

Text Dependent Voice Recognition System using MFCC and VQ for Security Applications

¹Ashwin Nair Anil Kumar

Heriot-Watt University, Dubai Campus
Dubai, U.A.E
aa1334@hw.ac.uk

²Senthil Arumugam Muthukumaraswamy

Department of Electrical and Electronics Engineering
Heriot-Watt University, Dubai Campus
Dubai, U.A.E
m.senthilarumugam@hw.ac.uk

Abstract—This paper presents the implementation of a practical voice recognition system using MATLAB (R2014b) to secure a given user's system so that only the user may access it. Voice recognition systems have two phases, training and testing. During the training phase, the characteristic features of the speaker are extracted from the speech signal and stored in a database. In the testing phase, the stored audio features of the test voice sample are compared with the voice samples in the database and determined if a match exists. For this research, Mel Frequency Cepstral Coefficients (MFCCs) were chosen to represent the feature vectors of the user's voice as it accurately simulates the behavior of the human ear. This characteristic of the MFCCs makes them an excellent measure of speaker characteristics. The feature matching process is then performed by subjecting the MFCCs to vector quantization using the LBG (Linde-Buzo-Gray) algorithm. In practical scenarios, noise is a major factor that adversely influences a voice recognition system. The paper addresses this issue by utilizing spectral subtraction to remove environmental noise affecting the speech signal thereby increasing the robustness of the system.

Keywords—*Speaker identification; voice recognition; mel frequency cepstral coefficients (MFCCs); vector quantization (VQ); spectral subtraction*

I. INTRODUCTION

Utilizing biometrics, i.e., the unique characteristics of a user such as fingerprints, voice, vein patterns, etc. is an efficient methodology of improving security. This is because biometric data cannot be forgotten or copied and is more difficult to hack when compared to traditional security systems.

Speech is the most natural form of communication for humans and the speech of every individual is unique. This is mainly due to biological factors such as size of vocal tract or due to behavioral characteristics such as accent, speaking speed, etc.

A voice recognition system is a biometric system that can identify an individual via the unique acoustic characteristics of the individual's voice. This system has a wide variety of applications in the field of security such as granting access to a system, a secure location and providing services such as telephone banking, voice dialing, surveillance, etc.

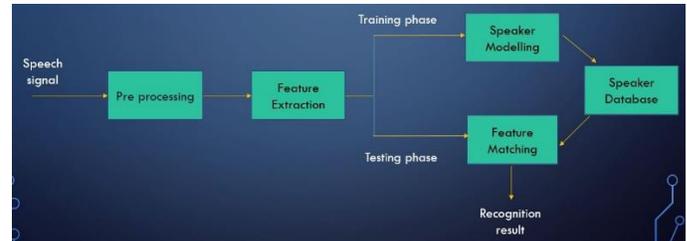


Fig. 1. System methodology.

Voice recognition systems have two phases, namely, training and testing. During the training phase, the speaker registers into the system by providing their voice sample so that the system can be trained to recognize them in the future. In the testing phase, the voice sample provided is compared with the existing samples in a database. If a match is obtained, the system will provide the user access. This paper aims to design a text dependent voice recognition system to be used to secure a given user's system so that only the user may access it. The system will also support a multi user database which the user should want to provide access to select a few.

The methodology of the voice recognition process illustrated in Fig. 1 consists mainly of three processes, namely, pre-processing, feature extraction and feature matching.

- *Pre-processing:*

The input speech signal is sampled and cut into frames. Speech is a highly variable signal and is difficult to analyze as it is. However, in very short intervals of time (in milliseconds) the signal appears to be stationary thereby facilitating analysis. After cutting into frames, the signal is then windowed to remove any discontinuities between the frames and reduce spectral leakages.

- *Feature Extraction*

The unique characteristics of an individual's voice is extracted and stored in a database for future referencing.

- *Feature Matching*

The audio characteristics of a voice sample are compared with the voice samples in the database and determine the existence of a match.

II. PRE-PROCESSING

The pre-processing stage of the voice recognition process consists of four steps as illustrated in Fig. 2:

- Sampling
- Frame Blocking
- Windowing
- Noise Removal.

A. Sampling

In order to analyze the analog speech signal, it has to be digitized (sampled and quantized) so as to convert it into a discrete time signal. Per the Nyquist Shannon theorem, signal $x(t)$ of frequency f_{max} needs to be sampled at a frequency of $2f_{max}$ for accurate signal realisation and to avoid aliasing. To have a discrete value signal, the sampled values are quantized which leads to a significant reduction of data.

Bit Depth/Bit Resolution is the number of bits used to store an audio sample. The number of bits utilized depends on the number of quantization levels used during the analog to digital conversion. Usually voice recognition systems encode samples with 8 or 16 bits depending on the available processing power. As the number of encoding bits increases more memory is required to store the audio file. For example, if an audio file was encoded using 8 bits there would be 256 (2^8) quantization levels. However, if the file was encoded using 9 bits, there would be 512 (2^9) quantization levels. Every bit of resolution doubles the number of quantization levels utilized. As quality of audio file is increased, the sampling frequency and file size increases as well. For this paper, the signal was sampled at 22.05 kHz with 8 bit encoding to achieve optimal clarity.

B. Frame Blocking

An important characteristic of speech is its property of quasi-stationarity, i.e., the signal while being highly variable in real time is stationary or considered to be stationary at very small intervals (in the order of milliseconds). Therefore, in order to analyze the signal, it must be split into frames of a few milliseconds. Another advantage of frame based analysis is that it improves the efficiency of the system by analyzing groups of samples (in a frame) as opposed to analyzing each sample separately.

Each frame overlaps with the previous frame so as to ensure a smooth transition of signal from one frame to another, i.e., less discontinuities. Large overlapping creates smoother transition of signals between frames but results in a smaller time shift in the signal which requires higher processing power. The ideal amount of overlapping is between 50% and 70%.

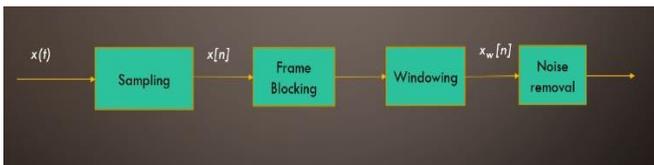


Fig. 2. Pre-processing stage.

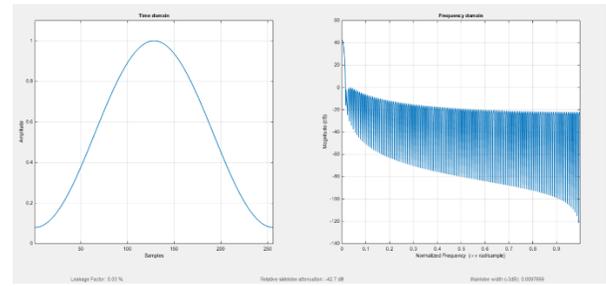


Fig. 3. 256-point Hamming window in time domain (left) and frequency domain (right).

The selection of frame length, i.e. number of samples that constitute a single frame is a crucial matter. If the frame length is too short, it does not possess enough samples to get a reliable estimate. However, if the frame length is too long, the signal varies too much throughout the frame thus defeating the purpose of framing in the first place. The ideal frame length lies in the range of 20ms – 40ms [1].

The continuous speech signal is blocked into frames of N samples each with adjacent frames being separated by M samples. The initial frame consists of N samples. The second frame begins M samples after the first one and overlaps it by $N - M$ samples. This process continues until the entire speech signal is within one or more frames. Referring previous studies [2], [3], typical number of samples for frame length (N) and overlap (M) are 256 and 100, respectively. This corresponds to 40ms frames separated by 15ms, i.e., a frame rate of 66.7Hz. The choice of number of samples per frame (N) was selected as 256 as it offers a compromise between poor frequency resolution and poor time resolution.

C. Windowing

The next step in the pre-processing phase is windowing. Windowing the frames is necessary to reduce any discontinuities present at the beginning and end of each frame. Windowing the signal is done by multiplying the window function ($w(n)$) with each frame. A simple rectangular window cannot be used as it abruptly cuts off the signal at its boundaries. To accomplish this task, a tapered window function needs to be applied as it removes discontinuities by tapering the signal value towards zero at the boundaries. The choice of the type of tapered windows is not a critical aspect of this process as any good window function would perform this task [4]. The commonly used Hamming window function was chosen for this task and is defined in (1).

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

Where, N represents the sample width.

Since each frame consists of 256 samples, a 256-point Hamming window as illustrated in Fig. 3 was utilized for this paper.

D. Noise Removal

Spectral subtraction was the method adopted to remove the noise from the speech signal. Spectral subtraction as discussed before is a noise suppression technique wherein an estimate of

the noise is obtained and subtracted from the speech signal. The general equation for the process is given in (2).

$$x(m) = y(m) - n(m) \quad (2)$$

Where, $y(m)$ represents the speech signal, $x(m)$ represents desired noise free signal and $n(m)$ represents the noise.

As per the spectral subtraction algorithm proposed by Boll [5], the speech signal is segmented into overlapping frames and windowed. The magnitude spectra of the windowed speech signal are calculated and the noise estimate calculated during the non-speech activity is subtracted from it. This is followed by the suppression of any residual noise present. Finally, a time domain waveform is recalculated from the modified magnitude which is then overlap added to the previous data to generate the noise free speech signal. A flowchart of this algorithm is illustrated in Fig. 4.

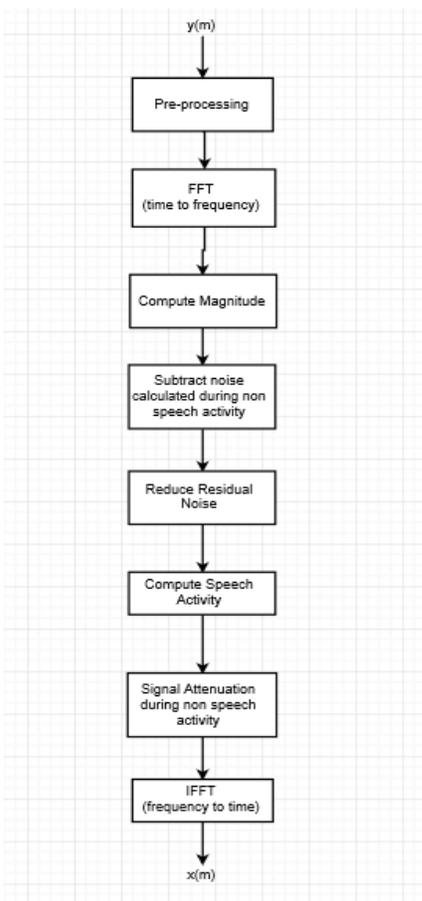


Fig. 4. Flowchart for Boll's spectral subtraction.

III. FEATURE EXTRACTION USING MFCC (MEL FREQUENCY CEPSTRAL COEFFICIENTS)

The mel cepstrum is obtained on applying a linear cosine transform of a log power spectrum on a non-linear mel scale. The coefficients that make up this cepstrum are called Mel Frequency Cepstral Coefficients (MFCCs). Extracting MFCCs from the speech signal consists of a few processes as illustrated in Fig. 5.

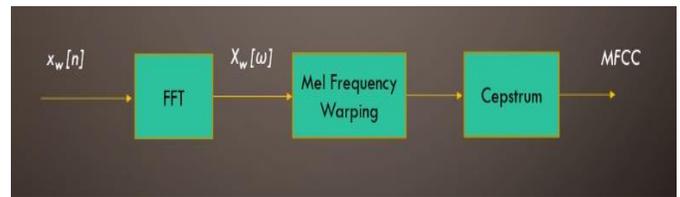


Fig. 5. Feature extraction methodology (MFCCs).

A. Fast Fourier Transform (FFT)

The first step in the feature extraction phase is to convert the speech signal into the frequency domain using the Fourier transform. The framed and windowed signal from the pre-processing stage is subjected to a Discrete Fourier Transform (DFT) in order to convert each speech frame from the time domain into frequency domain.

The DFT of the pre-processed speech frame is defined as:

$$X_w(k) = \sum_{n=0}^{N-1} x_w(n) e^{-\frac{j2\pi kn}{N}}, \quad k = 0, 1, 2, \dots, N-1 \quad (3)$$

Where, $x_w(n)$ = pre-processed speech frame and N is the length of the DFT.

The Fast Fourier Transform (FFT) is used as it is a fast algorithm for implementing the Discrete Fourier Transform (DFT) to obtain the frequency spectrum.

B. Mel Frequency Warping

Once the signal is converted into the frequency domain via the Fast Fourier Transform (FFT), the next step in the process is to warp the obtained frequency power spectrum according to the mel scale so as to convert it into the mel spectrum. To accomplish this, the power spectrum of the speech signal is weighted by a series of triangular filter frequency responses whose bandwidths and central frequencies match those of the auditory critical band filters. The filters convert the signal spectrum into a representation akin to the behavior of the human ear, i.e. the mel scale. The human ear is much more sensitive to lower frequencies (<1kHz) than the higher frequencies.

This is seen in the mel scale and is also why the filters are mel spaced in order to model this behavior (more filters in the lower frequency region than in the higher ones). The filters are called mel-scale filters and are collectively called mel-scale filter bank. (4) is used to convert the frequency spectrum to the mel spectrum where m represents mels and f represents the frequency.

$$m = 2595 * \log_{10} \left(1 + \frac{f}{700} \right) \quad (4)$$

Using (4), the linearly spaced bins of the frequency spectrum obtained using FFT, are converted into the mel spaced bins of the mel scale. The triangular mel filter bank is then designed per the new bins and are applied to the frequency spectrum to produce the mel spectrum. An example of the mel filter bank consisting of 24 filters used by Davis and Mermelstein [6] is shown in Fig. 6.

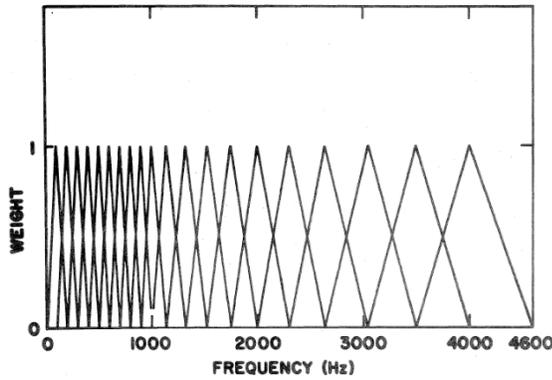


Fig. 6. Mel filterbank with 24 filters utilized by Davis and Mermelstein [6].

C. Cepstrum

A cepstrum is obtained by taking the inverse Fourier transform of the logarithm of an estimated spectrum of a signal. Cepstral features are significantly better at characterizing speech than ordinary spectral features. A key property of the cepstral domain is that the convolution of two signals can be easily expressed as the addition of their complex cepstra which facilitates easier analysis. This property is especially important considering speech is a convolution of two signals, the excitation sequence/input signal and the impulse response of the vocal tract.

The only remaining step after obtaining the log mel spectrum is to convert the spectrum back into the time domain. Since the mel spectrum coefficients (and their logarithm) are real numbers, the time domain conversion can be achieved by utilizing the even property of the real cepstrum whereby it is possible to express the inverse DFT in terms of cosine i.e. the Discrete Cosine Transform (DCT) [7].

The DCT expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. It is similar to the DFT, where the only difference is that the DCT only uses real (cosine) terms whereas the DFT uses complex exponentials (cosines and sines).

Applying the DCT to the log mel spectrum converts the mel spectrum into the mel cepstrum. The coefficients of this resulting mel cepstrum are called the Mel Frequency Cepstral Coefficients (MFCCs) which are the feature vectors used in this paper to characterize an individual.

IV. FEATURE MATCHING USING VECTOR QUANTISATION

Vector Quantization (VQ) is a process of mapping a large vector space to a finite number of regions in that space. Each region is termed a cluster and can be represented by its center/centroid called a 'code word'. The collection of all 'code words' is called a codebook.

Fig. 7 represents a conceptual diagram to demonstrate the recognition process. In the figure, only two speakers and two dimensions of the acoustic space are shown. The circles refer to the acoustic (feature) vectors of Speaker 1 and the triangles refer to the acoustic (feature) vectors of Speaker 2. During the training phase, a speaker specific codebook is generated for each speaker by clustering his/her training feature vectors. The

resulting code words (centroids) are shown by black circles (Speaker 1) and black triangles (Speaker 2), respectively. The distance between a feature vector and the nearest code word (centroid) is called a VQ distortion. The distortion is essentially a measure of how similar the provided testing data (testing voice sample) is to the existing training data (trained voice sample). Smaller distortion implies the testing sample is a closer match to the training sample. During the testing phase, the input voice sample is subjected to vector quantization using the trained codebooks and the VQ distortion is computed. The speaker corresponding to the codebook with the smallest total distortion is then identified.

A. The LBG Algorithm

The speaker specific codebook was designed using the Linde-Buzo-Gray algorithm, a binary split algorithm proposed by Linde et al. [8]. It initially generates a 1-vector codebook which is the centroid or mean of the entire training set. After which the centroid is split into two centroids or code words using the K-means clustering algorithm. This process is repeated as the two code words splits into four then eight and so on till the codebook of the required size is obtained. The algorithm is implemented in four steps [2], [8]:

1) Design a 1-vector codebook; this will be the centroid of the entire set of training vectors.

2) Size of the initial codebook is doubled by splitting the current codebook y_n according to the rule,

$$y_n^+ = y_n(1 + e)$$

$$y_n^- = y_n(1 - e)$$

Where n varies from 1 to the current size of the codebook, and e is a splitting parameter (e is typically chosen to lie within the range $0.01 \leq e \leq 0.05$).

3) For every training vector, find the code word (centroid) in the current codebook that is closest (least VQ distortion) and assign that vector to the corresponding cluster.

4) Repeat steps 2 and 3 until a C vector codebook is designed.

Fig. 8 showcases the algorithm flowchart where n is the number of centroids/code words, C is the desired number of vectors in the codebook, e is the splitting parameter and D is the distance measure (distortion).

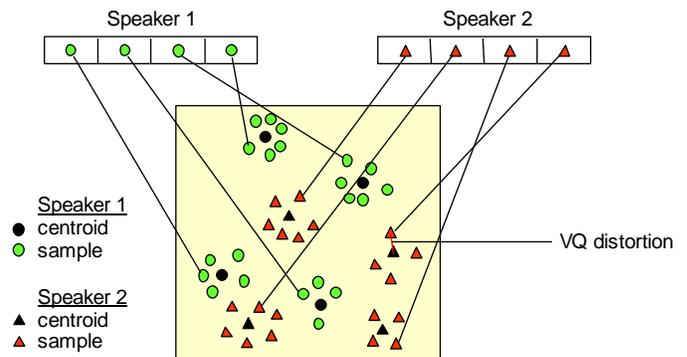


Fig. 7. Conceptual diagram illustrating vector quantization codebook formation [9].

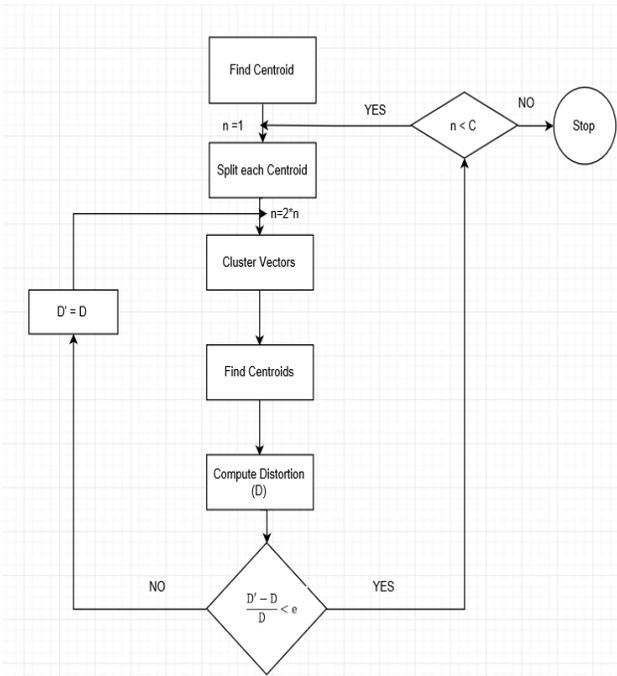


Fig. 8. Flowchart for the LBG algorithm to generate a C vector codebook.

V. RESULTS

System testing was done by registering six users (4 Male (M), 2 Female (F)) into the system and checking how accurately the system was able to identify them. When each speaker speaks their particular passphrase, a speaker specific codebook is generated and compared with the codebooks of the other speakers by measuring the Euclidean distance (VQ distortion) between them. The smaller the Euclidean distance (VQ distortion) between a specific codebook in the database and the current codebook, the likelier it is a match. Cross-correlation of the user speech waveform and the waveform within the database to check for similarity was used as a rejection measure.

For system evaluation, each registered user was asked to access the system 10 times leading to a total of 60 trials. The resulting system performance obtained under various noise conditions were tabulated in Tables 1 and 2. The system was evaluated using three parameters; **False Acceptance (FA)**, **False Rejection (FR)** and **Successful Acceptance (SA)**.

TABLE I. SYSTEM PERFORMANCE UNDER LOW NOISE

User	FA	FR	SA
User 1 (M)	1	0	9
User 2 (M)	0	1	9
User 3 (M)	1	1	8
User 4 (F)	1	2	7
User 5 (F)	2	1	7
User 6 (M)	0	2	8

$$\text{Percentage of accuracy} = \frac{\text{Total number of SA}}{\text{Total number of trials}} * 100 = \frac{48}{60} * 100 = 80\%$$

$$\text{False Acceptance Rate (FAR)} = \frac{\text{Number of FA}}{\text{Total number of trials}} * 100 = \frac{5}{60} * 100 = 8.33\%$$

$$\text{False Rejection Rate (FRR)} = \frac{\text{Number of FR}}{\text{Total number of trials}} * 100 = \frac{7}{60} * 100 = 11.67\%$$

TABLE II. SYSTEM PERFORMANCE UNDER MODERATE NOISE

User	FA	FR	SA
User 1 (M)	1	1	8
User 2 (M)	0	2	8
User 3 (M)	1	1	8
User 4 (F)	1	2	7
User 5 (F)	3	1	6
User 6 (M)	1	2	7

$$\text{Percentage of accuracy} = \frac{\text{Total number of SA}}{\text{Total number of trials}} * 100 = \frac{44}{60} * 100 = 73.3\%$$

$$\text{False Acceptance Rate (FAR)} = \frac{\text{Number of FA}}{\text{Total number of trials}} * 100 = \frac{7}{60} * 100 = 11.67\%$$

$$\text{False Rejection Rate (FRR)} = \frac{\text{Number of FR}}{\text{Total number of trials}} * 100 = \frac{9}{60} * 100 = 15\%$$

TABLE III. DISTANCE MEASURES FOR SUCCESSFUL ACCEPTANCES (SA) / FLAWED RECOGNITION

	User 1 (M)	User 2 (M)	User 3 (M)	User 4 (F)	User 5 (F)	User 6 (M)
User 1 (M)	2.5937	4.2561	6.4446	5.2558	5.6422	2.8973
User 2 (M)	6.9552	4.2585	7.9865	7.4125	7.3515	6.2199
User 3 (M)	4.6702	4.088	3.7403	3.9531	4.0345	4.844
User 4 (F)	5.2081	4.7506	5.4531	4.3072	4.5236	5.328
User 5 (F)	5.3241	4.1065	4.4309	3.9803	3.4562	5.6456
User 6 (M)	3.713	4.599	6.9025	6.2964	6.1602	3.0387

TABLE IV. DISTANCE MEASURES FOR FAILED ACCEPTANCES (FA) / FLAWED RECOGNITION

	User 1 (M)	User 2 (M)	User 3 (M)	User 4 (F)	User 5 (F)	User 6 (M)
User 1 (M)	3.5283	4.6288	6.1934	5.5778	5.7515	3.2276
User 2 (M)	3.1551	2.9969	2.6099	2.673	2.8466	3.167
User 3 (M)	3.8763	3.1881	3.1285	3.0833	3.1781	3.9292
User 4 (F)	3.2822	2.5961	3.0625	2.3471	2.284	3.298
User 5 (F)	5.6567	4.774	5.1242	4.1761	4.328	5.4264
User 6 (M)	4.9801	6.1043	10.0217	9.0526	9.1794	5.3394

The Euclidean distances calculated for each user resulting in accurate recognition and flawed recognition are shown in Tables 3 and 4. The users in columns (colored in orange) are the users in the database and the users in rows (colored in blue) are the users who are testing the system.

It can also be seen from Tables 1 and 2 that there is a slight drop in accuracy (successful acceptances) while analyzing female voice samples. This could be due to an inherent problem of utilizing the MFCCs. The MFCCs are based on the mel scale which is a scale that accurately approximates the human hearing system. Human hearing is more sensitive in the lower frequency regions (less than 1kHz) and less so in the higher frequency regions (greater than 1kHz). This is emulated by spacing filters according to the mel scale, i.e., more overlapping filters below 1kHz than above (refer Fig. 6). The presence of more filters in the lower frequency region gives the system more discerning power i.e. to identify the minor characteristics in this frequency range. On the other hand, the number of filters above 1kHz are less in number due to being spaced logarithmically. The system has less discerning power in this range and thus causes a slight accuracy drop of the system whilst analyzing the female users who speak at a higher frequency due to the smaller size of their vocal cords.

From Table 4, it can be observed that false acceptances occur due to very minute differences between the distance measures of the correct and incorrect users. While a specific minimum threshold for each user cannot be set due to the recalculation of the distance between speaker codebook and database codebooks, implementing a multiple trial system (where the user provides their voice sample twice or more) would fix this problem. This type of system would also prove a better rejection measure as well when compared to cross correlation. Cross correlation results in a large False Rejection Rate (FRR) for the system especially if environment is noisy.

VI. CONCLUSION

The voice recognition system was successfully designed and implemented. The system was successfully able to identify speakers with an accuracy of 80% in low noise environments and with an accuracy of 73.3% in moderate noise environments.

From the results tabulated in Tables 1 and 2, it can be observed that the accuracy of the system for distinguishing female users is slightly lower than that of the males. This is an inherent problem of utilizing the MFCCs. While the accuracy drop is not significant enough to warrant a change to the methodology, it is still a factor to be considered.

Higher accuracy rates could be achieved by either using two-factor authentication or by using different feature matching techniques such as Hidden Markov Models that would yield high accuracy but potentially at the cost of computation time and memory.

ACKNOWLEDGMENT

I would like to thank my project mentor Dr. Senthil Arumugam Muthukumaraswamy for his support and guidance throughout the life cycle of this project.

REFERENCES

- [1] Nijhawan, G. and Soni, M. (2014). Real Time Speaker Recognition System for Hindi Words. *IJIEEB*, 6(2), pp.35-40.
- [2] Rabiner, L. and Juang, B. (1993). *Fundamentals of speech recognition*. Englewood Cliffs, N.J.: PTR Prentice Hall.
- [3] Ahmed, R., Ahmmmed, R., Hosen, M. and Hasan, M. (2011). A Text Dependent Speaker Recognition using Vector Quantization. *Dhaka Univ. J. Eng. & Tech*, 1(2), pp.1-6.
- [4] Tomi, K. and Li, H. (2009). *An Overview of Text-Independent Speaker Recognition: from Features to Supervectors*. 1st ed.
- [5] Boll, S. (1979). Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(2), pp.113-120.
- [6] Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4), pp.357-366.
- [7] Quatieri, T. (2006). *Discrete-time speech signal processing*. 1st ed. Delhi: Pearson Education, pp.712-714.
- [8] Linde, Y., Buzo, A. and Gray, R. (1980). An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communications*, [online] 28(1), pp.84-95.
- [9] Soong, F., Rosenberg, A., Juang, B. and Rabiner, L. (1987). Report: A Vector Quantization Approach to Speaker Recognition. *AT&T Technical Journal*, 66(2), pp.14-26.