Speaker Identification
4. SPEAKER IDENTIFICATION

The proposed research work is a text dependent, closed set recognition system, used in healthcare environment for authenticating a person to use the speech recognition system. The techniques used for speaker identification and methods for improving the speed and accuracy of the system are discussed in this chapter.

4.1. NEED FOR SPEECH AUTHENTICATION TOOL IN ASR

Authentication is the process by which an individual proves his/her identity to the computer. The purpose of an authentication system is to protect against/granting system access to unauthorized individuals (Guideline for the Use of Advanced Authentication Technology Alternatives, 1994). A survey of healthcare professionals, conducted in May-June 2000 and published by the Medical Records Institute, reported that 59% (of 273 respondents) felt that access to patient record information by unauthorized users was a “major concern” regarding the security of patient data (Survey of Electronic Health Record Trends and Usage, 2000). Similarly, information security evaluations conducted at Medical Treatment Facilities (MTFs) as part of the Defense Healthcare Information Assurance Program (DHIAP) found that the security of patient information is at risk (Andrews et al., 2000).

A statement made by Marcel Koster about system security in the healthcare industry reflects the state of user authentication in the MTF environment today (Michael, 2000). He opined that too often, front-end access is restricted only with low-level safeguards like passwords, usernames, personal identification numbers (PINs), and ID cards. It follows from Koster’s statement that in most healthcare organizations entry to the system is compromised with a cascaded effect of compromised access to connected systems/applications. Sensitive patient details
require more protection than the one already existing. This problem is solved by using authentication.

Good security practices and emerging healthcare regulations indicate that controlling user access (i.e., authorize only those users with a need-to-know to access data and applications) to patient details is essential for auditing the actions taken by the authorized users. Effective user authentication establishes the foundation for the closely related security practices of access control and audit (Alexander and McElveen, 2007).

The passwords have several disadvantages as they are shared among users and users share active terminal sessions and therefore are often considered weak (i.e., easy to guess). They are often written down rather than committed to memory and users attempt to establish a single password for their accounts on multiple systems, creating the weakness that discovering one password exposes access to all accounts. Lack of consistent procedures for deactivating user accounts at the time of a user's duty reassignment or for reinitializing changing passwords also create problems in many hospitals. Using smartcard as a tool for authentication suffers from the fact that the instrument can be lost or stolen.

These authentication problems are the main motivation point for investigating an alternative method for authentication using speech signals to develop an effective, secure, and at the same time, user-friendly system. Authentication based on speech signal is probably the least subject to compromise, and interest in them is increasing while costs are decreasing. It could be used in computer login, a “key” to a physical facility, or in border control (Kinnunen, 2003). Voice identification is probably the least intrusive of the biometric authentication methods. It is best suited to use in areas without extensive background noise (particularly other human voices) and free of voice-altering
stress, in other words, it is well-suited for the authentication of doctors and practitioners.

4.2. TECHNIQUES USED IN THE PROPOSED RESEARCH WORK

This section explains the techniques that are used in the proposed algorithm. The techniques described here are also used in the speech recognition module explained in Chapter 5.

4.2.1. Feature Extraction

Feature extraction plays an important role in both enrolment and identification phase. A feature is an idiosyncratic element of speech and even though the features are not readily heard by the humans can be identified through digital analysis. In this research work, Mel Frequency Cepstral Coefficients (MFCC) technique is used to extract features from the speech signal. The MFCC is the most widely used feature in speaker identification (Wu and Cao, 2005; Molau et al., 2001). The process of MFCC is shown in Figure 4.1.

MFCC are coefficients that represent speech based on perception. Hamming Windowing was used to avoid problems that may arise due to truncation of signals. In this technique, the frequency bands are positioned logarithmically and closely approximate to the human system responses, which allow better processing of data.

The concept here is to minimize the spectral distortion by using the window to taper the signal on both ends thus reducing the side effects caused by signal discontinuity at the beginning and at the end due to framing. In the present research work, each frame is multiplied by a Hamming window function (Equation 4.1 and 4.2) and once the framed and windowed speech signal samples are obtained, the frame of N samples is converted from time to frequency domain using Fast Fourier Transform (FFT).
\[ y_i(n) = x_i(n) w(n), \ 0 \leq n \leq N-1 \] (4.1)

where

\[ w(n) = 0.54 - 0.46 \cos \left( \frac{2\pi n}{N-1} \right), \ 0 \leq n \leq N-1 \] (4.2)

The physco-acoustical 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 ‘\( t \)’ measured in Hz, a subjective pitch is measured on a scale called the ‘Mel Scale’. The Mel frequency scale is a linear frequency spacing below 1000 Hz and logarithmic spacing above 1kHz. As a reference point, the pitch of a 1 kHz tone, 40 dB above the perceptual hearing
threshold, is defined as 1000 Mels. Therefore, the following approximate formula is used to compute the Mels for a given frequency \( f \) in Hz:

\[
F_{\text{mel}} = \frac{1000}{\log 2} \left[ 1 + \frac{F_{\text{hertz}}}{1000} \right].
\]  

(4.3)

In the next step, critical band filters are applied to the Mel scale to map the log amplitudes of the spectrum onto the Mel scale using triangular overlapping windows. The Discrete Cosine Transform of the Mel log-amplitudes is performed to transform/convert the spectrum back to time domain. The resulting coefficients are the MFCC coefficients.

4.2.2. Vector Quantization

Quantization, the process of approximating continuous amplitude signals by discrete signals, is an important aspect of data compression or coding, the field concerned with the reduction of the number of bits necessary to transmit or store analogue data, subject to a distortion or fidelity criterion. The independent quantization of each single value or parameter is termed scalar quantization. In contrast, the joint quantization of a block of parameters is termed Vector Quantization (VQ). VQ plays an important role in describing the acoustic prototypes of speech signals that is very important in speaker pruning, speaker identification and recognition processes. This section gives a brief overview of the working of Vector Quantization.

Advantages of VQ

The key benefits of using VQ in Speaker Identification and Speech Recognition systems are:

- **Reducing storage for spectral analysis information** – Spectral analysis significantly reduces the information rate of the speech data. When
comparing the raw (uncoded) speech data with that of spectral analysis speech data, it can be seen that spectral analysis significantly reduces the required information rate. For example, 10-KHz sampled speech with 16-bit pitch amplitude requires 160 kpbs signal information rate to store the speech sample in PCM format. For the spectral analysis, consider vectors of dimension \( p = 10 \) using a 10 ms frame rate. If each spectral component is represented to 16-bit precision, the required storage is about 16 kbps. It is a 10-to-1 reduction over the uncompressed signal. Such compressions in storage rate are very remarkable and are very much desired.

- **Reducing 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 spectral similarity computation is often reduced to a table lookup of similarities between pairs of codebook vectors.

- **Discrete representation of speech sounds** – By associating a phonetic label (or possibly a set of phonetic labels or a phonetic class) with each codebook vector, the process of choosing a best codebook vector to represent a given spectral vector becomes equivalent to assigning a phonetic label to each spectral frame of speech. A range of recognition systems exists that exploit these labels so as to efficiently recognize speech.

As mentioned previously, the main function of VQ is to take a large set of feature vectors and produce a small set of measure vectors that represent the centroids of the distribution. Figures 4.2a and 4.2b shows the feature vectors before and after the application of VQ.
Functioning of VQ

The main function of VQ is to reduce data redundancy. This inevitably causes distortion between original data and transmitted data. A key point of VQ is to minimize the distortion. A set of parameters is quantized as a whole, minimizing the global distortion. The finite set of possible quantized vectors is stored in a codebook. After quantization, the input parameter vector is represented by the corresponding label of the codebook entry that shows the smallest distortion. The goal of VQ is to generate a number of VQ codewords from a large sample of training vectors such that the codewords can represent the distribution
of the training vectors and minimize the total distortion over all training vectors. Figure 4.3 shows a schematic diagram of a VQ technique that can be used in speaker and speech recognition systems.

Template speech data is first used to generate the codebook. The speech signal is segmented (windowed) into successive short frames and each frame of speech is represented by a vector of finite dimensionality. The vector is the result of either the filter-bank analysis or the LPC analysis which captures the time-variant spectral characteristics of the speech signal.

![VQ Diagram](image)

**Fig. 4.3. Process of VQ**

In vector quantization, the real-valued, continuous-amplitude d-dimensional vector ‘x’ is mapped to another real-valued, discrete (or continuous)-amplitude d-dimensional vector ‘z’. It is then said that ‘x’ is quantized to ‘z’.

\[
Z = q(x)
\]

(4.4)

where
• $x = (x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$ is a d-dimensional vector whose components are real random variables.

• $q(x)$ is the quantization operator

• $z = \{z_1, z_2, \ldots, z_d\}$ typically takes one of a finite set of values $Z = \{z_i, 1 \leq i \leq L\}$ where $z_i = \{z_{i1}, z_{i2}, \ldots, z_{id}\}^t$

The set $Z$ is referred to as the codebook, $L$ is the size of the codebook, and $\{z_i\}$ is the set of codewords. The size $L$ of the codebook is also called the number of levels in the codebook. To design a codebook, the d-dimensional space of the original random vector $x$ can be partitioned into $L$ regions or cells $\{C_i, 1 \leq i \leq L\}$ and each cell $C_i$ is associated with a vector $z_i$. The quantizer assigns the codeword $z_i$ if $x$ lies in $C_i$.

$$q(x) = z, \text{ if } x \in C$$

(4.5)

This codebook design process is also known as training or populating the codebook.

An example of partitioning a two-dimensional space ($d = 2$) for the purpose of vector quantization is shown in Figure 4.4 (Weiye, 1999). The shaded region enclosed by the bold lines is the cell $C_i$.

Any input vector $x$ that lies in the cell $C_i$ is quantized as $z_i$. The shapes of the various cells can be different, and the positions of the codewords corresponding to the cells are determined by minimizing the average distortion $D_i$ associated with the corresponding cells. The positions of the codewords within each cell are shown by dots in Figure 4.4.
When \( x \) is quantized as \( z \), a distortion measure \( d(x, z) \) can be defined between \( x \) and \( z \) to measure the quantization quality. The distortion measure between \( x \) and \( z \) is also known as a distance measure in the speaker and speech recognition context. The measure must be tractable in order to be computed and analyzed, and also must be subjectively relevant so that differences in distortion values can be used to indicate differences in speech signals. The most commonly used measure is the Euclidean distortion measure which assumes that the distortions contributed by quantizing the different parameters are equal.

### 4.2.3. Clustering

Clustering by definition produces localized group of items, which implies that the results depend on the used similarity measure. One of the typical methods for clustering huge volume of speech data is K-Means algorithm. The reason behind this is that it requires only \( O(k, N) \) computation for a given number of clusters \( k \) and sample size \( N \) (Deng et al., 2000).
The major disadvantage of using K-means algorithm for clustering is the optimal selection of the number of clusters, k. This pre-determination of the number of clusters is a strict restriction, which can cause serious fluctuation on the computation overhead and is one of the most essential issues in clustering. Normally, as a simple approach, the results of multiple runs with different K values are compared and the best one is chosen according to a criterion (for instance, the Schwarz Criterion). When K equals to the optimum number of clusters, the algorithm can correctly find out the clustering centres. Otherwise, it will lead to an incorrect clustering where it becomes difficult to locate the centres of the corresponding clusters. Instead, they are either at some boundary points among different clusters or at points biased from some cluster centres.

Because of the above-mentioned disadvantages, it is desirable to automatically calculate the optimal number of clusters (Min-Joung et al., 2003). Automatic detection of K while clustering is not a new concept. It has been used in many researches in data mining (Wu and Zhou, 2002; Yin et al., 2004), image processing (Zhang et al., 2004), etc. However, to the best of the knowledge, it is new to speech data clustering and is explained in the following section.

4.2.4. Modified K-Means Method

The procedure of K-means proposed by is as follows:

1. Get the initial K-features from the feature set and set each of them as initial clusters, so that each cluster will have at least one feature in it.
2. Allocate the remaining data to the nearest neighbourhood cluster centers.
3. Calculate the cluster centers and regard them as fixed seeds.
4. Repeat steps 1-3 to allocate all data to the nearest neighbour cluster seeds, until the cluster centers converge.
This research work proposes a modified K-means algorithm based on Bayesian Information criterion (BIC) to estimate the optimal value of K. The algorithm initially divides speech data into two clusters and continues to divide each of the clusters into two more clusters. Bayesian Information Criterion (Duffie, 1987; Schwarz, 1978) is used as the diversion criterion. The reason for using BIC are:

- BIC considers the selection among from exponential family of distribution
- BIC is based on prior probability rather than the distance between two distributions

The modified K-means algorithm is explained below.

Step 1 : Prepare p-dimensional data whose sample size in n

Step 2 : Set \(k_0 = 2\), where \(k_0\) is the initial number of clusters

Step 3 : Apply K-means to all data with setting \(k = k_0\), the divided clusters are named \(C_1, C_2, \ldots, C_{k_0}\)

Step 4 : For each cluster \(C_i\), apply k-means by setting \(k = 2\). The divided clusters are named as \(C_i^{(1)}\) and \(C_i^{(2)}\)

Step 5 : Assumption : The following p-dimensional normal distribution for the feature \(x_i\) contained in \(C_i\)

\[
f(\theta_i, x) = (2\pi)^{-p/2} |V_i|^{-1/2} \exp\left(\frac{1}{2} (x - \mu_i)^T V_i^{-1} (x - \mu_i)\right)
\]

then calculate BIC as

\[
BIC = -2 \log L(\hat{\theta}_i, x_i \in C_i) + q \log n_i
\]

where \(\hat{\theta}_i = [\hat{\mu}_i, \hat{V}_i]\) is the maximum likelihood estimate of the p-
dimensional normal distribution; \( \mu_i \) is \( p \)-dimensional means vector and \( V_i \) is \( p \times p \) dimensional variance-covariance matrix; \( q \) is the number of parameters dimension and it becomes \( 2p \) if covariance \( V_i \) is ignored. \( x_i \) is the \( p \)-dimensional data contained in \( C_i \); \( n_i \) is the number of elements contained in \( c_i \). \( L \) is the likelihood function which indicates \( L(.) = \prod f(.) \). The present research work ignores \( V_i \) and therefore \( q = 2p \).

**Step 6:** Assumption: The \( p \)-dimensional normal distributions and their parameters \( \theta_i^{(1)}, \theta_i^{(2)} \) for \( C_i^{(1)}, C_i^{(2)} \) respectively. The probability density function this 2-division model becomes

\[
G(g(\theta_i^{(1)}, \theta_i^{(2)}, x)) = \alpha_i \left[ f(\theta_i^{(1)}, x) \right] \left[ f(\theta_i^{(2)}, x) \right]^{-\delta_i}
\]

where

\[
\delta_i = \begin{cases} 
1, & \text{if } x \text{ is included in } C_i^{(1)} \\
0, & \text{if } x \text{ is included in } C_i^{(2)}
\end{cases}
\]

\( x_i \) will be included in either \( C_i^{(1)} \) or \( C_i^{(2)} \), \( \alpha_i \) is a constant which lets \( g(\theta, x) \) be a probability density function \( (1/2 \leq \alpha_i \leq 1) \). \( \alpha_i \) is approximated as below:

\[
\alpha_i = 0.5 / K \beta_i
\]

where \( \beta_i \) is a normalized distance measure between two clusters, shown as below

\[
\beta_i = \sqrt{\frac{||\mu_1 - \mu_2||^2}{|V_1| + |V_2|}}
\]

80
$K(.)$ stands for an lower probability of normal distribution. The BIC for this model is

$$\text{BIC}' = -2 \log L'(\hat{\theta}_i, x_i \in C_i) + q' \log n_i$$

where $\hat{\theta}_i = [\hat{\theta}_i^{(1)}, \hat{\theta}_i^{(2)}]$ is a maximum likelihood estimate of two $p$-dimensional normal distributions; since there are two parameters of mean and variance of each $p$ variable, the number of parameters dimension becomes $q' = 2 \times 2p = 4p$. $L'$ is the likelihood function which indicates $L'(.) = \prod g(.)$.

Step 7: If $\text{BIC} > \text{BIC}'$, then the two-divided model is preferred and $C_i$ is set to $C_{i}^{(1)}$. As for $C_{i}^{(2)}$, the $p$-dimensional data, cluster centers, log likelihood and the BIC are pushed onto the stack. Return to Step 4.

Step 8: If $\text{BIC} \leq \text{BIC}'$, the clusters are not divided any more and we stop the process. The stacked data is extracted and $C_i$ is set to $C_{i}^{(2)}$. Return to Step 4. When the stack is empty, proceed to Step 9.

Step 9: The 2-division procedure for $C_{i}$ is complete. The identification of the clusters are renumbered such that it becomes unique in $C_{i}$.

Step 10: The 2-division procedure initial $k_0$ divided clusters is completed. All the cluster are renumbered so that it becomes unique in $C_{i}$.

Step 11: Output the cluster identification number to which each feature is allocated, the center of each cluster, the log likelihood of each cluster and the number of elements in each cluster. This is the code book.
When a new IP is received, the features are extracted using MFCC. These extracted feature vectors are used to identify the appropriate cluster. The cluster is chosen such that the distance between feature vectors, extracted from the IP and cluster representative (i.e. centroid codebook) is minimal. The use of modified K-means clustering algorithm successfully prunes all the speaker models that do not match the IP. This reduction increases the speed of the matching algorithm.

4.2.5. Neyman-Pearson Likelihood Ratio Test

Voice based authentication is one of the candidates for secure systems, where Neyman-Pearson likelihood ratio test is predominantly used to improve their performance (Wenndt and Noga, 2004; Sargin et al., 2006). Existing techniques schemes result in higher computation and storage complexity, which renders them unsuitable for embedded applications that, must satisfy various real time constraints in a hospital environment. To address these concerns, this research work presents a doctor identification system using a multi-step approach based on five tests. The four tests are speech length, cross correlation, frequency multiplication, frequency cross correlation, peak signal comparison. The results of these tests are combined using the Neyman-Pearson Criterion.

The Neyman-Pearson criterion leads the test statistics for doctor identification as follows:

\[
T = \begin{cases} 
1 & \text{if } \frac{P(O|\lambda_{\text{tar}})}{P(O|\lambda_{\text{si}})} > T_\alpha \\
M & \text{if } \frac{P(O|\lambda_{\text{tar}})}{P(O|\lambda_{\text{si}})} < T_\alpha \\
n & \text{if } \frac{P(O|\lambda_{\text{tar}})}{P(O|\lambda_{\text{si}})} > T_\alpha \\
M & \text{if } \frac{P(O|\lambda_{\text{tar}})}{P(O|\lambda_{\text{si}})} < T_\alpha
\end{cases}
\]  

(4.6)
where $O$ is the speech data, $P(O/\lambda_{\text{tar}})$ are the likelihoods from template speech model and sample input speech data and $T_\alpha$ is the threshold chosen beforehand. Each test is associated with two probabilities. They are

(a) Probability of successful detection ($p$)
(b) Probability of unsuccessful detection ($u$)

In order for the test to be effective $p$ must be greater than $q$. If $q$ is greater than $p$, then the test should be reversed.

4.3. DOCTOR RECOGNITION SYSTEM

Doctor recognition systems require speech signals of a previously enrolled phrase to be compared with the spoken input to identify the speaker. The speech signals previously enrolled are called the “Speech Reference Template (SRT)” and the signal submitted for identification is called the “Input Phrase (IP)”.

To enable remote accessibility, the SIS works in a network environment with several nodes and a server. The speaker who wants to be identified speaks from a remote node. This IP is passed onto the server. The server has the SRT, which is examined with the IP received from the node and determines the identity of the speaker.

4.3.1. Enrolment and Identification Phase

SIS works in two phases: enrolment phase and identification phase. The enrolment phase and the identification phase used in the proposed work are illustrated in Figure 4.5 and Figure 4.6.
4.3.2. Speech Appraisal

Speech data (both IP and SRT) are captured from the remote computer by continuously recording the sound signals from the remote system. This might involve recording large numbers of frames which have both wanted and unwanted data. Unwanted data are those which do not have the required input signal and which are normally composed of silent or empty signals, where nothing is said. The first step is to remove these unwanted silent or empty signals.
Speech period detection task is performed to find out when the speech data starts or ends. While recording voice for identification from remote computer, it is very difficult to know when the actual data is sent by the node. To work out this dilemma, the server is made to continuously sample the sound data and the magnitude of these sample sound is examined. When the magnitude becomes high, it is considered as the start of the speech sample. Similarly, after a high magnitude sample, a sudden dip indicates the end of the sample data. Figure 4.7 shows the speech wave for the word “Hello World” after detection.
While analyzing the input signal, care must be taken to ensure that the system does not spend time examining empty space. Empty or silent data occur frequently while recording speech, which cannot be avoided. Removing silent speech greatly enhances the identification / execution time of the proposed algorithm (Fig. 4.8a and 4.8b).

Fig. 4.8a. Speech signal with silent speech

Fig. 4.8b. Speech signal after silent speech is removed

In the proposed system, the silent data is identified by computing of variations of the signal samples in speech frame, against the frame’s mean. If variations are big enough, the frame is considered as a speech frame, otherwise is taken as a silence. The algorithm used for silent region detection is given in Figure 4.9.

1. Compute the mean of the frame samples.
2. Collect the cumulative sum of absolute magnitude of differences between the samples
3. If this sum exceeds predefined threshold, then the frame is considered as a speech frame, otherwise as a silent frame.

Fig. 4.9 : Algorithm – Silent Detection
The process is represented in Equations (4.7) and (4.8):

\[
\mu = \frac{1}{N} \sum_{n=1}^{N} S_k(n) \tag{4.7}
\]

\[
\Omega = \sum_{n=1}^{N} |S_k(n) - \mu| \tag{4.8}
\]

where \(S_k(n)\) is the signal samples for frame \(k\), \(\mu\) is a mean and \(\Omega\) is a cumulative sum, compared with the threshold.

Although this method is quite simple, it has a prominent advantage of speeding up the identification process (Fig. 4.6) as the number of speech frames taken for comparison becomes small. For example, an input signal with 13,120 samples before cleaning was reduced to 2,602 samples after it was cleaned (Fig. 4.10).

The advantages of removing silent speech or empty space from the input signal are

- Speech signal becomes shorter
- Amount of time spent to perform calculations is reduced
- Speech signal after removal of silence has all the important data that are required to perform speech analysis.

The cleaned output signal is taken as input to the second step of speech appraisal, that is, the length of speech test. The length of speech is a measure that is very vital while testing speaker identity. Speech signal normally can have too much or too little data. Both of these types are useless for speaker identity. If the sample is very small, then the speech signal probably might not have the input data, instead has been initiated through background noise generated by motor or fan running sound that are near to the microphone in the remote environment.
Fig. 4.10: Input Speech Samples before and after cleaning
If this is the scenario, then these noise signals have to be ignored. Increase in the length of the input signal might be because of high pitch in the background noise that it results in constant recording or it might be because the user is issuing commands very quickly without any pause. The sample signal passes this step if the length of the sample is within a percent threshold of the length of the template. This test is mainly designed to prevent false positives.

4.3.3. Speaker Identification Process

When a new IP is received, the features are extracted using MFCC. Similarly, the SRT is also feature extracted and clustered using techniques explained in the previous section. The centroid of IP is used to identify the appropriate cluster from the SRT. The cluster is chosen in such a way that the similarity between IP and cluster representative (i.e. centroid codebook) is maximum. After identifying the cluster, the matching algorithm (explained below) is executed to identify the correct speaker. The use of modified K-means clustering algorithm successfully prunes all the speaker models that do not match the IP which increases the speed of the matching algorithm to a great extent.

4.3.4. Matching Algorithm

The second step of the system is speaker identification, where attempts are made to identify the speaker from the source input signals. This phase relies heavily on frequency analysis because each person has unique voice characteristics that can be isolated in the frequency domain. In order to confirm that the identity of the person is determined accurately, the proposed system performs four tests, which are applied in a multi-step fashion on the input signal frames. The results of the four tests are consolidated using Neyman-Pearson Likelihood ratio (explained in the previous section).
The four tests used are,

- Cross correlation test (Test 1)
- Frequency multiplication test (Test 2)
- Frequency cross correlation test (Test 3)
- Peak signal comparison test (Test 4).

The four tests are applied to the speech data (both SRT and IP) during speaker identification process. The four tests are named as Test 1, Test 2, Test 3 and Test 4 respectively in the dissertation.

The process of using these four tests in a multi-step fashion for speaker identification is illustrated in Figure 4.11.

1. Cross correlation Test (Test 1)

Test 1 uses cross correlation to identify speakers from the input signal. This test is very effective test, but requires a large amount of computation to perform. For example, if two samples each of length n=2000 frames are compared, a total of 4000000 operations are required to complete the operation. In spite of the computation overhead, the result of cross-correlation gives a good indication on the similarity of the input signal and template signal. After the computation of correlation, a high peak in the middle of the cross-correlation figure (Fig. 4.12) indicating that the two signals being correlated are similar.
IP and Cluster speaker models

Fourier Transform

Power Spectrum

Cross Correlation

Frequency Multiplication

Frequency Cross Correlation

Peak Signal Comparison

Neyman-Pearson likelihood ratio test

False

> TH

True

SPEAKER NOT RECOGNIZED

SPEAKER RECOGNIZED

Fig. 4.11: Matching Process
The above figure shows the cross correlation of the signal with itself (autocorrelation). In the figure, the peak value is close to 200, indicating that they are similar. To accept the signal, a threshold value is used. If the peak value is greater than this threshold value, then the signal is accepted otherwise ignored. Correlation, a measure of similarity between two signals, is frequently used in the analysis of speech and other signals. The cross-correlation between two discrete-time signals $y[n]$ and $x[n]$ is defined as the cross correlation is calculated using the formula,

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} y(n)x(n-l)$$

(4.9)

where $n$ is the sample index, and $l$ is the lag or time shift between the two signals. Since speech signals are not stationary, the similarity between signals was tested only over a short time duration (<30 ms). In this case, the cross-correlation is
computed only over a window of time samples and for only a few time delays $l=\{0,1,\ldots,P\}$.

**Fourier Transformation**

The Fourier transformation examines the power spectrum of the signals. It is computed using the formula,

$$x(e^{j\theta}) = \sum_{n=-\infty}^{\infty} x[n]e^{-jn\theta}$$

The power spectrum is computed by multiplying the Fourier transform coefficients by their complex conjugate. The equation given above has the restriction that the length of the spectrum to be equal to the signal. This restriction is overcome, by dividing the signal into smaller chunks, calculate the Fourier transform for each chunk and sum them cumulatively. This allows the resulting spectra to be of a specified length and it includes data from the entire sampling range. The pseudo code of modified Fourier transform is given in Figure 4.13 and Figure 4.14 shows a power spectrum of a sample signal.

```
procedure modified_FT
    for $i = 1$ to $n$
        $X(:, i) = 0$;
    end for;
    $j=0$;
    while ($j < \text{length}(x)-n$)
        for $i=1$ to $n$
            $\text{tmp}(i) = x(j+1)$;
        end for;
        $X = X + \text{fft}(\text{tmp}, n)I$
        $j = j + n$;
    end while;
```

**Fig. 4.13 Modified FT Algorithm**
2. Frequency Multiplication Test (Test 2)

Frequency multiplication is the most important test in speaker recognition. First, the normalized power spectrum of the sample is calculated for each template. This is done by identifying the locations in the power spectrum where both signals have peaks (Fig. 4.15). This test works because only peaks that show up in both signals will be passed on to the resulting calculation. Each element, then, is multiplied by the corresponding element in the power spectrum of the template. This value is stored. The values of the entire resulting spectrum are cumulatively added. The result is compared with a threshold. If the sum is greater than the threshold value, then the sample has passed the third test.
3. Frequency Cross Correlation Test (Test 3)

Frequency cross correlation test is similar to frequency multiplication test. The only difference is instead of performing a cross multiplication, a cross correlation is performed. The power spectrum of the sample is cross correlated with that of the template. The highest value from the correlation is compared against a threshold. If it is greater, then it passes the fourth test.

4. Peak Signal Test (Test 4)

The fourth test compares the number of peaks in the power spectrum of the sample and the template. This is done by counting the number of times the power spectrum is above a certain threshold. If this value of the sample signal is close to the template value, then it is accepted by fourth test.

The result of these tests ensures that the correct speaker is identified.
4.4. EXPERIMENTAL RESULTS

In this section, the simulation environment of the experimental results is discussed.

4.4.1. Simulation Environment

The enrolment phase takes care of the creation of speaker (doctor) database, which acts like a template during the authentication process. To ensure high identification rates and a large reliability of the identification results, each speaker has to register and has to be enrolled to the system, by speaking a certain amount of text, which is predefined.

The amount of speech from each of the enrolled speakers should be as high as possible to ensure a good performance within the identification process. On the other hand, the speech to be recorded in the enrolment phase should be as short as possible to minimize the inconvenience for the user (Bimbot et al., 2004). Keeping this in mind, during enrolment, the medical professionals were requested to utter a word or group of words that act as password in the ASSR system.

The number of words was limited to three words and care was taken to ensure that the speech signal was within 5 seconds duration. Speech signal is recorded in laboratory environment using the recorder facilities provided by the computer system. The sampling rate used on all speech samples throughout this research work is 8kHz and the compression format used to store speech data is “wav”. The database consisted of 10 sample speech signals from each speaker taken on different days, at different times, so as to accommodate voice variation that may occur due to external and internal factors.
4.4.2. Results

The performance of the speaker recognition algorithm is directly proportional with the success of the individual test algorithm’s result. Therefore, the analysis of the SIS system is started with the analysis of each of the four tests. The test results were obtained by calculating the probability of detection for different thresholds.

Figures 4.16 through 4.19 show the graphical representation of the result. Each point in the graph represents a different threshold. In the graphs, the x-axis represents the probability of false alarm and y-axis represents the detection probability. The slope indicates the maximum (100%) and minimum (0%) probability of successful detection. The aim of each test is to have plot values close to 100%. To find the probabilities associated with each threshold, a batch of samples (N=50) was given as input to the SIS system and the successful and unsuccessful detection of the speech signal was measured. This process was repeated for each threshold value chosen.

From the figures, it is clear that the performance order of the tests starting from the best result were test 1, test 2, test 3 and finally test 4. Test 1 and 2 had the best successful detection capacity without high false alarm rate (wrongly identifying or not identifying speech signal). These tests were given the highest weights when compared with all other tests combined. Test 3 also produced similar result, but not upto the mark when compared with test 1 and test 2. Test 4 was the least successful of the tests. They failed in the process of detecting non-speech signals.

After the result of each test is obtained, a threshold value is selected, by examining each plot and picking a value that is associated with the point closest to the top left corner. This ensures that an optimum threshold value is chosen for each test.
Fig. 4.16. Test 1 Results

Fig. 4.17. Test 2 Results
Fig. 4.18. Test 3 Results

Fig. 4.19. Test 4 Results
The results of all the tests are combined and are shown in Figure 4.20. This plot shows the success rate plotted against the false alarm rate for the entire algorithm. It can be seen from the figure that successful detection increases as the threshold value increases.

Another parameter that was used to evaluate the algorithm was the identification time. The identification time of the algorithm determines how quickly an input signal is recognized and mapped to an individual speaker in the system. Figure 4.21 shows the performance of the system when tested with different number of clusters.

It is evident from the figure, that the average time taken for identification is directly proportional with the number of candidate clusters selected for comparison. As the number of clusters increases, the time for matching process also increases, which in turn increases the execution time. Similar results were observed by Leung et al. (2004) who reported that the execution time grows with the number of speakers (in our case, clusters) known to the system in all closed set speaker identification system.
Fig. 4.20: Combined Results

Fig. 4.21: Pruning effect on Identification Time
4.5. SUMMARY

It is evident from the result that both the pruning algorithm and the matching algorithm performed satisfactorily with large set of speaker models. After correct identification, the system authenticates the person to use the speech recognition engine. The working of the proposed speech recognition engine and the results obtained while executing in a simulated environment is discussed in the next chapter.