Figure 3.4 on the spectrum. A useful way of thinking about this mel-wrapping filter bank is to view each filter as an histogram bin (where bins have overlap) in the frequency domain.

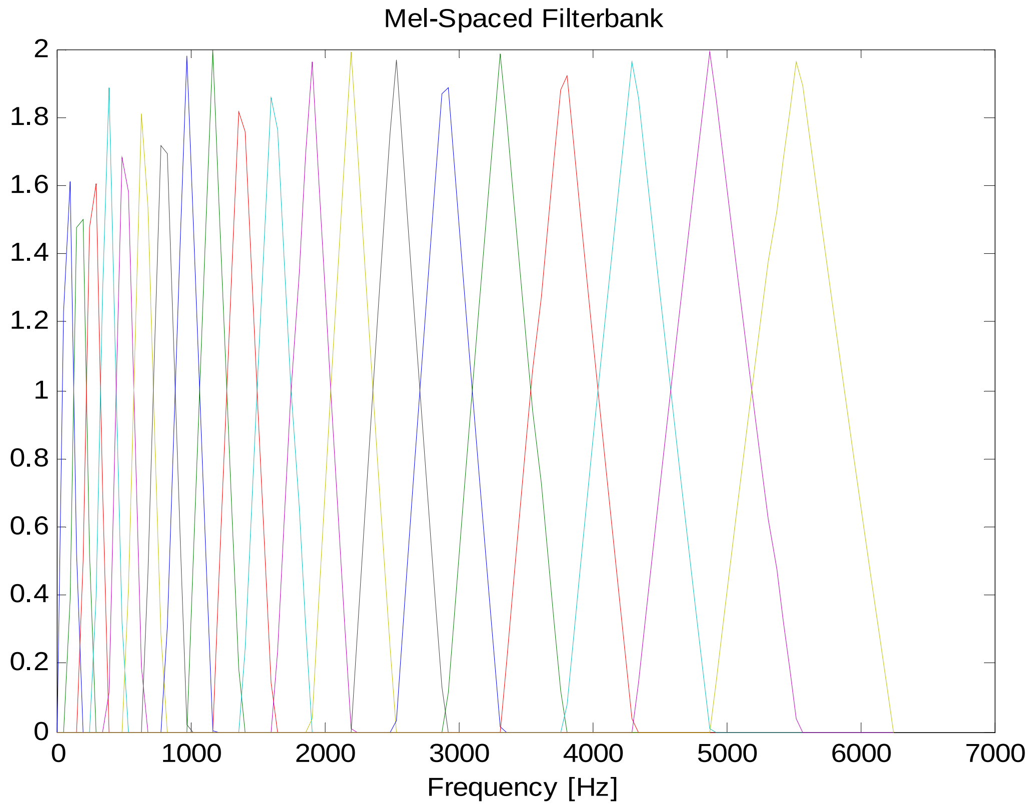


Figure 3.5. An example of mel-spaced filterbank for 20 filters

3.2.5 Cepstrum

As per the National Instruments, Cepstrum is defined as the Fourier transform of the logarithm of the autospectrum. It is the inverse Fourier transform of the logarithm of the power spectrum of a signal. It is useful for determining periodicities in the autospectrum. Additions in the Cepstrum domain correspond to multiplication in the frequency domain and convolution in the time domain. The Cepstrum is the Forward Fourier Transform of a spectrum. It is thus the spectrum of a spectrum, and has certain properties that make it useful in many types of signal analysis [3]. One of its most powerful attributes is the fact that any periodicities, or repeated patterns, in a spectrum will be sensed as one or two specific components in the Cepstrum. If a spectrum contains several sets of sidebands or harmonic series, they can be confusing because of overlap. But in the Cepstrum, they will be separated in a way similar to the way the spectrum separates repetitive time patterns in the waveform.

The Cepstrum is closely related to the auto correlation function. The Cepstrum separates the glottal frequency from the vocal tract resonances. The Cepstrum is obtained in two steps. A logarithmic power spectrum is calculated and declared to be the new analysis window. On that an inverse FFT is performed. The result is a signal with a time axis.

The word Cepstrum is a play **on spectrum**, and it denotes mathematically:

$$c(n) = \mathrm{ifft}(\log|\mathrm{fft}(s(n))|),$$

Where s(n) is the sampled speech signal, and c(n) is the signal in the Cepstral domain. The Cepstral analysis is used in speaker identification because the speech signal is of the particular form above, and the "Cepstral transform" of it makes the analysis incredibly simple. The speech signal s(n) is considered as the convolution of pitch p(n) and vocal tract h(n), then, c(n) which is the Cepstrum of the speech signal can be represented as..

$$c(n) = \mathrm{ifft}(\log(\ \mathrm{fft}(\ h(n)*p(n)\ )\ )\ )$$

$$c(n) = \mathrm{ifft}(\log(\ H(jw)P(jw)\ )\ )$$

$$c(n) = \mathrm{ifft}(\log(H(jw)) + \mathrm{ifft}(\log(P(jw)))$$

The key is that the logarithm, though nonlinear, basically just attenuates each spectrum. For human speakers, Fp, the pitch frequency, can take on values between 80Hz and 300Hz, so we are able to narrow down the portion of the Cepstrum where we look for pitch. In the Cepstrum, which is basically the time domain, we look for an impulse train. The pulses are separated by the pitch period, i.e. 1/Fp

In this final step, we convert the log Mel spectrum back to time. The result is called the Mel frequency cepstrum coefficients (MFCC). The cepstral representation of the speech spectrum provides a good representation of the local spectral properties of the signal for the given frame analysis. Because the Mel spectrum coefficients (and so their logarithm) are real numbers, we can convert them to the time domain using the Discrete Cosine Transform (DCT). Therefore if we denote those Mel power spectrum coefficients that are the result of the last step are $\tilde{S}_k$, $k = 1,2,...,K$, we can calculate the MFCC's, $\tilde{c}_n$, as

$$\tilde{c}_n = \sum_{k=1}^{K} (\log \tilde{S}_k) \cos\left[n\left(k - \frac{1}{2}\right)\frac{\pi}{K} \right], \qquad n = 1,2,...,K$$

Note that we exclude the first component, $\tilde{c}_0$, from the DCT since it represents the mean value of the input signal, which carried little speaker specific information.

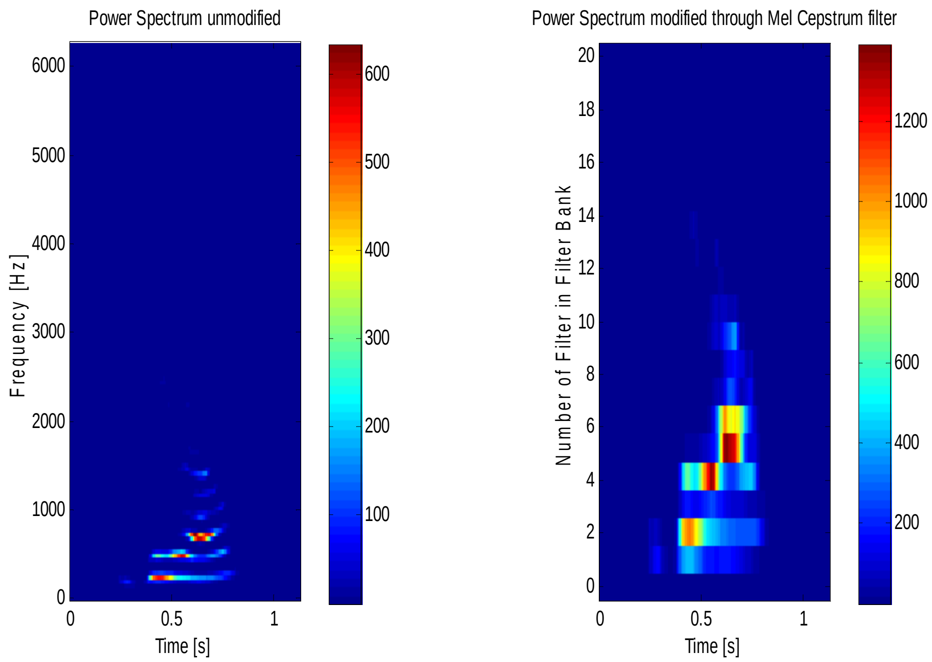


Figure:3.6 power spectrum modified through mel spaced filter bank

The resulted acoustic vectors i.e. the Mel Frequency Cepstral Coefficients corresponding to fifth and sixth filters were plotted in the following figure i.e. the figure

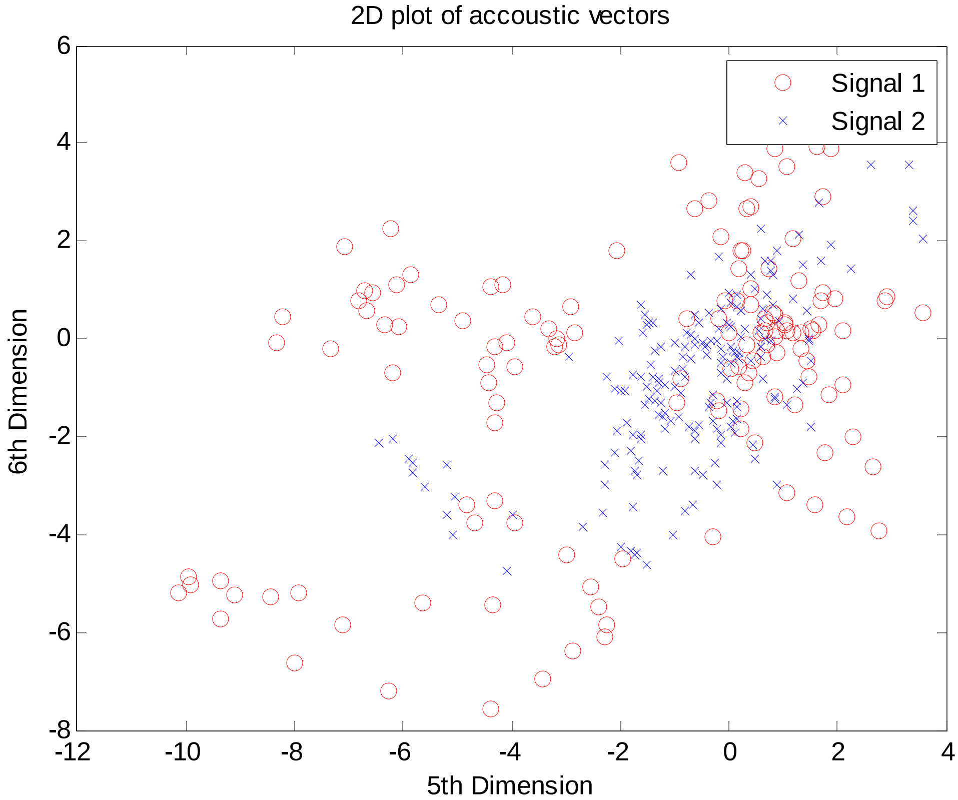


Figure 3.7 MFCCs corresponding to speaker 1 though fifth and sixth filters.

3.3 SUMMARY

By applying the procedure described above, for each speech frame of around 30msec with overlap, a set of mel-frequency cepstrum coefficients is computed. These are the result of a cosine transform of the logarithm of the short-term power spectrum expressed on a mel-frequency scale. This set of coefficients is called an acoustic vector. Therefore each input utterance is transformed into a sequence of acoustic vectors. In the next section we will see how those acoustic vectors can be used to represent and recognize the voice characteristic of the speaker.

Chapter 4

SPEAKER CODING USING VECTOR QUANTIZATION

4.1 INTRODUCTION

The problem of speaker recognition belongs to a much broader topic in scientific and engineering so called *pattern recognition*. The goal of pattern recognition is to classify objects of interest into one of a number of categories or classes. The objects of interest are generically called *patterns* and in our case are sequences of acoustic vectors that are extracted from an input speech using the techniques described in the previous section. The classes here refer to individual speakers. Since the classification procedure in our case is applied on extracted features, it can be also referred to as *feature matching*.

Furthermore, if there exist some set of patterns whose individual classes are already known, then one has a problem in *supervised pattern recognition*. This is exactly our case, since during the training session, we label each input speech with the ID of the speaker (S1 to S8). These patterns comprise the *training set* and are used to derive a classification algorithm. The remaining patterns are then used to test the classification algorithm; these patterns are collectively referred to as the *test set*. If the correct classes of the individual patterns in the test set are also known, then one can evaluate the performance of the algorithm.

The state-of-the-art in feature matching techniques used in speaker recognition includes Dynamic Time Warping (DTW), Hidden Markov Modeling (HMM), and Vector Quantization (VQ). In this project, the VQ approach will be used, due to ease of implementation and high accuracy. VQ is a process of mapping vectors from a large vector space to a finite number of regions in that space. Each region is called a *cluster* and can be represented by its center called a *codeword*. The collection of all codewords is called a *codebook.*

4.2 SPEAKER MODELLING

Using Cepstral analysis as described in the previous section, an utterance may be represented as a sequence of feature vectors. Utterances spoken by the same person but at different times result in similar yet a different sequence of feature vectors. The purpose of voice modeling is to build a model that captures these variations in the extracted set of features.

There are two types of models that have been used extensively in speaker recognition systems**:** stochastic models and template models .The stochastic model treats the speech production process as a parametric random process and assumes that the parameters of the underlying stochastic process can be estimated in a precise, well defined manner. The template model attempts to model the speech production process in a non-parametric manner by retaining a number of sequences of feature vectors derived from multiple utterances of the same word by the same person. Template models dominated early work in speaker recognition because the template model is intuitively more reasonable. However, recent work in stochastic models has demonstrated that these models are more flexible and hence allow for better modelling of the speech production process. The state-of-the-art in feature matching techniques used in speaker recognition includes Dynamic Time Warping (DTW), Hidden Markov Modeling (HMM), and Vector Quantization (VQ).

In a speaker recognition system, each speaker must be uniquely represented in an efficient manner. This process is known as vector quantization. Vector quantization is the process of mapping vectors from a large vector space to a finite number of regions in that space. Each region is called a *cluster* and can be represented by its center called a codeword. The collection of all codewords is called a codebook. The data is thus significantly compressed, yet still accurately represented. Without quantizing the feature vectors, the system would be too large and computationally complex. In a speaker recognition system, the vector space contains a speaker's characteristic vectors, which are obtained from the feature extraction described above. After the completion of vector quantization, only a few representative vectors remain, and these are collectively known as the speaker's *codebook*. The codebook then serves as delineation for the speaker, and is used when training a speaker in the system.

4.3 VECTOR QUANTIZATION

Vector quantization *(VQ)* is the process of taking a large set of feature vectors and producing a smaller set of feature vectors that represent the centroids of the distribution, i.e. points spaced so as to minimize the average distance to every other point. We use vector quantization since it would be impractical to store every single feature vector that we generate from the training utterance [8][11]. While the VQ algorithm does take a while to compute, it saves time during the testing phase, and therefore is a compromise that we can live with.

A vector quantizer maps *k-dimensional* vectors in the vector space R^k into a finite set of vectors $Y = \{y_i: i = 1, 2, ..., N\}$. Each vector y_i *is* called a code vector or a *codeword* and the set of all the codewords is called a *codebook*. Associated with each codeword, y_i, is a nearest neighbor region called Voronoi region, and it is defined by:

$$V_i = \left\{x \in R^k : \|x - y_i\| \leq \|x - y_j\|, for\,all\; j \neq i\right\}$$

The set of Voronoi regions partition the entire space R^k such that:

$$\bigcup_{i=1}^{N} V_i = R^k$$

$$\bigcap_{i=1}^{N} V_i = \phi$$

for all $i \neq j$

As an example, take vectors in the two dimensional case without loss of generality. Figure 3.6 shows some vectors in space. Associated with each cluster of vectors is a representative codeword. Each codeword resides in its own Voronoi region. These regions are separated with imaginary lines in figure 3.6 for illustration. Given an input vector, the codeword that is chosen to represent it is the one in the same Voronoi region.

The representative codeword is determined to be the closest in Euclidean distance from the input vector. The Euclidean distance is defined by:

$$d(x, y_i) = \sqrt{\sum_{j=1}^{k} (x_j - y_{ij})^2}$$

Where x_j is the jth component of the input vector, and y_{ij} is the jth is component of the codeword y_i.

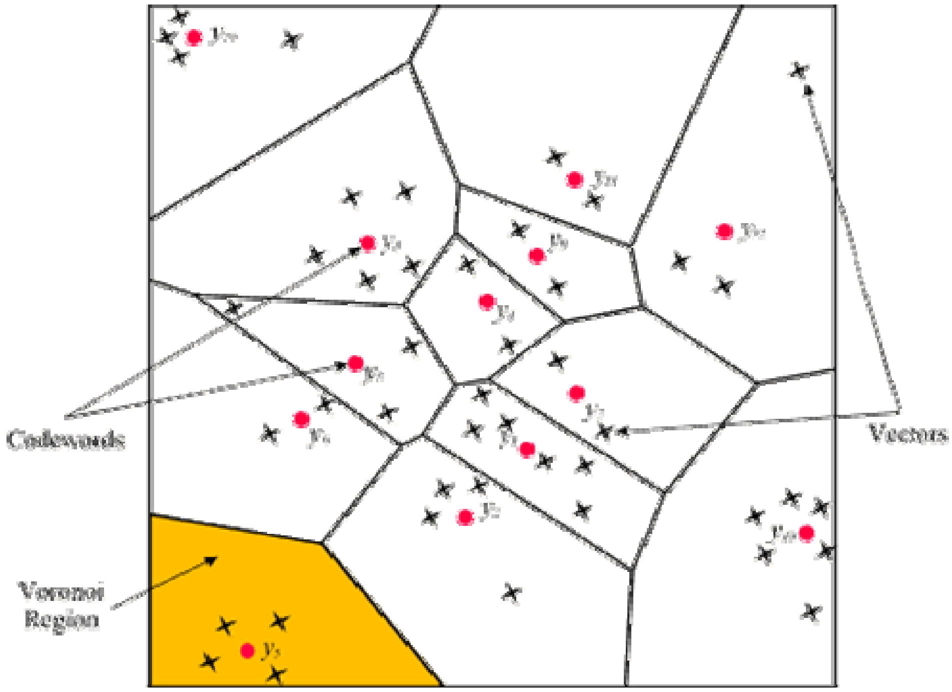


Figure 4.1: Codewords in 2-dimensional space. Input vectors are marked with an x, codewords are marked with red circles, and the Voronoi regions are separated with boundary lines.

There are a number of clustering algorithms available for use, however the one chosen does not matter as long as it is computationally efficient and works properly. A clustering algorithm is typically used for vector quantization, and the words are, at least for our purposes, synonymous. Therefore, LBG algorithm proposed by Linde, Buzo, and Gray is chosen. After taking the enormous number of feature vectors and approximating them with the smaller number of vectors, all of these vectors are filed away into a codebook, which is referred to as codewords.

The result of the feature extraction is a series of vector characteristics of the time-varying spectral properties of the speech signal. These vectors are 24 dimensional and are continuous. These can be mapped to discrete vectors by quantizing. However, as vectors are quantized, this is termed as Vector Quantization. VQ is potentially an extremely efficient representation of spectral information in the speech signal.

The key advantages of VQ are

- ➢ Reduced storage for spectral analysis information
- ➢ Reduced computation for determining similarity of spectral analysis vectors. In speech recognition, a major component of the computation is the determination of spectral similarity between a pair of vectors. Based on the VQ representation this is often reduced to a table lookup of similarities between pairs of codebook vectors.
- ➢ Discrete representation of speech sounds

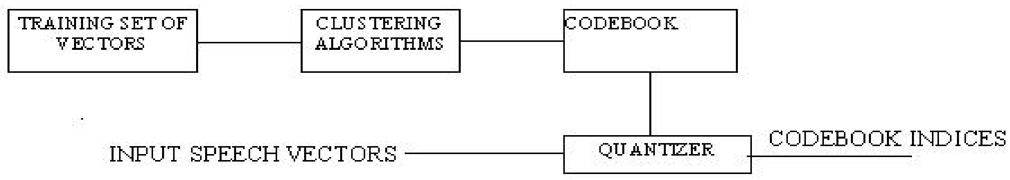


Figure 4.2.Block Diagram of the basic VQ Training and classification structure

Figure 4.1 shows a conceptual diagram to illustrate this recognition process. In the figure, only two speakers and two dimensions of the acoustic space are shown. The circles refer to the acoustic vectors from the speaker 1 while the triangles are from the speaker 2. In the training phase, a speaker-specific VQ codebook is generated for each known speaker by clustering his/her training acoustic vectors. The resultant codewords (centroids) are shown in Figure 4.1 by black circles and black triangles for speaker 1 and 2, respectively.

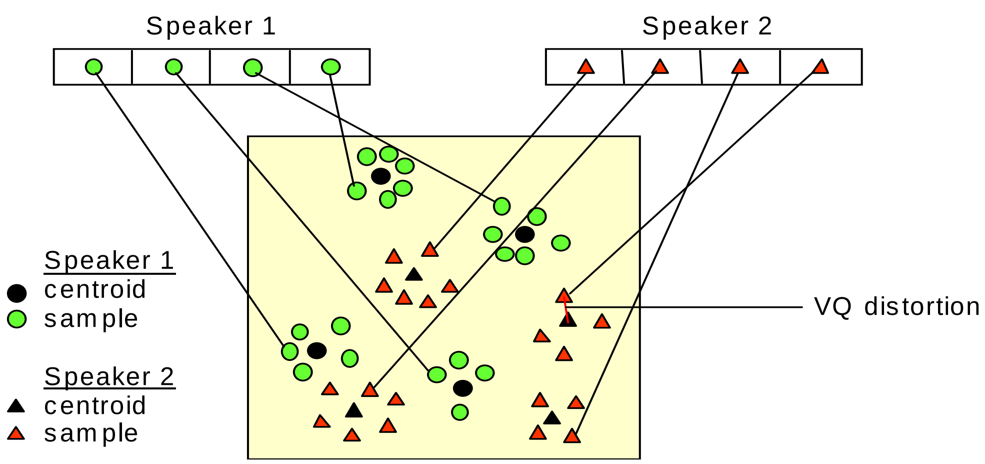


Figure 4.3. Conceptual diagram illustrating vector quantization codebook formation.

One speaker can be discriminated from another based of the location of centroids.

4.4 OPTIMIZATION WITH LBG

Given a set of I training feature vectors, {**a**1, **a**2…, **a***I*} characterizing the variability of a speaker, a partitioning of the feature vector space, {S1, S2,..., S*M*}is to be determined. For a particular speaker the whole feature space S, is represented as S = S1 U S2 U...U S*M*. Each partition, S*i*, forms a non-overlapping region and every vector inside S*i* is represented by the corresponding centroid vector, **b***i*, of S*i*. Each iteration of k moves the centroid vectors such that the accumulated distortion between the feature vectors is lessened. The more the iterations are , the less is the distortion. The algorithm takes each feature vector and compares it with every codebook vector, which is closest to it. That distortion is then calculated for each codebook vector j as [12]:

$$D_j = \sum_{i=1}^{I} (t_i - v_{j,i})^2$$

where v are the vectors in the codebook and t is the training vector. The minimum distortion value is found among all measurements. Then the new centroid of each region is calculated. If x is in the training set, and x is closer to vi than to any other codebook vector, assign x to Ci. The new centroid is calculated, where Ci is the set of vectors in the training set that are closer to vi than to any other codebook vector. The next iteration will recompute the regions according to the new centroids. The total distortion will now be smaller. Iteration continues until a relatively small percent change in distortion is achieved.After the enrollment session, the acoustic vectors extracted from input speech of a speaker provide a set of training vectors. As described above, the next important step is to build a speaker-specific VQ codebook for this speaker using those training vectors. There is a well-know algorithm, namely LBG algorithm [Linde, Buzo and Gray, 1980], for clustering a set of L training vectors into a set of M codebook vectors.

The algorithm is formally implemented by the following recursive procedure:

Design a 1-vector codebook; this is the centroid of the entire set of training vectors (hence, no iteration is required here).

Double the size of the codebook by splitting each current codebook $\mathbf{y}_n$ according to the rule

$$\mathbf{y}_n^+ = \mathbf{y}_n(1+\varepsilon)$$

$$\mathbf{y}_n^- = \mathbf{y}_n(1-\varepsilon)$$

where n varies from 1 to the current size of the codebook, and ε is a splitting parameter (we choose ε =0.01).

Nearest-Neighbor Search: for each training vector, find the codeword in the current codebook that is closest (in terms of similarity measurement), and assign that vector to the corresponding cell (associated with the closest codeword).

Centroid Update: update the codeword in each cell using the centroid of the training vectors assigned to that cell.

Iteration 1: repeat steps 3 and 4 until the average distance falls below a preset threshold

Iteration 2: repeat steps 2, 3 and 4 until a codebook size of M is designed.

Intuitively, the LBG algorithm designs an M-vector codebook in stages. It starts first by designing a 1-vector codebook, then uses a splitting technique on the codewords to initialize the search for a 2-vector codebook, and continues the splitting process until the desired M-vector codebook is obtained.

Figure 4.2 shows, in a flow diagram, the detailed steps of the LBG algorithm. "*Cluster vectors*" is the nearest-neighbor search procedure, which assigns each training vector to a cluster associated with the closest codeword. "*Find centroids*" is the centroid update procedure. "*Compute D (distortion)*" sums the distances of all training vectors in the nearest-neighbor search so as to determine whether the procedure has converged.

The resultant codebooks along with the MFCCs were shown in figure 4.3

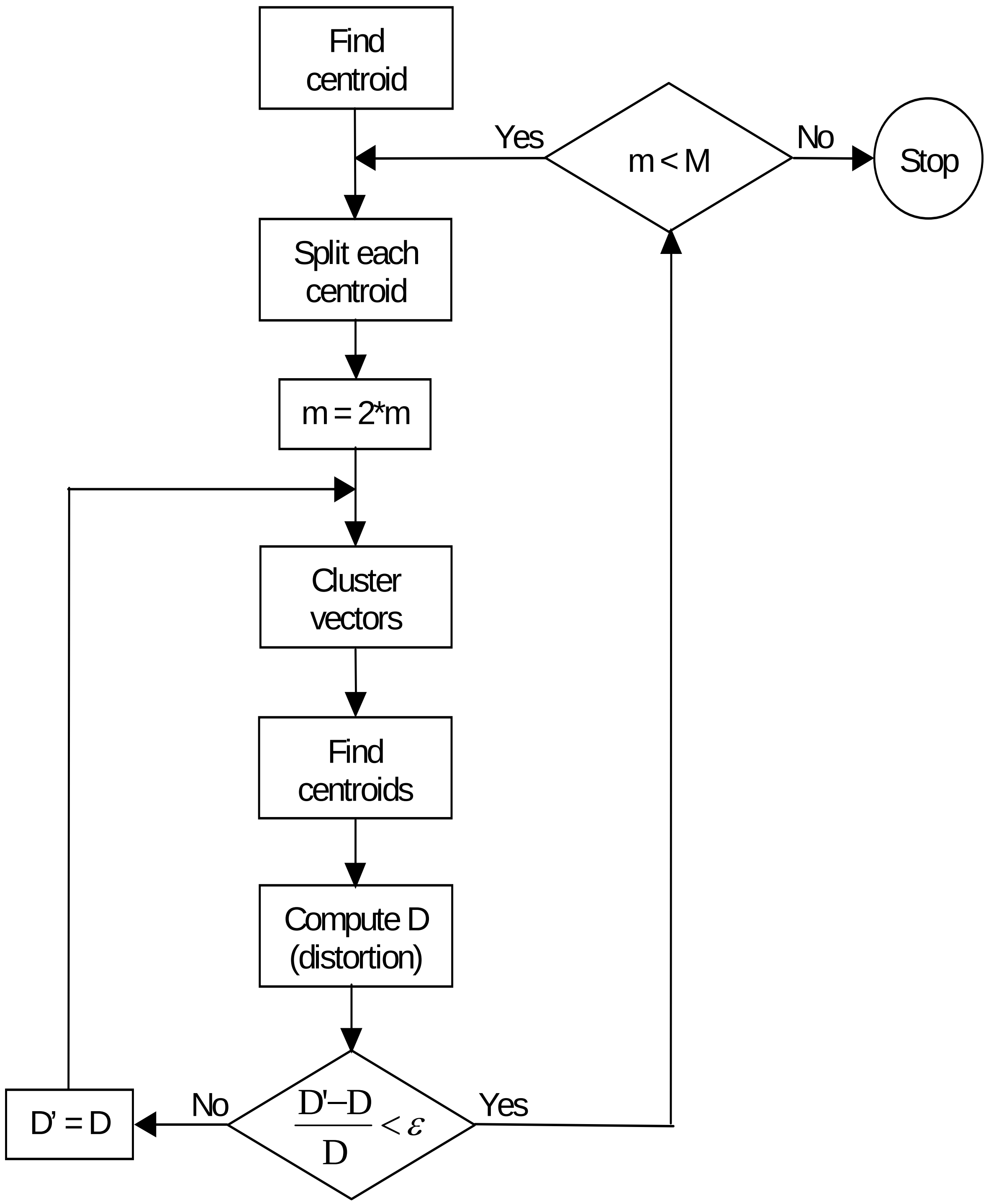
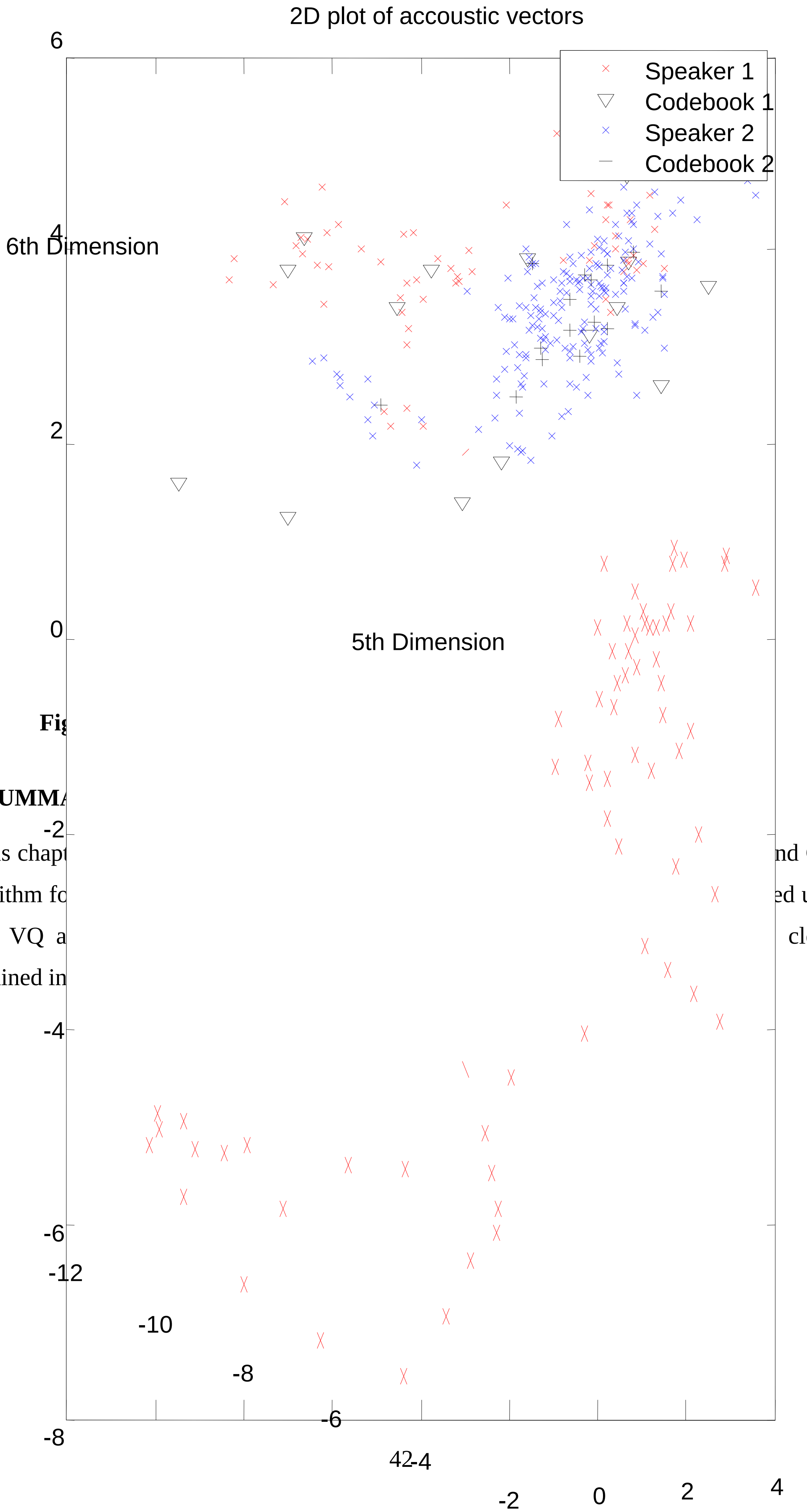


Figure 4.4. Flow chart showing the implementation of the LBG algorithm



Fi

4.5 SUMMA

In this chapt … nd Gray

algorithm fo … ed using

LBG VQ a … clearly

explained in

Chapter 5

SPEECH FEATURE MATCHING

5.1 DISTANCE CALCULATION

Figure 5.1 shows a conceptual diagram to illustrate this recognition process. In the figure, only two speakers and two dimensions of the acoustic space are shown. The circles refer to the acoustic vectors from the speaker 1 while the triangles are from the speaker 2. In the training phase, a speaker-specific VQ codebook is generated for each known speaker by clustering his/her training acoustic vectors. The result codewords (centroids) are shown in Figure 4.1 by black circles and black triangles for speaker 1 and 2, respectively. The distance from a vector to the closest codeword of a codebook is called a VQ-distortion. VQ distortion is nothing but the Euclidian distance between the two vectors and is given by the formula

$$d_E(x, y) = \sqrt{\sum_{i=1}^{\dim} (x_i - y_i)^2}$$

In the recognition phase, an input utterance of an unknown voice is "vector-quantized" using each trained codebook and the *total VQ distortion* is computed. The speaker corresponding to the VQ codebook with smallest total distortion is identified.

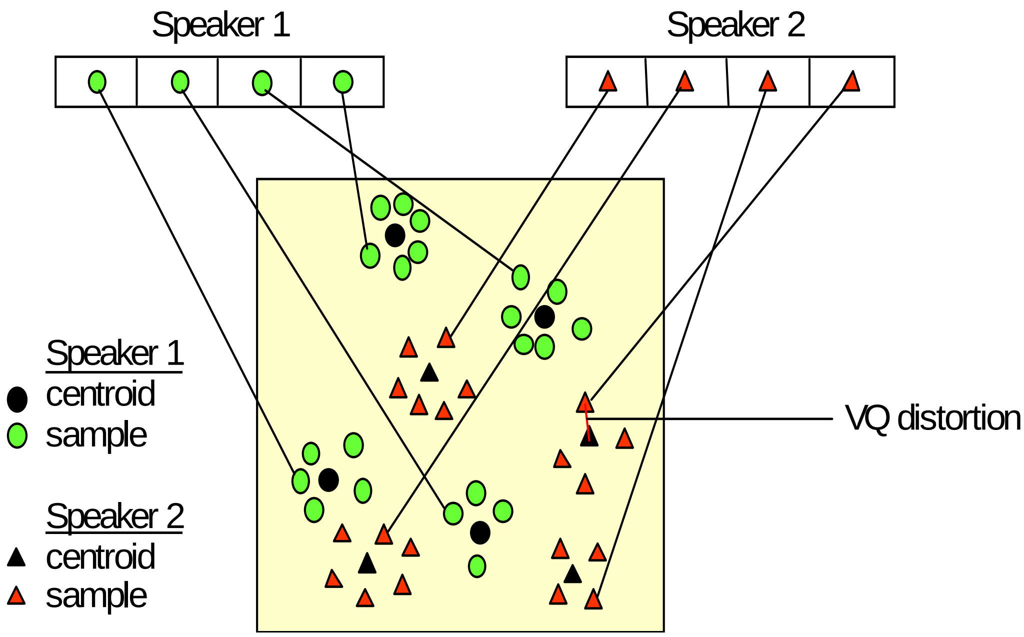


Figure 5.1. Conceptual diagram illustrating vector quantization codebook formation.

One speaker can be discriminated from another based of the location of centroids.

As stated above, in this project we will experience the building and testing of an automatic speaker recognition system. In order to implement such a system, one must go through several steps which were described in details in previous sections. All these tasks are implemented in Matlab.

5.2 SUMMARY

The speech feature matching is explained clearly and calculation of the Euclidean distance between the codebooks of each speaker is done which makes us to identify the corresponding speaker of the speech. Many other techniques are available for the feature matching but I employed only Euclidean distance because to my system simple and easy to understand.

Chapter 6

DIGITAL SIGNAL PROCESSING

5.3 INTRODUCTION TO DSP

Digital Signal processing (DSP) is one of the fastest growing fields of technology and computer science in the world. In today's world almost everyone uses DSPs in their everyday life but, unlike PC users, almost no one knows that he/she is using DSPs. Digital Signal Processors are special purpose microprocessors used in all kind of electronic products, from mobile phones, modems and CD players to the automotive industry; medical imaging systems to the electronic battlefield and from dishwashers to satellites.[17] DSP is all about analysing and processing real-world or analogue signals, i.e. the kind of signals that humans interact with, for example speech. These signals are converted to a format that computers can understand (digital) and, once this has happened, process. The following diagram shows the typical component parts of a DSP system.

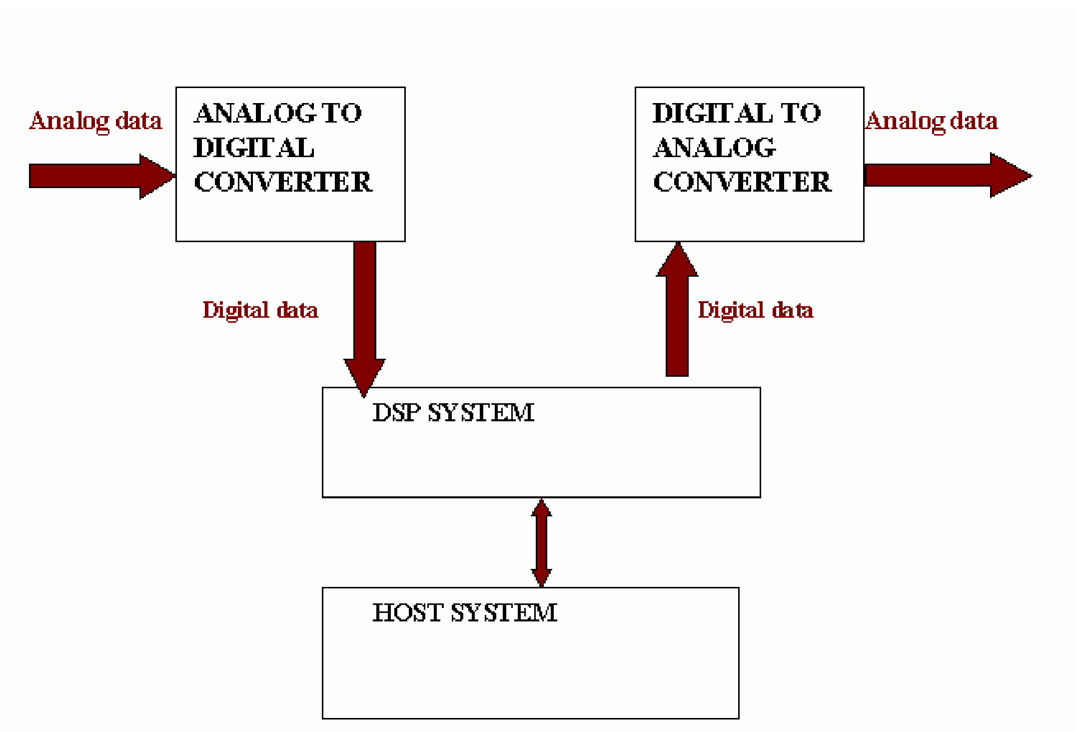


Figure.4.1. Typical components of a DSP system

In order to process analog signals with digital computers they must first be converted to digital signals using analog to digital converters. Similarly, the digital signals must be converted back to analog ones for them to be used outside the computer.

There are many reasons why we process these analog signals in the digital world. Traditional signal processing was achieved by using analogue components such as resistors, capacitors and inductors. However, the inherent tolerance associated with this components, temperature and voltage changes and mechanical vibrations can dramatically affect the effectiveness of analogue circuitry. On the other hand, DSP is inherently stable, reliable and repeatable.With DSP it is easy to chance, correct or update applications. Additionally, DSP reduces noise susceptibility, chip count, development time, cost and power consumption.

5.4 HOW DSPs ARE DIFFERENT FROM OTHER MICROPROCESSORS

DSP has many unique properties. It is a Super Mathematician thanks to its arithmetic logic units and its optimized multipliers. DSPs do really well in application where the data to be processed is arriving in a continuous flow, often referred to as a stream. It uses almost no power compared to a PC microprocessor. Next, some features that make DSP different from other microprocessors are going to be described:

1. High speed arithmetic: Most DSP operations require additions and multiplications together. DSP proccesors usually have hardware adders and multipliers which can be used in parallel within a single instruction, so both, an addition and a multiplication, can be executed in a single cycle. Thus, DSP processors arithmetic speed is very high compared with microprocessors.

2. Data transfer to and from real world. In a typical DSP application the processor will have to deal with multiple sources of data from the real world. In each case, the processor may have to be able to receive and transmit data in real time, without interrupting its internal mathematical operations. These multiple communications routes mark the most important distinctions between DSP processors and general purpose processors.

3. Multiple access memory architectures: Typical DSP operations require many simple additions and multiplications. To fetch the two operations in a single instruction cycle the two memory accesses should be able to operate simultaneously. For this reason DSP processors usually support multiple memory accesses in the same instruction cycle.

4. Digital Signal Processors also have the advantage of consuming less power and being relatively cheap.

5.5 INTRODUCTION TO THE TMS320C6000 PLATFORM OF DIGITAL SIGNAL PROCESSORS

The DSP architecture is a well defined but quite complex hardware structure that needs much time to be explained in detail. An overview of this architecture is going to be exposed here in order to make it as much understandable as possible.

The TMS320C6000 family of processors from the company Texas Instruments is designed to meet the real-time requirements of high performance digital signal processing. With a performance of up to 2000 million instructions per second (MIPS) at 250 MHz and a complete set of development tools, the TMS320C6000 DSPs offer cost effective solutions to higher-performance DSP programming challenges.

The TMS320C6000 DSPs give the system architects unlimited possibilities to differentiate their products. High performance, easy use, and affordable pricing make the TMS320c6000 platform the ideal solution for a large number of applications (multichannel multifunction applications such as: pooled modems, wireless local loop base stations, multichannel telephony systems, etc). First of all, a DSP device must be considered as a specific microprocessor whose components have been linked in a clever way to process faster.

The TMS320C6xxx family are processors currently running at a clock speed of up to 300MHz (225MHz in the TMS320C6713 case). The C62xx processors are fixed-point processors whereas the C67xx are floating-point processors. These refer to the format used to store and manipulate numbers withing the devices. Figure 24 shows the main components of the TMS320C6000 DSP under a block diagram form.

It is composed of:

1. External Memory Interface (EMIF) to access external data at the specified address.
2. Memory, which is the internal memory where a set of instructions and data values can be stored (FFT algorithm for example).
3. Peripherals are the possible connectable devices that can be associated with the DSP (DMA/EDMA, Serial port, Timer/Counter…).
4. Internal buses; they allow the components to quickly communicate together differentiating addresses and data.
5. CPU, which is the most important component since it performs all the operation.

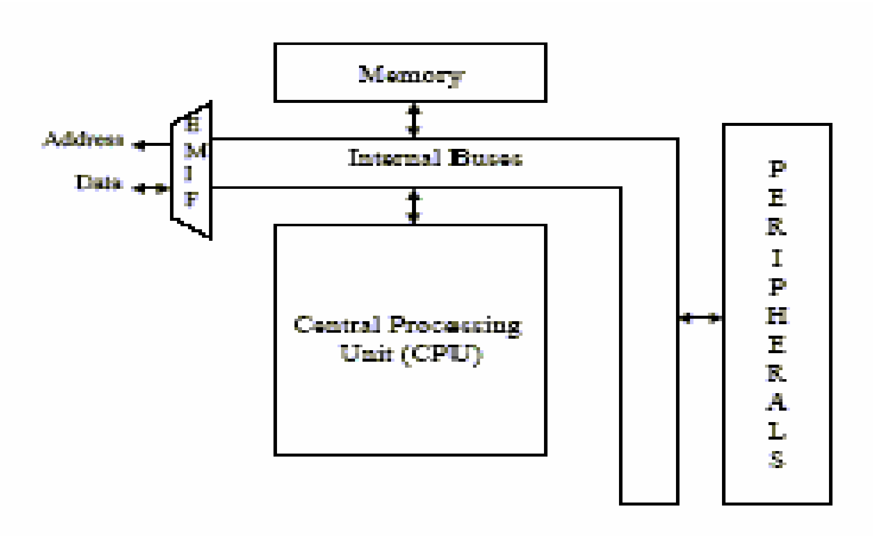


Figure .4.2.TMS320C6000 block diagram

5.6 TMS320C6713 DSP Description

The TMS320C67 DSPs (including the TMS320C6713 device) compose the floating–point DSP generation in the TMS320C6000 DSP platform. The TMS320C6713 (C6713) device is based on the high-performance, advanced VelociTI very-long-instruction-word (VLIW) architecture developed by Texas Instruments (TI), making this DSP an excellent choice for multichannel and multifunction applications.

The DSK features the TMS320C6713 DSP, a 225 MHz device delivering up to 1800 million instructions per second (MIPs) and 1350 MFLOPS. This DSP generation is designed for applications that require high precision accuracy. The C6713 is based on the

TMS320C6000 DSP platform designed to fit the needs of high-performing high-precision applications such as pro-audio, medical and diagnostic. Other hardware features of the TMS320C6713 DSK board include:

- Embedded JTAG support via USB.
- High-quality 24-bit stereo codec.
- Four 3.5mm audio jacks for microphone, line in, speaker and line out.
- 512K words of Flash and 8 MB SDRAM.
- Expansion port connector for plug-in modules.
- On-board standard IEEE JTAG interface.
- +5V universal power supply.

The DSP environment used in this project is the TMS320C6713 DSK. Figure 25 shows the architecture of the TMS320C6713 DSK. Key features include:

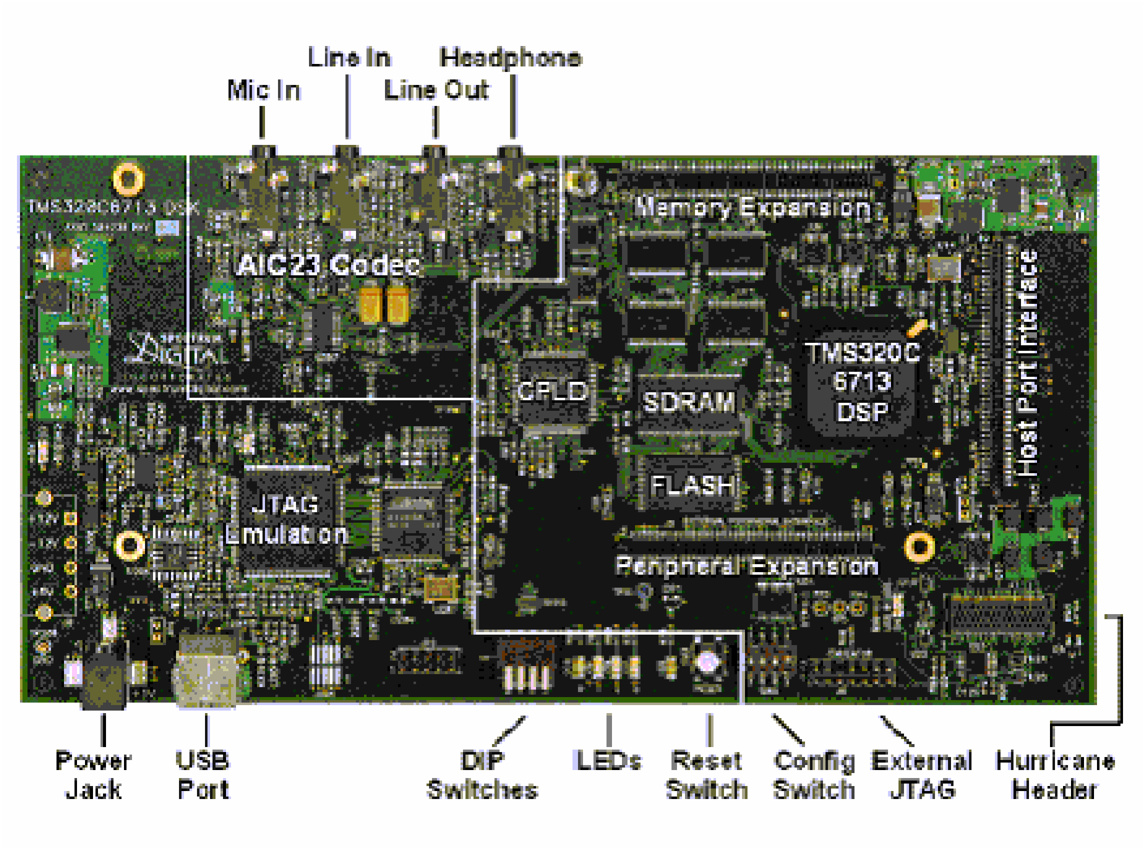


Figure .4.3. Architecture of the TMS320C6713 DSK

5.7 DSP IMPLEMENTATION

This project implements an automatic speaker recognition system [46–50]. *Speaker recognition* refers to the concept of recognizing a speaker by his/her voice or speech samples. This is different from speech recognition. In automatic speaker recognition, an algorithm generates a hypothesis concerning the speaker's identity or authenticity. The speaker's voice can be used for ID and to gain access to services such as banking, voice mail, and so on. Speaker recognition systems contain two main modules: *feature extraction* and *classification*.

1. Feature extraction is a process that extracts a small amount of data from the voice signal that can be used to represent each speaker. This module converts a speech waveform to some type of parametric representation for further analysis and processing. Short-time spectral analysis is the most common way to characterize a speech signal. The Mel-frequency Cepstrum coefficients (MFCC) are used to parametrically represent the speech signal for the speaker recognition task. The steps in this process are shown in Figure 10.53:

(a) Block the speech signal into frames, each consisting of a fixed number of samples.

(b) Window each frame to minimize the signal discontinuities at the beginning and end of the frame.

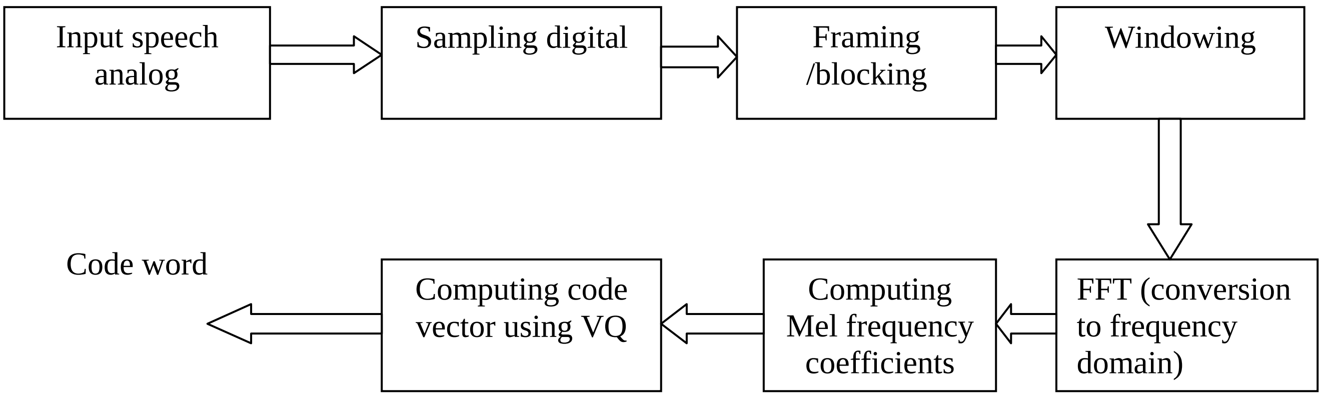


FIGURE 10.53. Steps for speaker recognition implementation.

(c) Use FFT to convert each frame from time to frequency domain.

(d) Convert the resulting spectrum into a Mel-frequency scale.

(e) Convert the Mel spectrum back to the time domain.

2. Classification consists of models for each speaker and decision logic necessary to render a decision. This module classifies extracted features according to the individual speakers whose voices have been stored. The recorded voice patterns of the speakers are used to derive a classification algorithm. Vector quantization (VQ) is used. This is a process of mapping vectors from a large vector space to a finite number of regions in that space. Each region is called a *cluster* and can be represented by its center, called a *codeword*. The collection of all clusters is a *codebook*. In the training phase, a speaker-specific VQ codebook is generated for each known speaker by clustering his/her training acoustic vectors. The distance from a vector to the closest codeword of a codebook is called a *VQ distortion*. In the recognition phase, an input utterance of an unknown voice is vector-quantized using each trained codebook, and the total VQ distortion is computed. The speaker corresponding to the VQ codebook with the smallest total distortion is identified.

Speaker recognition can be classified with identification and verification. *Speaker identification* is the process of determining which registered speaker provides a given utterance. *Speaker verification* is the process of accepting or rejecting the identity claim of a speaker. This project implements only the speaker identification (ID) process. The speaker ID process can be further subdivided into *closed set* and *open set*. The *closed set* speaker ID problem refers to a case where the speaker is known a priori to belong to a set of *M* speakers. In the *open set* case, the speaker may be out of the set and, hence, a “none of the above” category is necessary. In this project, only the simpler closed set speaker ID is used.

Speaker ID systems can be either *text-independent* or *text-dependent*. In the *text independent* case, there is no restriction on the sentence or phrase to be spoken, whereas in the *text-dependent* case, the input sentence or phrase is indexed for each speaker. The text-dependent system, implemented in this project, is commonly found in speaker verification systems in which a person’s password is critical for verifying his/her identity.

In the *training phase*, the feature vectors are used to create a model for each speaker. During the *testing phase*, when the test feature vector is used, a number will be associated with each speaker model indicating the degree of match with that speaker’s model. This is done for a set of feature vectors, and the derived numbers can be used to find a likelihood score for each speaker’s model. For the speaker ID problem, the feature vectors of the test utterance are passed through all the speakers' models and the scores are calculated. The model having the best score gives the speaker's identity (which is the decision component).

This project uses MFCC for feature extraction, VQ for classification/training, and the Euclidean distance between MFCC and the trained vectors (from VQ) for speaker ID. Much of this project was implemented with MATLAB [47].

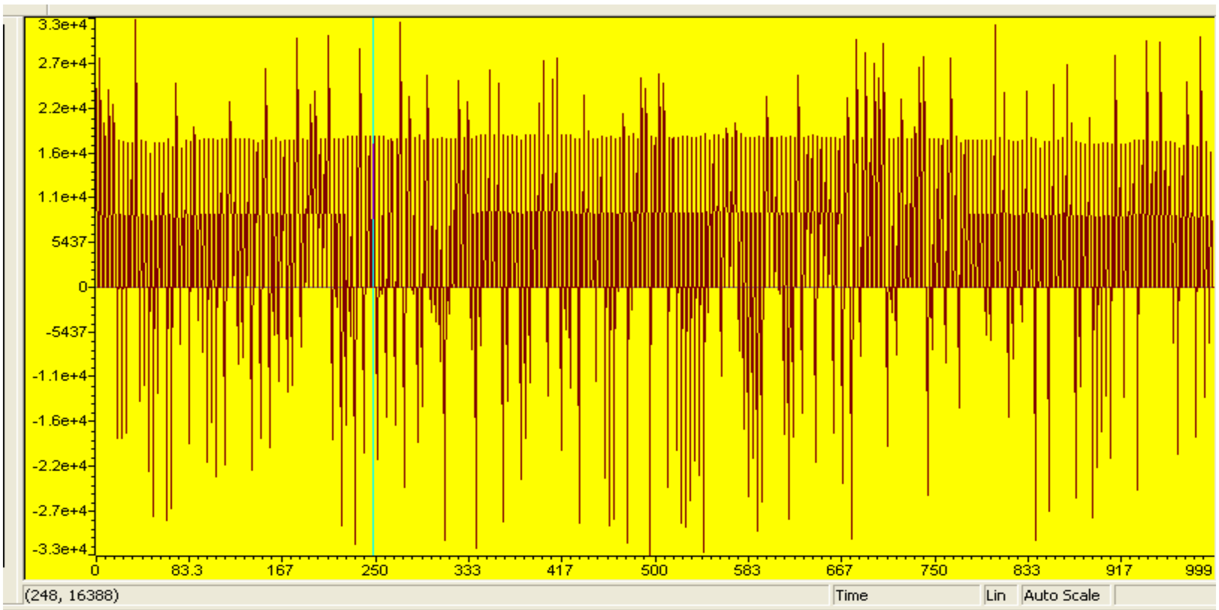


Figure10.54. Speech samples DSP Processor

The above diagram is speech sample. Speech is taken 'zero'. I taken 8 speakers.

5.8 Features on TMS320C6713 PROCESSOR

- Floating point device.
- Operating frequency 225MHz.
- 264kb internal cache, 16Mb SDRAM, 512 Kb flashes.
- 1800 MIPS or 1350 MFLOPS per second
- TLV320AIC23 (AIC23) onboard stereo codec for I/O operations (range 8 KHz – 96 KHz).
- +5v universal power supply

Chapter 7

RESULTS

MATLAB RESULTS

Speech signals corresponding to eight speakers i.e. S1.wav, S2.wav, S3.wav, S4.wav, S5.wav, S6.wav, S7.wav, and S8.wav in the training folder are compared with the speech files of the same speakers in the testing folder. The matching results of the speakers are obtained as follows.

6.1 WHEN ALL VALID SPEAKERS ARE CONSIDERED

Speaker 1 matches with speaker 1
Speaker 2 matches with speaker 2
Speaker 3 matches with speaker 3
Speaker 4 matches with speaker 4
Speaker 5 matches with speaker 5
Speaker 6 matches with speaker 6
Speaker 7 matches with speaker 7
Speaker 8 matches with speaker 8

6.2 WHEN THERE IS AN IMPOSTER IN PLACE OF SPEAKER 4

Speaker 1 matches with speaker 1
Speaker 2 matches with speaker 2
Speaker 3 matches with speaker 3
Speaker 4 is an imposter and corresponding distance is 1.060407e+001
Speaker 5 matches with speaker 5
Speaker 6 matches with speaker 6
Speaker 7 matches with speaker 7
Speaker 8 matches with speaker 8

6.3 EUCLIDEAN DISTANCES BETWEEN THE CODEBOOKS OF SPEAKERS

Distance between speaker 1 and speaker 1 is 2.582456e+000
Distance between speaker 1 and speaker 2 is 3.423658e+000
Distance between speaker 1 and speaker 3 is 6.691428e+000
Distance between speaker 1 and speaker 4 is 3.290923e+000
Distance between speaker 1 and speaker 5 is 7.227603e+000
Distance between speaker 1 and speaker 6 is 6.004165e+000
Distance between speaker 1 and speaker 7 is 6.388921e+000
Distance between speaker 1 and speaker 8 is 3.990130e+000

Distance between speaker 2 and speaker 1 is 3.644154e+000
Distance between speaker 2 and speaker 2 is 2.023527e+000
Distance between speaker 2 and speaker 3 is 5.932640e+000
Distance between speaker 2 and speaker 4 is 3.962964e+000

Distance between speaker 2 and speaker 5 is 6.041227e+000
Distance between speaker 2 and speaker 6 is 5.033079e+000
Distance between speaker 2 and speaker 7 is 5.120361e+000
Distance between speaker 2 and speaker 8 is 4.053674e+000

Distance between speaker 3 and speaker 1 is 6.208796e+000
Distance between speaker 3 and speaker 2 is 5.631654e+000
Distance between speaker 3 and speaker 3 is 2.000804e+000
Distance between speaker 3 and speaker 4 is 5.191537e+000
Distance between speaker 3 and speaker 5 is 3.464318e+000
Distance between speaker 3 and speaker 6 is 3.608015e+000
Distance between speaker 3 and speaker 7 is 4.014857e+000
Distance between speaker 3 and speaker 8 is 4.323667e+000

Distance between speaker 4 and speaker 1 is 3.280098e+000
Distance between speaker 4 and speaker 2 is 3.713952e+000
Distance between speaker 4 and speaker 3 is 5.298161e+000
Distance between speaker 4 and speaker 4 is 2.499871e+000
Distance between speaker 4 and speaker 5 is 5.865334e+000
Distance between speaker 4 and speaker 6 is 4.805346e+000
Distance between speaker 4 and speaker 7 is 5.314957e+000
Distance between speaker 4 and speaker 8 is 3.441053e+000

Distance between speaker 5 and speaker 1 is 6.978178e+000
Distance between speaker 5 and speaker 2 is 6.136129e+000
Distance between speaker 5 and speaker 3 is 3.579665e+000
Distance between speaker 5 and speaker 4 is 5.900074e+000
Distance between speaker 5 and speaker 5 is 2.078398e+000
Distance between speaker 5 and speaker 6 is 3.537214e+000
Distance between speaker 5 and speaker 7 is 3.579846e+000
Distance between speaker 5 and speaker 8 is 5.079328e+000

Distance between speaker 6 and speaker 1 is 5.776238e+000
Distance between speaker 6 and speaker 2 is 5.380254e+000
Distance between speaker 6 and speaker 3 is 3.690566e+000
Distance between speaker 6 and speaker 4 is 4.937416e+000
Distance between speaker 6 and speaker 5 is 3.420030e+000
Distance between speaker 6 and speaker 6 is 2.990975e+000
Distance between speaker 6 and speaker 7 is 3.429637e+000
Distance between speaker 6 and speaker 8 is 4.257454e+000

Distance between speaker 7 and speaker 1 is 5.701679e+000
Distance between speaker 7 and speaker 2 is 4.847096e+000
Distance between speaker 7 and speaker 3 is 4.125191e+000
Distance between speaker 7 and speaker 4 is 5.077611e+000
Distance between speaker 7 and speaker 5 is 3.453576e+000

Distance between speaker 7 and speaker 6 is	2.844346e+000
Distance between speaker 7 and speaker 7 is	1.706408e+000
Distance between speaker 7 and speaker 8 is	4.496219e+000

Distance between speaker 8 and speaker 1 is	3.791221e+000
Distance between speaker 8 and speaker 2 is	3.783202e+000
Distance between speaker 8 and speaker 3 is	3.965679e+000
Distance between speaker 8 and speaker 4 is	3.363752e+000
Distance between speaker 8 and speaker 5 is	4.687492e+000
Distance between speaker 8 and speaker 6 is	3.931799e+000
Distance between speaker 8 and speaker 7 is	4.254214e+000
Distance between speaker 8 and speaker 8 is	2.386353e+000

The Plots for clearly understanding the difference between the Euclidean distances of speakers were given below.

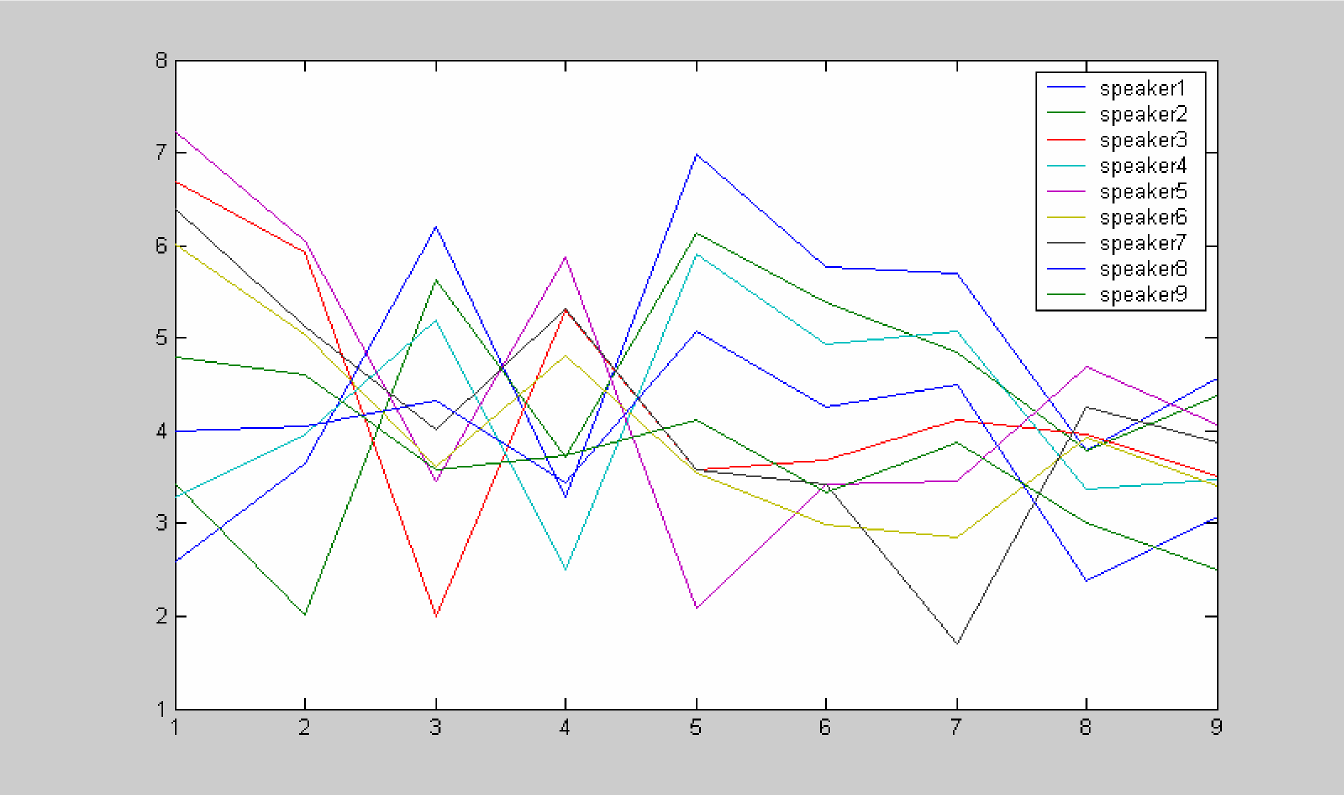


Figure 6.1 Plot for the difference between the Euclidean distances of speakers

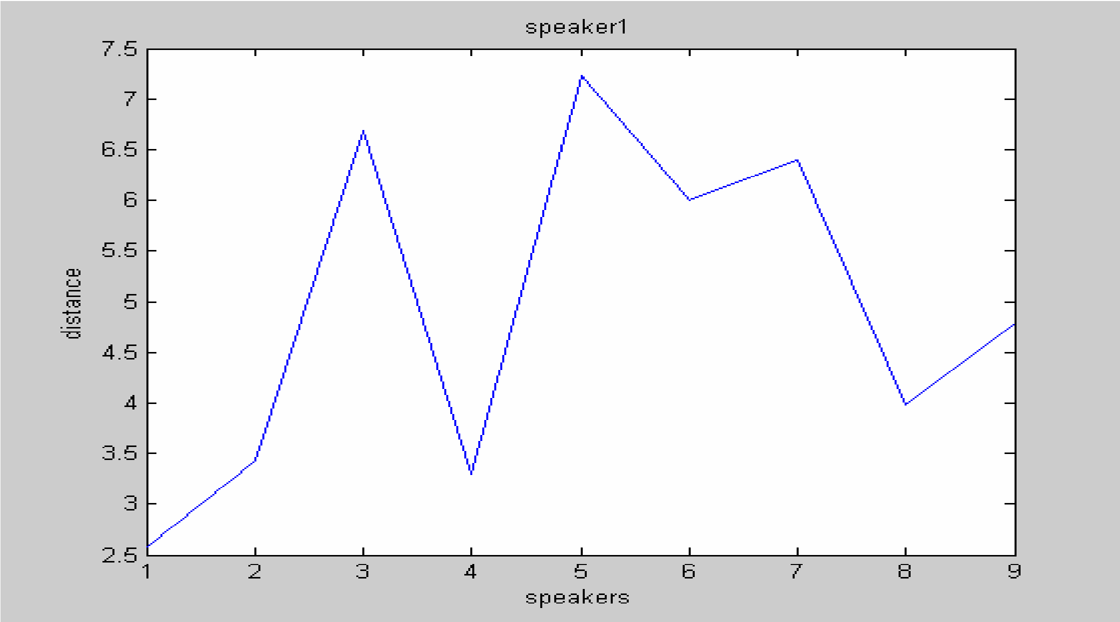


Figure 6.2 Plot for the Euclidean distance between the speaker 1 and all speakers

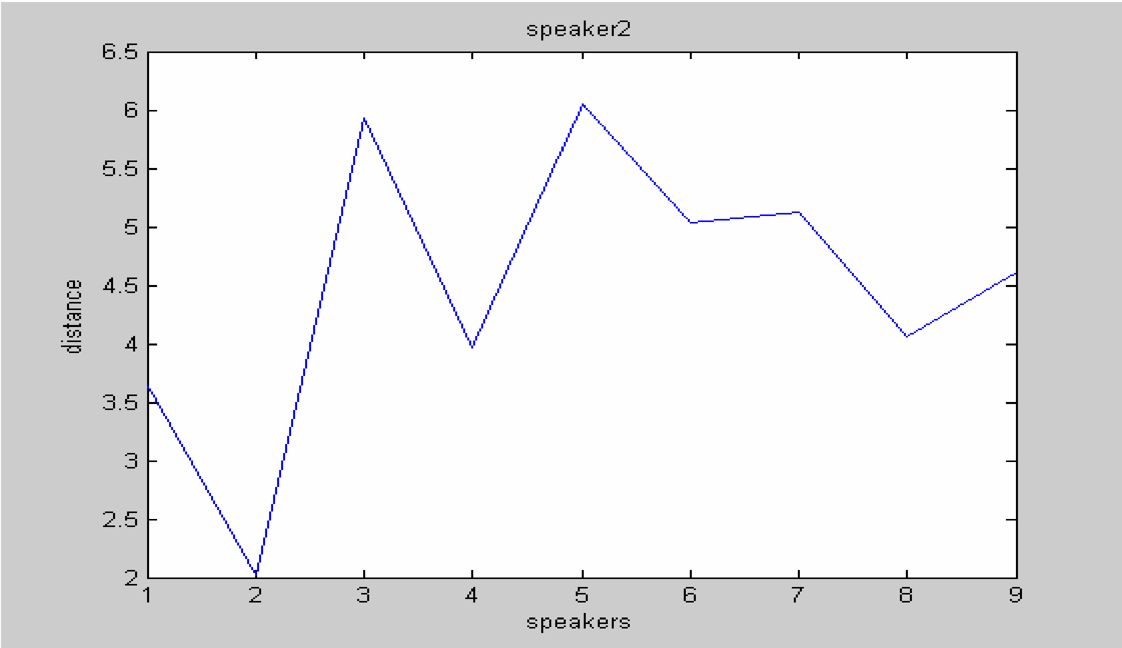


Figure 6.3 Plot for the Euclidean distance between the speaker 2 and all speakers

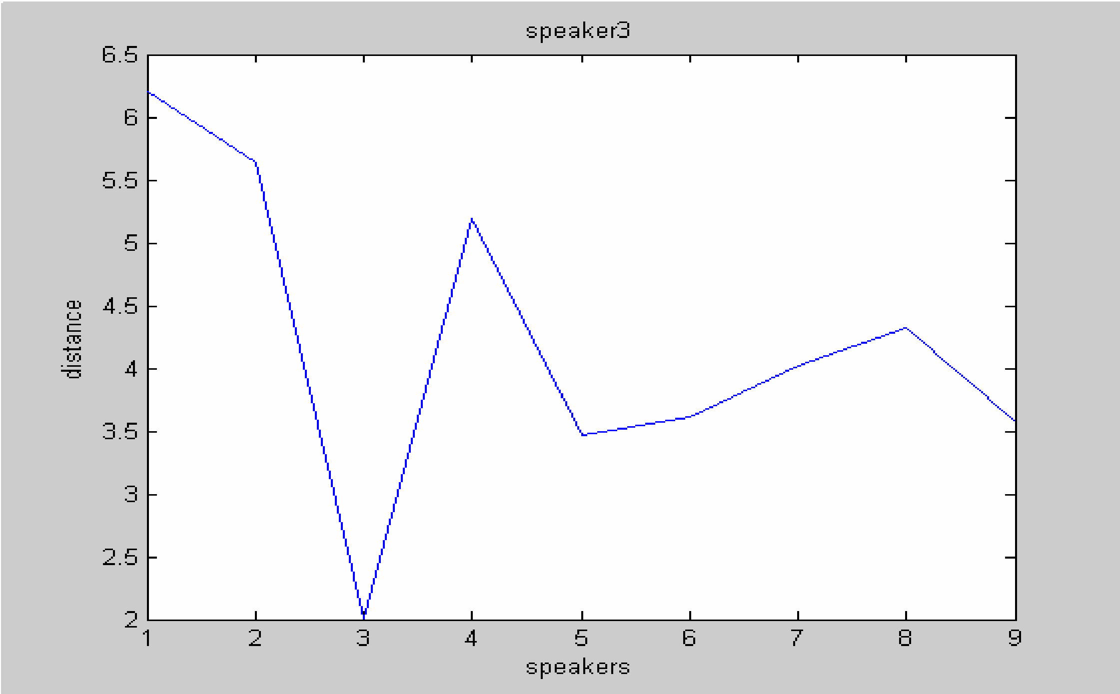


Figure 6.4 Plot for the Euclidean distance between the speaker 3 and all speakers

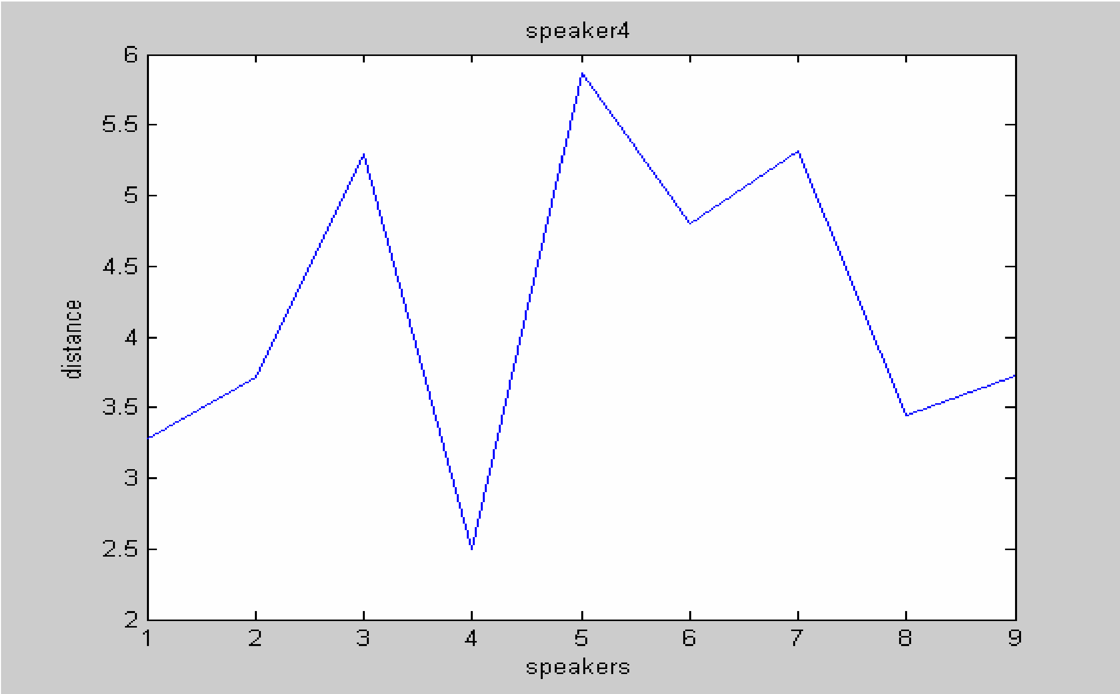


Figure 6.5 Plot for the Euclidean distance between the speaker 4 and all speakers

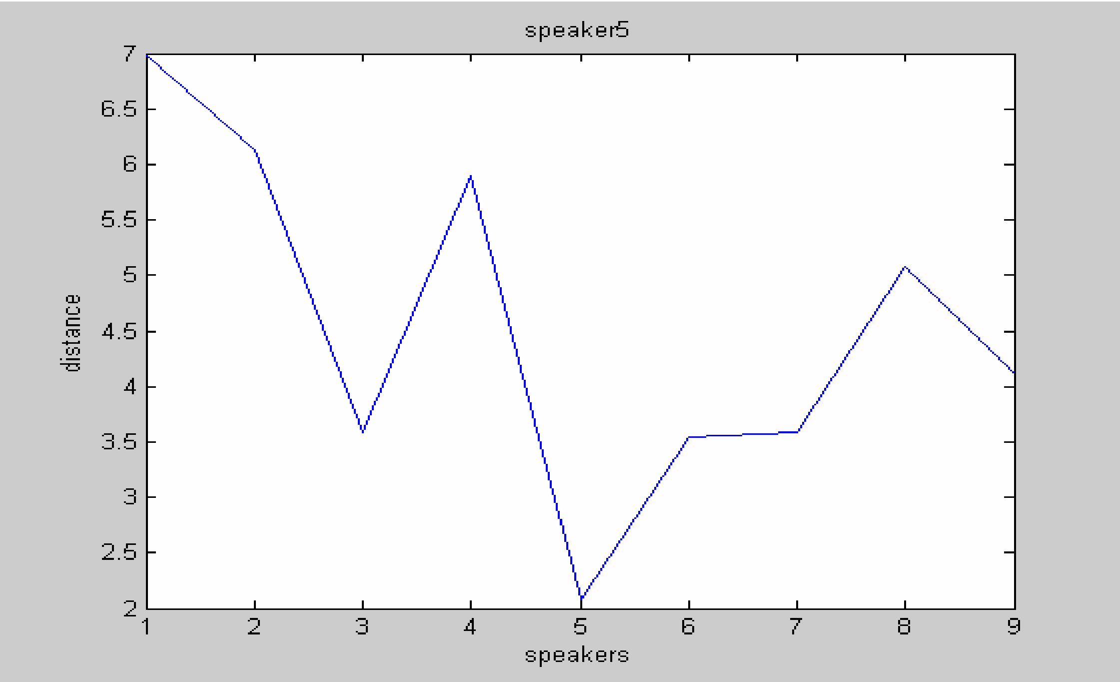


Figure 6.6 Plot for the Euclidean distance between the speaker 5 and all speakers

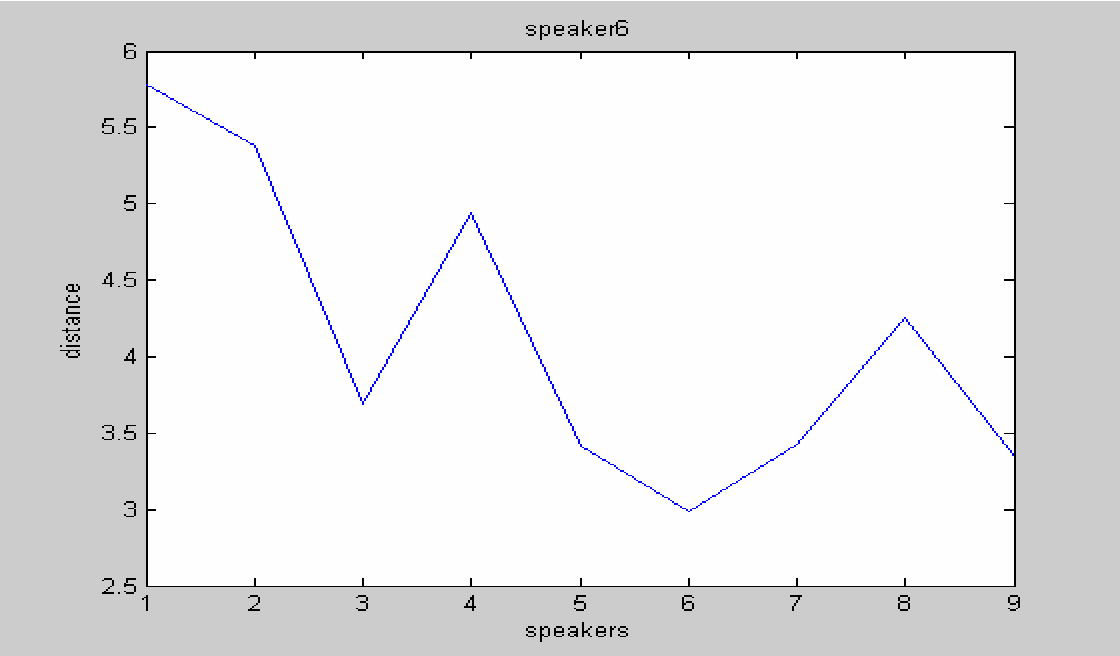


Figure 6.7 Plot for the Euclidean distance between the speaker 6 and all speakers

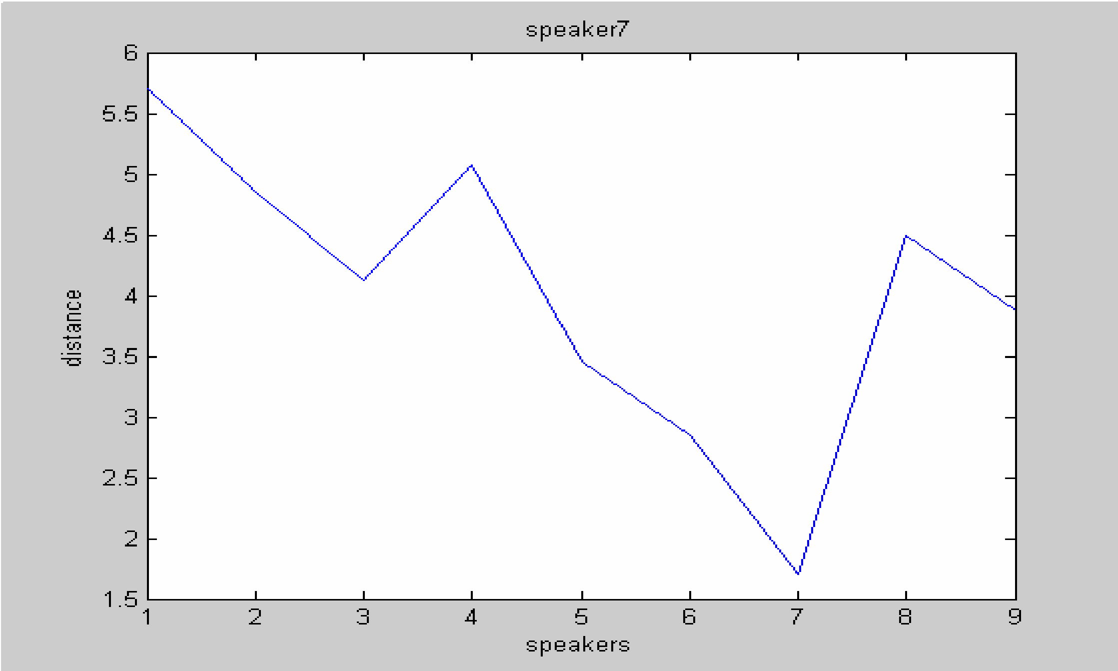


Figure 6.8 Plot for the Euclidean distance between the speaker 7 and all speakers

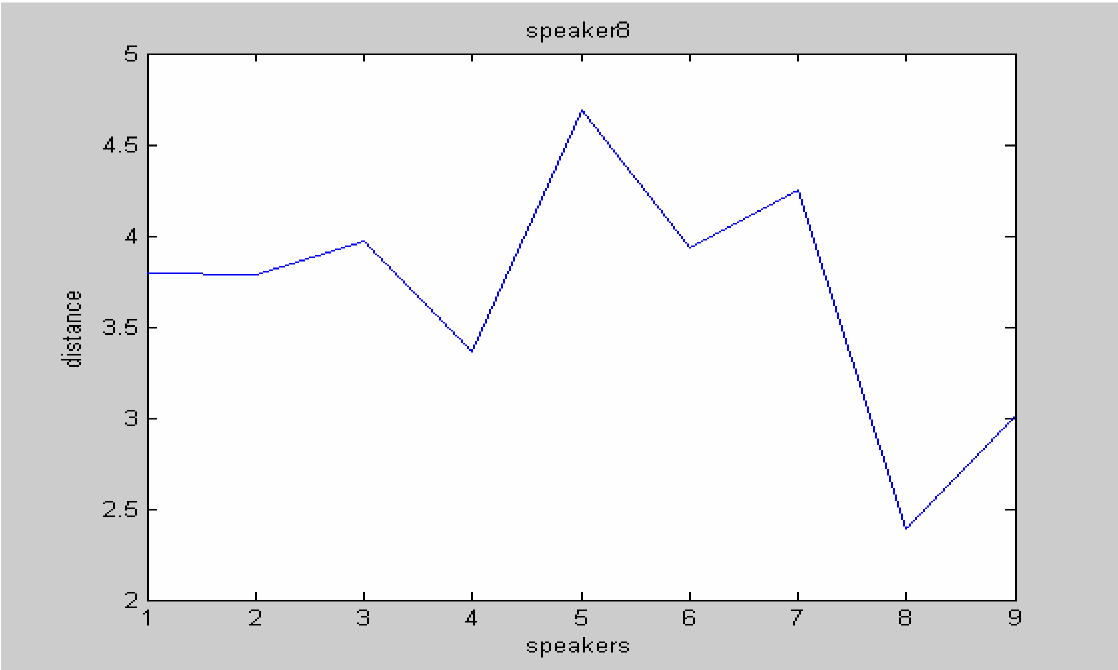


Figure 6.9 Plot for the Euclidean distance between the speaker 8 and all speakers

DSP RESULTS

The design is first tested with MATLAB. A total of eight speech samples from eight different people (eight speakers, labeled S1 to S8) are used to test this project. Each speaker utters the same single digit, *zero*, once in a training session (then also in a testing session). A digit is often used for testing in speaker recognition systems because of its applicability to many security applications. This project was implemented on the C6711 DSK and can be transported to the C6713 DSK. Of the eight speakers, the system identified six correctly (a 75% identification rate). The identification rate can be improved by adding more vectors to the training codewords. The performance of the system may be improved by using two-dimensional or fourdimensional VQ by changing the quantization method to dynamic time wrapping or hidden Markov modeling.

On kit board every program runs this much of clock cycles given clearly below.

1 clock cycle = 1/255 MHz

- FFT AND POWER SPECTRUM ESTIMATION = 77639929 Clock cycles.
- FFT = 71986984 Clock cycles;
- POWER SPECTRUM= fft & power spectrum estimation – fft

  = 5652945 clock cycles.
- MEL_FREQ. SPECTRUM = 51923798 clock cycles.
- LOG_ENERGY (log of the power spectrum)= 217615 clock cycles.
- MEL_FREQ_SPECTRUM + LOG_ENERGY = 52141413 clock cycles.
- MFCC_COEFF = 83508618 clock cycles.
- MEL_COEFF = (mfcc_coeff) – (mel_freq_spectrum +log_energy)

  = 31367205 clock cycles.

- Actual MFCC_SPECTRUM = 43650346 clock cycles.
- Actual MFCC_SPECTRUM + LOG_ENERGY = 43869656 clock cycles.
- Actual LOG_ENERGY = (mfcc_spectrum) - (mfcc_spectrum + log_energy)

  = 219310 clock cycles.
- Overall program time = 168918997(almost less than 1sec)

LESS THAN 1SEC GENERATION IS COMPLETING THE PROGRAM.

Chapter 8

CONCLUSION AND FUTURE WORK

CONCLUSION

The results obtained in this project using MFCC and VQ are applaudable. I have computed MFCCs corresponding to each speaker and these are vector quantized. The VQ distortion between the resultant codebook and MFCCs of an unknown speaker is taken as the basis for determining the speaker's authenticity. Here I used MFCCs because they follow the human ear's response to the sound signals.

The performance of this model is limited by a single coefficient having a very large VQ distortion with the corresponding codebook. The performance factor can be optimized by using high quality audio devices in a noise free environment. There is a possibility that the speech can be recorded and can be used in place of the original speaker .This would not be a problem in our case because the MFCCs of the original speech signal and the recorded signal are different. Psychophysical studies have shown that there is a probability that human speech may vary over a period of 2-3 years. So the training sessions have to be repeated so as to update the speaker specific codebooks in the database.

Finally I conclude that although the project has certain limitations, its performance and efficiency have outshined these limitations at large.

FUTURE WORK

There are a number of ways that this project could be extended. Perhaps one of the most common tools in speaker identification is the *Hidden Markov Model (HMM)*. This uses theory from statistics in order to (sort of) arrange our feature vectors into a Markov matrix (chains) that stores probabilities of state transitions. That is, if each of our codewords were to represent some state, the HMM would follow the sequence of state changes and build a model that includes the probabilities of each state progressing to another state. This is very useful, since it gives us another way of looking at speaker-unique characteristics.

This system is a text dependent system we can design a system which is text independent where the speaker is given with a freedom that he can speak anything in order to get recognized. There are also other weighting algorithms (and indeed VQ techniques) that I could have implemented, one of which is suggested in. It would be an interesting study to determine the effectiveness of more complicated methods such as this one.

Chapter 9

APPLICATIONS

Automatic Speaker Recognition Systems are being used in many applications such as.

- Time and Attendance Systems
- Access Control Systems
- Telephone-Banking/Booking
- Biometric Login to telephone aided shopping systems Information and Reservation Services
- Security control for confidential information Forensic purposes
- Voice command and control
- Voice dialing in hands-free devices
- Credit card validation or personal identification number (PIN) entry in security systems

REFERENCES

[1] Campbell, J.P., Jr.; **"Speaker recognition: a tutorial"** Proceedings of the IEEE Volume 85, Issue 9, Sept. 1997 Page(s):1437 – 1462.

[2] Seddik, H.; Rahmouni, A.; Sayadi, M.; **"Text independent speaker recognition using the Mel frequency cepstral coefficients and a neural network classifier"** First International Symposium on Control, Communications and Signal Processing, Proceedings of IEEE 2004 Page(s):631 – 634.

[3] Childers, D.G.; Skinner, D.P.; Kemerait, R.C.; **"The cepstrum: A guide to processing"** Proceedings of the IEEE Volume 65, Issue 10, Oct. 1977 Page(s):1428 – 1443.

[4] Roucos, S. Berouti, M. Bolt, Beranek and Newman, Inc., Cambridge, MA; **"The application of probability density estimation to text-independent speaker identification"** IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '82. Volume: 7, On page(s): 1649- 1652. Publication Date: May 1982.

[5] Castellano, P.J.; Slomka, S.; Sridharan, S.; **"Telephone based speaker recognition using multiple binary classifier and Gaussian mixture models"** IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 Volume 2, Page(s) :1075 – 1078 April 1997.

[6] Zilovic, M.S.; Ramachandran, R.P.; Mammone, R.J **"Speaker identification based on the use of robust cepstral features obtained from pole-zero transfer functions"**.; IEEE Transactions on Speech and Audio Processing, Volume 6, May 1998 Page(s):260 - 267

[7] Davis, S.; Mermelstein, P, **"Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences"** , IEEE Transactions on Acoustics, Speech, and Signal Processing Volume 28, Issue 4, Aug 1980 Page(s):357 – 366

[8] Y. Linde, A. Buzo & R. Gray, **"An algorithm for vector quantizer design"**, IEEE Transactions on Communications, Vol. 28, issue 1, Jan 1980 pp.84-95.

[9] S. Furui, **“Speaker independent isolated word recognition using dynamic features of speech spectrum”**, IEEE Transactions on Acoustic, Speech, Signal Processing, Vol.34, issue 1, Feb 1986, pp. 52-59.

[10] Fu Zhonghua; Zhao Rongchun; **“An overview of modeling technology of speaker recognition”**, IEEE Proceedings of the International Conference on Neural Networks and Signal Processing Volume 2, Page(s):887 – 891, Dec. 2003.

[11] Moureaux, J.M., Gauthier P, Barlaud, M and Bellemain P.”**Vector quantization of raw SAR data”**, IEEE International Conference on Acoustics, Speech, and Signal Processing Volume 5, Page(s):189 -192, April 1994.

[12] Nakai, M.; Shimodaira, H.; Kimura, M.; **“A fast VQ codebook design algorithm for a large number of data”**, IEEE International Conference on Acoustics, Speech, and Signal Processing, Volume 1, Page(s):109 – 112, March 1992.

[13] Xiaolin Wu; Lian Guan; **”Acceleration of the LBG algorithm”** IEEE Transactions on Communications, Volume 42, Issue 234, Part 3Page(s):1518 - 1523, February/March/April 1994.

[14] B. P. Bogert, M. J. R. Healy, and J. W. Tukey: **"The quefrency alanysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-cepstrum, and saphe cracking".** Proceedings of the Symposium on Time Series Analysis (M. Rosenblatt, Ed) Chapter 15, 209-243. New York: Wiley, 1963.

[15] Martin, A. and Przybocki, M., **“The NIST Speaker Recognition Evaluations: 1996-2000”**, Proc. OdysseyWorkshop, Crete, June 2001

[16] Martin, A. and Przybocki, M., **“The NIST 1999 Speaker Recognition Evaluation—An Overview”**, Digital Signal Processing, Vol. 10, Num. 1-3. January/April/July 2000

[17] Claudio Becchetti and Lucio Prina Ricotti, “Speech Recognition”, Chichester: John Wiley & Sons, 2004.

[18] John G. Proakis and Dimitris G. Manolakis, “Digital Signal Processing”, New Delhi: Prentice Hall of India. 2002.

[19] Rudra Pratap. Getting Started with MATLAB 7. New Delhi: Oxford University Press, 2006

[20] R. Chassaing, *DSP Applications Using C and the TMS320C6x DSK, Wiley, New York*, 2002.