Channel Compensation for Speaker Recognition Systems

A thesis submitted in fulfilment of the requirements for the degree of Master of Engineering

Katrina Lee Neville
B Eng.

School of Electrical and Computer Engineering

Science, Engineering and Technology Portfolio

RMIT University

November 2006

Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; and, any editorial work, paid or unpaid, carried out by a third party is acknowledged.

Katrina Lee Neville

November 2006

Acknowledgements

I would like to thank my supervisor, Dr. Margaret Lech, for her support during my Masters research, it was very much appreciated and I wouldn't be here without her.

I would also like to thank Assoc. Prof. Zahir Hussain who helped teach me how to effectively use Matlab in my research and the most efficient ways of implementing algorithms in Matlab.

Thanks also go to my parents, my brother and my sister who have also been very supportive during my time researching and working towards completing this thesis.

Also thanks to the guys in the Digital Signal Processing groups who have been in and out of the DSP research centres during my time completing my Masters, these include, from the postgraduate research group; Fawaz Al-Qahtani, Tasso Athanasiadis, Jusak Jusak, Jimmy Lau, Kevin Lin and Amin Sadik, you guys really gave me a lot of advice during my research and helped me get used to the life of a researcher. Also thank you to the DSP undergraduate students: Geoff Lethbridge, Allen Ling and Tom Targownik for being great company (and entertainment) while you were working on your projects in our labs.

And lastly I'd like to thank the other staff and researchers from around the RMIT School of Electrical and Computer Engineering who have helped me and given me advice on thesis writing and researching in general, particularly the researchers from the sensors research group; Glenn, Sam and Sasi.

Abstract:

This thesis attempts to address the problem of how best to remedy different types of channel distortions on speech when that speech is to be used in automatic speaker recognition and verification systems.

Automatic speaker recognition is when a person's voice is analysed by a machine and the person's identity is worked out by the comparison of speech features to a known set of speech features. Automatic speaker verification is when a person claims an identity and the machine determines if that claimed identity is correct or whether that person is an impostor.

Channel distortion occurs whenever information is sent electronically through any type of channel whether that channel is a basic wired telephone channel or a wireless channel. The types of distortion that can corrupt the information include time-variant or time-invariant filtering of the information or the addition of 'thermal noise' to the information, both of these types of distortion can cause varying degrees of error in information being received and analysed.

The experiments presented in this thesis investigate the effects of channel distortion on the average speaker recognition rates and testing the effectiveness of various channel compensation algorithms designed to mitigate the effects of channel distortion.

The speaker recognition system was represented by a basic recognition algorithm consisting of: speech analysis, extraction of feature vectors in the form of the Mel-Cepstral Coefficients, and a classification part based on the minimum distance rule.

Two types of channel distortion were investigated:

- Convolutional (or lowpass filtering) effects
- Addition of white Gaussian noise

Three different methods of channel compensation were tested:

- Cepstral Mean Subtraction (CMS)
- RelAtive SpecTrAl (RASTA) Processing
- Constant Modulus Algorithm (CMA)

The results from the experiments showed that for both CMS and RASTA processing that filtering at low cutoff frequencies, (3 or 4 kHz), produced improvements in the average speaker recognition rates compared to speech with no compensation. The levels of improvement due to RASTA processing were higher than the levels achieved due to the CMS method.

Neither the CMS or RASTA methods were able to improve accuracy of the speaker recognition system for cutoff frequencies of 5 kHz, 6 kHz or 7 kHz.

In the case of noisy speech all methods analysed were able to compensate for high SNR of 40 dB and 30 dB and only RASTA processing was able to compensate and improve the average recognition rate for speech corrupted with a high level of noise (SNR of 20 dB and 10 dB).

Publications

1. K. Neville, J. Jusak, Z. M. Hussain and M. Lech, "Performance of Text – independent Remote Speaker recognition Algorithm over Communication Channels with Blind Equalisation", in *Proceedings of TENCON 2005*, Melbourne, Australia, November 21 – 24, 2005,.

2. K. Neville, “Blind Channel Equalisation for Speaker Recognition Systems,” Conference Presentation for RMIT University School of Electrical and Computer Engineering Conference, September 2005.

3. K. Neville, F. Al-Qahtani, Z. M. Hussain and M. Lech, “Recognition of Modulated Speech over OFDMA”, in *Proceedings of TENCON 2006*, Hong Kong, November 14-17, 2006.

Table of contents

List of figures

List of tables

Definition of terms

Definitions of important acronyms and terms used in the field of this research

AWGN: Additive White Gaussian Noise

CMA: Constant Modulus Algorithm

CMS: Cepstral Mean Subtraction

DTFT: Discrete Time Fourier Transform

FFT: Fast Fourier Transform

ISI: Inter-Symbol Interference

LMS: Least Mean-Squared

LPF: Low Pass Filter

LTI: Linear Time Invariant

LTV: Linear Time Variant

MFCC: Mel-Frequency Cepstral Coefficients

pdf: Probability density function

PSD: Power Spectral Density

RASTA: RelAtive SpecTrA

SNR: Signal to Noise Ratio

UBM: Universal Background Model

VQ: Vector Quantisation

Chapter 1 - Introduction

1.1 Problem Statement

With security of personal details becoming more and more of an issue for people in today's society people want companies to make sure the best possible preventative measures are in place to prevent the possibility of identity fraud occurring.

Telephone banking in particular is becoming more and more popular, the potential issues with this type of banking is the relative ease in which people can break into the system if a password is leaked and gets into the wrong hands.

Banking customer's are expecting more and more security to be introduced to try and prevent this from occurring, possible solutions being researched and implemented are biometric 'fingerprints.' One biometric 'fingerprint' that could be particularly useful over the telephone is Speaker Recognition and Speaker Verification.

Speaker Recognition is the process of a machine recognising who a person is from their voice by comparing the unique features in that person's voice to a database of features from known speakers. The theory behind Speaker Recognition is that by just listening to people's voices humans are able to recognise who a person is (assuming they have heard their voice before in the past). Therefore if humans can recognise people from their voices so, in theory, should machines, if certain unique features can be isolated and used by the machine for comparison [1],[2]. Speaker verification on the other hand is the process of a machine ensuring a person is who they say they are by statistically comparing the speaker's voice to the voice of the person they claim to be and calculating the probability that they belong to the same person [3],[4]. Again this is something humans are capable of doing, so in theory machines should also be able to do it too, and very possibly improve upon the accuracy of verification.

The main problem with these processes when used in conjunction with a telephone and a telephone channel is the effect channel distortion has on the features in a person's voice. Telephone channels remove the frequencies stored in a person's voice, above 3 KHz and below 300 Hz, so when listening to a person speaking on a telephone the speech tends to sound different to what it would in a face to face situation. This effect is also evident when trying to process speech using a computer. Automatic speaker recognition is primarily based on frequency-domain analysis, therefore any loss of this frequency information can effectively destroy the speaker recognition, speaker verification and many other speech processing applications.

For the effective use of speaker recognition technologies these effects need to be mitigated before the technologies can be accepted by companies and the general public.

This research attempts to improve upon the already existing technologies already out there and compare what methods of speech enhancement are already in the field to attempt to mitigate the effects of Channel distortion on speech features.

1.2 Contribution of the Thesis

This thesis analyses different channel compensation and equalisation methods that could be used in speaker recognition and verification systems when speech is sent through channels. Channel distortion is a major problem for these types of systems since only the smallest amount of distortion to speech can potentially cause unique features in a person's voice to be changed and necessary information for recognition destroyed.

One of the channel equalisation methods studied in the experiments, namely the Constant Modulus Algorithm (CMA), is a channel equalisation method not specifically aimed at speaker recognition systems. Unlike the other two channel compensation methods researched and implemented in this thesis (the Cepstral Mean Subtraction algorithm and the RASTA processing method) little research has been conducted on the CMA algorithm in regards to the effect this algorithm could have on the Cepstral speech features extracted from speakers and the potential improvements to the quality of speech this algorithm could provide in these types of systems. This algorithm is of particular interest since it is also being used on speech that has been converted into binary digits and sent over wireless channels, which is a very practical application of this type of information.

This thesis attempts to shed light on the issues surrounding the effects channel distortion has on the Cepstral features and hence the effects it has on speaker recognition systems and attempts to compare the performance of the Constant Modulus Algorithm with other very well known speech processing algorithms used for channel and microphone distortion compensation in speaker recognition and verification systems. These methods include Cepstral Mean Subtraction (CMS) and RelAtive SpecTrAl (RASTA) Processing.

1.3 Scope

This thesis studies different channel compensation and equalisation methods available to effectively reduce errors in speech data sent over different channels for the purpose of increasing accuracy of speaker recognition systems.

This research focuses on both direct channel equalisation, and channel compensation during the feature extraction phase of the speaker recognition system. The block diagram of the direct channel equalisation is illustrated in Figure 1.1. The block diagram of the channel compensation applied after the feature extraction is illustrated in Figure 1.2.

Figure 1.1: Channel equalisation directly after channel (before data is entered into speaker recognition system)

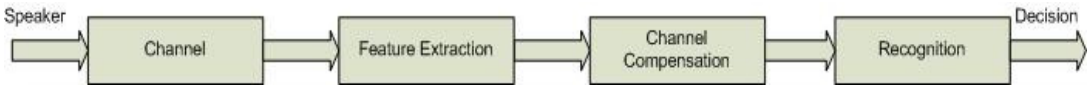

Figure 1.2: Channel compensation during feature extraction phase of speaker recognition system

The following three effects channels have on speech characteristics will be considered:

1. Addition of white (Gaussian) noise.
2. Convolutional channel distortion, and
3. Loss of frequency information due to channel band limiting (filtering).

The effectiveness of the channel compensation techniques will be tested on a speaker recognition system, where the speech features extracted from an unknown speaker will be compared with a set of known speaker's features. It will be assumed that the same type of channel equalisation technology could be implemented in a speaker verification system since the feature extraction phase is almost identical in both cases, and the changes occur in the classification and recognition phases of the two systems.

Other factors that can affect the quality of the features extracted from a person's voice include illness, aging, oral prosthetics and anything that alters the shape of the oral cavity [4]. The effects of these factors on speaker recognition are beyond the scope of this thesis, only the channel effects on speaker recognition will be considered.

1.4 Outline of the Thesis

This thesis aims to analyse and evaluate the effects that channel distortion and noise have on speaker recognition and verification systems. It also aims to evaluate algorithms used for equalisation and compensation of the distorted speech in order to improve the effectiveness of speaker recognition and verification systems over different channels.

This thesis will be laid out and presented in the following manner:

Chapter 2: Speaker Recognition and Verification Theory Overview.

In this chapter, current technologies used in speaker recognition and verification systems will be reviewed. Firstly the differences between the two systems will be discussed and then the potential applications of these systems will be presented.

The block diagrams containing the main components of these two systems will be presented. The individual components of these block diagrams will be discussed in detail.

The purpose of the pre-processing of speech before feature extraction will be explained and common pre-processing algorithms will be presented and discussed.

Speech activity detection will be presented and the role it plays in the efficiency of a speech processor will be discussed. Two different approaches to speech activity detection will be analysed and the strengths and weaknesses these algorithms will be listed.

Feature extraction algorithms including the Mel-Cepstral Coefficients will be then discussed. It will be explained what these features represent and why they are useful for identification and verifications of people from their voices.

Finally the speaker classification algorithm will be discussed, this being the Minimum-Distance classification algorithm.

Chapter 3: Channel Effects and Equalisation Techniques

In this chapter the main types of channel effects on speech will be discussed. It will be explained how these effects could corrupt information contained in speech.

Several different channel equalisation algorithms described in the literature will be introduced.

The discussed channel equalisation methods will include Cepstral Mean Subtraction and RASTA methods, which are techniques specifically used in speaker recognition and verification.

These techniques will be discussed in detail and their strengths and weaknesses analysed. In particular it will be explained how these algorithms function in different speech processing applications and for what kinds of distortions these algorithms are designed to work the best.

Chapter 4: Speaker Recognition using Blind Channel Equalisation Methods.

This chapter will discuss the concept of blind channel equalisation, what it is and what it means to speech processing.

Two commonly used blind channel equalisation algorithms will be introduced; the Least Mean-Squared (LMS) algorithm and the Constant Modulus Algorithm. It will be discussed how these algorithms can be applied to channel equalisation in speaker verification and recognition applications. These algorithms are used in general channel equalisation for many types of channel transmitted information and many purposes.

Chapter 5: Experiment and Results.

This chapter will firstly discuss the experimental design, software and algorithms used in this study.

The source of the speech data, size of the speech database, language and gender of speakers will be explained.

The different channel compensation methods used in this study will be discussed. The structure of the algorithms used will be outlined with important information about the programming of the system.

The second part of this chapter will present the results obtained from the experiments based on the proposed speaker recognition system and the channel compensation methods discussed in Chapters 3 and 4. Graphs showing the recognition rate for each speaker used in the experiments will be included.

Chapter 6: Conclusions and Future Research Directions.

In this chapter research summary and concluding remarks will be presented as well as future research directions stemming from this research.

Chapter 2 - Speaker Recognition and Verification Theory Overview

2.1 Introduction

Speaker recognition and verification is becoming an increasingly important area of research of recent times with public security becoming more and more of a concern. Both speaker recognition and verification systems have potential use in different areas of public security with speaker recognition determining a person's identity from a known set of speakers and speaker verification on determining whether a person is who they claim to be by working out the probability of their voice features belonging to the voice of person they are claiming to be or not.

A speaker recognition system has potential use in situations where only a closed set of people are using the system. Possible applications include a person's voice being used to activate personal settings for, cars or computers where the speaker recognition can be used to determine who is attempting to use the system.

A speaker verification system on the other hand could be potentially useful to ensure security of telephone banking and telephone access to personal details from organisations, particularly with the addition of text dependence into the system to have the double security of a password plus the speaker dependent voice features.

In the following sections of this chapter the outline and components of the speaker recognition and verification systems will be presented and each component will be discussed with details. Different approaches realising these components will be presented and their usefulness for different applications will be analysed.

2.2 General Speaker Verification System

Speaker Verification is a process of determining whether a person is who he or she claims to be or an impostor [1].

This speaker verification system operates in the following way: In the training phase, an average or Universal Background Model (UBM) containing the features from the voices of people who are not the claimed speaker is created. The features stored in the UBM are extracted from approximately 1-2 hours of speech [5]. During this phase features are also extracted from the claimed speaker and the characteristic model of the claimant is created. This phase is also called 'enrolment' into the system.

At the testing phase characteristic features are extracted from the claimant (the unknown person), next the background model as well as the model of the claimed speaker are combined and a likelihood ratio test is performed by the system.

A decision is then made by the system on whether the voice of the claimant is of the person he or she claims to be or of someone else [4].

Block diagrams of the speaker verification training and testing phases are illustrated in Figure 2.1 and Figure 2.2 respectively.

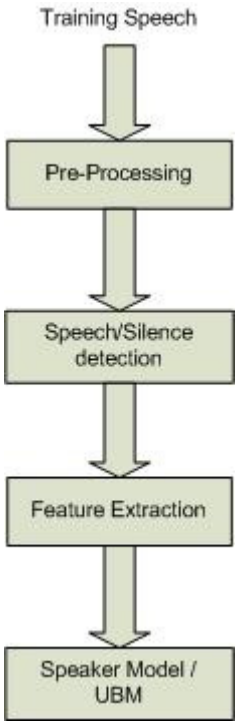

Figure 2.1: Block diagram of speaker verification training phase

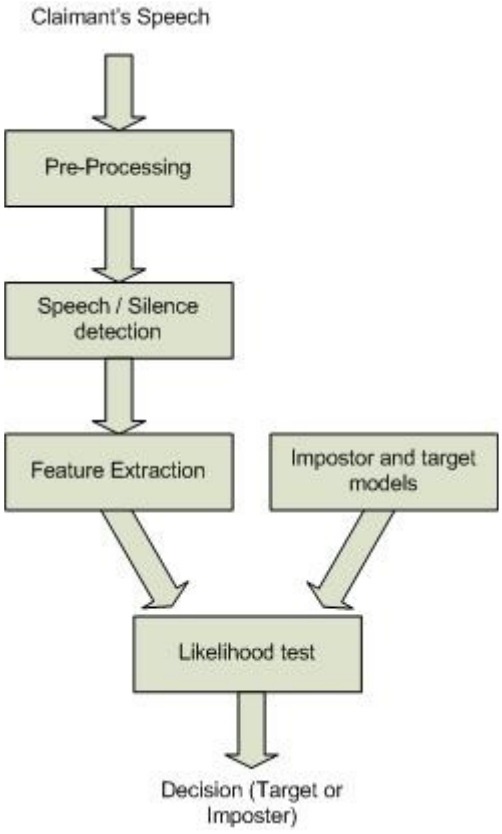

Figure 2.2: Block diagram of speaker verification testing phase

2.3 General Speaker Recognition System

Speaker recognition or identification is a process of determining who a person is from his or her voice features. This is achieved by comparing an unknown speaker's voice features to a database of known speakers and then determining whose features match the unknown speakers features the closest [1].

The speaker recognition system operates in the following way: at the training phase, features are extracted from all the people who are to use the system, these features are then stored.

At the testing phase features are extracted from the unknown speaker and compared to the features of all the system users stored in the database [2].

A decision is then made by the system about the unknown speaker's identity.

The speaker recognition system is very similar to the speaker verification system with only a few small differences. The block diagrams of the training and

testing phases of a speaker recognition system are shown in Figure 2.3 and Figure 2.4 respectively.

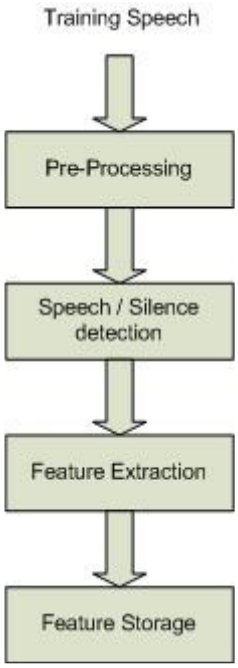

Figure 2.3: Block diagram of speaker recognition/identification training phase

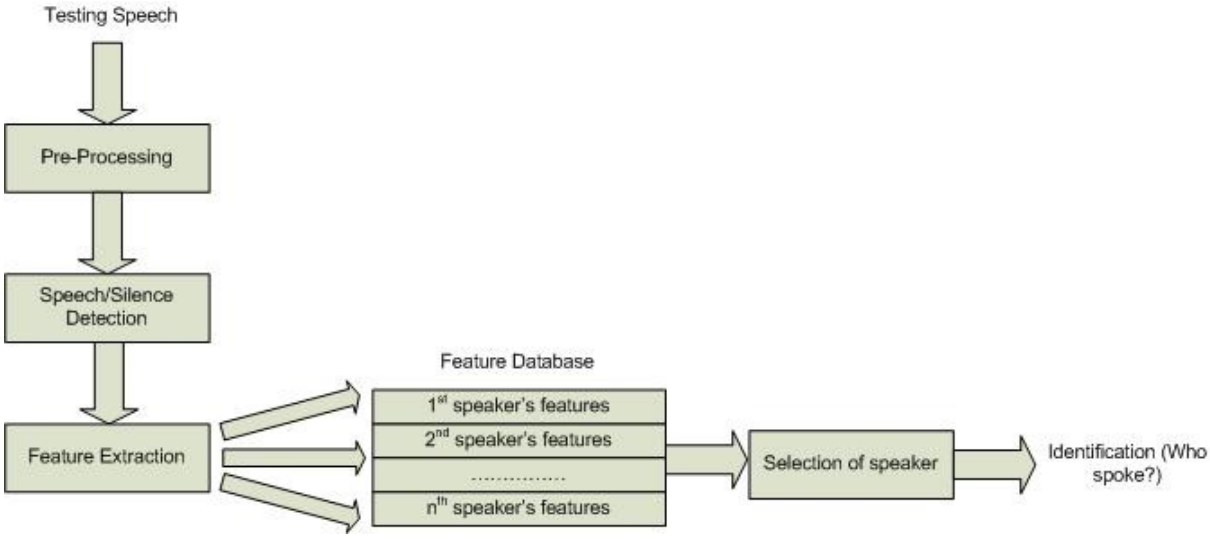

Figure 2.4: Block diagram of speaker recognition/identification testing phase

2.4 Common Pre-Processing Methods

2.4.1 Pre-Emphasis Filter

Before extracting features from the speech it needs to be pre-processed to remove any unwanted distortion such as a low frequency noise. This is achieved by using a pre-emphasis filter. The pre-emphasis filter is used to emphasise the speech frequency bands containing the first formants, which are essential for the speech intelligibility [6]. A commonly used pre-emphasis filter in speech signal processing is a first order high pass filter that has the transfer function of:

$$H(z) = 1 - \frac{15}{16} z^{-1} \tag{2.1}$$

2.5 Speech Activity Detection (Speech / Silence Detection) techniques

On average, speech utterances tend to consists of around 20%-25% silence, these segments of silence appear at the start of the utterance as well as at the end of the utterance, between words and also very small silence segments appear between syllables in words [7]. Since silence segments contain no useful information about a person's identity, which is needed for speaker recognition, removing it should not decrease the accuracy of a speaker recognition system and should improve the overall efficiency of the system. Another downside of having silence in amongst the speech needing to be processed is that keeping the silence takes up storage space and increases the computational effort since features are extracted from the silence as well as the speech. Therefore, it is essential to remove the silence intervals before feature extraction takes place.

One issue with the speech / silence detection is the presence of a background noise in speech recordings. The background noise can often make it difficult to detect the start and endpoints of certain words and phrases, particularly when the start or the end sound blends in with the background noise, such as, for example, the sound of *f* or *v* [8]. A speech processing algorithm therefore needs to be able to detect silence intervals even when the silence intervals contain a background noise.

Techniques derived for the purpose of speech-silence detection in the presence of noise include algorithms designed to detect energy content in the signal, rate of zero crossing of the signal and statistical rules of speech behaviour. Many of these techniques can be adapted to account for changes in intensity of the noise but their effectiveness can diminish when Signal to Noise Ratios fall below around 25 - 30 dB.

2.5.1 Rabiner and Sambur Algorithm

L. R. Rabiner and M. R. Sambur [8] proposed an algorithm to determine the start and end-points of utterances. This algorithm requires that the first 100 ms of a speech recording contain silence. The algorithm uses this time to calculate the zero crossing rate and the short time energy of the silence segment so it can initialise the system and set up appropriate threshold values for speech silence detection.

This algorithm determines the thresholds in the following manner. The short time speech energy over 10 ms windows is calculated using the following equation:

$$E(n) = \sum_{i=-50}^{50} | s(n+i) | \tag{2.2}$$

Where s(n) are the speech samples of the utterance being processed with the sampling frequency assumed to be 10 kHz.

By using equation (2.2) the values of the peak energy within the speech segments (IMX) and the energy during the 100 ms silence segment (IMN), can be calculated and the energy thresholds can then be determined. The energy threshold equations are shown in equations (2.3), (2.4), (2.5) and (2.6).

$$I1 = 0.03 * (IMX - IMN) + IMN \tag{2.3}$$

$$I2 = 4 * IMN \tag{2.4}$$

$$ITL = Min(I1, I2) \tag{2.5}$$

$$ITU = 5 * ITL \tag{2.6}$$

The zero-crossing rate is determined by the number of times per 10 ms that the signal crosses zero during the silence segment, this value is then checked against

the zero-crossing rate of unvoiced speech (25 crossings per 10 ms) to determine what zero-crossing threshold should be used. This is achieved from the equation (2.7) [8]:

$$IZCT = Min(IF, \overline{IZC} + 2\sigma_{IZC}) \qquad (2.7)$$

Where IF is the zero-crossing rate of unvoiced speech, $\overline{IZC}$ is the mean zero-crossing rate during the silence and σ_{IZC} is the standard deviation of the zero-crossing rate during the silence.

Figure 2.5 shows a flowchart of the way this algorithm determines endpoints [8].

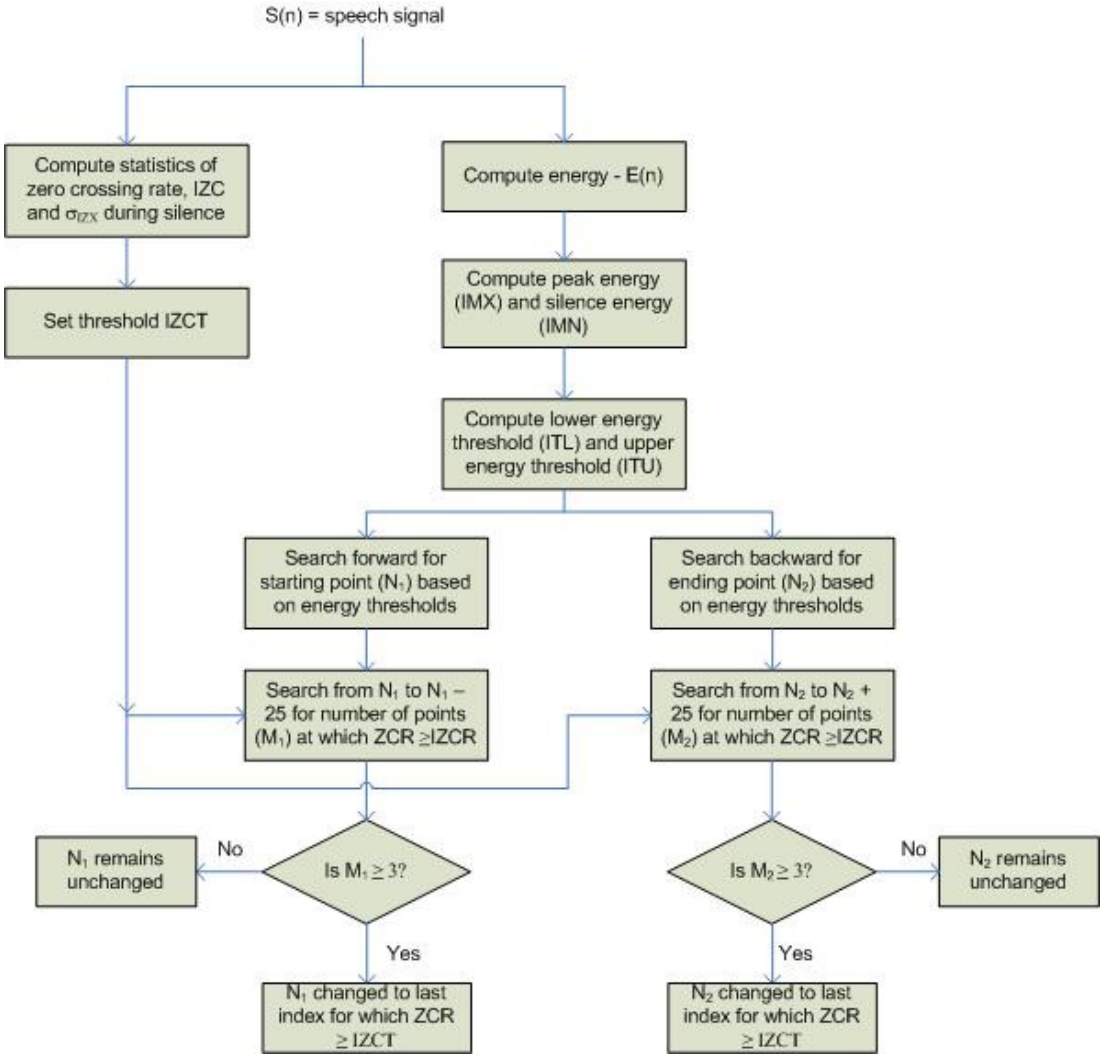

Figure 2.5: Flowchart of the Rabiner and Sambur speech endpoint detection algorithm

At the beginning of this algorithm the start-point of the speech utterance is estimated by determining where the energy of the signal increases beyond the first energy threshold (ITL), this point is taken initially as the start-point unless the energy level again falls below ITL before exceeding the second energy level (ITU). The algorithm then searches the samples for 250 ms before this estimated start-point and sees whether the zero-crossing rate increased past the zero-crossing threshold determined from the silence segment (IZCT). If it did, then the algorithm determines how many times this occurred, if it occurred 3 or more times then the start-point is changed to the first time at which the zero-crossing threshold was exceeded.

The end-point is then determined similarly. It is firstly estimated by detecting the time when the energy level drops off to the silence energy threshold (ITL) and then the next 250 ms are tested to determine the starting point for which the zero-crossing rate exceeds the silence threshold level; the new end-point is then altered accordingly.

2.5.2 Rule Based Adaptive Endpoint detection

Rule Based Adaptive Endpoint Detection as presented in [6] takes a different approach to the Rabiner and Sambur algorithm in that this algorithm attempts to adapt itself to any change over time in the noise energy levels in the signal. It also works on statistical inferences on the general behaviour of speech.

Assumptions made about speech and its behaviour, determine how the speech is to be processed for endpoint detection using this algorithm. It is assumed that:

- 99.9% of continuous speech segments contain talk intervals of less than 2.0 seconds in duration.
- 99.56% of continuous speech segments contain gaps of less than 150 ms.
- Speech energy can only increase the signal level above the background acoustic level.

Using these assumptions three ‘metrics’ are generated representing: the speech energy level, background noise energy level and the minimum energy level. The speech, noise and minimum noise energy levels are shown in figure 2.5.

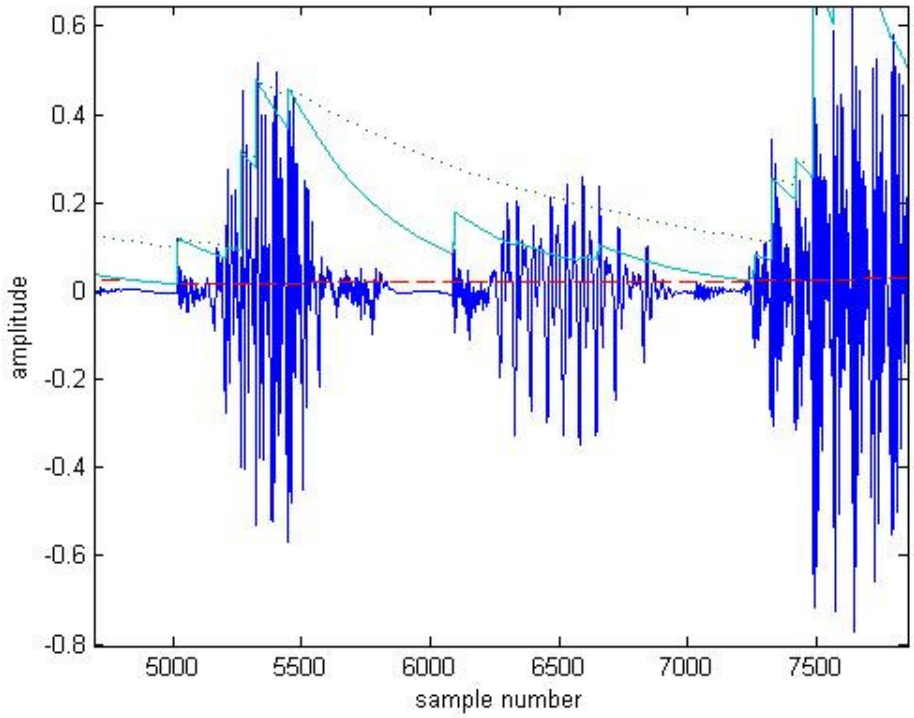

Figure 2.5: Adaptive endpoint detection algorithm metrics

Figure 2.5 shows the speech signal with the three metrics plotted on top, these metrics are as follows: the speech energy level metric (dotted line), the noise energy metric (solid line) and the minimum noise level metric (dashed line).

These three metrics assume the speech is sampled at 8 kHz and are calculated using the following rules [6]: firstly the speech energy metric (s) is defined. This metric will show the peak values of the noise during the duration of the utterance:

$$\begin{aligned} &\text{if } u(k) > s(k-1) \\ &s(k) = u(k) \end{aligned} \tag{2.8}$$

$$\begin{aligned} &\text{if } u(k) \leq s(k-1) \\ &s(k) = (1 - B_s)u(k) + B_s s(k-1) \end{aligned} \tag{2.9}$$

Where u is the absolute value of the original speech and B_s is the decay time constant set at 0.9992

The noise metric $n(k)$ is then defined; this metric is to show the current level of the background noise:

$$\text{if } n(k) > u(k-1)$$
$$n(k) = u(k) \tag{2.10}$$

$$\text{if } u(k) \le s(k-1)$$
$$n(k) = (1 - B_n)u(k) + B_n n(k-1) \tag{2.11}$$

Where and B_s is the decay time constant set at 0.9922

The final metric, the minimum noise energy level metric, *tn(k)*, is then defined by:

$$\text{if } tn(k-1) > n(k)$$
$$tn(k) = (1 - B_t)n(k) + B_t tn(k-1) \tag{2.12}$$

$$\text{if } tn(k-1) \le n(k)$$
$$tn(k) = n(k) \tag{2.13}$$

Where and B_t is the final decay time constant set at 0.999975

These metrics are then used to detect the silence in speech segments by choosing the following threshold levels: speech threshold $T_s = 2$, noise threshold $T_n = 1.414$ and Minimum threshold level T_{min} = the level that is 40 dB below the maximum allowable signal [6]. The following speech-silence detection rules are then applied to the signal:

$$\text{if } (s(k) > T_s tn(k) + T_{\min})$$
$$\text{segment is speech} \tag{2.14}$$

$$\text{if } (s(k) < T_n tn(k) + T_{\min})$$
$$\text{segment is noise} \tag{2.15}$$

$$\text{if } (T_n tn(k) + T_{\min} \le s(k) \le T_s tn(k) + T_{\min})$$
$$\text{no change} \tag{2.16}$$

The positive aspect of this technique is its adaptability to changes in background noise levels. The downside is that it doesn't take into account the spoken sounds that could be lost in the background noise, for example *f* or *v,* which the Rabiner and Sambur algorithm did take into account, therefore it's still susceptible to deleting speech segments of those types of sounds.

2.6 Speech Segmentation

For speaker recognition / verification purposes speech needs to be segmented into small frames before short time spectral analysis can be performed and speaker dependent features can be extracted from each frame. Short time analysis is required in speech signal processing since just calculating a Fourier transform on the whole speech signal would make it impossible to be able to characterise changes in the spectral content over time, therefore time varying components of the speech would not be able to be considered [9].

The most common way this is achieved is using a Hamming Window of 20 ms length with a 10 ms overlap [7]. An example of this is shown in figure 2.6 with the Hamming windows shown as dashed lines against the speech signal.

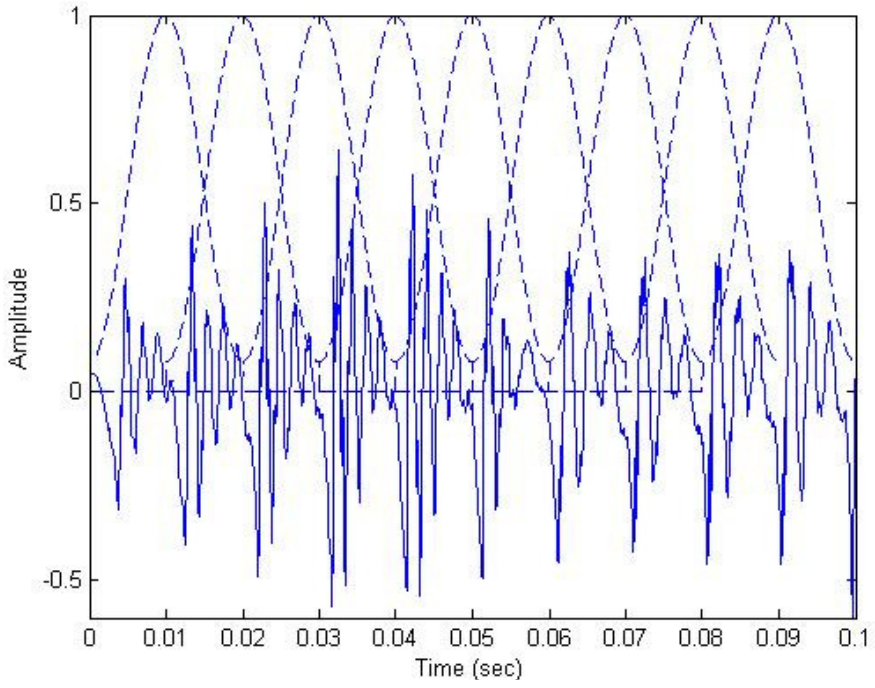

Figure 2.6: Example of using Hamming windows to segment a speech utterance into 20 ms frames with 10 ms overlap.

2.7 Current Feature Extraction Techniques

For automatic speaker recognition to be able to occur certain features need to be extracted from the speech being used in the system, these features need to be unique for every individual speaker being enrolled in the system.

Many features about a person's voice contributes to it being unique, some of these features include, dialect, syntax usage and speech style, these features are considered high level information and are the types of features humans use to aid in recognising who a speaker is.

Machines on the other hand are unable to use these types of features easily for recognition so low level features are extracted and used in automatic speaker recognition, these features are based on spectral analysis of the speech signal, formant frequencies, voice pitch frequency and bandwidth [10].

In this section spectral features will be analysed, particularly Cepstral analysis and the extraction of the Mel-frequency Cepstral coefficients from speech.

2.7.1 Cepstral Feature Extraction

The Mel-Frequency Cepstrum is the discrete cosine transform of the log-spectral energies of a speech segment where the spectral energy is calculated using logarithmically spaced filters with increasing bandwidths [7],[11].

Cepstral analysis has proven to be an effective feature extraction technique as the extracted features depend on the structure of a person's vocal tract. This makes the Cepstral analysis very effective in extracting features in noisy speech [12]. Before Cepstral analysis can be performed the speech needs to be pre-processed as explained in Sections 2.4, 2.5 and 2.6.

After the pre-processing and the windowing of the speech Cepstral feature extraction can begin. For each frame of speech the Short Time Fourier Transform is calculated and the absolute value of it is computed and passed into a mel-scale filter bank. The Short Time Fourier Transform for the n-th window is given as:

$$X(n,\omega) = \sum_{k=-\infty}^{\infty} x[k]w[n-k]e^{-j\omega n} \tag{2.17}$$

Where w[n] is the analysis window (the Hamming window).

The mel-scale is a logarithmic scale that is designed to match the human auditory perception of pitch. The scale was introduced by S.S. Stevens, J.E. Volkman and E.B. Newman in 1937 and was determined through an experimental testing of human pitch and loudness perception. For a given frequency f in Hz, the corresponding mel-scale frequency Mel(f) can be calculated as:

$$Mel(f) = 2595 \log_{10}(1 - \frac{f}{700}) \tag{2.18}$$

Davis and Mermelstein [13] introduced the use of this scale in creating a filter-bank for the extraction of Mel-Frequency Cepstral Coefficients. Figure 2.7 shows a mel-scale filterbank with 20 filters logarithmically spaced according to the mel-scale. These filters are used to extract speech features.

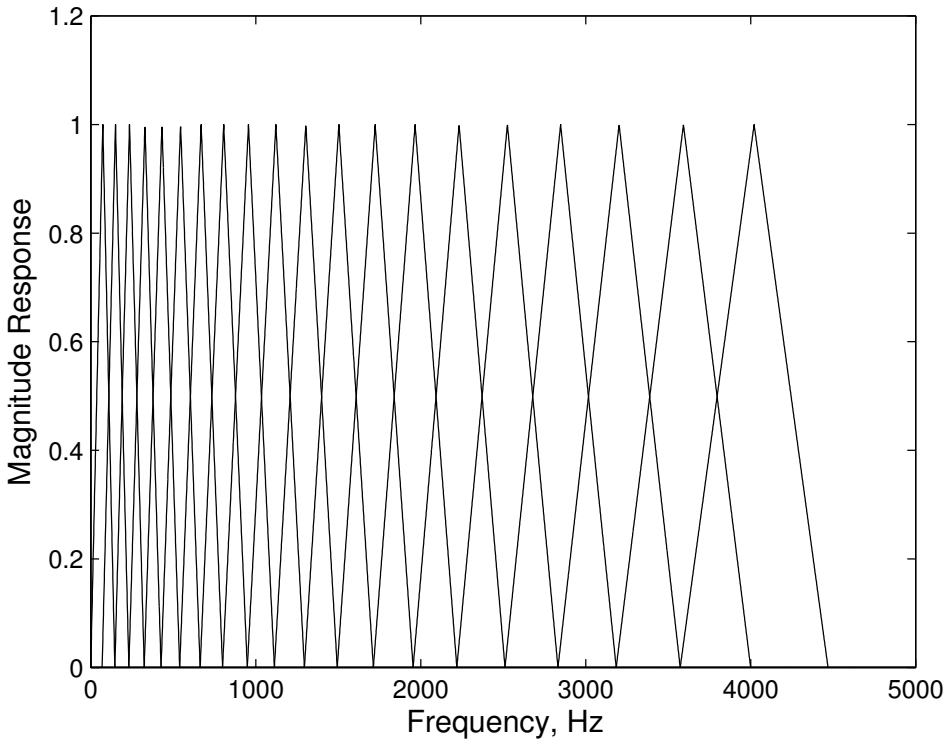

Figure 2.7: Triangular Mel-scale filterbank containing 20 logarithmically spaced filters

The speech frames are passed through the mel-scale filterbank and the log energy of the outputs are calculated. The Mel-Frequency Cepstral Coefficients (MFCCs) are then found using the Equation 2.19:

$$MFCC = \sum_{k=1}^{20} X_k \cos\left[i(k - \frac{1}{2}) \frac{\pi}{20} \right] \qquad \text{i} = 1,2,3,\ldots,\text{M} \ , \quad \text{k} = 1,2,3,\ldots,\text{P} \tag{2.19}$$

Where X_k is the log energy of the output from the k^{th} filter, M is the number of MFCCs and P is the number of filters in the filterbank.

A finite number of MFCCs (between 12 to 20) are calculated for each frame of speech and stored in a database to be used in the recognition phase of a speaker recognition or verification system. In theory, each speaker should have a unique combination of coefficients after his or her voice has been processed in this way since everyone's voice contains different frequency components.

By using the mel-scale this technique is one of the better ways of extracting unique frequency characteristics from a person's voice, therefore it is one of the most commonly used feature extraction processes.

2.8 Current Feature Classification Techniques

2.8.1 Minimum Distance Classification

As its name suggests the Minimum Distance Classifier takes the feature coefficients from the unknown speaker and compares the distance between them and the coefficients taken from known speakers.

Equation 2.20 can be used to calculate the distance between these two sets of coefficients:

$$Distance = \frac{1}{N-1} \sum_{n=1}^{N-1} (\overline{C}^{ts}[n] - \overline{C}^{tr}[n])^2 \tag{2.20}$$

Where N is the total number of feature coefficients, $\overline{C}^{ts}[n]$ is the mean of the testing coefficients, and $\overline{C}^{tr}[n]$ is the mean of the known coefficients (training coefficients) [9].

For speaker verification the speaker's identity is confirmed when the distance exceeds a pre-defined threshold. For speaker recognition/identification the speaker is identified as a person whose coefficients are at the closest –distance to the coefficients of the unknown speaker.

2.8.2 Vector Quantisation

While the minimum distance classifier takes an average of all the speech features extracted from the speakers over all frames and classifies the speech based on these averages, Vector Quantisation (VQ) is able to categorise speech over different acoustic classes [9].

Vector quantisation uses the *k*-nearest neighbour clustering algorithm to determine centroids for each acoustic class within the training speech. These centroids become the basis of the recognition system.

At the testing phase features are extracted from the test speech segments, the distance between the testing feature vectors and the trained centroids are calculated by using a distance measure. The identity of the speaker is then determined by which centroids are nearest to the testing feature vectors [11].

This concept of vector quantisation in illustrated in Figures 2.8 and 2.9.

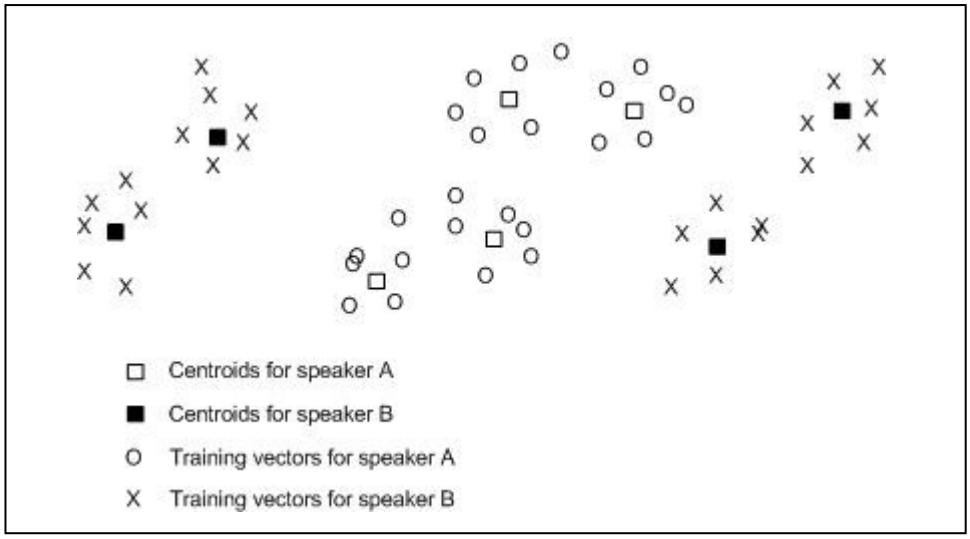

Figure 2.8: Illustration of vector quantisation at training phase.

Figure 2.8 shows the way in which the centroids are determined in order to represent individual speaker's acoustic patterns.

At training vector quantisation of speakers occurs by taking a *k*-dimensional feature vector $\mathbf{x} = (x_0, x_1, ...x_{k-1})$ representing a speaker and mapping each of these

input feature vectors to centroid vectors or codewords used to represent the region of the vector space that the feature vectors fall into.

The centroid vector chosen for each feature vector is determined by minimising the distortion between the original feature vector $\mathbf{x}$ and the centroid vector $\hat{\mathbf{x}}$ using Equation 2.21.

$$d(\mathbf{x},\hat{\mathbf{x}}) = \sum_{i=0}^{k-1} | x_i - \hat{x}_i |^2 \tag{2.21}$$

Where $d(\mathbf{x},\hat{\mathbf{x}})$ is the distortion between $\mathbf{x}$ and $\hat{\mathbf{x}}$ [14],[15].

These codewords representing each individual speaker are then gathered and stored as a codebook to represent each speaker for the speaker recognition system to refer to.

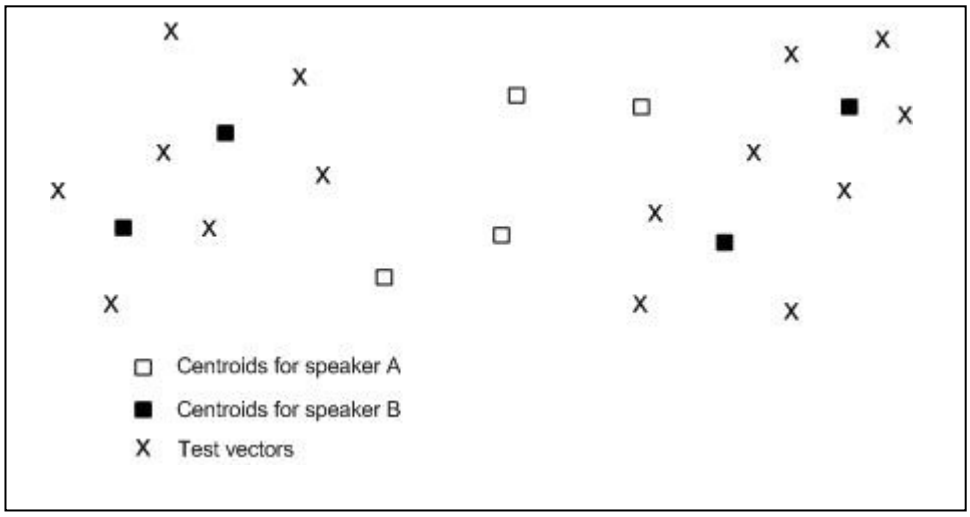

Figure 2.9: Illustration of vector quantisation used at testing phase. Speaker B has been identified in this case.

At the recognition phase the vector quantiser determines the speaker's identity by taking the testing feature vectors, determining which centroids these vectors map to and then identifies which speaker's codebook those centroids match the closest to, this is returned then as the speaker's identity [15].

2.9 Summary

This chapter has discussed the basic structures of both speaker verification and speaker recognition systems and what purposes each system can be used for.

This chapter also analysed and discussed each component in each of these systems and described some common algorithms used for these components. The algorithms described in this section include;

- The pre-emphasis filter, which is used to remove any unwanted low frequency noise in a speech segment
- Two different methods of speech activity detection, including the Rabiner and Sambur algorithm which uses energy and zero-crossing rate to detect speech activity and the rule based adaptive endpoint detection algorithm which detects speech using thresholds determined from statistical assumptions of speech.
- Description of the importance of speech segmentation and windowing of speech segments in preparation of feature extraction.
- A description of the Cepstral feature extraction algorithm which extracts Mel-Frequency Cepstral coefficients (MFCCs). The MFCCs have been proven to be a reliable indication of unique features in a person's voice.
- Two different feature classification algorithms. Classification algorithms are used in the recognition/verification stage of the systems where a decision needs to be made on a speaker's identity. The two algorithms discussed were;
  - The Minimum Distance Classifier where all features are averaged and the distance between known and unknown features are measured and a decision is made on identity from the distance between these averaged features.
  - The Vector Quantisation classifier where speech features are clustered and assigned codewords rather than averaged and a decision on identity is made by comparing the codewords representing a known speaker and the codewords representing an unknown speaker and seeing if they match.

Chapter 3 – Channel Effects and Equalisation Techniques

3.1 Introduction

Channel effects, as mentioned in Chapters 1 and 2, are major causes of errors in speaker recognition and verification systems. In this chapter some common channel effects will be discussed in detail and common methods of compensation and equalisation of these effects will be presented.

3.2 Common Channel Effects

3.2.1 Bandlimiting

All communication channels have a limited bandwidth, which means that only signal frequencies that fall within this bandwidth can be transmitted through the channel.

Human voice has a frequency range of approximately 200 Hz – 4 kHz. It is possible that important speaker dependent information is stored in this entire range. Therefore, it is very important that as much of the spectrum can be preserved, so as many speaker dependent features as possible can be extracted, even after the speech has been transmitted through a communication channel.

For common, landline telephone systems, the frequency range is between 300 Hz – 3.4 kHz, therefore some of the upper and lower frequency components contained in the voice signal are removed. This can significantly reduce efficiency of speaker recognition systems working over telephone lines.

3.2.2 Additive White Gaussian Noise

The most common distortion effect on signals being sent through channels is the additive white Gaussian noise, as represented in Figure 3.1.

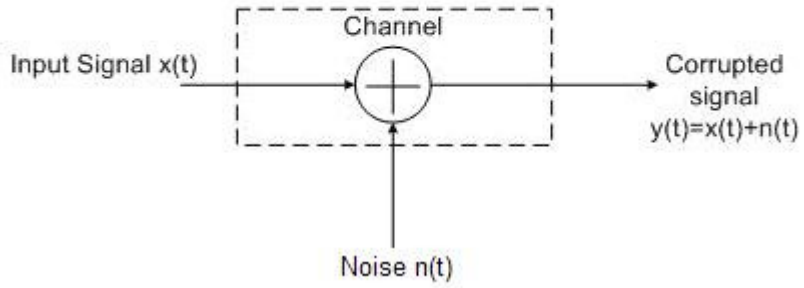

Figure 3.1: Diagram of additive noise channel

Addition of noise to the signal can be caused by many factors including electronic components in a communication system, thermal interference as well as environmental factors such as storms and radiation in the atmosphere (mainly in wireless transmission).

White noise is defined as an uncorrelated random noise process with spectral power spread equally over all frequencies, for channels this entire frequency range is in actuality the bandwidth of the channel and for discrete time signals this bandwidth is equal to half the sampling frequency of the signal [16]. This means that its power spectral density (PSD) is constant over all frequencies contained within the channel's bandwidth:

$$PSD = \frac{\eta}{2}\Pi_{2B}(f) \tag{3.1}$$

where $\frac{\eta}{2}$ is the average power of the noise and Π_{2B} is the rectangular pulse function with width 2B [17].

A Gaussian noise represents a random signal with the probability density function pdf(n) given as a Gaussian function:

$$pdf(n) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(n-m)^2/2\sigma^2} \tag{3.2}$$

Where m is the mean value, usually assumed equal to zero, and σ is the standard deviation.

There are many other non-white types of noise which can distort a signal, these include coloured noise, where the noise power is not evenly distributed over the entire spectrum but concentrated in certain ranges of the bandwidth and impulsive noise which consists of random bursts of noise of short duration [16].

3.2.3 Linear Time-Invariant filtering

In addition to the white Gaussian noise, convolutional (or filtering) effects are often present in channels. One of the easier convolutional effects to analyse and compensate for is the Linear Time-Invariant (LTI) convolutional distortion. This type of distortion is constant over time. The block diagram of a Linear Time-Invariant filtering channel is presented in Figure 3.2:

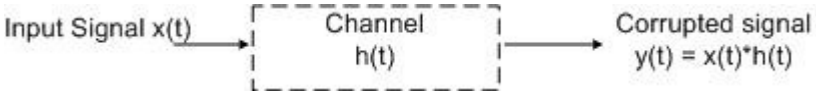

Figure 3.2: Diagram of Linear Time-Invariant filtering channel.

Assuming that x(t) is an input signal, the output y(t) of a channel can be in general described as: $h(\tau;t)$

$$y(t) = x(t) * h(t) + n(t). \tag{3.3}$$

where h(t) is the channel impulse response function and n(t) is the white Gaussian noise [18].

3.2.4 Linear Time-Variant filtering

The Linear Time-Variant (LTV) channel distortion is similar to LTI filtering except that the impulse response of the filter, $h(\tau;t)$, changes over time. The block diagram of a Linear Time Variant filtering channel is presented in Figure 3.3:

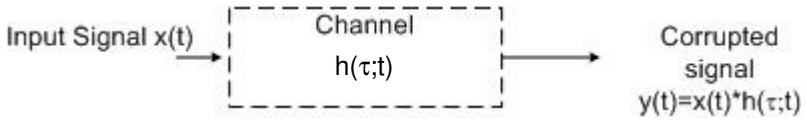

Figure 3.3: Diagram of Linear Time-Variant (fading) filtering channel

Assuming that x(t) is an input signal, the output y(t) of a LTV channel can be in general described as:

$$y(t) = x(t) * h(\tau ; t) + n(t). \quad (3.4)$$

where $h(\tau ; t)$ is the channel impulse response at time t due to an impulse applied at time $(t - \tau)$ [18].

3.3 Channel Equalisation Methods

Channel Compensation methods discussed in this chapter include: Cepstral Mean Subtraction, RASTA Processing, Least Mean-Squared Filtering and the Constant Modulus Algorithm. This section focuses on the compensation methods which can be integrated in the feature extraction phase of a speaker recognition or verification system, these include the Cepstral Mean Subtraction Method and the RASTA processing.

Least Mean-Squared and the Constant Modulus Algorithm will be discussed in more detail in the next chapter.

3.3.1 Cepstral Mean Subtraction

The Cepstral Mean Subtraction is often used during the feature extraction phase of speaker recognition/verification systems to compensate for convolutional channel distortion of voice signals. The convolutional channel distortion can be

caused by different microphones used between the testing and training phases or different transmission channels used during testing and training [19],[20].

The Cepstral Mean Subtraction method assumes that the time average of all speech signals is zero and the convolutional effects due to the channel are uniform over time (ie. time-invariant) [9],[21]. Therefore, it does not provide a perfect solution for eliminating channel effects because speech does not necessarily have a zero mean and often there are time-variant channel effects due to external factors that can affect speech signals. Despite these drawbacks, the Cepstral Mean Subtraction method can be relatively effective and useful.

The convolutional channel effect results in a distorted speech signal y[n] given as:

$$y[n] = x[n] * h[n] \tag{3.5}$$

Where x[n] is the clean speech and h[n] is the channel impulse response causing distortion to the speech. With Short-Time Fourier Transform applied to y[n] using a window w[pL-k] (where L is the window length and p = 1, 2, 3…) this distorted signal can be referred to in the frequency domain by equation 3.6 [9]:

$$Y(pL,\omega) = X(pL,\omega)H(\omega) \tag{3.6}$$

Equation 3.6 shows that the convolutional distortion applied to the clean speech has a multiplicative character in the frequency domain, therefore it is not easy to isolate the channel distortion H(ω) from the speech signal X(pL,ω).

To aid in isolating the convolutional distortion, a logarithmic operation can be performed. By taking the log of both sides of Equation 3.6 the signal Y(pL,ω) can be represented as a sum of two logarithms; the log of the speech and the log of the convolutional distortion:

$$\log[Y(pL,\omega)] = \log[X(pL,\omega)] + \log[H(\omega)] \tag{3.7}$$

Assuming that the convolutional distortion H(ω) is time-invariant, it is now easier to isolate the channel distortion from the speech.

Firstly this is achieved by calculating the inverse Fourier Transform of $\log[Y(pL,\omega)]$ given by Equation 3.8:

$$\hat{y}[n,\omega] = \Im^{-1}(\log[X(pL,\omega)] + \Im^{-1}(\log[H(\omega)]) \tag{3.8}$$

And finally a Cepstral lifter (l[n]) is applied to remove the mean of $\hat{y}[n,\omega]$. This results in a signal $\hat{x}[n,\omega]$ given as:

$$\hat{x}[n,\omega] = l[n]\hat{y}[n,\omega] \tag{3.9}$$

The Cepstral lifter l[n] is a function defined as:

$$l[n] = \begin{cases} 0 & \text{for n} = 1 \\ 1 & \text{elsewhere} \end{cases} \tag{3.10}$$

The Cepstral lifter when applied to $\hat{y}[n,\omega]$ removes the 0^{th} value in $\hat{y}[n,\omega]$ and leaves the remaining values intact [9],[22].

While the Cepstral Mean Subtraction is relatively effective in removing convolutional distortion, it is not able to compensate for additive channel distortion [23],[21]. Therefore, it is not capable of removing an additive channel distortion such as white Gaussian noise, which occurs commonly in transmission channels.

3.3.2 RASTA Processing

RASTA, which stands for Relative Spectral Processing, is another channel compensation technique, RASTA was proposed by H. Hermansky and N. Morgan [23]. This speech processing technique acts in a similar way to Cepstral Mean Subtraction, in that it attempts to compensate for convolutional distortion due to mismatched microphones or channels and attempts to eliminate any DC component within the channel distorted signal [24]. One of the differences between Cepstral Mean Subtraction and RASTA is that RASTA assumes that the convolutional effects on the speech due to the channel are non-uniform over time, whereas the Cepstral Mean Subtraction does not take into effect varying convolutional distortion and assumes uniform convolutional effects [9]. RASTA Processing also is designed to

help take into account the additive channel distortion caused by the addition of white Gaussian noise [25].

The process of RASTA processing can be outlined as follows; the short time Fourier Transform is firstly taken of the distorted speech segment y[n], then the logarithmic transform is taken of the speech's spectrum.

The logarithmically transformed speech is then passed through the RASTA IIR filter which has the following transfer function H(z) [23],[9]:

$$H(z) = \frac{2 + z^{-1} - z^{-3} - 2z^{-4}}{1 - 0.98z^{-1}} \tag{3.11}$$

The filter amplitude response is shown in Figure 3.4.

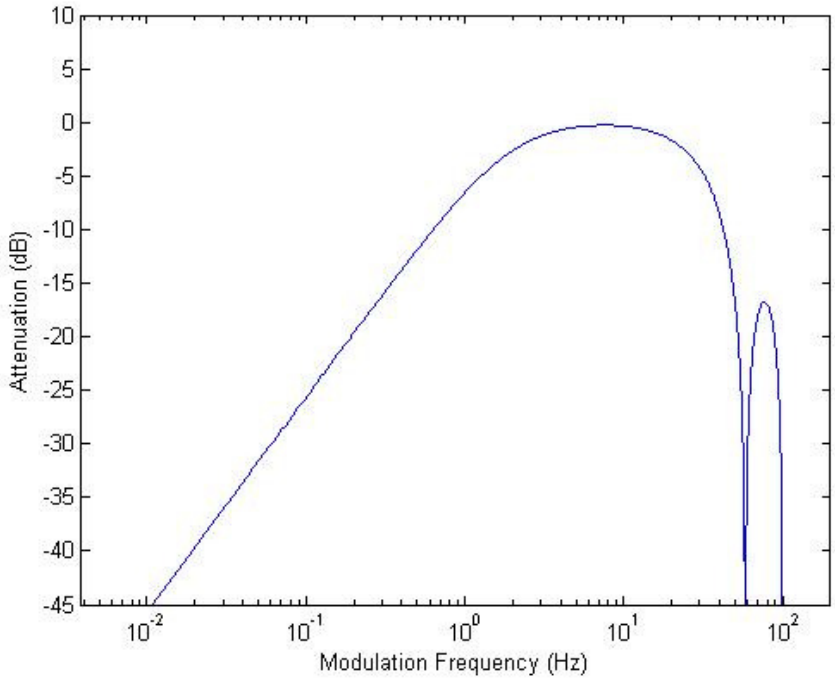

Figure 3.4: RASTA filter response

This RASTA filter effectively removes time-variant convolutional distortion caused by transmission channels by having a high attenuation at low modulation frequencies at and near DC [24],[9].

The RASTA filter also reduces the effect of additive white Gaussian noise more effectively than Cepstral Mean Subtraction but its effectiveness at reducing noise can be improved again by implementing the J-RASTA processing algorithm [26],[23].

J-RASTA is very similar to RASTA processing except that a J factor is introduced at the logarithmic transform stage. Therefore the transformation is calculated as in Equation 3.12:

$$y = \ln(1 + Jx) \tag{3.12}$$

Where x is the speech segment and J is a factor dependent on the characteristics of the noise corrupting the speech. This is calculated using Equation 3.13:

$$J = \frac{1}{C.E_{noise}} \tag{3.13}$$

Where E_{noise} is the mean energy of the noise corrupting the signal and C is a constant chosen to achieve the best possible reduction of noise distortion. In the paper [23] the optimal C value for the author's experiments was found to be C=3, but this value can change depending on experimental conditions.

Using this type of transform on the signal being processed increases the accuracy of the system for compensation of noise distortion above the plain log transform used in RASTA processing.

3.4 Summary

This chapter discussed some common channel effects known to causer corruption to speech signals and some algorithms used to attempt to mitigate these effects in speaker recognition and verification systems.

The channel effects discussed in this section include:

- Bandlimiting where a signal can be distorted by a filtering effect from the medium the signal is being sent through.
- Additive White Gaussian Noise (AWGN) which is distortion caused by electrical, thermal and/or environmental factors where random signal distortion is added to a signal during transmission through a vulnerable channel.
- Linear Time–Invariant and Linear Time-Variant filtering which are filtering effects that cause convolutional distortion to a signal being transmitted.

Two channel compensation methods were also discussed in this section; Cepstral Mean Subtraction which is designed to remove Linear Time-Invariant filtering from speech features and RASTA Processing designed to remove Linear Time-Variant distortion from speech features.

Chapter 4 - Speaker Recognition using Blind Channel Equalisation Methods

4.1 Introduction

Blind Channel Equalisation is an invaluable method to compensate for channel distortion when the channel impulse response $h[n]$ is unknown.

Channel equalisation with a known impulse response is relatively easy and can be achieved by designing a matched filter with a response that is the inverse of the known channel's response, [27],[16],[17]. Unfortunately this is not always possible in practice and particularly when a channel is noisy, non-linear or time-variant [16].

One method used to equalise an unknown channel distortion is by sending a training sequence which is known to both the sender and receiver. This technique is known as a supervised channel equalisation technique. A flowchart of this method is shown in Figure 4.1.

The receiver receives the distorted training sequence and then adapts its inverse filter coefficients to compensate for the distortion that has occurred to the training sequence. This is an effective method; however it has very high bandwidth and power requirements [28].

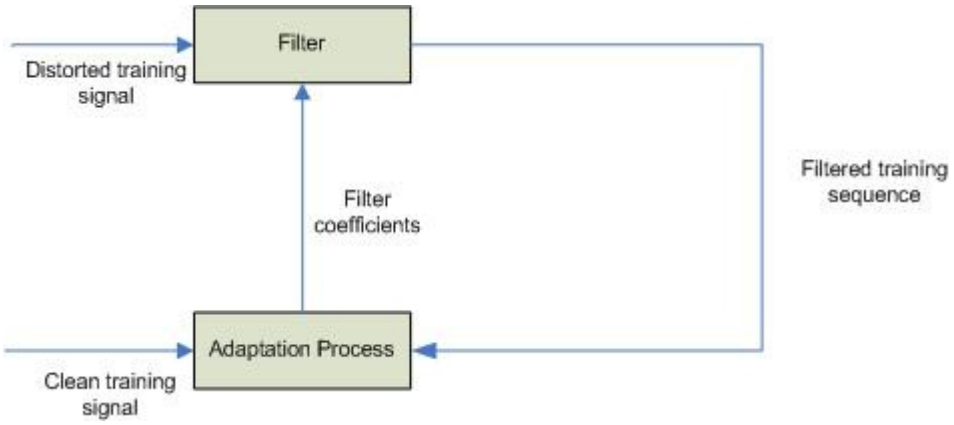

Figure 4.1: Flowchart of a supervised channel equalisation technique

Blind, or unsupervised channel equalisation methods can be implemented to adapt a filter's coefficients, and hence it's response, to equalise the corrupted information with no knowledge of the channel's impulse response or the training sequence being sent over the channel [27]. This type of equalisation uses statistics to retrieve the signal.

A basic diagram of an adaptive blind equalisation system is shown in Figure 4.2.

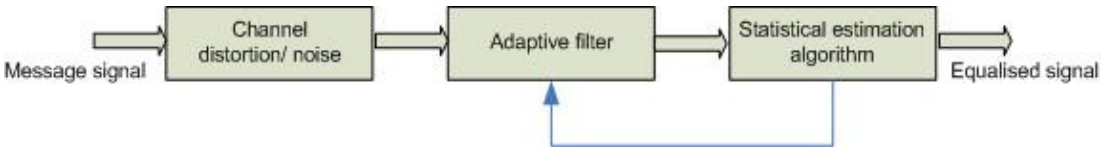

Figure 4.2: Basic system block diagram for an adaptive blind equaliser

There are many examples of this type of system used in digital signal processing applications. Two of these methods including the Least Mean-Squared algorithm and the Constant Modulus Algorithm will be discussed in this Chapter.

4.2 Adaptive Blind Equalisation Algorithms

4.2.1 Least Mean-Squared Adaptive Filtering

Least Mean-Squared filtering (LMS) is an adaptive filtering algorithm used for discrete time signals. The LMS algorithm uses a feedback system to reduce noise and channel distortion by changing the coefficients on a filter to minimise the error between the filtered signal $\hat{x}(n)$ and the expected or desired response $d(n)$ [29],[17].

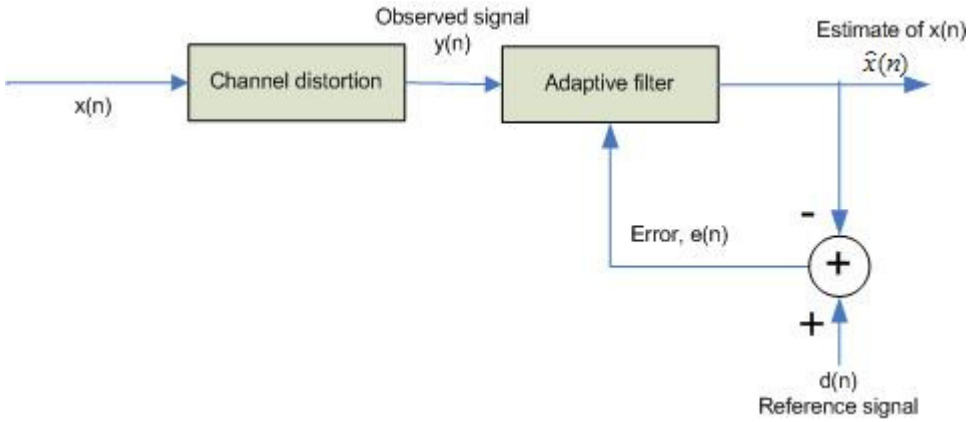

Figure 4.3: Least Mean-Squared system block diagram

As shown in Figure 4.3 the filter's coefficients are altered by taking the output of the filter $\hat{x}(n)$ and subtracting that value away from the desired signal $d(n)$.

The desired signal $d(n)$ is a signal chosen to have properties as near to what is expected of the message $x(n)$ as possible, this may take the form of a training sequence known both to the sender and receiver or in the case of noise cancellers $d(n)$ can be a delayed version of the observed signal $y(n)$ [17].

After subtracting $d(n)$ from $\hat{x}(n)$ the resulting value, $e(n)$, is the estimation error which is fed back into the filter to determine the coefficient changes needed. The estimation error $e(n)$ is given as:

$$e(n) = d(n) - \hat{x}(n) \tag{4.1}$$

The changes in the filter's coefficient values are determined iteratively using Equation 4.2:

$$a_n = a_{n-1} + \mu.e(n)y(n) \tag{4.2}$$

Where a are the filter's coefficients at the nth instant and μ is the convergence accuracy coefficient which affects the convergence speed and accuracy of the system.

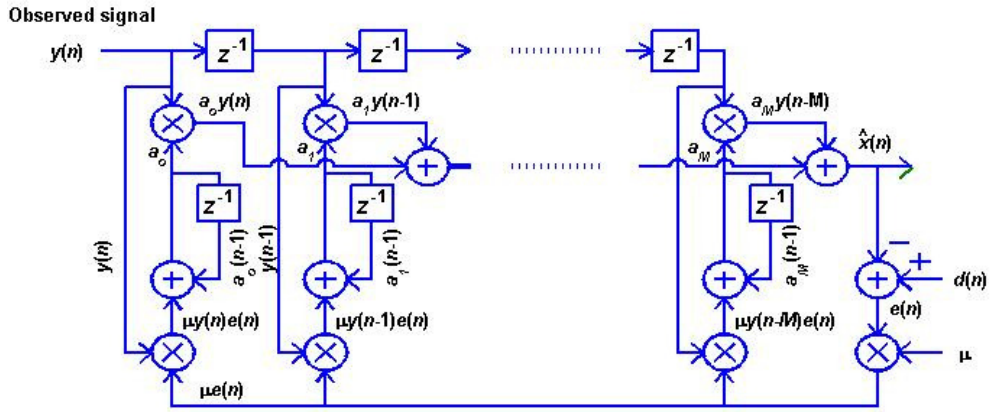

Figure 4.4: Least Mean-Squared adaptive filter implementation diagram

Figure 4.4 shows the implementation of an M+1 tap adaptive filter within an LMS filtering system. M+1 represents the filter's length and the number of components needed in the filter, the larger M is the better the estimation of the filter coefficients but the more delay will occur in the output.

The convergence accuracy coefficient, μ determines both the accuracy and speed of convergence of the filter's coefficients. Small values of μ will cause the system to be more accurate but slower to converge while larger values of μ will cause the system to converge quickly but be less accurate [17]. Convergence will hold accurately as long as the following criterion is met for the chosen value of μ [29]:

$$0 < \mu < \frac{2}{tap\ input\ power} \tag{4.3}$$

Where the tap input power is equal to the sum of the mean squared values of the tap inputs in the filter, this is shown in Equation 4.4.

$$tap\ input\ \ power = \sum_{i=0}^{M} \overline{a}_i^2 \tag{4.4}$$

4.2.2 Constant Modulus Algorithm (CMA)

The Constant Modulus Algorithm (CMA) is a blind channel equalisation method, meaning that there is no assumed knowledge of the impulse response of the transmission channel and there is no reference or training data that can be used to equalise the channel distortion.

The CMA algorithm assumes that the received signal $x(n)$ is a binary output from a wireless channel of unknown impulse response. At the receiver end of this system there is a linear filter used to equalise the received signal, this received signal will be corrupted with white Gaussian noise and inter-symbol interference (ISI) [30].

Similarly to the Least-Means Squared algorithm, the CMA algorithm uses an iterative technique in order to determine the optimal filter coefficients to effectively compensate for the distortion in the channel.

In the CMA case the filter's coefficients are updated using the following stochastic gradient descent algorithm [27],[28].

$$\mathbf{f}(n+1) = \mathbf{f}(n) + \mu r^*(n)\psi_{CMA}(n) \tag{4.5}$$

Where: **f**(n) represents a vector of filter coefficients

μ is the step-size parameter

ψ_{CMA} is the error function of the CMA calculated as:

$$\psi_{CMA}(y_n) = y_n^*(\gamma - |\ y_n\ |^2) \tag{4.6}$$

Where $\gamma = E[|\ x(n)\ |^4] / E[|\ x(n)\ |^2]$

The Constant Modulus Algorithm uses the Godard cost function as shown in Equation 4.7 [27],[29] and seeks to minimise this cost function to achieve ideal equalisation for the received signal [28].

$$J(n) = E[(|\ y(n)\ |^p\ - R_p)^2] \tag{4.7}$$

Where $R_p = \frac{E[|\ x(n)\ |^{2p}]}{E[|\ x(n)\ |^p]}$ and is chosen so that the gradient of the function J(n) is zero when perfect equalisation is achieved.

The benefits of using blind equalisation similar to what has been discussed in this section is that instead of sending training sequences down the channel first to determine the distortion caused by the channel, the adaptive filter can be used on the signal itself and adapt to the channel to equalise the signal during the transmission process. This is a much more efficient equalisation technique than supervised channel equalisation, particularly in respect to the efficient use of bandwidth since training sequences need not be used.

4.3 Summary

This chapter covered adaptive channel equalisation methods used to filter a received signal. Adaptive filtering involves iteratively altering a filter's impulse response in order to minimise error in a received signal, therefore reducing the effects channel distortion have on a signal.

The adaptive filtering methods discussed in this chapter included;

- The Least Means Squared adaptive filtering which uses a reference signal to alter a filter's coefficients to minimise the error between the reference signal and the distorted received signal.
- The Constant Modulus Algorithm which is a blind equalisation method that assumes no knowledge of the impulse response of the channel and also uses no training sequence to initialise the filter coefficients

Chapter 5 - Experiment and Results

5.1 Introduction

All the experiments were performed as text-independent speaker recognition experiments, meaning that the semantic information included in the speech was not taken into account during the speaker recognition process.

The aim of the speaker recognition experiments was to determine who the speaker was rather than making sure that the query speaker was who he or she claimed to be.

The results are presented in the following manner. Firstly an analysis is performed on clean undistorted speech before analysing speech corrupted with noise and filtering effects with no equalisation. Statistical analyses of the effects these distortions play on the recognition rate are discussed for these cases. The second part of the results focus on statistical comparison of speech that has been corrupted with channel distortion and equalised using Cepstral Mean Subtraction (CMS), RelAtive SpecTral processing (RASTA) and the Constant Modulus Algorithm (CMA) and finally the preferred methods of equalisation are discussed.

The experiments described in this chapter were set up in the following manner.

5.2 Test data

Clean speech samples from 10 people (5 male and 5 female) were used. The speech was sampled at a rate of 16,000 Hz. For each speaker 6 samples of speech of duration of 1 to 4 seconds were analysed. The samples represented six different utterances. One utterance was used in the training phase of the system and five other utterances were used in the testing phase.

Both the testing and training utterances represented clean speech containing no channel distortion such as filtering or noise.

All the speech samples were selected from the Berlin emotional speech database [31] containing voices portraying happiness, sadness, anger, neutral, boredom, fear and disgust. Only neutral speech recordings were used in these experiments. The emotional aspect of speech was not taken into account.

5.3 Speaker recognition system structure

The experimental algorithms were developed and tested using Matlab (version 7.1) programming language.

The flowcharts of the training and testing systems used in the experiments are illustrated in Figure 5.1 and 5.2 respectively.

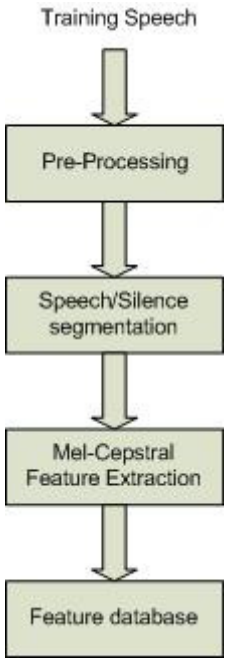

Figure 5.1: Flowchart of the speaker recognition training system used in the experiments

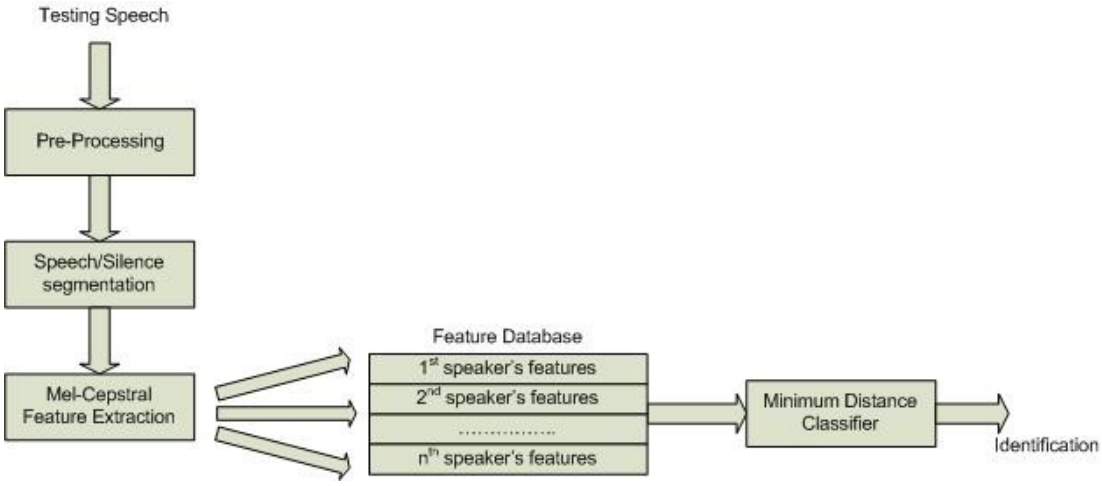

Figure 5.2: Flowchart of the speaker recognition testing system used in the experiments

The training and testing systems used in the experiments were based on the Mel-frequency Cepstral Coefficient features extracted from each speaker as described in section 2.7.1. The use of Mel-Frequency Cepstral Coefficient features in speaker recognition has been proven to be very effective in being able to identify individual speakers from the individual phonetic and frequency characteristics in their speech [9],[13].

5.4 Training Procedure

The training system consisted of a pre-processing component, involving the entering of raw data containing the speech to be trained into the system. Figures 5.3 and 5.4 shows an example of a speech segment used in the system and a time-frequency plot of its STFT.

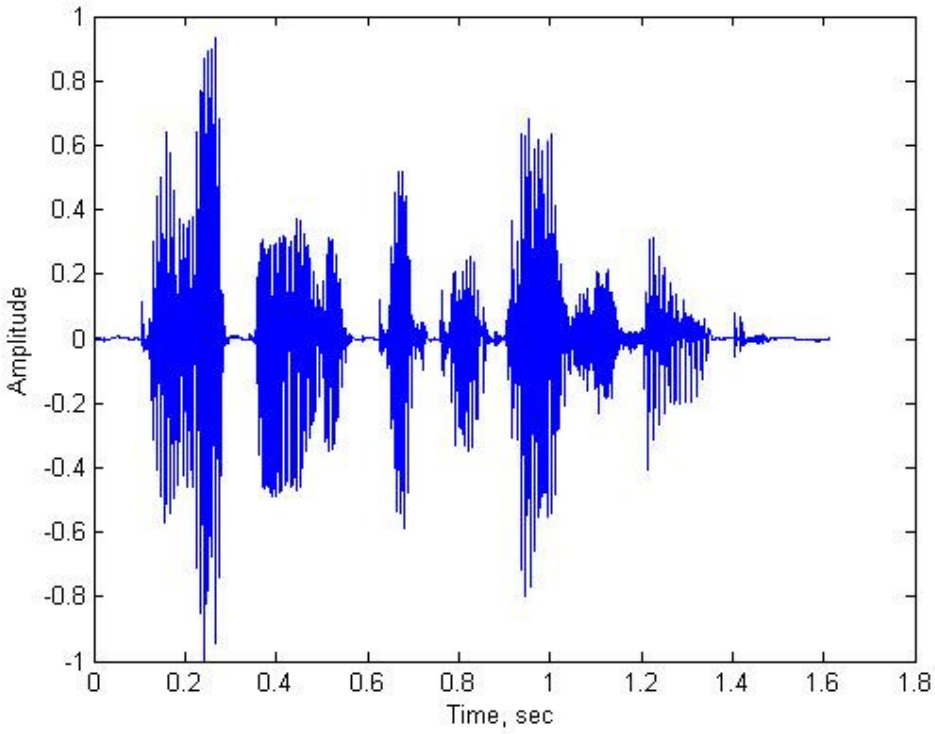

Figure 5.3: Undistorted speech sample used in the speaker recognition system

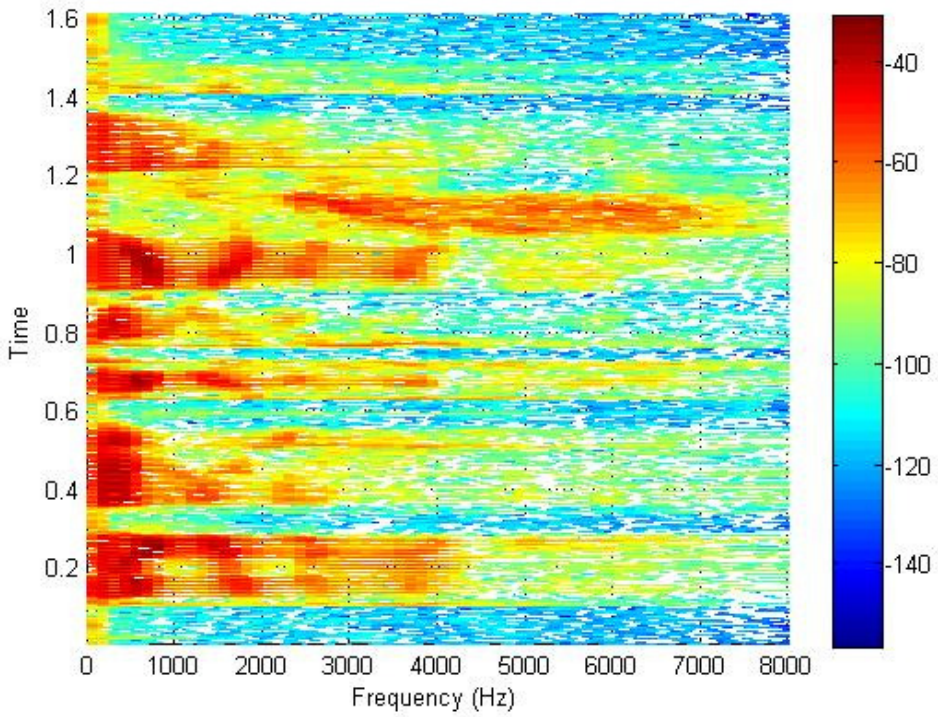

Figure 5.4: Time Frequency plot of an undistorted speech sample used in the speaker recognition system

This speech was firstly pre-emphasised using a first order high pass filter given in Equation 2.1 emphasising important high frequency information in the speech and reducing the effect of low frequency background noise such as machine and air-conditioner noise, which could affect the accuracy of speaker recognition (see section 2.4.1).

Because speech utterances are made up of many silence segments as well as speech segments for efficiency of the system, silence intervals were removed in the pre-processing phase. Silence intervals are unnecessary as they contain no useful information about the speaker's identity and takes up processing time and computer storage space.

The silence detection and removal in this research involved a technique similar to the Rule Based Adaptive Endpoint detection discussed in section 2.5.2 and proposed in [6]. This technique uses speech and noise energy metrics to represent the levels of speech and noise throughout a spoken utterance. The silence/noise intervals are then detected using an adaptive thresholding scheme. The silence segments are then removed.

After pre-processing the training speech was segmented into 20ms frames with a frame overlap of 10ms. The short time spectral analysis was then performed on a frame-to-frame basis and the Mel-Cepstral Coefficients were calculated. The feature extraction method used in the experiments was the same as described in the section 2.7.1.

These features were averaged over all frames and stored in a library of speaker models.

5.5 Testing Procedure

5.5.1 Speaker recognition based on clean speech

To test the system firstly three clean speech utterances from the testing set for each speaker were processed in the same manner as the training speech and had their MFCCs extracted (see Figure 5.2). These coefficients were then compared to the coefficients extracted earlier in the training phase using the Minimum Distance Classifier method as discussed in Section 2.8.1. In this way the system was able to determine which speaker was most likely to have been the one to have uttered the test phrases from the group of speakers trained into the system.

5.5.2 Speaker recognition based on distorted speech

The next stage was to test how well the system can perform after the clean speech is corrupted by channel effects. In this experiment, the test speech was distorted in a way simulating the effects of channel filtering and/or addition of white noise.

The effect of channel filtering was analysed using a Butterworth filter with five different cutoff frequencies, fc = 7, 6, 5, 4 or 3 kHz. A Butterworth filter of order 9 was used in this experiment because this type of filter has no ripple in the pass band region which would have caused extra unwanted distortion to the speech.

The frequency response of the low-pass filter used in the experiments is illustrated in Figure 5.5

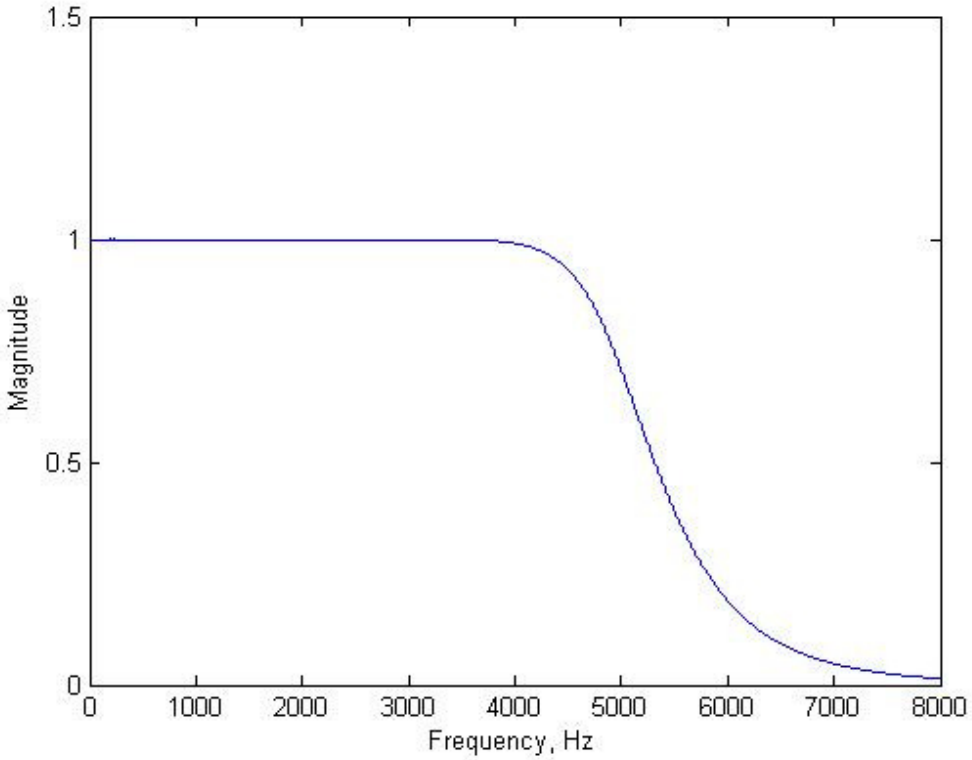

Figure 5.5: Frequency response of a 9th order low pass filter with cutoff frequency = 5 kHz

The second effect analysed in this experiment was the addition of Gaussian noise to the speech. A vector of random noise at different power levels were generated and added to the speech files to simulate the addition of channel noise at Signal to Noise Ratios of 10, 20, 30 and 40 dB. An example of a noisy speech signal with a Signal to Noise Ratio of 30 dB is shown in Figure 5.6 and it's time – frequency plot is shown in Figure 5.7.

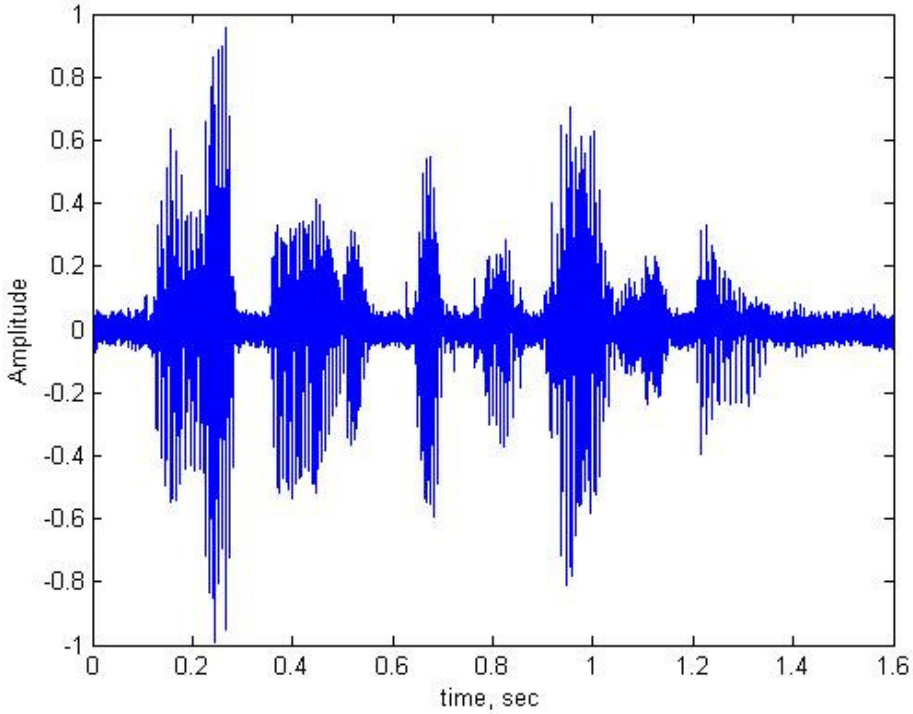

Figure 5.6: Noisy speech sample used in experiments. Signal to Noise Ratio is 30 dB

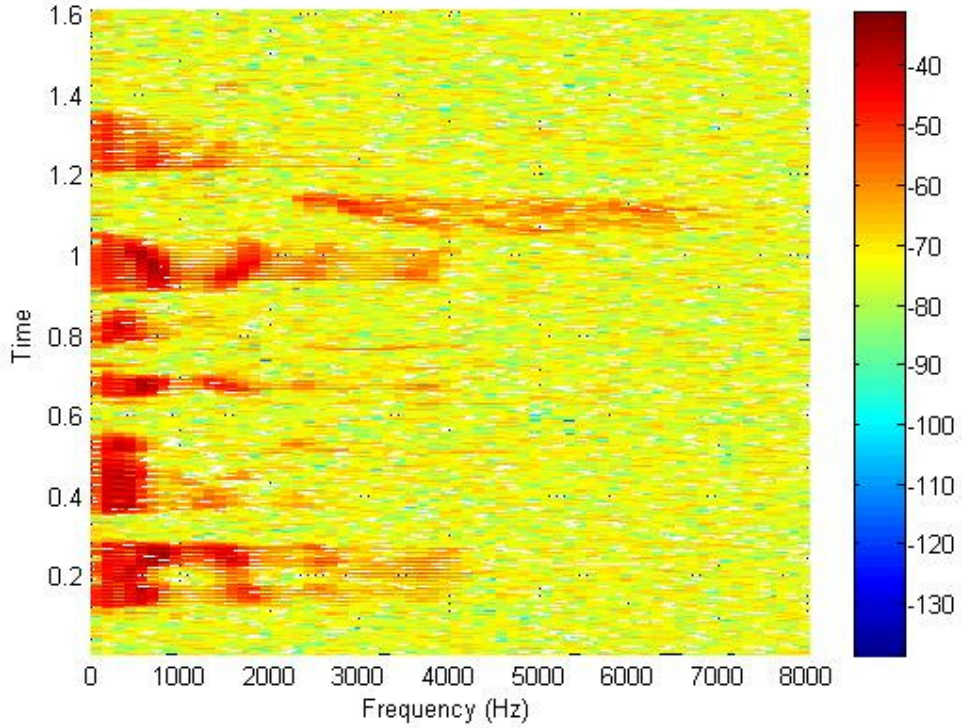

Figure 5.7: Time-frequency plot of noisy speech sample used in experiments. Signal to Noise Ratio is 30 dB

The same procedure as described in recognising clean speech (Section 5.4.2) was then used to extract the features from the corrupted speech segments and then compared with the clean features stored in the library of speaker models.

5.5.3 Speaker recognition based on distorted speech with channel compensation

Two channel compensation algorithms: the Cepstral Mean Subtraction (CMS) algorithm and the RASTA algorithm were used to compensate for the convolutional channel distortion.

To test the effectiveness of the Cepstral Mean Subtraction Algorithm (CMS) and the RASTA algorithm the channel compensation part of the system based on CMS or RASTA was added before the extraction of the Mel-Frequency Cepstral Coefficients. This is illustrated in Figure 5.8.

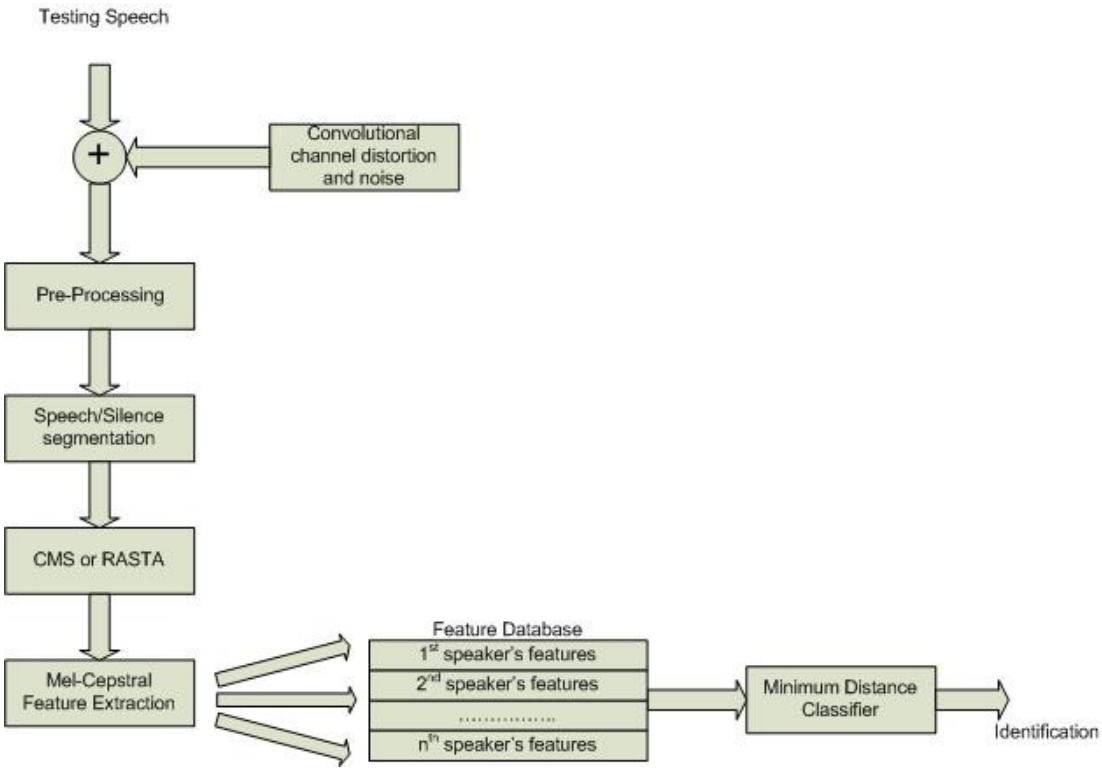

Figure 5.8: Flowchart of speaker recognition based on distorted speech with channel distortion compensated using either the CMS or RASTA algorithm.

CMS and RASTA are channel compensation algorithms designed to be used before the feature extraction phase rather than before the pre-processing phase.

To compensate for the channel noise distortion as well as for the convolutional distortion the Constant Modulus Algorithm (CMA) was used.

To test the effectiveness of the CMA, the test utterances were converted from decimal wav files into binary text files.

The binary files consisted of samples of the .wav files converted into 16 bit binary numbers.

The binary test files were then distorted by low pass filtering and the addition of white noise (See Figure 5.9). The effects of channel distortion were then equalised by the CMA algorithm. The equalised speech files were then converted back into decimal wav files and used in the speaker recognition system.

The accuracy of the channel equalisation algorithms was measured by mean squared error between the equalised speech and the original speech before the addition of the channel distortion.

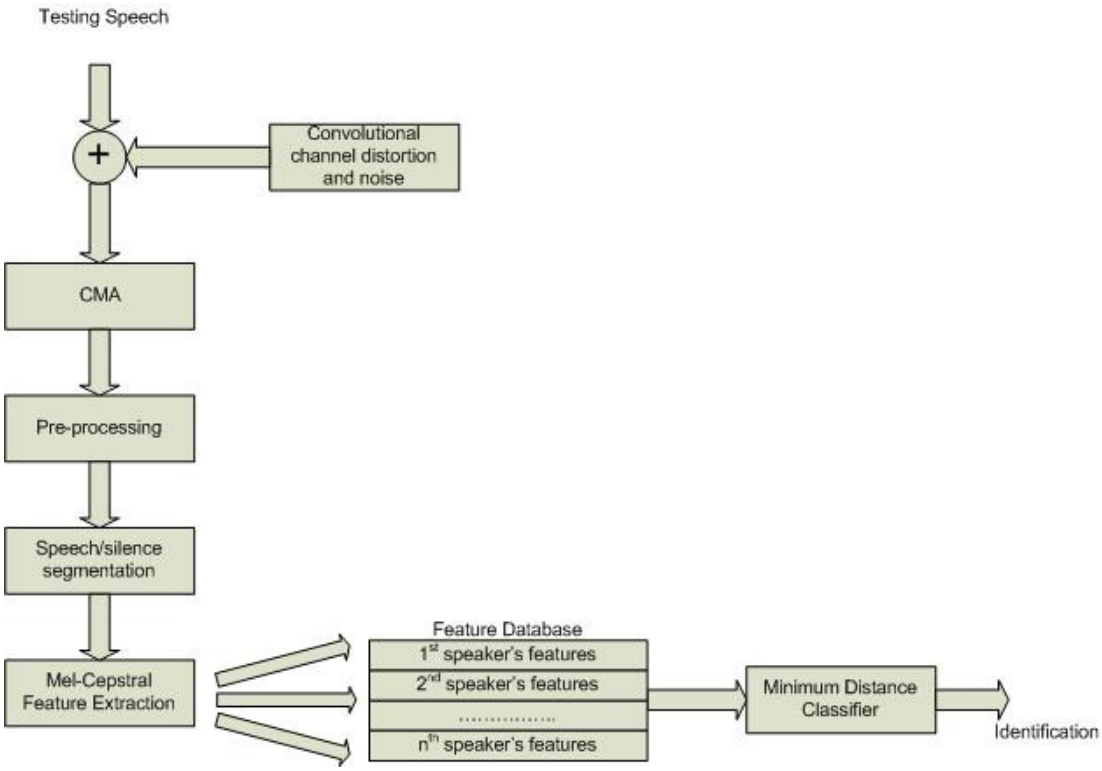

Figure 5.9: Flowchart of speaker recognition based on distorted speech with channel distortion compensated by the CMA algorithm.

5.6 Performance Measure

The performance of the speaker recognition algorithms tested in the experiments was measured by calculating the percentage of times the speakers were correctly identified over all the trials.

The following percentage recognition rates were measured:

1. Correct recognition rate for clean speech
2. Correct recognition rate for speech with channel distortion
3. Correct recognition rate for speech with channel distortion and channel compensation

The channel algorithms were assumed to give reasonable performance if the recognition rate for the equalised speech was noticeably better than the recognition rate for the distorted speech.

5.7 Test Results and Discussion

5.7.1 Results of speaker recognition for clean speech

The test data used in this experiment represented clean speech with no addition of channel distortion in the form of either noise or low pass filtering. Figure 5.10 shows the summary of speaker recognition results obtained for clean speech. The recognition rates in Figure 5.10 represent the percentage of correct classifications obtained over all tested speech samples.

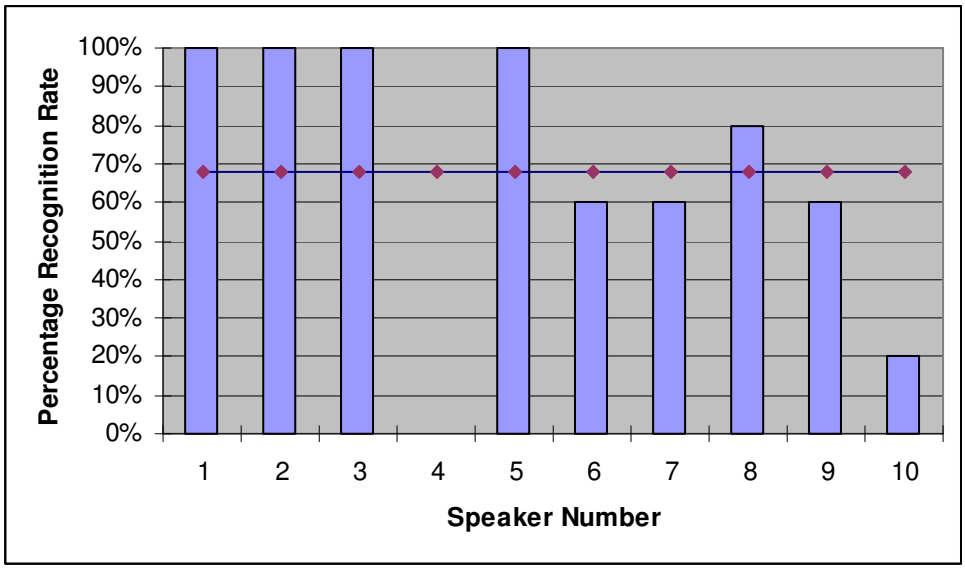

Figure 5.10: Percentage of speakers recognised from clean speech with overall average percentage recognition rate shown in pink

As indicated in Figure 5.10 the recognition rate is relatively high. For four speakers (speakers 1, 2, 3 and 5), the recognition rate is 100%, for four speakers (speakers 6, 7, 8 and 9), the recognition rate is equal or more than 60%. There was one case with recognition rate of 0% (speaker 4), which can be attributed to a number of potential problems with the speech, including the small number of training samples used in the experiments (only 1 to 4 seconds of speech) and/or the speaker's recording quality being inferior as this speaker had a lower pitched voice than the other speakers and the high pass filtering used during the pre-processing phase could have affected the low frequency characteristic features enough for this speaker to not be recognised. This issue may be able to be rectified if using a more complicated feature classification scheme or different features extracted from the speaker. The result for speaker number 10 is also low (only 20% recognition rate).

The speaker recognition rates could also be affected by the fact that the recognition had a text-independent character and different utterances were used during the training and testing phases. The text-dependent systems are usually expected to perform better for small training samples since the matching semantic speech information is used as an additional cue during the recognition process.

A larger number of training samples of longer duration would enable the system to obtain statistically more accurate characteristics of speakers during the training phase. This would lead to better recognition rates.

However, this experiment was designed to test the effects of channel distortion on a basic speaker recognition system; therefore an improvement of the system performance was outside the scope of this research.

The speaker recognition results based on clean speech were produced as a reference data allowing observing the effects of different types of channel distortion on the speaker recognition rates.

The results for speakers 4 and 10 were not treated as outliers and were kept as valid in order to observe if there will be any change in these speaker's recognition rates when channel distortion is introduced and when equalisation is applied.

5.7.2 Results of speaker recognition for distorted speech

The results presented in this section relate to the accuracy of the speaker recognition system after channel distortion has been applied. No equalisation or compensation method has been applied to the distorted speech in this section.

The following types of distortion have been analysed; low pass filtering and the addition of white Gaussian noise.

5.7.2.1 Results of speaker recognition for low pass filtered speech

The low pass filtering was expected to have some effect on the recognition rate, since the removal of the high frequency components of speech would reduce the number of Mel-Cepstral Coefficients and thus, reduce the amount of speaker-dependent characteristic information available to the system.

Table 5.1 shows a summary of the speaker recognition rates obtained for low pass filtered speech with different cutoff frequencies.

Figures 5.11 through to 5.13 show the individual results for the lowpass filters with the respective cutoff frequencies of 7, 6 and 5 kHz.

The recognition rates in Figures 5.11 through to 5.13 and Table 5.1 represent percentage of correct classifications obtained over all tested speech samples.

TABLE 5.1: PERCENTAGE OF SPEAKERS RECOGNISED FROM LOWPASS FILTERED SPEECH

Speaker number	Cutoff frequency of a lowpass Butterworth filter					
	Clean	7kHz	6kHz	5kHz	4kHz	3kHz
1	100%	80%	80%	40%	0%	0%
2	100%	100%	100%	100%	0%	0%
3	100%	80%	100%	100%	0%	0%
4	0%	0%	0%	60%	20%	0%
5	100%	100%	100%	60%	0%	0%
6	60%	80%	80%	60%	0%	0%
7	60%	60%	60%	60%	0%	0%
8	80%	80%	80%	0%	0%	0%
9	60%	80%	80%	80%	0%	0%
10	20%	40%	20%	20%	0%	0%
average	**68%**	**70%**	**72%**	**58%**	**0%**	**0%**

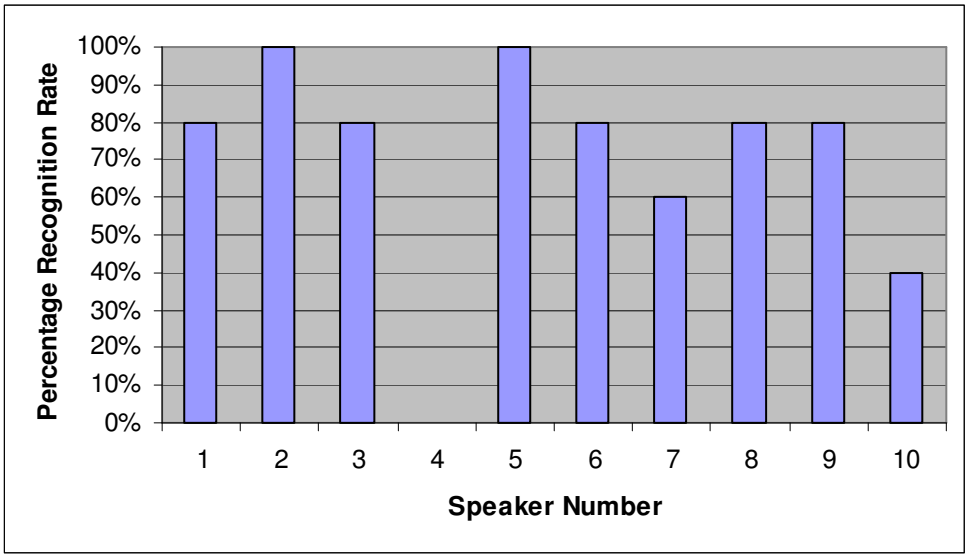

Figure 5.11: Percentage of speakers recognised from low pass filtered speech with cutoff of 7 kHz

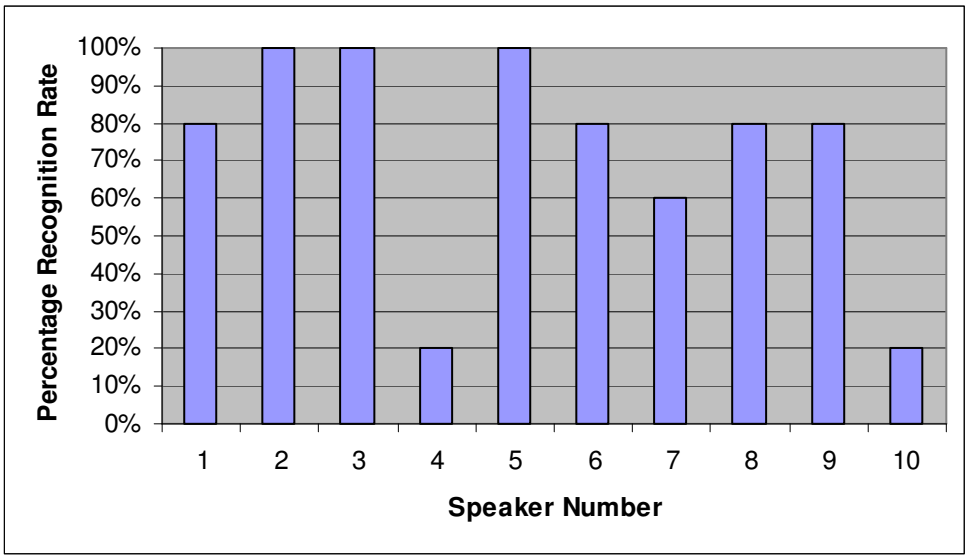

Figure 5.12: Percentage of speakers recognised from low pass filtered speech with cutoff of 6 kHz

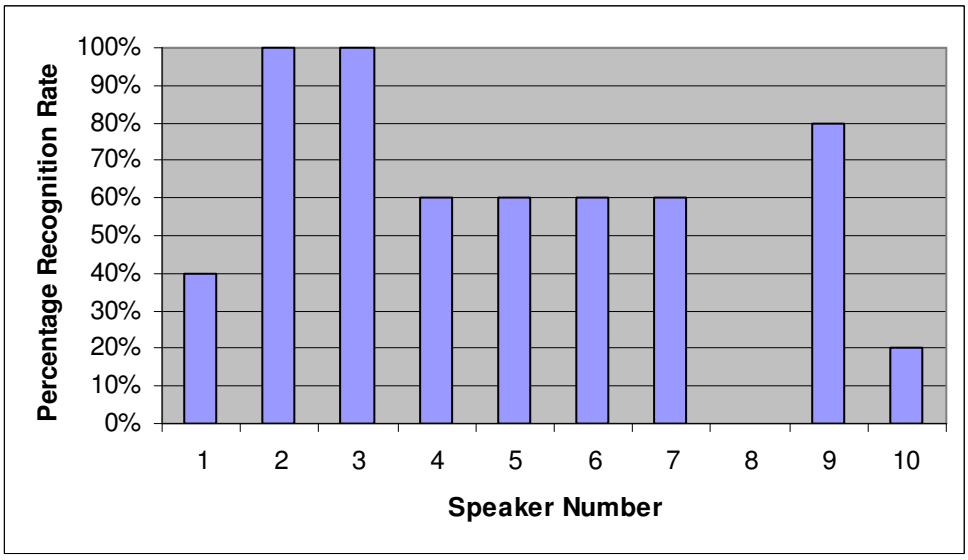

Figure 5.13: Percentage of speakers recognised from low pass filtered speech with cutoff of 5 kHz

The results of the paired t-test for the recognition rates based on clean speech versus the recognition rates based on low pass filtered speech are given in Table 5.2.

TABLE 5.2: PAIRED T-TEST FOR CLEAN SPEECH VERSUS LOW PASS FILTERED SPEECH (ALPHA =0.05)

	Clean speech versus lowpass speech				
	Cutoff 7kHz	Cutoff 6kHz	Cutoff 5kHz	Cutoff 4kHz	Cutoff 3kHz
Pearson correlation	0.912	0.9389	0.2902	Undefined	Undefined
t stat	-0.4286	-1	0.7851	6.0526	6.0526
P(T<=t) one-tail	0.3392	0.1717	0.2263	9.4943E-05	9.4943E-05
t critical one-tail	1.8331	1.8331	1.8331	1.8331	1.8331
P(T<=t) two-tail	0.6783	0.3434	0.4525	0.0002	0.0002
t critical two-tail	2.2622	2.2622	2.2622	2.2622	2.2622

Both Figures 5.11 and 5.12 show very similar results to what was obtained using clean speech. This is confirmed by the t-test which shows that there is no significant difference in the mean recognition rates between the clean speech and the low pass filtered speech with cutoff frequencies of 7, 6 and 5 kHz. For the cutoff frequencies below 5 kHz the difference does become significant.

It seems, therefore, in speech that frequencies above 5 kHz do not play an important role in the process of speaker recognition. These results would be expected as human conversational speech has an upper frequency limit of approximately 5 kHz; therefore it is likely that only speech characteristics at frequencies below 5 kHz are used in the speaker recognition process. This is confirmed in Table 5.1, which shows a rapid decline in the average recognition rates of the system as those important frequencies above 5 kHz are removed.

These results indicate that the telephone systems with bandwidths reduced to the range of 300 Hz to 3kHz may almost certainly provide significant difficulties in the process of automatic speaker recognition.

5.7.2.2 Results from recognising speakers after speech has had noise added

White Gaussian noise is also a very important factor affecting speaker recognition systems. White noise can be attributed to many different environmental sources such as the weather, storms or solar radiation. Factors such as the proximity to electrical wires and other electromagnetic devices can also cause noise in signals.

The results presented in this section show the effect noise has on speaker recognition systems.

As a speech-silence-noise segment detection algorithm is normally used in the pre-processing phase of the speaker recognition system, very noisy segment of speech could be detected as not containing any speech information and, hence, being removed and rendering the system useless. Therefore the speech-silence-noise detection algorithm was disabled on very noisy segments to prevent the entire utterance being detected as noise. Disabling this algorithm did not affect the features being extracted only the amount of time the system took to process the speech segments.

Table 5.3 shows the summary of speaker recognition results obtained for noisy speech with different values of Signal-to-Noise Ratio (SNR). Figures 5.14 through to 5.17 show the individual results for the SNR values of 40 dB, 30 dB, 20 dB and 10 dB respectively.

The recognition rates in Figures 5.14 through to 5.17 and Table 5.3 represent the percentage of correct classifications obtained over all tested speech samples.

TABLE 5.3: PERCENTAGE OF SPEAKERS RECOGNISED FROM NOISY SPEECH.

Speaker number	Signal – to – Noise Ratio				
	Clean	40 dB	30 dB	20 dB	10 dB
1	100%	80%	80%	40%	0%
2	100%	100%	100%	100%	100%
3	100%	80%	80%	20%	20%
4	0%	0%	0%	60%	0%
5	100%	100%	100%	20%	20%
6	60%	60%	40%	20%	0%
7	60%	60%	20%	0%	0%
8	80%	80%	60%	40%	20%
9	60%	80%	80%	80%	60%
10	20%	20%	20%	20%	40%
average	**68%**	**66%**	**58%**	**40%**	**26%**

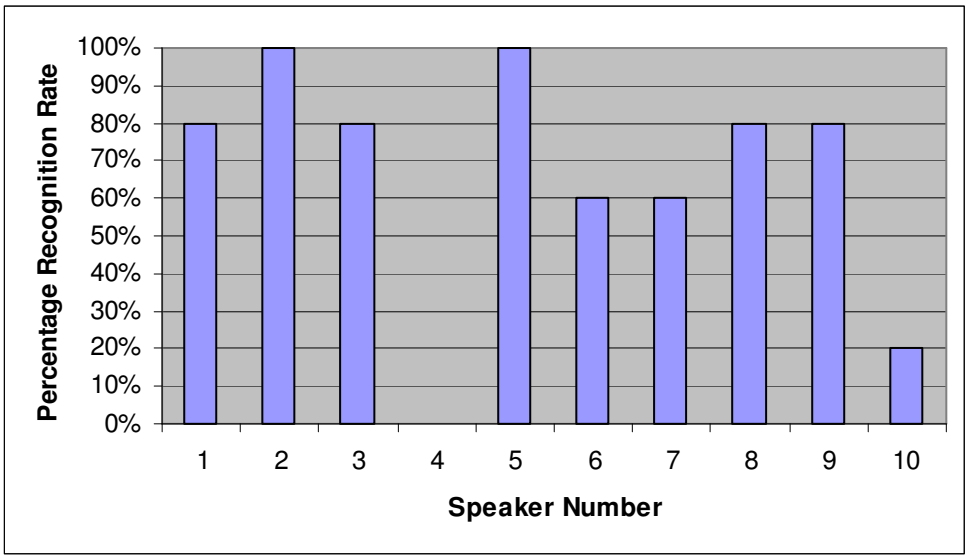

Figure 5.14: Percentage of speakers recognised from noisy speech with signal to noise ratio of 40 dB

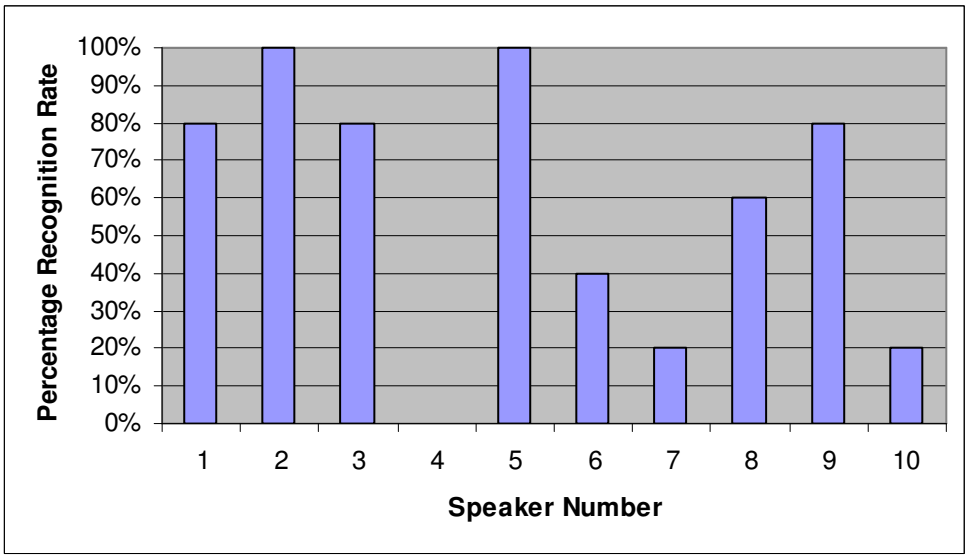

Figure 5.15: Percentage of speakers recognised from noisy speech with signal to noise ratio of 30 dB

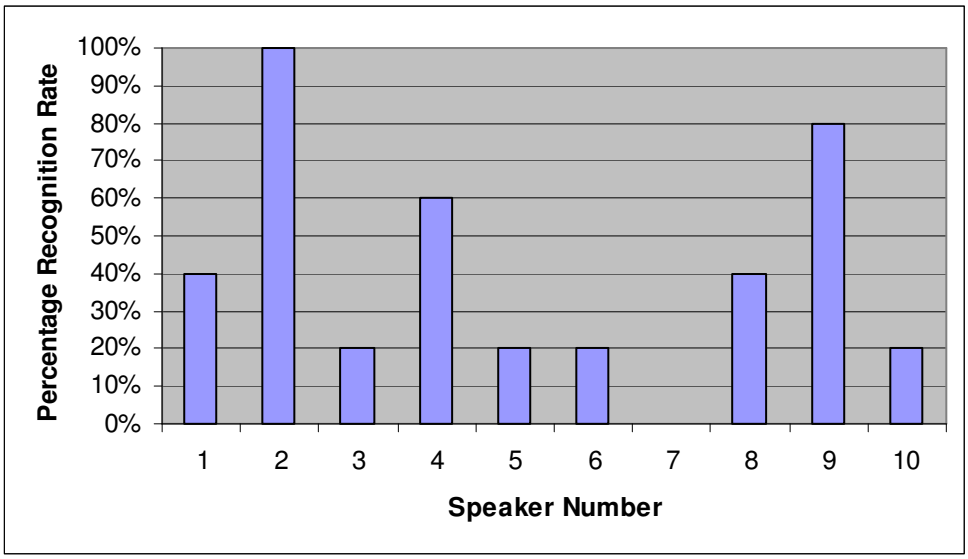

Figure 5.16: Percentage of speakers recognised from noisy speech with signal to noise ratio of 20 dB

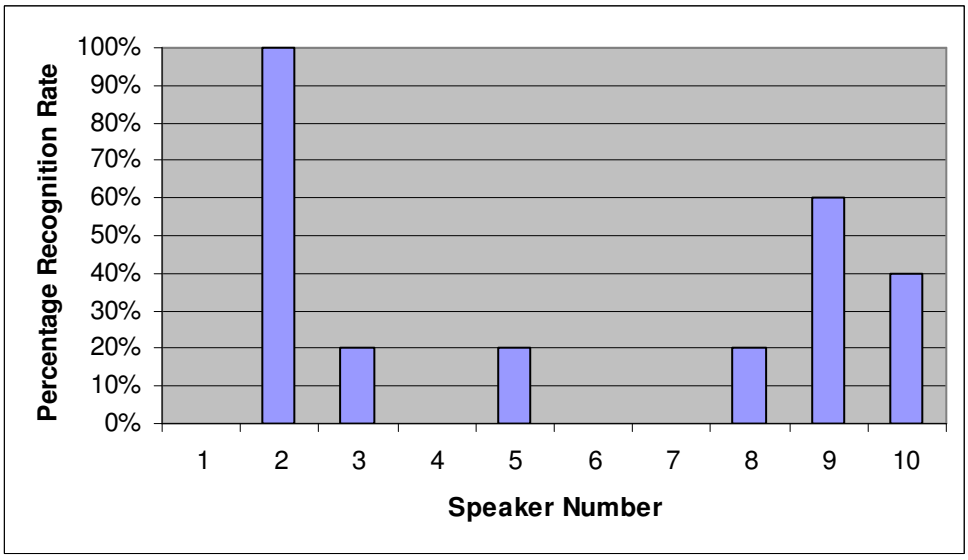

Figure 5.17: Percentage of speakers recognised from noisy speech with signal to noise ratio of 10 dB

The results of the paired t-test for the recognition rates based on clean speech versus the recognition rates based on noisy speech are given in Table 5.4.

TABLE 5.4: PAIRED T-TEST FOR CLEAN SPEECH VERSUS NOISY SPEECH

	Clean speech versus noisy speech			
	SNR=40dB	SNR=30dB	SNR=20dB	SNR=10dB
Pearson correlation	0.9479	0.8866	0.04	0.2217
t stat	0.5571	1.8605	1.9091	3.1151
P(T<=t) one-tail	0.2955	0.0479	0.0443	0.0062
t critical one-tail	1.8331	1.8331	1.8331	1.8331
P(T<=t) two-tail	0.5911	0.0957	0.0886	0.0124
t critical two-tail	2.2622	2.2622	2.2622	2.2622

Subjective listening tests showed that the gradual decrease in SNR from 40 dB to 10 dB was resulting in systematic reduction of speech intelligibility and at the same time in the reduction of human ability to recognise the speakers. At SNR=10 dB, the speakers and the semantic contents of the speech were practically unrecognisable to human listeners.

As expected the noise being added to the speech had an almost immediate effect on the recognition rate. As indicated in Table 5.4, there is a decrease in the average recognition rates over all values of SNR. The t-test results in Table 5.4 indicate that the decline in recognition rates for SNR $\leq$ 30 dB is statistically significant.

5.7.3 Test Results for Equalised speech

The results presented in this section show the effects different channel equalisation techniques have on the accuracy of speaker recognition systems.

The following equalisation methods are analysed in this section; Cepstral Mean Subtraction (CMS), the RASTA algorithm and the Constant Modulus Algorithm (CMA).

5.7.3.1 Using Cepstral Mean Subtraction (CMS) Algorithm for Compensation of low pass filtering effects

Tables 5.5 and 5.6 shows the summary of speaker recognition results obtained for the low pass filtered speech and the results using the CMS algorithm compensating for the results of low pass filtering. Figures 5.18 through to 5.22 show the individual results for the low pass filters with cutoff frequencies of 7, 6, 5, 4 and 3 kHz respectively.

The recognition rates in Figures 5.18 to 5.22 and Tables 5.5 and 5.6 represent the percentage of correct classifications obtained over all tested speech samples.

TABLE 5.5: COMPARISON OF AVERAGE RECOGNITION RATE FOR LOWPASS FILTERED SPEECH AND LOWPASS FILTERED SPEECH USING CMS.

	Cutoff frequency of a lowpass Butterworth filter				
	7kHz	6kHz	5kHz	4kHz	3kHz
Average without compensation	70%	72%	58%	0%	0%
Average using CMS	48%	44%	46%	38%	22%

TABLE 5.6: PERCENTAGE OF SPEAKERS RECOGNISED FROM LOWPASS FILTERED SPEECH WITH CMS COMPENSATION.

Speaker number	Cutoff frequency of a lowpass Butterworth filter				
	7kHz	6kHz	5kHz	4kHz	3kHz
1	80%	40%	0%	20%	60%
2	0%	0%	0%	0%	0%
3	100%	100%	100%	80%	0%
4	20%	40%	40%	20%	0%
5	100%	100%	100%	100%	0%
6	80%	60%	60%	80%	100%
7	60%	60%	80%	20%	20%
8	40%	40%	80%	60%	0%
9	0%	0%	0%	0%	40%
10	0%	0%	0%	0%	0%
average	**48%**	**44%**	**46%**	**38%**	**22%**

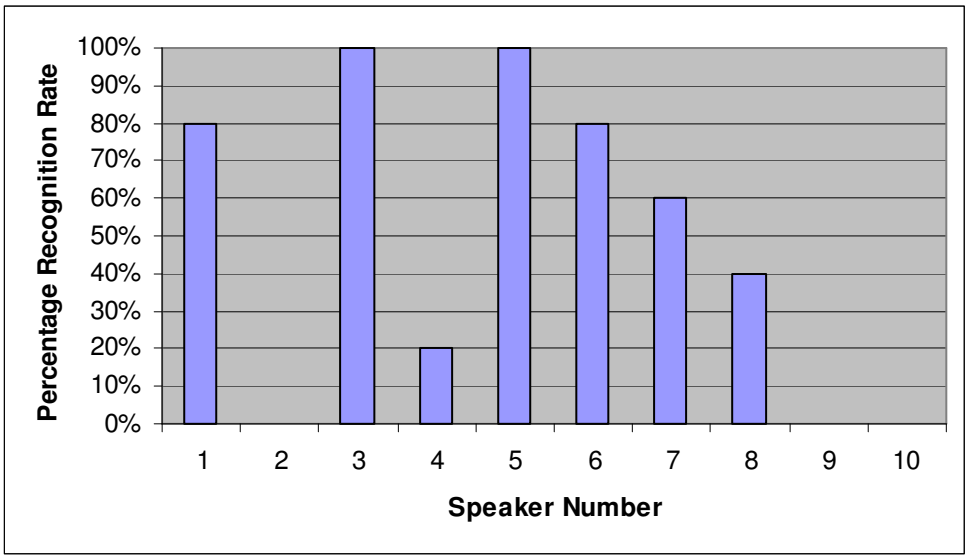

Figure 5.18: Percentage of speakers recognised from CMS equalised speech with low pass filter (cutoff = 7 kHz)

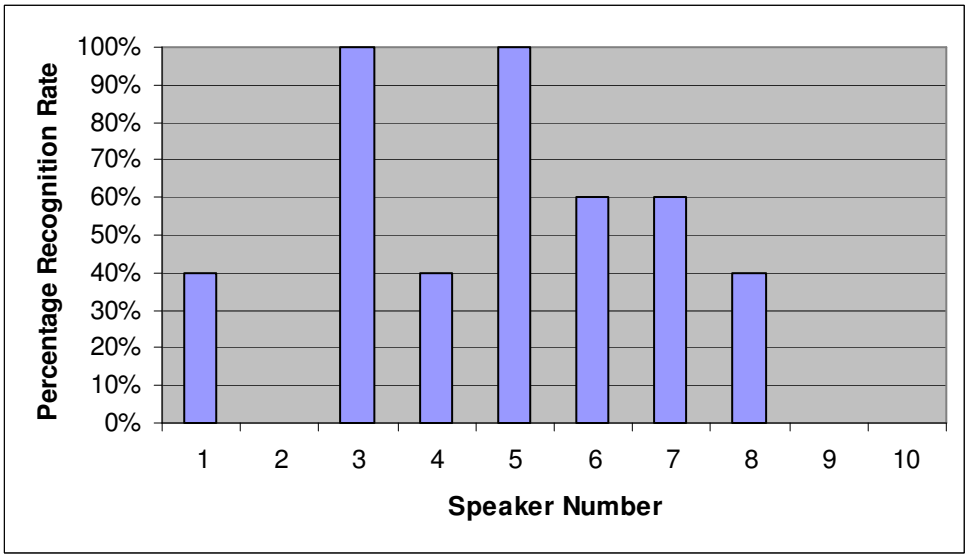

Figure 5.19: Percentage of speakers recognised from CMS equalised speech with low pass filter (cutoff = 6 kHz)

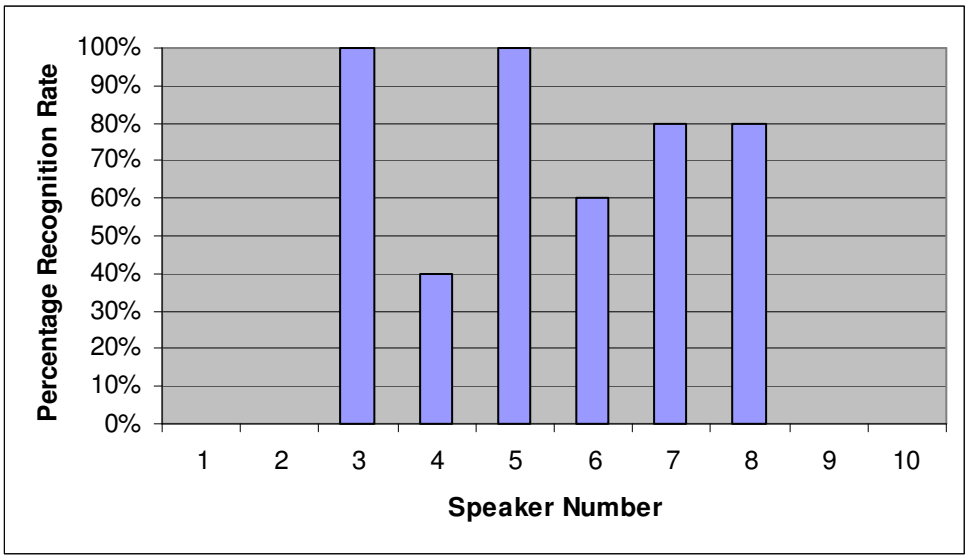

Figure 5.20: Percentage of speakers recognised from CMS equalised speech with low pass filter (cutoff = 5 kHz)

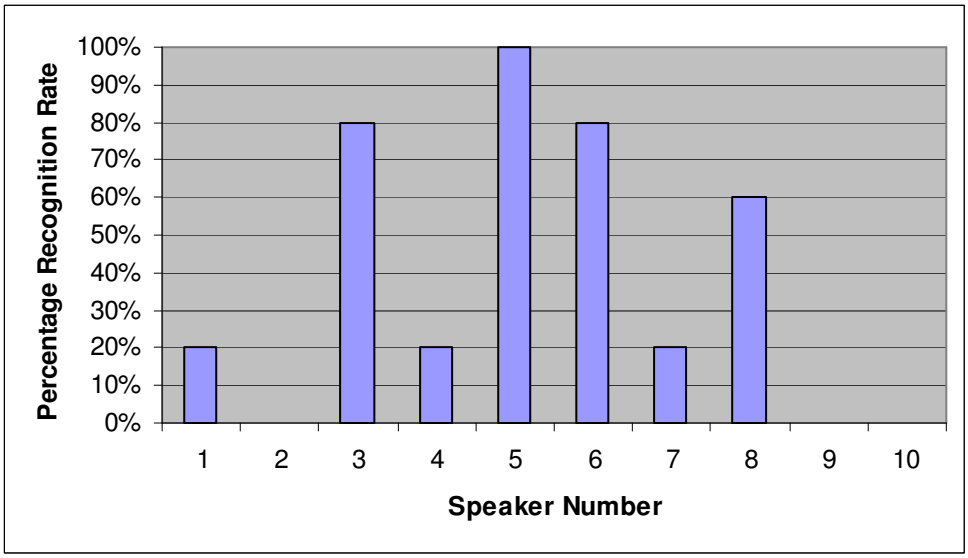

Figure 5.21: Percentage of speakers recognised from CMS equalised speech with low pass filter (cutoff = 4 kHz)

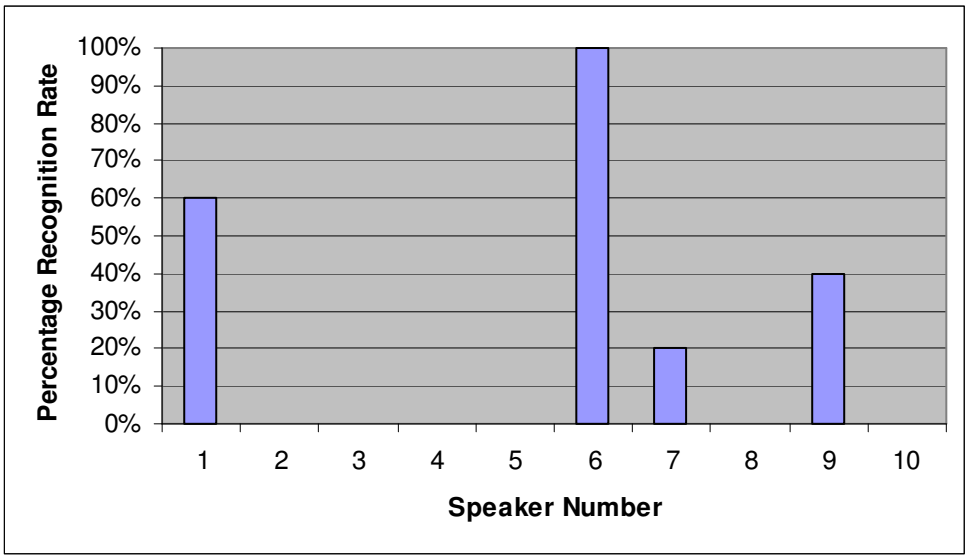

Figure 5.22: Percentage of speakers recognised from CMS equalised speech with low pass filter (cutoff = 3 kHz)

The average recognition rates for the CMS-equalised speech shown in Tables 5.5 and 5.6 show increased values compared to the corresponding average recognition rates for low pass filtered speech with no compensation for cutoff frequencies of 4 kHz and 3 kHz.

The results of the paired t-test for the recognition rates based on uncompensated low pass filtered speech versus the recognition rates based on CMS-equalised speech are given in Table 5.7.

TABLE 5.7: PAIRED T-TEST FOR LOWPASS FILTERED SPEECH VERSUS CMS-EQUALISED SPEECH

	lowpass speech versus CMS-equalised speech				
	Cutoff 7kHz	Cutoff 6kHz	Cutoff 5kHz	Cutoff 4kHz	Cutoff 3kHz
Pearson correlation	0.3564	0.386	0.0097	Undefined	Undefined
t stat	1.6732	2.3333	0.7093	-3.1425	-2.0121
P(T<=t) one-tail	0.0643	0.0223	0.2481	0.0059	0.03754
t critical one-tail	1.8331	1.8331	1.8331	1.8331	1.8331
P(T<=t) two-tail	0.1286	0.0445	0.4961	0.0119	0.07508
t critical two-tail	2.2622	2.2622	2.2622	2.2622	2.2622

Comparison of the results for the low pass filtered speech in Table 5.5 indicates that at frequencies of 5 kHz, 6 kHz and 7 kHz that there is a decline in the average recognition rate with a statistically significant value (Table 5.7) at 6 kHz.

The t-test results in Table 5.7 and the average recognition rates in Table 5.5 indicate that the increase of average speaker recognition rates due to CMS compensation is statistically significant for low pass filtered speech with cutoff frequencies of 3 kHz and 4 kHz.

The Cepstral Mean Subtraction method appears to compensate very well for errors due to filtering at very low cutoff frequencies such as 3 and 4 kHz. But CMS does not seem to compensate well for the low pass filtering effects with higher cutoff frequencies above 4 kHz.

This implies that the CMS technique is useful for improvement or speaker recognition rates when the speech is transmitted over channels with very narrow bandwidths. For wider bandwidths, this type of channel compensation could be detrimental to the speaker recognition system.

Another effect that CMS has on the speech is to remove the natural time invariant convolutional effects on speech which are not due to the channel. As discussed in Section 3.3.1 this compensation method assumes the speech signal has a zero mean which is not necessarily correct in most cases, therefore removal of this convolutional effect could also severely impact on the individuality of the speech features and therefore the accuracy of the speaker recognition system.

As stated in section 5.7.1 if longer segments of speech were used in the experiments or text-dependent rather than text-independent speech was used the result would have been expected to have improved yet again.

5.7.3.2 CMS compensation for white Gaussian noise

Tables 5.8 and 5.9 shows the summary of speaker recognition results obtained for noisy speech and the results using the CMS algorithm compensating for the results of additive white Gaussian noise. Figures 5.23 through to 5.26 show the individual results for the noisy speech with Signal-to-Noise Ratios of 40 dB, 30 dB, 20 dB and 10 dB respectively.

The recognition rates in Figures 5.23 through to 5.26 and Tables 5.8 and 5.9 represent the percentage of correct classifications obtained over all tested speech samples.

TABLE 5.8: COMPARISON OF AVERAGE RECOGNITION RATE FOR NOISY SPEECH AND NOISY SPEECH WITH CMS.

	Signal – to – Noise Ratio			
	40 dB	30 dB	20 dB	10 dB
Average without compensation	66%	58%	40%	26%
Average using CMS	38%	30%	10%	4%

TABLE 5.9: PERCENTAGE OF SPEAKERS RECOGNISED FROM NOISY SPEECH EQUALISED BY THE CMS ALGORITHM.

Speaker number	Signal – to – Noise Ratio			
	40 dB	30 dB	20 dB	10 dB
1	80%	80%	20%	0%
2	0%	0%	0%	0%
3	60%	40%	0%	0%
4	40%	20%	60%	40%
5	60%	20%	0%	0%
6	40%	40%	0%	0%
7	80%	60%	0%	0%
8	20%	40%	0%	0%
9	0%	0%	20%	0%
10	0%	0%	0%	0%
average	**38%**	**30%**	**10%**	**4%**

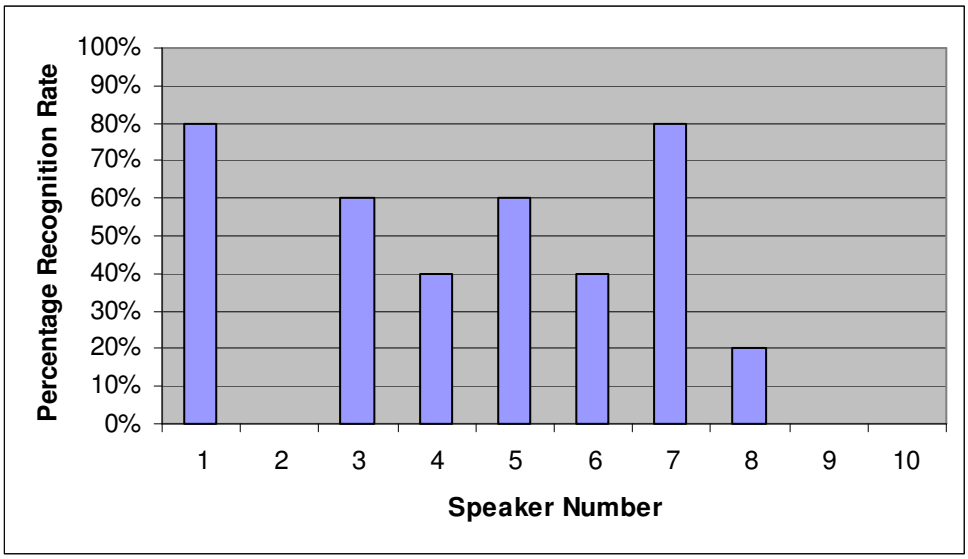

Figure 5.23: Percentage of speakers recognised from CMS equalised speech with additive noise (SNR = 40 dB)

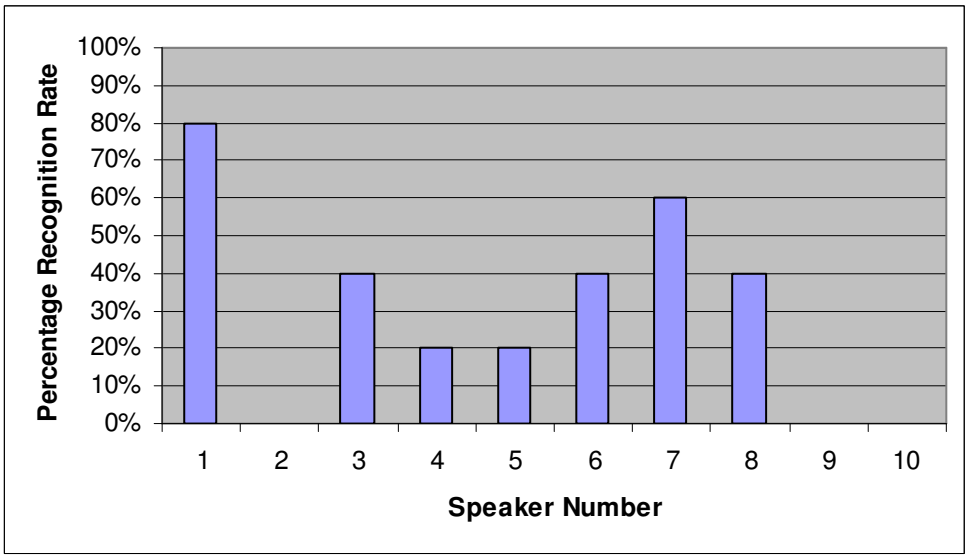

Figure 5.24: Percentage of speakers recognised from CMS equalised speech with additive noise (SNR = 30 dB)

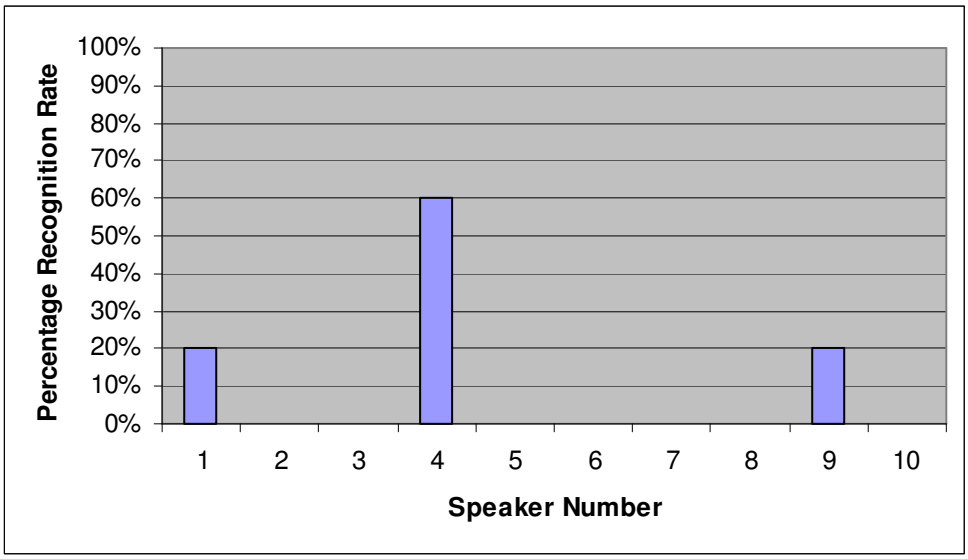

Figure 5.25: Percentage of speakers recognised from CMS equalised speech with additive noise (SNR = 20 dB)

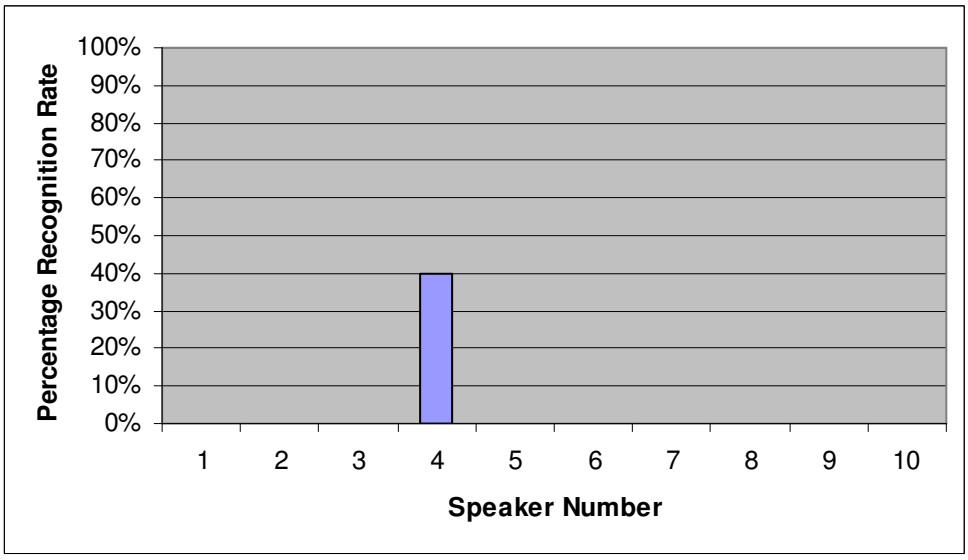

Figure 5.26: Percentage of speakers recognised from CMS equalised speech with additive noise (SNR = 10 dB)

The results of the paired t-test for the recognition rates based on uncompensated noisy speech versus the recognition rates based on CMS-equalised speech are given in Table 5.10.

TABLE 5.10: PAIRED T-TEST FOR NOISY SPEECH VERSUS CMS-EQUALISED SPEECH

	Noisy speech versus CMS-equalised speech			
	SNR=40dB	SNR=30dB	SNR=20dB	SNR=10dB
Pearson correlation	0.0979	-0.0687	0.3656	-0.2791
t stat	2.0397	1.9091	3.1429	1.8193
P(T<=t) one-tail	0.0359	0.0443	0.0059	0.0511
t critical one-tail	1.8331	1.8331	1.8331	1.8331
P(T<=t) two-tail	0.0718	0.0886	0.0119	0.1022
t critical two-tail	2.2622	2.2622	2.2622	2.2622

Comparison of the results for the noisy speech in Table 5.8 indicates that, for all of the tested values of SNR there is a decline in the average recognition rates.

The t-test results in Table 5.10 and the average recognition rates in Table 5.8 indicate that the decrease of the average speaker recognition rates due to the CMS compensation is statistically significant for all of the tested values of SNR.

In conclusion, the CMS channel equalisation technique does not provide effective compensation for noisy speech, and it does not improve the performance of this speaker recognition system in this case.

5.7.6 Using RASTA Processing for Channel Compensation

RASTA processing of speech as discussed in Section 3.3.2, was tested as an alternative to CMS, as it was specifically developed to compensate for both convolutional (filtering) as well as additive signal distortion (corruption by noise).

5.7.6.1 Using RASTA processing for low pass filtering effects

Table 5.11 and 5.12 shows the summary of speaker recognition results obtained for low pass filtered speech and the results using the RASTA algorithm compensating for the results of the low pass filtering. Figures 5.27 through to 5.31 show the individual results for the low pass filters with cutoff frequencies of 7, 6, 5, 4 and 3 kHz respectively.

The recognition rates in Figures 5.27 through to 5.31 and Tables 5.11 and 5.12 represent the percentage of correct classifications obtained over all tested speech samples.

TABLE 5.11: PERCENTAGE OF SPEAKERS RECOGNISED FROM LOWPASS FILTERED SPEECH WITH RASTA PROCESSING.

	Cutoff frequency of a lowpass Butterworth filter				
	7kHz	6kHz	5kHz	4kHz	3kHz
Average without compensation	70%	72%	58%	0%	0%
Average using CMS	48%	44%	46%	38%	22%
Average using RASTA Processing	60%	56%	58%	56%	26%

TABLE 5.12: PERCENTAGE OF SPEAKERS RECOGNISED FROM LOWPASS FILTERED SPEECH WITH RASTA COMPENSATION.

Speaker number	Cutoff frequency of a lowpass Butterworth filter				
	7kHz	6kHz	5kHz	4kHz	3kHz
1	80%	60%	60%	60%	0%
2	80%	80%	80%	80%	0%
3	100%	100%	100%	80%	0%
4	0%	0%	0%	0%	0%
5	100%	100%	100%	80%	0%
6	80%	80%	80%	80%	100%
7	20%	40%	40%	20%	0%
8	40%	40%	40%	40%	0%
9	60%	40%	40%	80%	100%
10	40%	20%	40%	40%	60%
average	**60%**	**56%**	**58%**	**56%**	**26%**

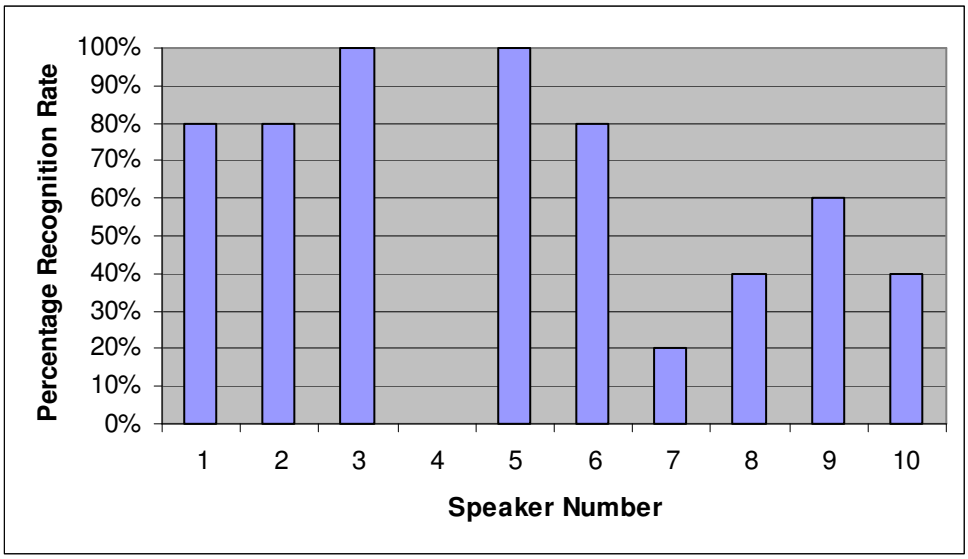

Figure 5.27: Percentage of speakers recognised from RASTA equalised speech with low pass filter (cutoff = 7 kHz)

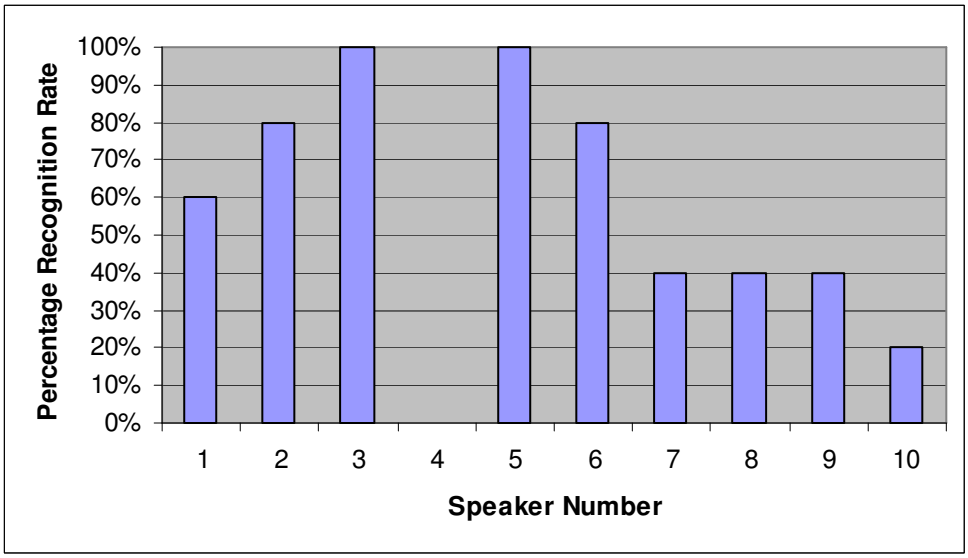

Figure 5.28: Percentage of speakers recognised from RASTA equalised speech with low pass filter (cutoff = 6 kHz)

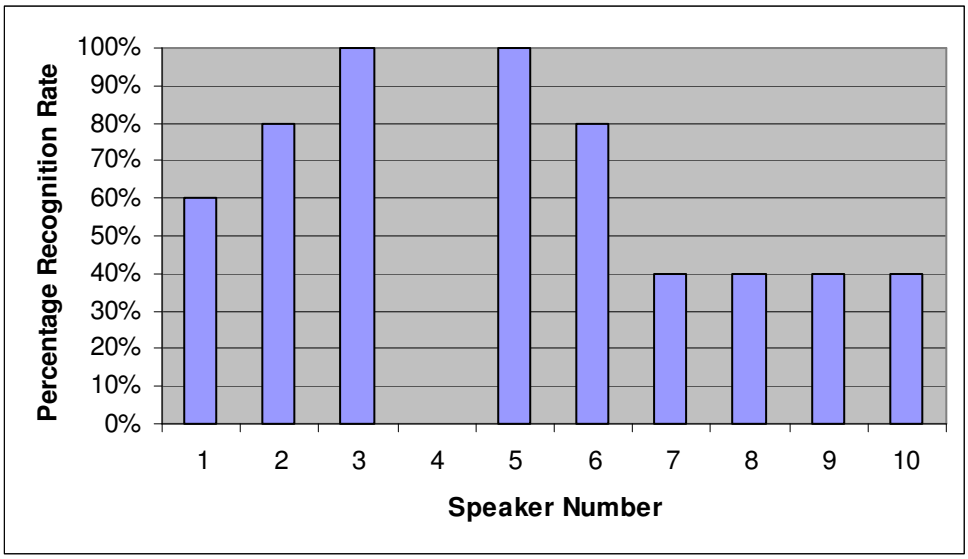

Figure 5.29: Percentage of speakers recognised from RASTA equalised speech with low pass filter (cutoff = 5 kHz)

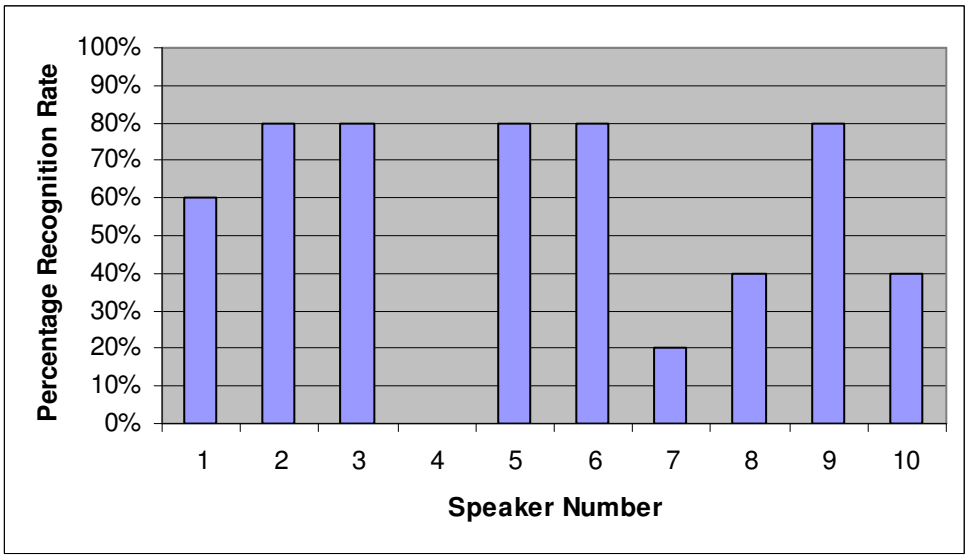

Figure 5.30: Percentage of speakers recognised from RASTA equalised speech with low pass filter (cutoff = 4 kHz)

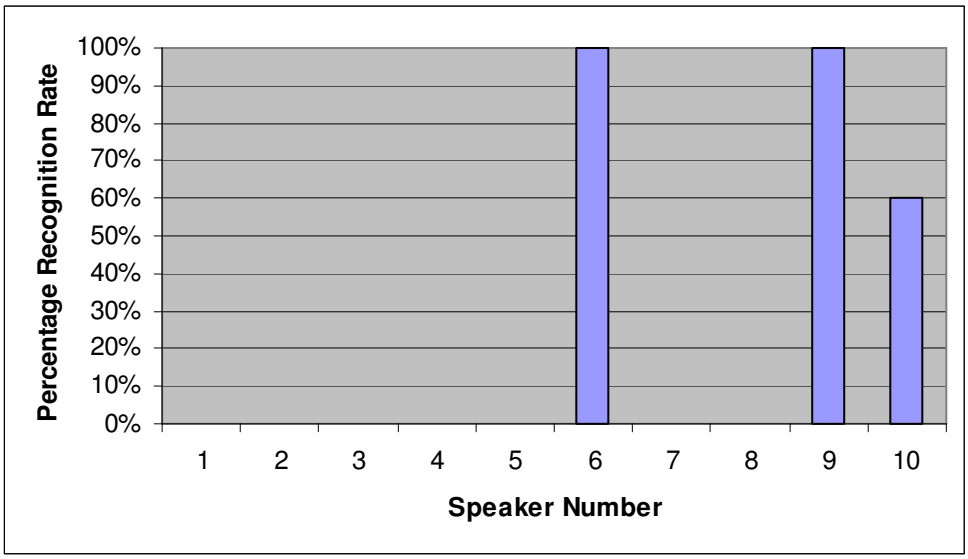

Figure 5.31: Percentage of speakers recognised from RASTA equalised speech with low pass filter (cutoff = 3 kHz)

The results of the paired t-test for the recognition rates based on uncompensated low pass filtered speech versus the recognition rates based on RASTA processed speech are given in Table 5.13.

TABLE 5.13: PAIRED T-TEST FOR LOWPASS SPEECH VERSUS RASTA-EQUALISED SPEECH

	lowpass speech versus RASTA-equalised speech				
	Cutoff 7kHz	Cutoff 6kHz	Cutoff 5kHz	Cutoff 4kHz	Cutoff 3kHz
Pearson correlation	0.823	0.8839	0.4323	Undefined	Undefined
t stat	1.627	3.2071	-2E-16	-6	-1.9007
P(T<=t) one-tail	0.0691	0.0054	0.5	0.0001	0.0449
t critical one-tail	1.8331	1.8331	1.8331	1.8331	1.8331
P(T<=t) two-tail	0.1382	0.0107	1	0.0002	0.0898
t critical two-tail	2.2622	2.2622	2.2622	2.2622	2.2622

Comparison of the results for the low pass filtered speech and RASTA processed speech in Table 5.11 indicates that at frequencies of 7 kHz and 6 kHz the RASTA processing shows a decline in the average recognition rate with a

statistically significant value (Table 5.13) at 6 kHz. At a cutoff frequency of 5 kHz the recognition rate remained the same as for uncompensated speech. However at the cutoff frequencies of 4 kHz and 3 kHz, there is a significant increase of the average recognition rate due to RASTA processing.

As indicated in Table 5.11, for all cutoff frequencies the RASTA processing method produces higher average recognition rates compared to the CMS method.

5.7.6.2 Using RASTA processing for white Gaussian noise

Tables 5.14 and 5.15 shows the summary of speaker recognition results obtained for noisy speech and the results using the RASTA algorithm compensating for the results of additive white Gaussian noise. Figures 5.32 through to 5.35 show the individual results for the noisy speech with Signal-to-Noise Ratios of 40 dB, 30 dB, 20 dB and 10 dB respectively.

The recognition rates in Figures 5.32 through to 5.35 and Tables 5.14 and 5.15 represent the percentage of correct classifications obtained over all tested speech samples.

TABLE 5.14: PERCENTAGE OF SPEAKERS RECOGNISED FROM NOISY SPEECH EQUALISED WITH THE RASTA ALGORITHM.

	Signal – to – Noise Ratio			
	40 dB	30 dB	20 dB	10 dB
Average without compensation	66%	58%	40%	26%
Average using CMS	38%	30%	10%	4%
Average using RASTA Processing	56%	54%	48%	30%

TABLE 5.15: PERCENTAGE OF SPEAKERS RECOGNISED FROM NOISY SPEECH EQUALISED WITH THE RASTA ALGORITHM.

Speaker number	Signal – to – Noise Ratio			
	40 dB	30 dB	20 dB	10 dB
1	80%	60%	60%	0%
2	80%	100%	80%	100%
3	100%	80%	60%	20%
4	0%	0%	0%	0%
5	100%	80%	40%	20%
6	80%	80%	60%	60%
7	0%	40%	60%	0%
8	20%	20%	20%	0%
9	60%	60%	80%	80%
10	40%	20%	20%	20%
average	**56%**	**54%**	**48%**	**30%**

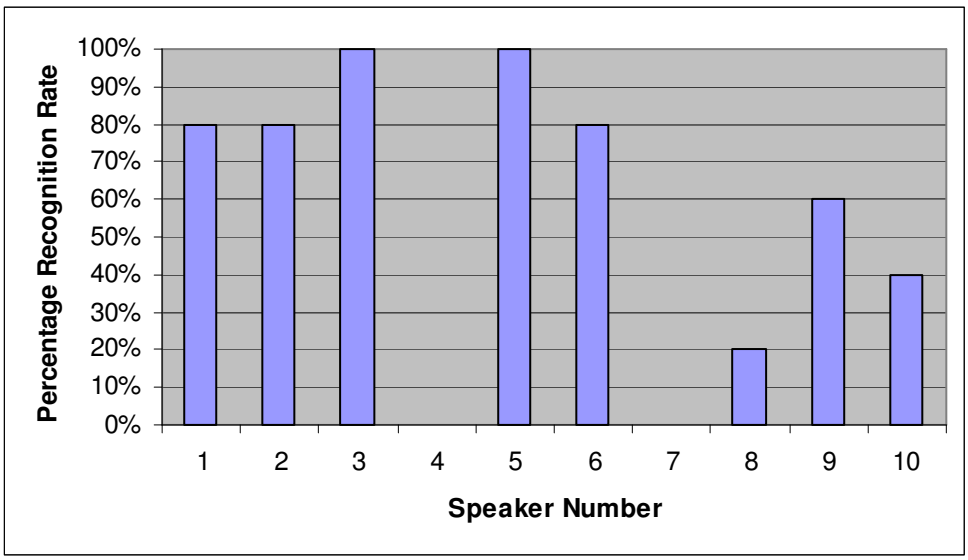

Figure 5.32: Percentage of speakers recognised from RASTA equalised speech with additive noise (SNR = 40 dB)

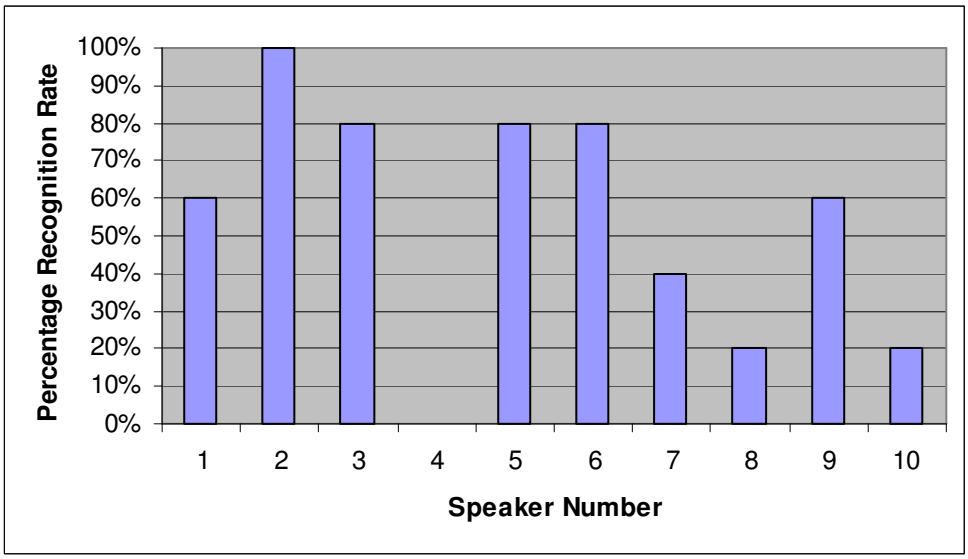

Figure 5.33: Percentage of speakers recognised from RASTA equalised speech with additive noise (SNR = 30 dB)

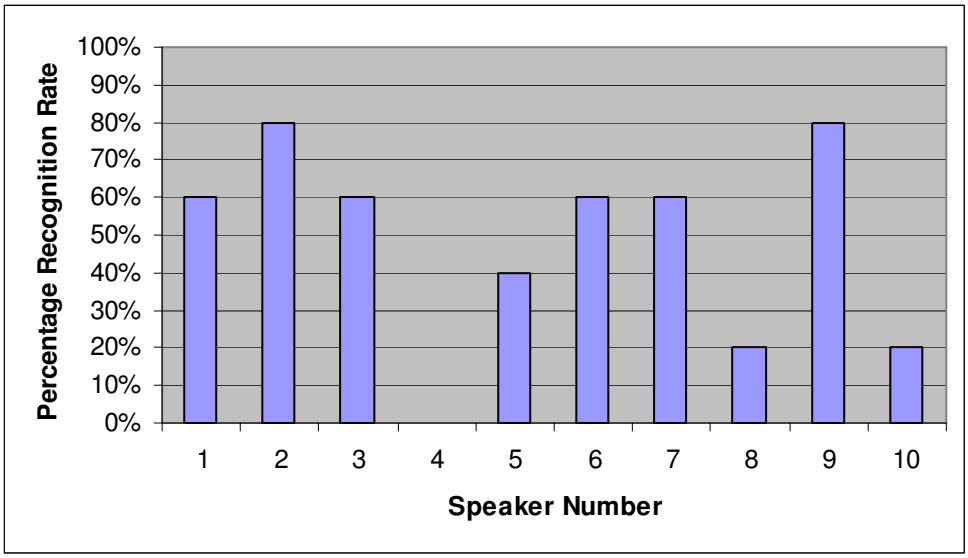

Figure 5.34: Percentage of speakers recognised from RASTA equalised speech with additive noise (SNR = 20 dB)

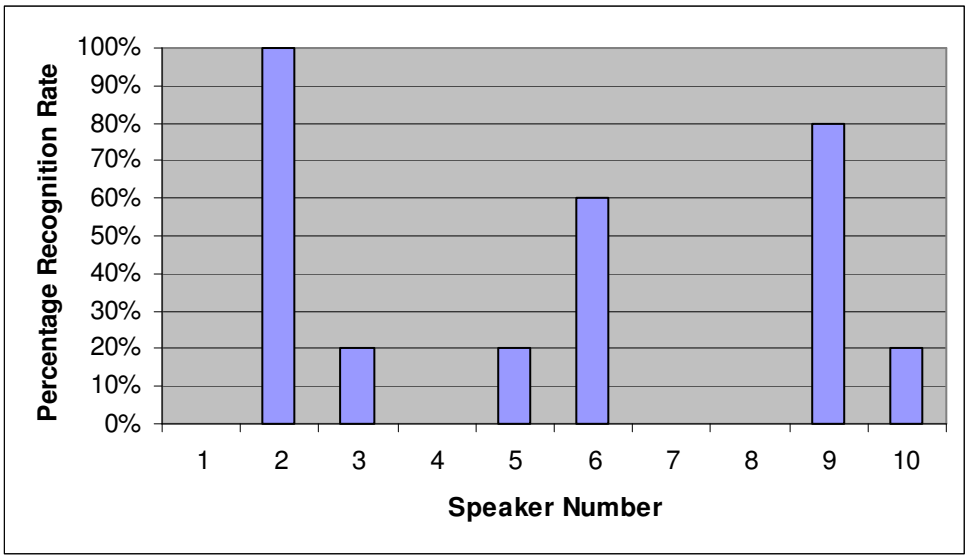

Figure 5.35: Percentage of speakers recognised from RASTA equalised speech with additive noise (SNR = 10 dB)

The results of the paired t-test for the recognition rates based on uncompensated noisy speech versus the recognition rates based on RASTA processed speech are given in Table 5.16.

TABLE 5.16: PAIRED T-TEST FOR NOISY SPEECH VERSUS RASTA-EQUALISED SPEECH

	Noisy speech versus CMS-equalised speech			
	SNR=40dB	SNR=30dB	SNR=20dB	SNR=10dB
Pearson correlation	0.6536	0.7844	0.2632	0.793
t stat	1.0476	0.5571	-0.7121	-0.5571
P(T<=t) one-tail	0.1611	0.2955	0.2472	0.2955
t critical one-tail	1.8331	1.8331	1.8331	1.8331
P(T<=t) two-tail	0.3221	0.5911	0.4945	0.5911
t critical two-tail	2.2622	2.2622	2.2622	2.2622

Comparison of the results for the noisy speech and the RASTA equalised speech in Table 5.14 indicate that at SNR values of 40 dB and 30 dB the RASTA processing shows small statistically insignificant decline in the average recognition rate, however, at SNR values of 20 dB and 10 dB, there is a small increase of the average recognition rate due to the RASTA processing.

As indicated in Table 5.14, for all tested values of SNR, the RASTA processing method produces higher average recognition rates compared to the CMS method.

In summary, both the RASTA and CMS methods produced improvements in the average recognition rates for the low cutoff frequencies of the filtered speech and for the low SNR values of the noisy speech.

The levels of improvements for RASTA were higher than for CMS particularly with corruption due to additive noise as was expected from the theory saying that RASTA took this sort of distortion into account rather than just compensating for convolutional distortion (filtering) alone.

Unfortunately, for low pass filtered speech with higher cutoff frequencies both RASTA and CMS reduced the speaker recognition rates slightly.

5.7.7 Using CMA Algorithm for Channel Equalisation

The CMA compensation algorithm was tested on speech corrupted by both a low pass filter as well as the addition of white noise.

The low pass filter impulse response used in these experiments was given as the vector: **c** = [0.04, -0.05, 0.07, -0.21, - 0.5, 0.72, 0.36, 0, 0.21, 0.03, 0.07], illustrated in Figure 5.36 and amplitude response illustrated in Figure 5.37.

As illustrated in Figure 5.37 the cutoff frequency of the low pass filter was about 7500 Hz.

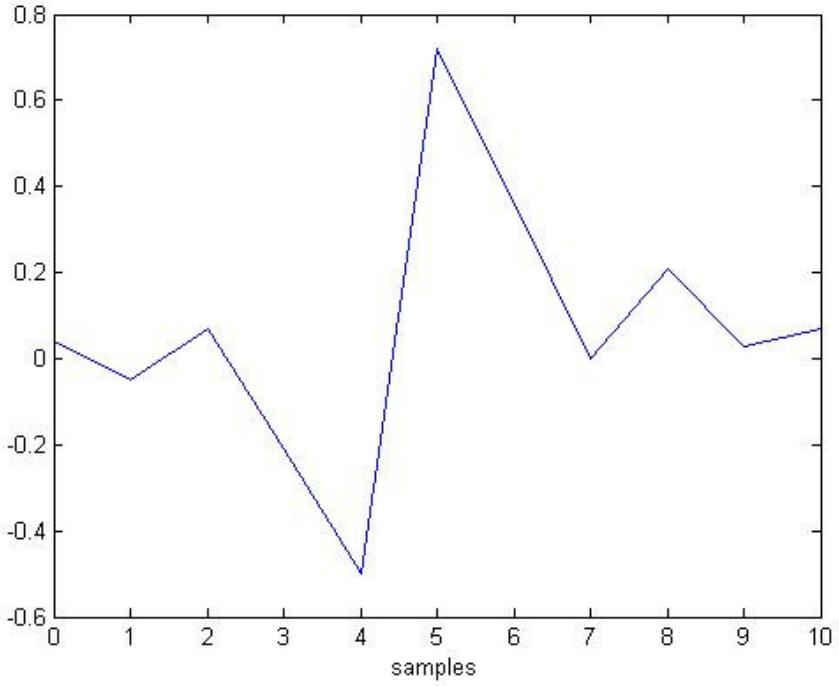

Figure 5.36: Channel impulse response used in channel simulation for CMA algorithm

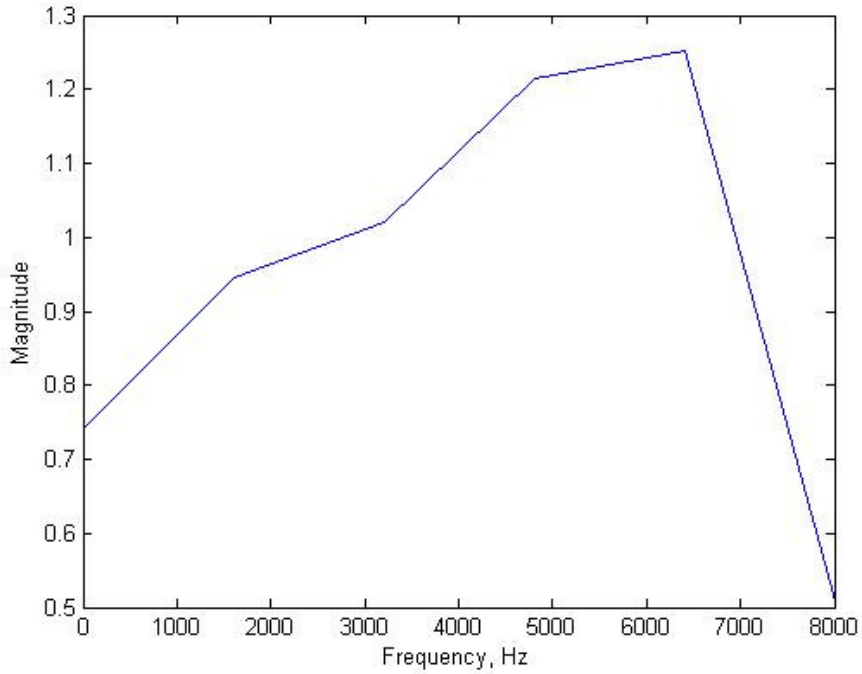

Figure 5.37: Channel amplitude response used in channel simulation for CMA algorithm

The level of noise used in the channel simulation had the SNR values of 10, 20, 30 and 40 dB. When using SNR values below 20 dB, the speech-silence-noise detection algorithm was disables to prevent all the speech being deleted (section 5.7.2).

5.7.7.1 CMA compensation results for the low pass filtered and noisy speech

Tables 5.17 and 5.18 shows the summary of speaker recognition results obtained for noisy speech and the results using the CMA compensation algorithm. Figures 5.38 through to 5.41 show the individual results for the noisy speech with Signal-to-Noise Ratios of 40 dB, 30 dB, 20 dB and 10 dB respectively.

The recognition rates in Figures 5.38 through to 5.41 and Tables 5.17 and 5.18 represent the percentage of correct classifications obtained over all tested speech samples.

TABLE 5.17: PERCENTAGE OF SPEAKERS RECOGNISED FROM NOISY SPEECH EQUALISED WITH THE CMA ALGORITHM.

	Signal – to – Noise Ratio			
	40 dB	30 dB	20 dB	10 dB
Average without compensation	66%	58%	40%	26%
Average using CMS	38%	30%	10%	4%
Average using RASTA Processing	56%	54%	48%	30%
Average using CMS Algorithm	56%	52%	52%	8%

TABLE 5.18: PERCENTAGE OF SPEAKERS RECOGNISED FROM NOISY SPEECH EQUALISED WITH THE CMA ALGORITHM.

Speaker number	Signal – to – Noise Ratio			
	40 dB	30 dB	20 dB	10 dB
1	80%	80%	80%	0%
2	100%	80%	80%	20%
3	80%	80%	100%	0%
4	0%	0%	0%	0%
5	80%	60%	40%	0%
6	80%	80%	60%	0%
7	20%	20%	20%	0%
8	60%	20%	40%	0%
9	60%	80%	100%	60%
10	0%	20%	0%	0%
average	**56%**	**52%**	**52%**	**8%**

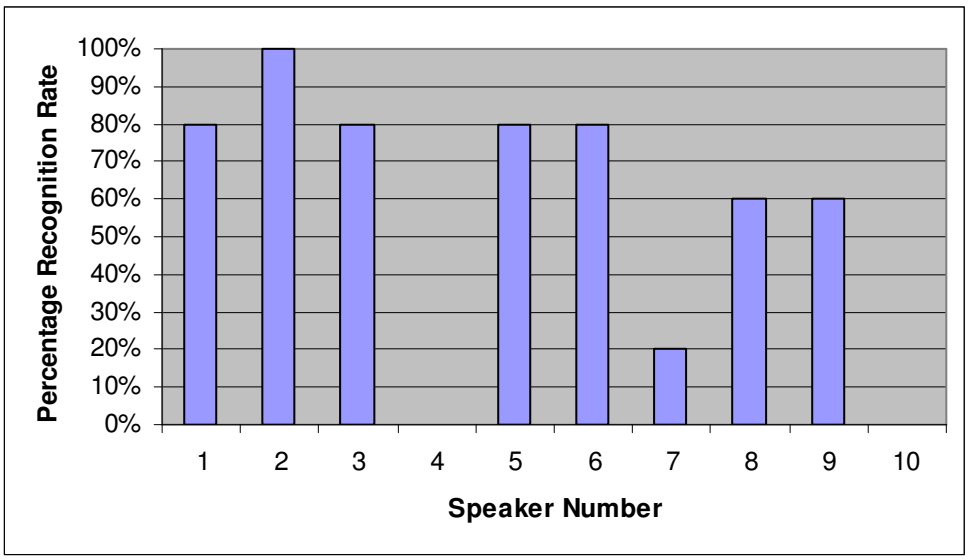

Figure 5.38: Percentage of speakers recognised from CMA equalised speech with additive noise (SNR = 40 dB)

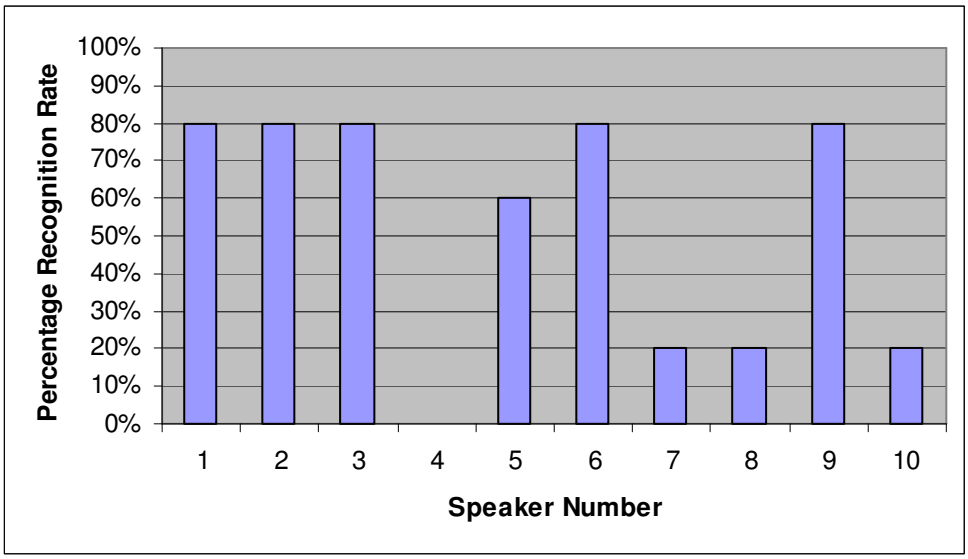

Figure 5.39: Percentage of speakers recognised from CMA equalised speech with additive noise (SNR = 30 dB)

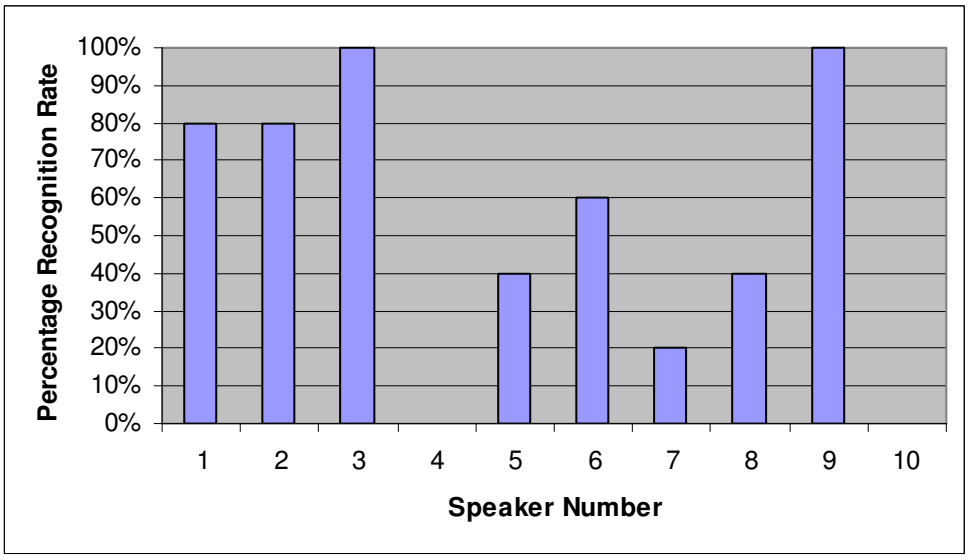

Figure 5.40: Percentage of speakers recognised from CMA equalised speech with additive noise (SNR = 20 dB)

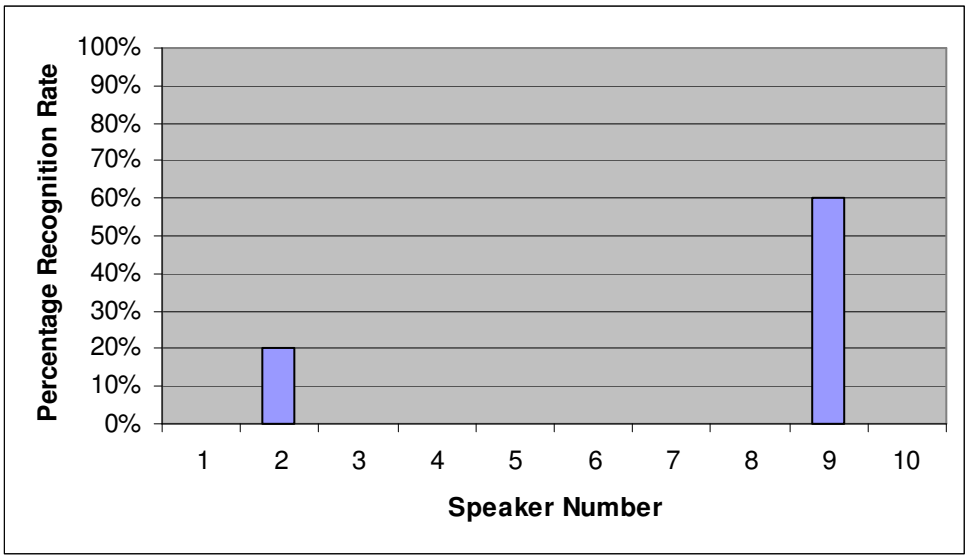

Figure 5.41: Percentage of speakers recognised from CMA equalised speech with additive noise (SNR = 10 dB)

As discussed in Section 5.7.1, at high cutoff frequencies (above 6 kHz), the recognition rates are similar to these for clean speech, thus the CMA results in this experiment were expected to be affected mostly by the addition of noise and were compared to the results for noisy speech from Section 5.7.2.

The results of the paired t-test for the recognition rates based on uncompensated noisy speech versus the recognition rates based on the CMA-equalised speech are given in Table 5.19.

TABLE 5.19: PAIRED T-TEST FOR NOISY SPEECH VERSUS CMA-EQUALISED SPEECH

	Noisy speech versus CMA-equalised speech			
	SNR=40dB	SNR=30dB	SNR=20dB	SNR=10dB
Pearson correlation	0.8836	0.7758	0.3746	0.6185
t stat	1.86052	0.8182	-0.97	2.2119
P(T<=t) one-tail	0.04787	0.2172	0.1786	0.0271
t critical one-tail	1.8331	1.8331	1.8331	1.8331
P(T<=t) two-tail	0.09573	0.4344	0.3572	0.0543
t critical two-tail	2.2622	2.2622	2.2622	2.2622

Comparison on the results in Tables 5.17 through to 5.19 show that as the SNR value increased so did the accuracy of the speaker recognition system.

The results in Table 5.17 and Table 5.19 show that only for SNR=20 dB does the CMA algorithm give a higher recognition rate compared to uncompensated noisy speech, however, this improvement is statistically insignificant.

Chapter 6 – Conclusions and Future Research Directions

6.1 Introduction

The experiments presented in this thesis investigate the effects of channel distortion on the average speaker recognition rates and testing the effectiveness of various channel compensation algorithms designed to mitigate these channel effects.

The speaker recognition system was simulated using a basic recognition algorithm consisting of the following components: speech analysis calculating feature vectors in the form of Mel-Frequency Cepstral Coefficients and the classification component based on the minimum distance algorithm.

Two types of channel distortion were investigated:

- Convolutional (or low pass filtering) effects,
- Addition of white Gaussian noise.

Three types of channel compensation algorithms were tested:

- Cepstral Mean Subtraction (CMS),
- RelAtive SpecTrAl (RASTA) Processing,
- Constant Modulus Algorithm (CMA).

6.2 Effects of Low Pass Filtering on Recognition rates

The results show that for low pass filtering the speech segments there is no significant difference in the mean recognition rates between the clean speech and the low pass filtered speech with cutoff frequencies of 7, 6 and 5 kHz. For speech filtered with low pass filter cutoff frequencies below 5 kHz, the average recognition rates for the filtered speech drops significantly to a zero recognition rate.

It indicates that the spectral features of speech above 5 kHz do not play an important role in the process of speaker recognition. This result would be expected as human conversational speech has an upper frequency limit of approximately 5 kHz;

therefore it is likely that only speech characteristics at frequencies below 5 kHz are used in speaker recognition by humans.

6.3 Effect of white Gaussian noise on speaker recognition rates

The speaker recognition tests based on noisy speech showed that a gradual decrease in the SNR from 40 dB to 10 dB resulted in a systematic reduction of average recognition rates.

For all values of SNR the speaker recognition rates for noisy speech were lower than those for clean speech. The decline in recognition rates were statistically significant for SNR less than or equal to 30 dB

For situations where there is a high level of noise distorting the speech these experiments showed that a compensation method would be needed for an effective recognition rate to be achieved.

6.4 Results of Cepstral Mean Subtraction Compensation

6.4.1 CMS compensation of low pass filtered speech

A comparison of the results obtained from the experiments using no compensation methods and experiments using CMS to compensate for low pass filtering indicate that at frequencies of 5 kHz, 6 kHz and 7 kHz there was a decline in the average recognition rate after compensation with a statistically significant value at 6 kHz. This result could have occurred due to the CMS algorithm removing the natural mean in the speech due to speaker variability in addition to removing the convolutional effects caused by the low pass filters.

For low pass filters with cutoff frequencies below 5 kHz, the average recognition rates after CMS compensation were higher than before compensation. The increase of the average speaker recognition rates due to the CMS compensation was statistically significant for the cutoff frequencies of 3 kHz and 4 kHz.

The Cepstral Mean Subtraction compensation method proved to compensate very well for the effects of low cutoff frequencies (below 4 kHz). CMS does not

seem to compensate well for the low pass filtering effects with the cutoff frequencies above 4 kHz.

6.4.2 CMS compensation of noisy speech

Comparison of the speaker recognition rates for uncompensated noisy speech and noisy speech with CMS compensation indicates that for all tested SNR values there was a decline in the average recognition rates.

Over all SNR values, the decrease of the average speaker recognition rates was statistically significant.

It was found that that the CMS channel compensation algorithm is ineffective in noisy situations and does not improve the average speaker recognition rates.

6.5 Results of RASTA compensation

6.5.1 RASTA compensation of Low Pass filtered speech

A comparison of the results obtained from the experiments of uncompensated low pass filtered speech and experiments using RASTA processing to compensate for low pass filtering indicate that at cutoff frequencies of 7 kHz, 6 kHz and 5 kHz, the RASTA method shows a decline in the average recognition rates with a statistically significant decrease at 6 kHz. At the lower cutoff frequencies of 4 kHz and 3 kHz, there is a statistically significant increase of the average recognition rates after RASTA processing.

It was also observed that, for all of the tested values of cutoff frequencies RASTA processing performed better than the CMS compensation method.

6.5.1 RASTA compensation of noisy speech

A comparison of the results obtained from the experiments of uncompensated noisy speech and experiments using RASTA processing to compensate for white Gaussian noise indicate that, at SNR values of 40 dB and 30 dB the RASTA processing method shows a small and statistically insignificant decline in the average recognition rates, however at SNR clues of 20 dB and 10 dB, there is a small, also

statistically insignificant increase of the average recognition rate after RASTA processing.

Additionally it was observed that for all tested values of SNR, the RASTA compensation method produced a higher recognition rate compared to the CMS compensation method.

6.6 Results of Constant Modulus Algorithm on noisy speech

The tests of the CMA compensation for noisy speech showed that at only SNR = 20 dB was there any improvement in recognition rates compared to the uncompensated noisy speech, however this improvement was found to be statistically insignificant.

6.7 Comparison of methods and summary

In summary, out of the three different channel compensation methods analysed it was shown that both RASTA and the CMS method produced improvements in the average speaker recognition rates for the low cutoff frequencies (4 kHz and 3 kHz) compared to the low pass filtered speech without compensation. The levels of improvements due to RASTA compensation were higher than the levels of improvements due to the CMS compensation method.

Neither the CMS or RASTA methods were able to improve the accuracy of the speaker recognition system for cutoff frequencies of 5 kHz, 6 kHz or 7 kHz.

In the case of noisy speech, all methods analysed were unable to compensate for high SNR of 40 dB and 30 dB and only RASTA processing was able to compensate and improve the average recognition rates for speech corrupted with a high level of noise (SNR of 20 dB and 10 dB).

6.8 Future research directions

Future research directions stemming from this research could include testing the channel equalisation methods from this work using more complex speaker recognition classifiers such as using Vector Quantisation (VQ), Gaussian Mixture Models (GMM) or the Hidden Markov Models (HMM) which are much more

complicated classifiers than the minimum distance classifier used in these experiments. These methods also rely on statistical rules for classification which could increase the performance of the system.

The impact of illness, age, prosthetics as well as many other problems that can affect the shape of the oral cavity and vocal tract is also another area that could be researched to aid in improving speaker recognition and verification systems.

Another field stemming from the work is on the transmission of speech through wireless channels includes implementing modulation algorithms and compression schemes on the testing speech in addition to the channel simulations completed in this research. This could be a beneficial area of research since wireless communication systems are being used on a much wider scale and use modulation and compression schemes on the data for transmission, which could cause potential errors in frequency information stored in speech in addition to the distortion occurring from the channel analysed in this work.

References

[1] Atal B. S., "Automatic Recognition of Speakers from Their Voices." *Proceedings of the IEEE.* Volume 64, No. 4, pp: 460 – 475. Apr 1976.

[2] Cheun R. S., "Feature Selection via Dynamic Programming for Text-Independent Speaker Identification." *IEEE transactions on Acoustics, Speech and Signal Processing,* Volume 26, No. 5, pp: 397-403, 1978.

[3] Zhonghua F., Rongchun Z., "An Overview of Modelling Technology of Speaker Recognition", in *Proceedings of the IEEE International Conference on Neural Networks & Signal Processing,* pp: 887 – 891, Dec 2003.

[4] Reynolds D. A. "An Overview of Automatic Speaker Recognition Technology." in *Proceedings of IEEE International Conference of Acoustics, Speech and Signal Processing, ICASSP,* Volume 4: pp: 4072 – 4075, May 2002.

[5] Reynolds D. A. "Channel Robust Speaker Verification via Feature Mapping." in *Proceedings of the IEEE International Conference of Acoustics, Speech and Signal Processing,* ICASSP 2003. Volume: 2, pp: 53 – 56, 2003.

[6] Lynch Jr. J. F. Josenhans J. G. Crochiere R. E. "Speech/Silence Segmentation for Real-Time Coding via Rule Based Adaptive Endpoint Detection." in *Proceedings of the IEEE International Conference of Acoustics, Speech and Signal Processing, ICASSP 1987.* Volume. 12 , Apr 1987 pp. 1348 – 1351.

[7] Reynolds, D. A., Quatieri T. F., Dunn R. B., "Speaker Verification Using Adapted Gaussian Mixture Models," *Digital Signal Processing*, Volume 10, pp. 19-41, 2000.

[8] Rabiner L. R. Sambur M. R., "An Algorithm for Determining the Endpoints of Isolated Utterances." *The Bell Systems Technical Journal.* Volume 54, No 2, pp: 297 – 315, 1975.

[9] Quatieri T. F., *Discrete-Time Speech Signal Processing: Principles and Practice,* Prentice Hall, New Jersey, 2002.

[10] Doddington G. R., "Speaker Recognition – Identifying People by their Voices." in *Proceedings of the IEEE.* Volume 73, No. 11, pp: 1651 – 1664, Nov 1985.

[11] Gish H., Schmidt M. "Text-independent speaker identification." *IEEE Signal Processing Magazine,* Volume 11, Issue 4, pp: 18 – 32, Oct. 1994.

[12] Reynolds D.A., "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models", *IEEE Transactions on Speech and Audio Processing,* pp: 72-83, Jan 1995.

[13] Davis S. B. Mermelstein P. "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences." *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Volume: 28, No 4, pp: 357 – 366 Aug 1980.

[14] Linde Y., Buzo A., Gray R. M., "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications*. Volume COM-28, No. 1. Jan 1980.

[15] Soong F. K., Rosenberg A. E., Rabiner L. R., Juang B. H., "A Vector Quantization Approach to Speaker Recognition" *Proceedings of IEEE International Conference of Acoustics, Speech and Signal Processing, ICASSP 1985.* Volume 1. pp: 387-390, Apr 1985.

[16] Vaseghi S. V., *Advanced Digital Signal Processing and Noise Reduction,* 3rd Edition, John Wiley & Sons, West Sussex, England, 2006.

[17] Hussain Z. M. *Digital Signal Processing*, RMIT Press, 2005.

[18] Proakis J.G., Salehi M. *Communication Systems Engineering*, 2nd Edition, Prentice Hall, New Jersey, 2002.

[19] Atal B. S., "Effectiveness of Linear Prediction Characteristics of the Speech Wave for Automatic Speaker Identification and Verification." *Journal of the Acoustics Society of America.* Volume 55, No. 6, pp: 1304 – 1312, June 1974.

[20] Heck L. P., Konig Y., Sonmez M. K., Weintraub M., "Robustness to telephone handset distortion in speaker recognition by discriminative feature design." *Speech Communication* Volume 31 (2-3): pp: 181-192 Jun 2000.

[21] Gish H., Karnofski K., Krasner M., Rouscos S., Schwartz R., Wolf J., "Investigation of Text-Independent Speaker Identification over Telephone Channels." in *Proceedings of IEEE International Conference of Acoustics, Speech and Signal Processing, ICASSP 1985,* Volume 10. pp: 379-382, Apr 1985.

[22] Childers, D.G., Skinner, D.P., Kemerait, R.C., "The Cepstrum: A guide to processing." in *Proceedings of the IEEE* Volume 65, Issue: 10, pp: 1428 – 1443, Oct. 1977.

[23] Hermansky H. Morgan N. "RASTA Processing of Speech." *IEEE Transactions on Speech and Audio Processing.* Volume 2, No. 4, pp: 578 – 589. Oct 1994.

[24] DeVeth J. Boves L. "Phase Corrected RASTA for Automatic Speech Recognition Over the Phone," in *Proceedings of IEEE International Conference of Acoustics, Speech and Signal Processing, ICASSP 1997*. Volume: 2, pp. 1239 – 1242, Apr 1997.

[25] Gong Y. F., "Speech Recognition in Noisy Environments - A Survey." *Speech Communication*, Vol. 16 Issue. 3, pp. 261 – 291, Apr 1995.

[26] Hardt D., Fellbaum K. "Spectral Subtraction and RASTA-Filtering in Text-Dependent HMM Based Speaker Verification*."* in *Proceedings of IEEE International Conference of Acoustics, Speech, and Signal Processing, ICASSP 1997*, Volume: 2, pp. 867-870, Apr 1997.

[27] Jusak J., "Blind Channel Equalisation for Mobile Communications." Dissertation for PhD (Communication Engineering), RMIT University, Aug 2005.

[28] Johnson C.R., Schniter P., Endres T. J., Behm J. D., Brown D. R., Casas R. A., "Blind Equalization Using the Constant Modulus Criterion: A Review," in *Proceedings of the IEEE*, Volume 86, No. 10, pp: 1927-1950 Oct 1998.

[29] Haykin S. Adaptive Filter Theory, 3rd Edition, Prentice Hall, New Jersey, 1996.

[30] Neville K. Jusak J. Hussain Z. M. Lech M. "Performance of a Text-Independent Remote Speaker Recognition Algorithm over Communication Channels with Blind Equalisation." in *Proceedings of IEEE TENCON Conference*, Nov 2005.

[31] *EMO-DB*, Retrieved (June 3rd, 2005), from http://pascal.kgw.tu-berlin.de/emodb/index-1024.html.

Appendix A – Source Code

A.1 Training

A.1.1 Preprocessing

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This code: loads wav files, pre-emphasises, removes silence from and saves
%  these as 1 second long files ready for the feature extraction phase
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

A=10; %Number of speakers

for i=1:A
   clear silence result
   string1 = 'speaker';
   string2=num2str(i);
   string3 = '.wav';
   string4=[string1 string2 string3];
   [x,fs,bits]=wavread(string4); %Input speech


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   %Filtering Signal with pre-emphasis filter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   v=preemph(x);

   N=length(x);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   %Speech/Silence Detection
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   u=abs(v);
   maxlevel=max(u); %Maximum speech input into system

      Bs=0.9992; %Decay time constant for speech metric
      Bn=0.99722; %Decay time constant for noise metric
      Bt=0.999975; %Decay time constant for silence metric

   %Speech Metric
   s(1)=u(1);

   for k=2:N
      if u(k) > s(k-1)
         s(k)=u(k);
      else
         s(k)=((1-Bs)*u(k))+(Bs*s(k-1));
```

```
    end
end

%Noise Metric
n(1)=u(1);

for k=2:N
    if u(k) > n(k-1)
        n(k)=u(k);
    else
        n(k)=((1-Bn)*u(k))+(Bn*n(k-1));
    end
end

%Silence Metric

tn(1)=u(1);

for k=2:N
    if tn(k-1) < n(k)
        tn(k)= ((1-Bt)*n(k))+(Bt*tn(k-1));
    else
        tn(k) = n(k);
    end
end

    Ths= 4; %Speech Threshold
    Thn= 2.828; %Noise Threshold
    Tmin= 0.001;

%Speech/silence decision
for k=1:N
    if s(k) > Ths*tn(k)+Tmin
        result(k)=1;
    end
    if s(k) < Thn*tn(k)+Tmin | tn(k)==0
        result(k)=0;
    end
    if Thn*tn(k)+Tmin <= s(k) <= Ths*tn(k)+Tmin
        result(k)= 0;
    end
end

silence=find(result==0);

%v(silence)=[]; %Removes silence from pre-emphasised speech
N=length(v);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   %Splitting the speech into 1 second long files
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

   for k=fs:fs:N
      y=v((k-(fs-1)):k);
      string5=num2str((k/fs),'%02d');
      string6='preprocessed';
      str=[string6 string2 string5 string3];
      wavwrite(y,fs,str)
   end
end
```

A.1.2 Pre-Emphasis Filter Algorithm

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This function pre-emphasises speech with high-pass filter:  v(k)=x(k)-0.95x(k-1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function y = preemph(x)

N=length(x);
a=15/16;
y(1)=x(1);
for k=2:N
   y(k)=x(k)-a*x(k-1);
end
```

A.1.3 Feature Extraction

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This code takes in 1 second wav files produced in preprocess.m, extracts the Mel-
Frequency Cepstral Coefficients and saves these as dat files to be used in classifier
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear
A=20; %Number of speakers
a=20; %Number of seconds per speaker
Coef=20; %Number of Cepstral Coefficients

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Signal input
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:A

   string1 = 'd:\katrina\research\Training\textindepend\speaker'; %this creates
constant part of filename
   string2 = num2str(i); %this converts number i into a string
```

```
   for k=1:a
      string3 = num2str(k,'%02d');
      string4 = '.wav'; %this adds the file extension to the filename
      string5=[string1 string2 string3 string4]; %this concatenates the four strings into
one string
      [x,fs,bits]=wavread(string5); %Input speech
      fs=11000;
      N=length(x);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %Extracting Mel-Cepstral Coefficients
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      signal_duration=N;
      window_length=0.02*fs; %length of window (should be 20ms)
      window_overlap=0.01*fs; %overlap of frames (should be 10ms)
      ceps=mfcc_1(x,window_overlap,window_length,Coef,12000);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      %Saving Mel-Frequency Coefficients
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

      string6='.dat';
      string7= [string1 string2 string3 string6];
      fid = fopen(string7,'w');

      for n=1:Coef
         fprintf(fid,'%4.6f ',ceps(n,:));
         fprintf(fid,'\n');
      end
   end
   fclose('all');
end
```

A.1.4 Mel-Frequency Cepstral Coefficient Algorithm

```
% mfcc.m
% Calculates cepstral coefficients for sequence y, using window length N,
% window step size M (for overlap between blocks), and order P (= number of cep
% coeff's wanted).

function ccep=mfcc_1(y,M,N,P,fs);
Nt=length(y); % total speech length
N2=N/2;
F=fs/N; % frequency step
f=F*(-N2:N2-1); % frequency vector for one block
H=zeros(20,N);Le=zeros(1,20);coef=zeros(P,ceil(Nt/N));ccep=zeros(1,P);
%............................................
% % Start & end of trianglar filters
% Formula used is: Mel(f)=2595*log10(1+f/700) with Range of 4000 Hz.
```

```
fo = [0 69 146 231 324 426 539 663 799 949 1113 1295 1494 1713 1954 2219 2511
2832 3185 3573];
fe = [146 231 324 426 539 663 799 949 1113 1295 1494 1713 1954 2219 2511 2832
3185 3573 4000 4469];
%.....................................
fc=fo+(fe-fo)/2; % centers of filters
B=fe-fo; % band-widths
for k=1:20
   Box1=stepfun(f,fo(k))-stepfun(f,fe(k));
   Box2=stepfun(f,-fe(k))-stepfun(f,-fo(k));
   H(k,:)=abs(1-abs(f-fc(k))/(B(k)/2)).*Box1+abs(1-abs(f+fc(k))/(B(k)/2)).*Box2; %
k-th +ve/-ve triangle
end
% for k=1:20
%     H(k,:)=abs(1-(f-fc(k))/(2*B(k))).*(stepfun(f,fo(k))-stepfun(f,fe(k)))+abs(1-
(f+fc(k))/(2*B(k))).*(stepfun(f,-fe(k))-stepfun(f,fo(k))); % k-th +ve/-ve triangle
% end
%.....................................


ns=1; %start point
ne=N; %end point
m=1;
while ne <= Nt
   ym=y(ns:ne); % m-th block
   yw=hamming(N).*ym; % windowed m-th block
   Yw=abs(fftshift(fft(yw)))/fs;

for j=1:20 % mel filters outputs
Yf(j,:)=H(j,:).*Yw.';
Ef=sum(Yf(j,:).^2); % Energy o/p of j-th filter
Le(j)=log(Ef); % log-energy output of the j-th filter
end

V=[1:20];
for i=1:P  % P is the number of coeff required
 coef(i,m)=sum(Le.*cos(i*(V-.5)*pi/20));
end

m=m+1;
ns=1+(m-1)*M; % new start
ne=ns+N-1; % new end
end; % go back for a new frame (block)

ccep= coef;
```

A.1.5 Averaging the Features

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This code takes in Mel-frequency cepstral coefficients from each speaker and
calculates the average of the feature vectors over multiple analysis frames.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear, clc

A=10; %Number of Speakers
a=10; %Number of seconds per speaker
Coef=20; %Number of Cepstral Coefficients

for j=1:A

   for i=1:a


      string1='d:\katrina\research\training\textindepend\';
      string2='speaker';
      string3=num2str(j);
      string4=num2str(i,'%02d');
      string5='.dat';
      string6=[string1 string2 string3 string4 string5];
      load (string6)

   end

if j==1
    speaker = [speaker101 speaker102 speaker103 speaker104 speaker105
speaker106 speaker107 speaker108 speaker109 speaker110];

 elseif j==2
    speaker = [speaker201 speaker202 speaker203 speaker204 speaker205
speaker206 speaker207 speaker208 speaker209 speaker210];

 elseif j==3
    speaker = [speaker301 speaker302 speaker303 speaker304 speaker305
speaker306 speaker307 speaker308 speaker309 speaker310];

 elseif j==4
    speaker = [speaker401 speaker402 speaker403 speaker404 speaker405
speaker406 speaker407 speaker408 speaker409 speaker410];

 elseif j==5
    speaker = [speaker501 speaker502 speaker503 speaker504 speaker505
speaker506 speaker507 speaker508 speaker509 speaker510];

 elseif j==6
```

```
    speaker = [speaker601 speaker602 speaker603 speaker604 speaker605
speaker606 speaker607 speaker608 speaker609 speaker610];

 elseif j==7
    speaker = [speaker701 speaker702 speaker703 speaker704 speaker705
speaker706 speaker707 speaker708 speaker709 speaker710];

 elseif j==8
    speaker = [speaker801 speaker802 speaker803 speaker804 speaker805
speaker806 speaker807 speaker808 speaker809 speaker810];

 elseif j==9
    speaker = [speaker901 speaker902 speaker903 speaker904 speaker905
speaker906 speaker907 speaker908 speaker909 speaker910];

 elseif j==10
    speaker = [speaker1001 speaker1002 speaker1003 speaker1004 speaker1005
speaker1006 speaker1007 speaker1008 speaker1009 speaker1010];


for i=1:Coef
   avg(:,i)=mean(speaker(i,:));
end

       string7='mean';
       string8= [string1 string7 string3 string5];
       fid = fopen(string8,'w');
for n=1:Coef
   fprintf(fid,'%4.6f ',avg(:,n));
end
fclose('all');
end
```

A.2 Testing

A.2.1 Minimum Distance Classifier Algorithm

```
clear, clc
A=10; %Number of trained speakers
Coef=20; %Number of Cepstral Coefficients

string1='d:\katrina\research\testing\textindepend\AvFeatVec\';

for i=1:A
   string2='mean';
   string3=num2str(i);
   string4='.dat';
   string5=[string1 string2 string3 string4];
   load(string5)
end

results(1,20)=zeros;

for i=1:A
   string6= 'testmean';
   string7 = num2str(i);
   string8= [string1 string6 string7 string4];
   meantestsamp = load(string8);

C2=(1/(Coef))*sum((mean2(1:Coef)-meantestsamp(1:Coef)).^2);
C3=(1/(Coef))*sum((mean3(1:Coef)-meantestsamp(1:Coef)).^2);
C4=(1/(Coef))*sum((mean4(1:Coef)-meantestsamp(1:Coef)).^2);
C5=(1/(Coef))*sum((mean5(1:Coef)-meantestsamp(1:Coef)).^2);
C6=(1/(Coef))*sum((mean6(1:Coef)-meantestsamp(1:Coef)).^2);
C7=(1/(Coef))*sum((mean7(1:Coef)-meantestsamp(1:Coef)).^2);
C8=(1/(Coef))*sum((mean8(1:Coef)-meantestsamp(1:Coef)).^2);
C9=(1/(Coef))*sum((mean9(1:Coef)-meantestsamp(1:Coef)).^2);
C10=(1/(Coef))*sum((mean10(1:Coef)-meantestsamp(1:Coef)).^2);


C=[C1 C2 C3 C4 C5 C6 C7 C8 C9 C10];
[g,Speaker]=min(C);
results(1,i)=i;
results(2,i)=Speaker;
end
results
```

Appendix B – Test Results

Results with no Equalisation/Compensation

			LPFs					Noise			
Speaker #	Gender	Clean	3kHz	4 kHz	5 kHz	6 kHz	7 kHz	10 dB	20 dB	30 dB	40 dB
1	M	100%	0%	0%	40%	80%	80%	0%	40%	80%	80%
2	F	100%	0%	0%	100%	100%	100%	100%	100%	100%	100%
3	F	100%	0%	0%	100%	100%	80%	20%	20%	80%	80%
4	M	0%	0%	0%	60%	20%	0%	0%	60%	0%	0%
5	M	100%	0%	0%	60%	100%	100%	20%	20%	100%	100%
6	M	60%	0%	0%	60%	80%	80%	0%	20%	40%	60%
7	F	60%	0%	0%	60%	60%	60%	0%	0%	20%	60%
8	F	80%	0%	0%	0%	80%	80%	20%	40%	60%	80%
9	M	60%	0%	0%	80%	80%	80%	60%	80%	80%	80%
10	F	20%	0%	0%	20%	20%	40%	40%	20%	20%	20%

Channel Compensation with Cepstral Mean Subtraction

		LPFs					Noise			
Speaker #	Gender	3kHz	4 kHz	5 kHz	6 kHz	7 kHz	10 dB	20 dB	30 dB	40 dB
1	M	60%	20%	0%	40%	80%	0%	20%	80%	80%
2	F	0%	0%	0%	0%	0%	0%	0%	0%	0%
3	F	0%	80%	100%	100%	100%	0%	0%	40%	60%
4	M	0%	20%	40%	40%	20%	40%	60%	20%	40%
5	M	0%	100%	100%	100%	100%	0%	0%	20%	60%
6	M	100%	80%	60%	60%	80%	0%	0%	40%	40%
7	F	20%	20%	80%	60%	60%	0%	0%	60%	80%
8	F	0%	60%	80%	40%	40%	0%	0%	40%	20%
9	M	40%	0%	0%	0%	0%	0%	20%	0%	0%
10	F	0%	0%	0%	0%	0%	0%	0%	0%	0%

Channel Equalisation with CMA

		Noise			
Speaker #	Gender	10 dB	20 dB	30 dB	40 dB
1	M	0%	80%	80%	80%
2	F	20%	80%	80%	100%
3	F	0%	100%	80%	80%
4	M	0%	0%	0%	0%
5	M	0%	40%	60%	80%
6	M	0%	60%	80%	80%
7	F	0%	20%	20%	20%
8	F	0%	40%	20%	60%
9	M	60%	100%	80%	60%
10	F	0%	0%	20%	0%

Channel Compensation with RASTA Processing

		LPFs					Noise			
Speaker #	Gender	3kHz	4 kHz	5 kHz	6 kHz	7 kHz	10 dB	20 dB	30 dB	40 dB
1	M	0%	60%	60%	60%	80%	0%	60%	60%	80%
2	F	0%	80%	80%	80%	80%	100%	80%	100%	80%
3	F	0%	80%	100%	100%	100%	20%	60%	80%	100%
4	M	0%	0%	0%	0%	0%	0%	0%	0%	0%
5	M	0%	80%	100%	100%	100%	20%	40%	80%	100%
6	M	100%	80%	80%	80%	80%	60%	60%	80%	80%
7	F	0%	20%	40%	40%	20%	0%	60%	40%	0%
8	F	0%	40%	40%	40%	40%	0%	20%	20%	20%
9	M	100%	80%	40%	40%	60%	80%	80%	60%	60%
10	F	60%	40%	40%	20%	40%	20%	20%	20%	40%

Appendix C – Formulas and Tables

C.1 Mel-Scale Filters

Range (Hz)	4000			
Mel Range	2146.06			
Mel increments	107.30			
	Mel-scale		**Linear scale**	
Filter	Start	Finish	Start	Finish
1	0.00	214.61	0.00	146.83
2	107.30	321.91	69.92	231.43
3	214.61	429.21	146.83	324.47
4	321.91	536.52	231.43	426.80
5	429.21	643.82	324.47	539.36
6	536.52	751.12	426.80	663.16
7	643.82	858.43	539.36	799.33
8	751.12	965.73	663.16	949.10
9	858.43	1073.03	799.33	1113.84
10	965.73	1180.34	949.10	1295.02
11	1073.03	1287.64	1113.84	1494.31
12	1180.34	1394.94	1295.02	1713.50
13	1287.64	1502.25	1494.31	1954.59
14	1394.94	1609.55	1713.50	2219.77
15	1502.25	1716.85	1954.59	2511.43
16	1609.55	1824.15	2219.77	2832.22
17	1716.85	1931.46	2511.43	3185.06
18	1824.15	2038.76	2832.22	3573.15
19	1931.46	2146.06	3185.06	4000.00
20	2038.76	2253.37	3573.15	4469.49

$$Mel(f) = 2595 \log_{10}(1 + \frac{f}{700})$$