Speaker Recognition in Reverberant Environments

by

Joseph Gammal

Proposed thesis

Faculty of Graduate Studies and Research

Master of Applied Science (M.A.Sc.)

Ottawa-Carleton Institute For Electrical and Computer Engineering

Department of Systems and Computer Engineering

Faculty of Engineering and Design

Carleton University

Ottawa, Ontario

November, 2004

Your file Votre référence
ISBN: 0-494-00739-7
Our file Notre référence
ISBN: 0-494-00739-7

Abstract

This thesis compares several speaker recognition algorithms in reverberant environments. The Gaussian Mixture Model, Auto-regressive vector model, covariance based models and Multi layer perceptron are compared.

The methods are compared when there is a mismatch between the training and test speech due to the non-reverberant nature of the training speech and the reverberant nature of the test speech.

In order to counteract the effects of reverberation, training was performed using reverberant speech. Average recognition accuracy improved by 9.8% for the GMM, 53% for AR-Itakura, 18.8% for sphericity measure, 18.1% for the divergence shape measure and 15.9% for AR-AGS.

A method was proposed to create a set of reverberant models for each speaker using speech reverberated to different degrees. A novel technique was proposed to determine which reverberant model for each speaker best matches the reverberant test speech. 98.7% classification accuracy was obtained using an Auto-regressive vector model adapted for this purpose.

Acknowledgements

First and foremost I would like to thank God for making my studies possible.

I would like to acknowledge Professor Rafik Goubran for his help, direction, unwavering support and commitment throughout the work of my thesis. His efforts are greatly appreciated.

I would like to acknowledge Darren Russ, Danny Lemay, Chris Welch and Jacques Chauvin for their technical support in network and computer system issues. In addition I would like to acknowledge James Gordy for sharing his expertise in speech processing. I would like to acknowledge Ahmad Rami Abu-El-Quran for his help in formatting this document.

I would like to acknowledge my parents, family and friends for remaining at my side throughout my graduate school experience.

I would like to acknowledge the financial assistance from CITO, Mitel, and the Ontario Government.

I would like to thank my undergraduate and graduate teachers at Carleton and Ottawa University. A special thanks to Mohammed El-Tanany for piquing my interest in DSP and Richard Danserau for his help in the DSP graduate course that went beyond the call of duty.

Finally I would like to offer my best wishes to all the graduate students of the Carleton DSP lab especially Yasser Mahgoub may God grant him success.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ARVM	Auto-Regressive Vector Model
AGS	Arithmetic Geometric Sphericity Measure
CMS	Cepstral Mean Subtraction
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DS	Divergence Shape
EBF	Elliptical Basis Functions
EM	Estimation Maximization
FFT	Fast Fourier Transform
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
IDFT	Inverse Discrete Fourier Transform
LBG	Linde, Buzo, Gray
LPC	Linear Predictive Coding
LPCC	Linear Predictive Cepstral Coefficients
MFCC	Mel Frequency Cepstral Coefficients
MLP	Multiple Layer Perceptron
MLS	Maximum Length Sequence
RT60	Reverberation Time 60 seconds

RBF	Radial Basis Functions
SM	Sphericity Measure
SNR	Signal to Noise Ratio
VAD	Voice Activity Detector
VQ	Vector Quantization

Chapter One
Introduction

1.1 Speaker Recognition

Speaker recognition is the problem of selecting from a set of known speakers, the identity of the speaker who uttered a speech segment. The problem is classified as closed set speaker recognition if the set of known speakers is limited and the speech is uttered by one of the speakers in the set. The problem is classified as open set speaker recognition if the possibility exists that the speech segment was produced by a speaker not included in the set of known speakers [1], [2]. Another problem that is similar to open set speaker recognition is speaker verification. In this case the objective is to verify the identity of a speaker. The speech that is offered to the system may belong to a registered speaker or it may be that of an imposter. The speaker is making an identity claim and the speaker may be a registered speaker and the identity that the speaker claims may be the true identity of the speaker. In this case the verification system should accept the speaker. The scenario also exists where

the claimant may not be the same person as is being claimed, the speaker would therefore be an imposter and it is the responsibility of the verification system to determine that the claimant is an imposter. The system will make a decision whether to accept that the speech belongs to the unknown speaker or to reject this claim. Another possible use of speaker verification lies in the forensic field where the system must determine whether or not a suspect's speech captured over a telephone does or does not belong to the suspect. Closed set speaker recognition may also be useful in forensic applications to determine the most likely candidate to have uttered a segment of speech if the identify of the speaker is unknown but is likely to belong to one of a limited number of suspects. This thesis addresses the problem of closed set speaker recognition and speaker verification.

The most commonly used research methodology in the field of speaker recognition is to collect a large set of speech utterances from various speakers. This large set is divided into two subsets. The first one is used to train the speaker recognition system and the second is used to test it.

Closed set speaker recognition and speaker verification can each be divided into two different applications or problem areas. These areas are text-dependent and text-independent closed set speaker recognition and speaker verification. In the text-dependent case, the training speech and test speech consist of the same phrases. The words used and the order in which they are stated are the same. In the text-independent case the phrases used for training and testing are different. Text-

dependent closed set speaker recognition or speaker verification can be conducted with higher accuracy than the text independent case. Also the amount of test speech required to conduct text dependent speaker recognition is less than that required for the text-independent case. This thesis will be in the area of text-independent speaker recognition.

1.2 Problem Statement and Thesis Objectives

The performance of most speaker recognition systems is severely degraded when the training speech and the testing speech are recorded in different acoustic environments. For example when the training speech is collected using a handset and the testing speech is collected in a reverberant hands-free environment. Very little research has been devoted to the effect of mismatch between training and test speech resulting from the existence of reverberation. This thesis compares the degradation in several speaker recognition algorithms resulting from a mismatch in training and testing due to reverberation. The degradation in recognition for a number of commonly used algorithms and feature vector types is determined and compared. The effect of training with reverberant speech from rooms that are different from that used for the testing is investigated as a possible method to counteract the mismatch. A method is proposed to determine from the reverberated speech, which reverberated training speech should be used. It is effectively a reverberation classifier.

1.3 Thesis Contributions

This thesis contributes to speaker recognition research in the area of mismatch due to reverberation. The specific contributions are as follows:

1) The performance of a number of speaker recognition algorithms was compared on clean non-reverberant speech. The impact of the various parameters on each algorithm was investigated.
2) The degradation of the performance of each method and feature vector combination using reverberated test speech originating from different rooms was compared. A publication highlighting these results was published [71].
3) The effect of training with reverberated speech uttered in different rooms than the training speech was performed in order to counteract the degradation. A paper demonstrating this method of combating reverberation was submitted to the International Measurement Technology Conference 2005 for publication [72].
4) A new method was devised to classify reverberant speech and determine which room it originated from.

1.4 Thesis Outline

The thesis is organized into nine chapters. Chapter 1 presents a general introduction to the speaker recognition. It contains a statement of the objectives of this thesis. A summary of the thesis contributions is also included.

Chapter 2 contains an overview of speaker recognition algorithms. A detailed description is given of how the speech is processed in order to perform recognition. It describes how the speech is transformed from a raw speech file such as a wave file into short segments or frames. It gives details of how features are extracted from these segments or frames and presented to the speaker recognition algorithms. A description is given of each speaker recognition algorithm used.

Chapter 3 presents the experimental setup of the speaker recognition systems that were implemented. Included in this chapter is a description of the model used to simulate reverberation. A description is given of the speaker recognition database that was used. The setup of an experiment conducted in a real reverberant room is also described. The method used in order to conduct closed set speaker recognition and verification trials is described.

Chapter 4 contains a comparison of the speaker recognition algorithms used. The algorithms are compared based on their performance in closed set speaker recognition and speaker verification trials.

Chapter 5 deals with the effect of reverberation on speaker recognition algorithms. The effect on recognition performance as reverberation is introduced into the test speech is quantified for the different algorithms. The performance degradation for each algorithm is demonstrated.

Chapter 6 contains the results of speaker recognition experiments conducted in a real reverberant room. The results here are compared to the results obtained when recognition is performed on simulated room impulse responses.

Chapter 7 presents a method of counteracting the degradation in recognition performance caused by reverberation. As was already stated chapter 5 presents recognition trials where the test speech is reverberated and the training speech is not. Chapter 7 contains the results of experiments where the training speech is reverberated in order to counteract the reverberation in the test speech. Specifically the premise of training with reverberated speech originating from different rooms than the test speech is investigated. This process of training with reverberated speech will be referred to as cross training.

Chapter 8 introduces a novel algorithm that determines the source of reverberation in a segment of reverberated speech. Here closed set speaker recognition algorithms are trained using reverberated speech. These algorithms then classify reverberated test speech and determine from which room they are most likely to have originated from. The performance of different speaker recognition algorithms and speech features used for this same purpose are compared.

Chapter 9 concludes the thesis and discusses possible future research directions.

Chapter Two
Overview of Speaker Recognition

2.1 Training and Testing Overview

The processing in closed set speaker recognition or speaker verification is usually divided into two stages: the first stage is training the second is testing. Training is the process of creating models that capture the attributes of different speaker's speech. These models will differentiate different speakers from each other. A simplified illustration of this process is shown in figure 2.1.

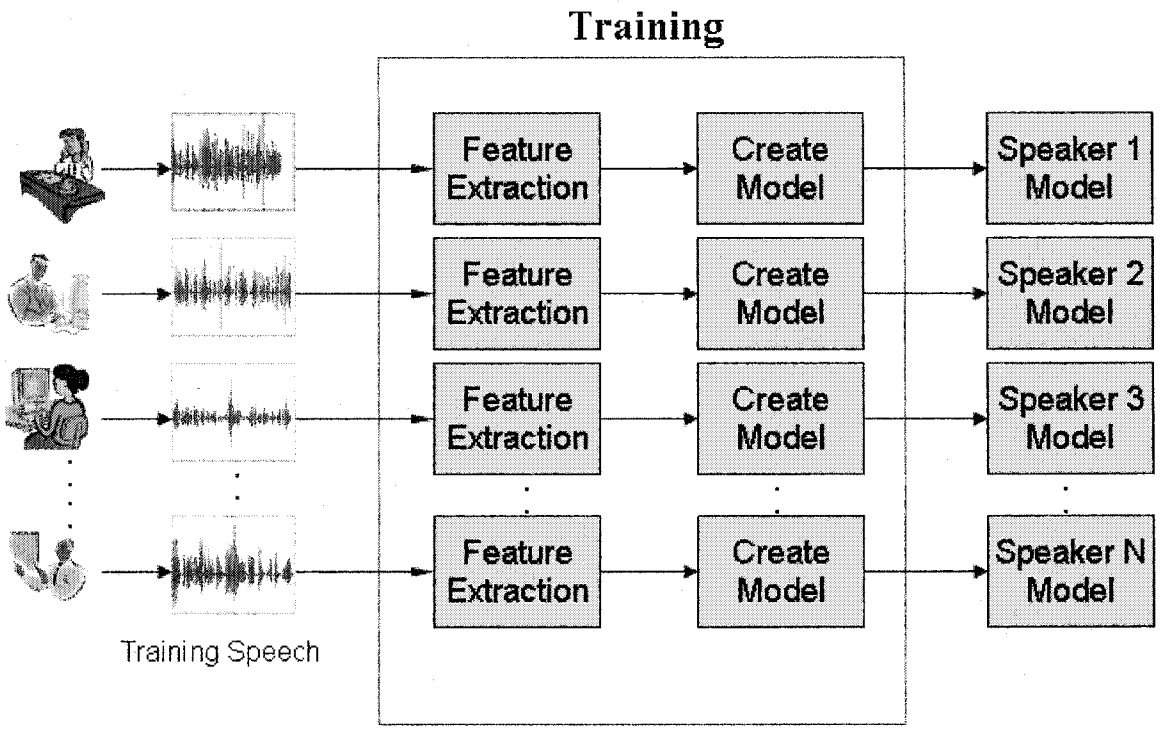

Figure 2.1: Training Speaker Models

For closed set speaker recognition, testing is the process of evaluating an utterance of speech from an unknown speaker and determining which speaker model is nearest to the utterance. This requires the generation of a probability, or distance for the unknown speech relative to each speaker model. The nearest speaker model usually has the largest probability or shortest distance from the unknown speech. This process is illustrated in figure 2.2.

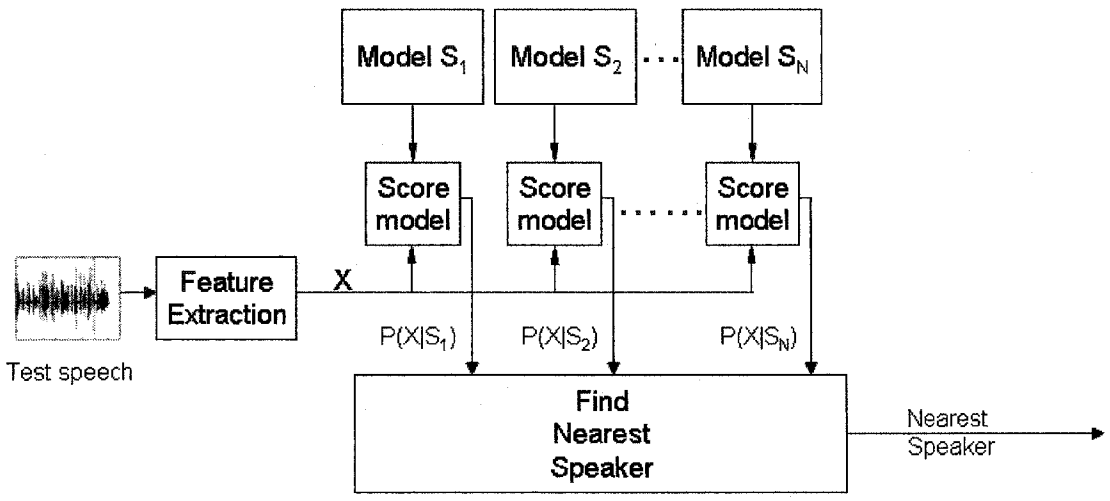

Figure 2.2: Speaker Recognition Testing

Speaker verification is similar to closed set speaker recognition in that the same training process can be used for both. The difference between closed set speaker recognition and speaker verification is in the way testing is performed. As was stated earlier a test utterance is compared to each speaker model, a likelihood that the utterance was generated by each model is generated. This is closed set recognition. For verification the issue is not which model the speech is closest to. Rather the issue becomes whether or not the speech is close enough to a speaker model that the system can conclude that the speech belongs to that speaker.

In a real world system where a speaker must be verified, the speaker who must be verified will be referred to as the test speaker as this speaker utters the test speech. The test speaker claims to be one of the speakers known to the system. If the test speaker is not an imposter, meaning that the identity claim is true then the system will possess a model for that speaker. The system will have been trained to recognize that

speaker's voice. Whether the test speaker is an imposter or not the system will retrieve the model associated with the claimed identity. The test speech will be compared to this speaker model and the distance from the model to the test speech will be generated. This distance will be compared to a threshold, if the distance is less than the threshold the speaker will be accepted. Some systems do not use a distance to the speaker model but rather use a probability that the model generated the test speech. For these systems a probability is compared to a threshold instead of a distance being compared to a threshold. If the probability is greater than the threshold the speaker will be accepted, otherwise the speaker will be rejected.

Additional actions may be involved in this process depending on the system used. This additional processing is known as normalization and will be discussed in chapter 3.

2.2 Front End Processing

The goal of the front end of speaker recognition systems is to transform raw speech such as that contained in wave files to a different form before the recognition algorithms can process it. This new form will be referred to as feature vectors. Feature vectors contain speaker specific characteristics. The feature vectors contain information that can be used to discriminate between speakers. There are a number of reasons for using feature vectors instead of the original speech. Consider a segment of speech that is 30 seconds to 2 minutes long. These are lengths that are

commonly used for testing and training respectively. If the speech is sampled at 8Khz then the number of resulting samples will be between 240,000 for the 30-second segment and 960,000 samples for the 2-minute segment. This is a relatively large amount of data that would require a large amount of time to process by the recognition algorithms. Unprocessed raw speech in this form does not emphasize the speaker specific information contained in speech. The speaker specific information contained in speech that is not emphasized in raw speech is the spectrum of the speech. The spectrum of the speech contains information about the shape of the vocal tract of the speaker; this varies from one speaker to another [57].

The spectrum of the speech is therefore the desired characteristic of the speech that should be emphasized in the feature vectors that are extracted from the speech. The spectrum of the speech varies as the speaker speaks. It is necessary to capture snap-shots of the spectrum as it changes. The raw speech must therefore be segmented into pieces known as frames before the spectrum of these frames can be computed. A 20ms frame size is adequate for this purpose [57].

The spectrum of each frame, unless further processing is performed, contains as many samples as the original frame if not more. This information must then be compressed to counteract the problem that there is too much data to process in a digitized wave file. The information in the spectrum can then be compressed using two approaches. The first is to characterize the spectrum of each frame by computing the coefficients of a filter that has the same overall spectrum of the frame. The

second is to use what is known as a filter-bank. A filter-bank groups adjacent parts of the spectrum together into bands. The spectral energy in each of these bands is then summed and represented as a single value. These values, one from each band, characterize the spectrum. The number of filters in the filter bank and therefore the resulting number of outputs from the filter bank is in the order of 19 to 24 filters. This is much less than the original number of samples in the spectrum of the frame. The number of samples in a 20ms frame is 160 samples, so compression of the spectrum results.

Returning to the first approach of compressing the information in the spectrum, one approach of accomplishing this is to use what is known as an LPC (linear prediction coefficients) model. This generates the coefficients of a filter whose frequency domain representation is similar to the spectrum of the speech in the frame. The number of LPC coefficients ranges between 10 and 20. This method of estimating the spectrum also results in a compression of the data that represents the original spectrum of the frame.

For a single speech frame, the outputs of the filter bank or the LPC coefficients are a vector quantity. These will be referred to as filter bank vectors and LPC vectors respectively.

An advantage of using filter banks or spectra derived from LPC analysis is not limited to the fact that there is a reduction in the dimensions of the data. Filter bank

and LPC derived spectra smooth the spectrum and emphasize the spectrum's envelope.

The elements in the filter bank or LPC vectors are correlated with the other elements in the respective vector [27], [23]. This has implications when measuring the distance between two filter bank vectors or two LPC vectors. The distance measures required are more complicated than if the vector elements were not correlated with each other.

In order to de-correlate the elements of the filter bank and LPC vectors a transformation is performed on them. This transformation converts the vectors to what are known as cepstral vectors. The filter bank vectors are transformed into what are known as Mel-cepstral (MFCC) vectors and the LPC vectors are transformed into what are known as LPC cepstral vectors (LPCC). These are known as cepstral representations and they are commonly used for speaker recognition [10], [11], [12], [13], and [14].

The details of how the transformations takes place are included in the up coming sections.

2.2.1 Feature Extraction Process

The purpose of the front end processing is to transform the raw speech contained in wave files into cepstral vectors. This is the desired result, but in order to perform this process properly some other processing must take place first.

The speech must be filtered with a pre-emphasis filter. Pre-emphasis compensates for the fact that as the frequency increases in speech the spectrum tends to decay. The pre-emphasis filter flattens the speech spectrum [15]. This is a common practice in speaker recognition. The output of the pre-emphasis filter is sub-dived into frames. Frames from silence periods must be discarded so the silences between utterances and the silence periods at the start and end of speech files are removed. A hamming window must be applied to each speech frame as this counteracts what is known as edge effects [12]. Edge effects are phenomena that occur because the speech is segmented into blocks. Without the application of a hamming window, the spectrum of the speech frames would be distorted. Figure 2.3 contains an illustration of a hamming window. Not applying a hamming window is equivalent to using a rectangular window to window the speech. Each has its advantages and disadvantages. The spectrum of rectangular window has a narrower main lobe than the hamming window, this offers sharper resolution of the spectrum of the windowed frames. On the other hand the spectrum of the hamming window outside of the main lobe is attenuated by at least 20db more than that of the rectangular window.

Figure 2.3 illustrates the pre-processing of the speech. This illustration shows two parallel lines of processing. The upper line contains a voice activity detector that labels the frames as being part of a silence interval or not. The lower line of

processing illustrates the pre-emphasis, segmentation and the application of a hamming window.

These steps are performed when either LPCC or MFCC vectors are extracted. They are therefore common pre-processing steps that are common to both these parameterizations.

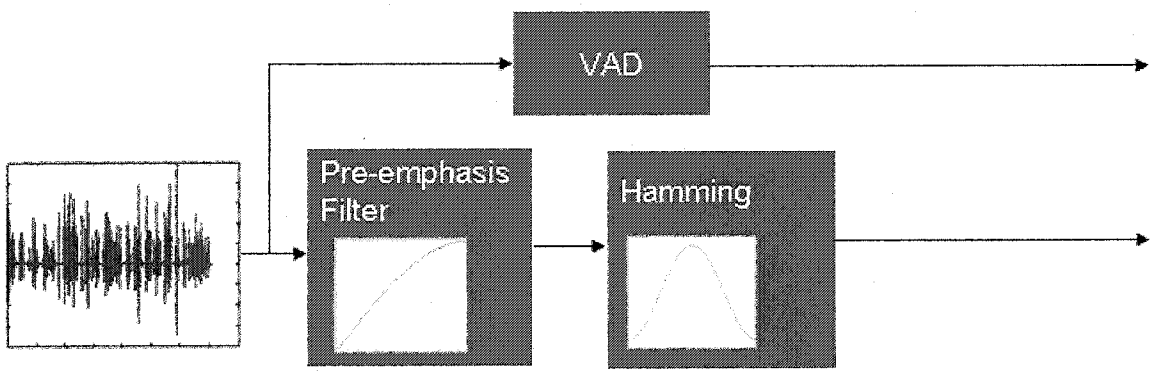

Figure 2.3: Common pre-processing [17]

The remainder of this section will be devoted to explaining the specific details of how the VAD is implemented and how the hamming window and pre-emphasis filter are applied.

Let $x(n), n = 1..N$ represent the input samples from a single speech file. These samples are part of a raw speech file specifically a wave file. A VAD partitions the speech signal into P frames of length between 20 and 32ms, with frame overlap between 25% and 50%, and labels each frame as being either on or off. P of course is dependent on the length of the input segment, the frame length used and the amount of overlap.

If an energy based VAD is used, an energy threshold is required. This threshold is compared to the energy of each frame to determine if the frame is silent or not. The energy threshold $E_{\min}$ is set to some fraction of the average, E_{avg}, of all the VAD frame energies. The i^{th} frame is labeled as on if $E_i > E_{\min}$ and off otherwise. The complete speech file is processed in this fashion first. The result is that each frame has a label indicating whether or not the frame is silent or not.

At this point two sets of data exist. The first set of data is the samples from the speech file. The second set of data is a set of VAD labels one for each frame from the speech file.

The next step is to apply the pre-emphasis filter to the samples from the file. The pre-emphasis filter used is also known as a differentiator and has the following system equation $y(n) = x(n) - \alpha x(n-1)$ where α is between 0.95 and 0.98 [16]. The magnitude response of a typical pre-emphasis filter is shown in figure 2.3.

The filtered samples are fragmented into frames with overlap. A hamming window is applied to each frame. One reason that the frames are overlapped is to preserve all the information in the speech. The hamming window has a peak at the center and decays towards the start and end of the frame. This means that towards the start of the frame and towards the end of the frame the information is de-emphasized. By overlapping the windows, information that is de-emphasized in one frame is emphasized in the next frame and visa-versa. The result is that all the information in the speech contributes equally to the final result.

The remainder of this section explains how the energy threshold and frame energy are computed. A frame has $K = F_s * Lms$ samples where F_s is the sampling rate and L is the frame length. The frame energies $E_i, i = 1..P$ for each frame are computed as follows:

$$E_i = \sum_{k=1}^{K} x((i-1) * \frac{K}{2} + k)^2 \tag{2.3}$$

The average frame energy is computed as follows:

$$E_{avg} = \frac{1}{P} \sum_{n=1}^{N} x(n)^2 \tag{2.4}$$

2.2.2 Mel-Warped Cepstral Coefficients

Figure 2.4 illustrates the processing that must be performed on the speech in order to produce Mel-cepstral (MFCC) vectors. It is based on the front-end system described in [19]. The first steps are identical to those shown in figure 2.3. The remainder of the steps will be explained in the following section.

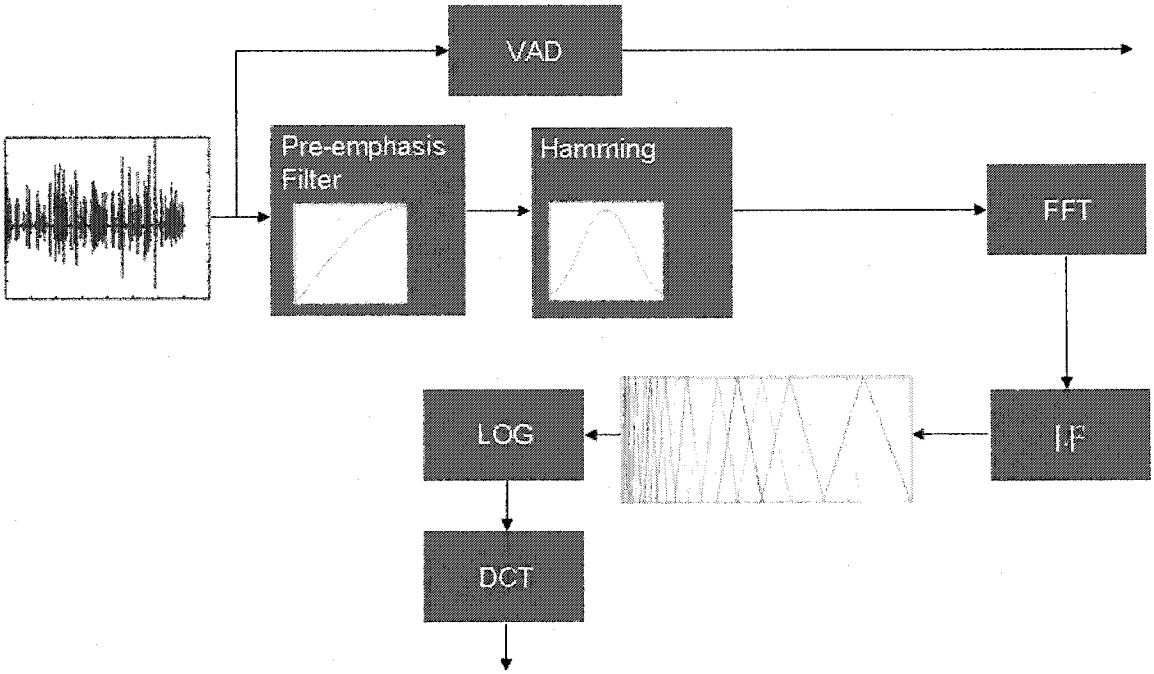

Figure 2.4: Mel Cepstrum processing

A Fast Fourier transform (FFT) is applied to each speech frame after the hamming window is applied, see figure 2.4. The power of each of the FFT coefficients is computed. This is performed by taking the magnitude squared of each coefficient. A Mel-frequency based filter bank is then applied. The Mel frequency bank is used to filter these squared coefficients. The Mel-frequency based filter bank is an approximation of the human auditory system. The details of the implementation of the filter bank are provided later in this section. The bark scale can also be used in order to warp the frequency scale to achieve an approximation of the human auditory filters. The bark scale gives specific locations for the positions of the center frequencies and widths of the filter bank filters. The mel scale allows for the

determination of the relative widths and spacing of the filters where the number of filters can be varied across a certain frequency band.

As can be seen in figure 2.4 the Mel-frequency bank is composed of a set of overlapping triangular filters. Each filter has an amplitude at a given frequency. Each coefficient that is outputted from the FFT also has an amplitude and an associated frequency. It follows that the power of a coefficient outputted from the FFT has the same frequency as the original coefficient. In order to filter the power spectrum of each frame with the triangular filters, each coefficient must be filtered by the triangular filter in whose band the coefficient falls. Since the triangular filters overlap, most frequency components fall within the frequency band of two filters. At each frequency within the band of a triangular filter, the filter has a corresponding amplitude. The power of the FFT coefficients are simply multiplied by the corresponding amplitudes (the amplitudes in the filters that correspond to their frequency) of the triangle filters within whose band they lie. This results in a set of scaled power spectrum coefficients. All the scaled power spectrum coefficients that lie within the band of the same triangular filter are summed. The result is one output for each filter. If the filter bank contains K filters there will be K outputs. These will be denoted as $X_i, i = 1..K$.

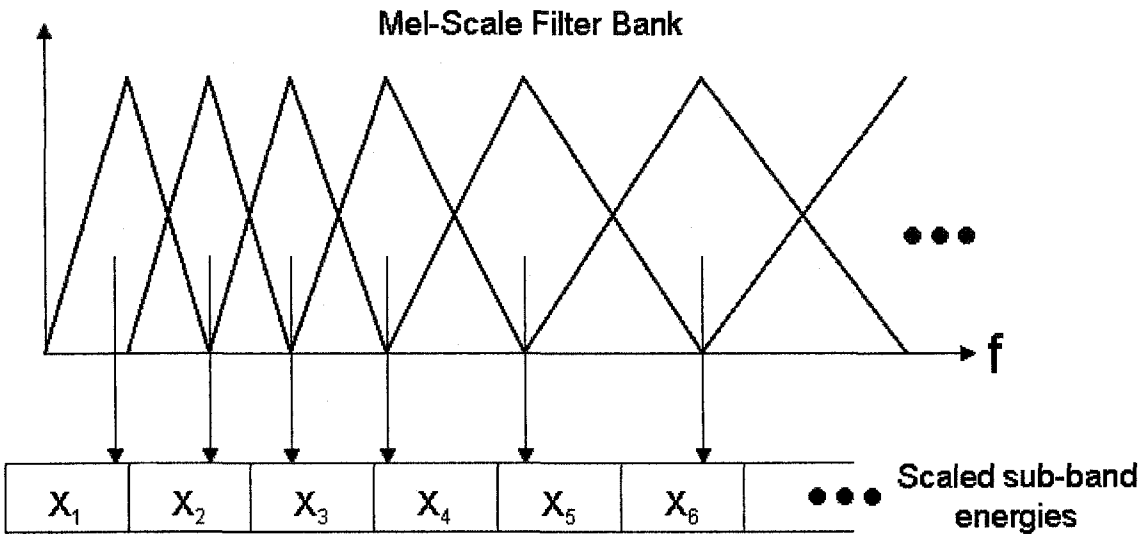

Figure 2.5: Mel scale energy binning [20]

The base ten logarithm of $X_i, i = 1..K$ is taken then a discrete cosine transform [19] is performed to compute the feature vector [19]:

$$MFCC_i = \sum_{k=1}^{K} \log(X_k) \cos[i(k-\tfrac{1}{2})\frac{\pi}{K}], \quad i = 1,2,\cdots M \tag{2.5}$$

where M is the length of the resulting feature vector. An effect of the DCT is to de-correlate the features [22], [23] and allow for the use of diagonal covariance matrices instead of full covariance matrices in the distance measure between two sets of filter-bank outputs [23]. Also the DCT compresses the information in the filter bank outputs resulting in a feature vector that is shorter than the output of the filter bank. The compression results from the projection of the filter bank outputs on the orthogonal DCT basis functions where the greater part of the variance in the information is stored in the earlier DCT coefficients.

The positioning and spacing of the centers of the triangular filters in the Mel filter bank is linear on the Mel frequency scale. This scale is shown in figure 2.6.

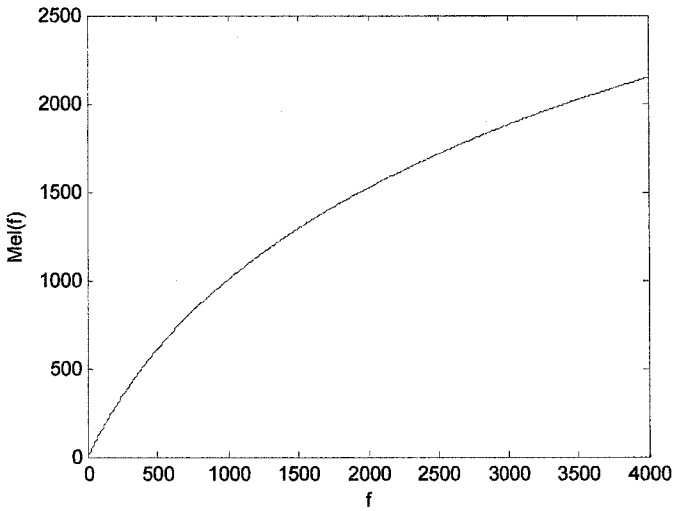

Figure 2.6: Mel frequency scale

The relationship between frequency and Mel scale frequency is as follows [21]:

$$\text{Mel(f)} = 2595\log_{10}(1+\frac{f}{700}) \tag{2.6}$$

If K triangular filters are used across a frequency band $f_{\min} \le f \le f_{\max}$ the corresponding minimum and maximum frequencies on the Mel scale $melf_{\min}, melf_{\max}$ are computed. The filter centers are computed as:

$$fc_i = 700 * \left[10^{\frac{(melf_{\min}+\frac{(melf_{\max}-melf_{\min})i}{K+1})}{2595}} - 1 \right], i = 1 \ldots K \tag{2.7}$$

As is depicted in figure 2.5 the maximum frequency of a triangle filter i is made to coincide with the center frequency of the next filter $i+1$. The minimum frequency of

a triangle filter i is made to coincide with the center frequency of the previous filter $i-1$. The maximum frequency of the last filter and the minimum frequency of the first filter coincide with $f_{\max}$ and $f_{\min}$ respectively of the complete filter bank's band. K is the number of filters.

2.2.3 LPC Cepstrum

The process outlined in figure 2.3 is common to both the processes that generate the MFCC and LPCC feature vectors. After the common process is performed, the autocorrelation method is used to generate the LPC coefficients. A recursion is then performed to determine the cepstral coefficients. For a speech frame $x_i, i = 1..N$ the p order LPC model is defined as $\hat{x}_n = -\sum_{i=1}^{p} \alpha_i x_{n-i}$. The p LPCC coefficients $C_i, 1 \le i \le p$ are computed as follows [24]:

$$C_i = -\alpha_i - \sum_{m=1}^{i-1} \frac{m}{i} C_i \alpha_{i-m}, 1 \le i \le p. \tag{2.8}$$

The following identity holds true for the cepstral coefficients derived from LPC analysis:

$$\sum_{n=-\infty}^{\infty} C_n z^{-n} = \log X(z) \tag{2.9}$$

Where X(z) is the z-transform of the speech derived from the all pole model. This is similar to the cepstral derivation through FFT coefficients which is:

$$C' = IDFT(\log(DFT(x)) \tag{2.10}$$

2.2.4 Delta Cepstrum

The data stored in the feature vectors discussed thus far, namely LPCC and MFCC, are classified as instantaneous spectral information as they contain information from a single speech frame. Another class of features containing data classified as transitional spectral information or regression coefficients are also commonly used for speaker recognition [13], [14]. These contain information extracted from more than one speech frame. These features are used in conjunction with instantaneous spectral information. Delta cepstral coefficients model the change over a specified time window of the instantaneous feature vectors. The generalized slope in time of cepstral coefficients in adjacent frames is measured by delta cepstral coefficients [26]. The first order delta cepstral coefficients are computed as [26]:

$$\Delta c_m(t) = \frac{\sum_{k=-K}^{K} k c_m(t+k)}{\sum_{k=-K}^{K} k^2} \tag{2.11}$$

Where the window length over which they are computed is $2K+1$ where in the case of this work K is 2. $c_m(t)$ is a vector containing instantaneous spectral information such as MFCC or LPCC coefficients. The dimensions of $\Delta c_m(t)$ are equal to that of $c_m(t)$. The coefficients $\Delta c_m(t)$ are appended to $c_m(t)$ to create the new feature vector of the form $[c_m(t)\ \Delta c_m(t)]$. The new feature vector is therefore twice the length of

the original feature vector $c_m(t)$. MFCC vector appended with delta cepstral coefficients will be referred to as MFCC+Δ; likewise LPCC vectors appended with delta cepstral coefficients will be referred to as LPCC+Δ.

2.2.5 Cepstral Mean Subtraction

It is a common practice in speaker recognition to perform cepstral mean subtraction or CMS on feature vectors before they are used in training or testing [17], [24], [40], [50], [52]. When speech passes through a telephone channel, the effect on cepstral feature vectors is to add an additive component to the cepstral vectors. CMS attempts to remove this additive component by subtracting the mean from the cepstral vectors [17]. CMS is performed by subtracting the mean from the cepstral vectors. If the original cepstral vectors before CMS is applied are denoted as $\vec{z}_i\{1 \le i \le T\}$ then the cepstral vectors after CMS is performed are computed as follows [17]:

$$\vec{z}_i^{comp} = \vec{z}_i - \frac{1}{T}\sum_{t=1}^{T}\vec{z}_t \tag{2.12}$$

Where T is the number of vectors. This is a very simple operation and its application to both LPCC and MFCC vectors in speaker recognition has become standard practice. The speech database that will be used in this thesis does not originate from a telephone channel. Even though no telephone channel exists, CMS is still applied in this thesis to all cepstral vectors simply because it is a standard practice in the research community. It is desirable to perform similar processing in

this thesis to that which is used in the research community. In addition the cepstral mean for a speaker is prone to change as time passes, removing the mean helps combat intersession variability. [17]

2.3 Recognition Methods

Several speaker recognition methods exist. For example, Gaussian Mixture Models, vector codebooks, hidden Markov Models, covariance models, auto-regressive vector models and neural network based models. Each of these methods creates a model of the feature vectors extracted from the speech. A GMM for example models them as a set of clusters, each cluster has an associated Gaussian distribution that is characterized by a mean and corresponding variances. A vector codebook creates a similar but simpler model than the GMM, it differs in the fact that the clusters are not modeled as Gaussian distributions. The HMM like the GMM models feature vector clusters using distributions. An HMM also includes some temporal information about the evolution over time of feature vectors. Covariance models, model the speech vectors using a single probability density function that include information about the shape of the distribution. Auto-regressive vector models model the evolution of feature vectors over time. The feature vector stream is modeled as an auto-regressive process. Neural networks of different types exist, some such as the multi-layer perceptron (MLP) are taught to learn about the differences between feature vectors of different speakers. The remainder of the

chapter is devoted to a detailed description of the implementation of GMM, covariance models, auto-regressive vector models and multi-layer perceptron for the purpose of speaker recognition.

2.3.1 Gaussian Mixture Models

GMMs or Gaussian Mixture Models are commonly used for speaker recognition and speaker verification. Similarly to VQ codebooks, the premise of a GMM is to model the clusters of speech feature vectors. The GMM models the distribution of feature vectors using a multimodal PDF. [17] The GMM attempts to perform an implicit segmentation or unsupervised clustering of the feature vectors [17].

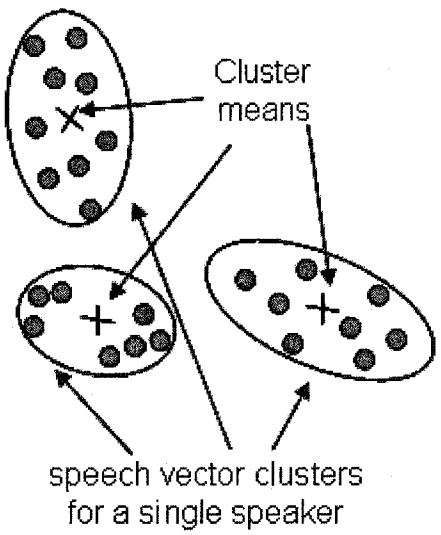

Figure 2.7: Clustering Of a Single Speakers Feature Vectors

Each of the ellipses is represented by a single Gaussian in the mixture model. All of the feature vectors both inside and outside each cluster contribute to the

calculation of the position and shape of each cluster's variance and mean. The vectors within a cluster and more generally the vectors nearer to the center of a cluster have the greatest contribution to the mean and variance of a cluster.

During training each speaker's speech will be modeled by a collection or mixture of Gaussians. Each of the Gaussians that makes up the mixture is specified by its mean, covariance and weight. Since the feature vectors extracted from the speech are vectors in R^M, each mean is a vector in R^M. The mixture weight is a scalar value that is constrained by the fact that the sum of the mixture weights of all the components is 1. The variance is a matrix; it can be either a diagonal matrix or a full covariance matrix. A model that uses diagonal covariance matrices can achieve the same effect as a model using full covariance matrices. This is achieved by using a larger number of diagonal covariance matrices. [17] It has been stated that using more components, i.e. a larger number of mixtures is better than using more complex components, specifically components using full covariance matrices. [31]

Each speaker is modeled by a unique Gaussian mixture model λ:

$$\lambda = \{\vec{\mu}_i, p_i, \Sigma_i\}\ i = 1,..L \tag{2.13}$$

Where L is the number of Gaussians, $\vec{\mu}_i$, p_i *and* Σ_i are mean, weight and covariance of mixture i respectively. The probability of a feature vector $\vec{x}$ is calculated using all the mixtures in the model as follows [17]:

$$p(\vec{x} \mid \lambda) = \sum_{i=1}^{L} p_i b_i(\vec{x}) \tag{2.14}$$

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{M/2} |\Sigma_i|^{1/2}} \exp\{-\frac{1}{2}(\vec{x} - \vec{u}_i)' \Sigma_i^{-1} (\vec{x} - \vec{u}_i)) \quad (2.15)$$

where M is the number of elements in the feature vector and $|.|$ is the determinant.

During training, the parameters of each speaker model must be determined from a sequence of training vectors X. First the vectors are grouped into clusters using the LBG algorithm outlined in section 2.3.2. The mean and variance of each cluster is then computed. The component density of each cluster is also computed as the ratio of the number of vectors in a cluster to the number of total vectors. The EM algorithm is then applied. The EM algorithm is an iterative algorithm that produces a new model after each iteration. Let λ^i represent the model produced after iteration i. The algorithm guarantees that $P(X|\lambda^{i+1}) \geq P(X|\lambda^i)$ [29]. The following equations extracted from [17] indicate how λ^i and X are used to compute λ^{i+1}. The parameters of the model λ^{i+1} are $\{\overline{\vec{\mu}}_i, \overline{p}_i, \overline{\Sigma}_i\}$ $i = 1,...,L$ while those of λ are $\lambda = \{\vec{u}_i, p_i, \Sigma_i\}$ $i = 1,..L$. N is the number of feature vectors in the training sequence, while L is the number of components in the GMM. The following equations are used to re-estimate the model parameters [17]:

$$\overline{p}_i = \frac{1}{N} \sum_{t=1}^{N} p(i | \vec{x}_t, \lambda) \quad (2.16)$$

$$p(i | \vec{x}_t, \lambda) = \frac{p_i b_i(\vec{x}_t)}{\sum_{k=1}^{L} p_k b_k(\vec{x}_t)} \quad (2.17)$$

$$\vec{\bar{u}}_i = \frac{\sum_{t=1}^{M} p(i \mid \vec{x}_t, \lambda)\vec{x}_t}{\sum_{t=1}^{M} p(i \mid \vec{x}_t, \lambda)} \tag{2.18}$$

$$\vec{\bar{\sigma}}_i = \frac{\sum_{t=1}^{M} p(i \mid \vec{x}_t, \lambda)\vec{x}_t^2}{\sum_{t=1}^{M} p(i \mid \vec{x}_t, \lambda)} - \vec{\bar{u}}_i^2 \tag{2.19}$$

During testing the probability that a given speaker generated a set of test vectors model must be computed. Assuming that the feature vectors in a test utterance are independent, this is of course a simplification, the probability of the test utterance, Y consisting of T feature vectors, given the speaker model λ is as follows [17]:

$$P(Y \mid \lambda) = \prod_{t=1}^{T} p(\vec{y}_t \mid \lambda) \tag{2.20}$$

The log likelihood is used instead [17]:

$$\log P(Y \mid \lambda) = \frac{1}{T}\sum_{t=1}^{T} \log p(\vec{y}_t \mid \lambda) \tag{2.21}$$

During testing $\log P(X \mid \lambda)$ will be computed for a number of speaker models. In the case of S speakers, the most likely speaker to have uttered the utterance Y will be i^* [17]:

$$i^* = \arg\max_{1 \le i \le S} \log P(Y \mid \lambda_i) \tag{2.22}$$

2.3.2 LBG Algorithm

The following section describes what are known as clustering algorithms. The creation of a GMM, HMM or VQ codebook requires that a set of representative means be calculated in order to characterize the clusters in the data. A number of variations of the k-means algorithm can be used for this purpose. These algorithms when applied to a set of p-dimensional feature vectors $\{\vec{x}_i, 1 \le i \le N\}$ will produce a p-dimensional codebook of size L: $\{\vec{z}_j, 1 \le j \le L\}$.

The LBG algorithm proposed in [28] is a clustering algorithm similar to the k-means algorithm. It is an iterative algorithm that begins with one large cluster containing all the feature vectors in the training set and doubles the number of clusters after each iteration. The training data is successively split into 2, 4, 8,...2^L clusters during successive iterations. The centroids or means of the 2^L clusters produced in the final L'th iteration become the resulting codebook. The algorithm is as follows [27]:

Step 1: Initialization: Set the number of clusters L to 1. Assign all the feature vectors in the training set to the initial cluster and compute the mean of all the vectors in the cluster. The result is a codebook with a single entry.

Step2: Splitting: Double the number of vectors in the current codebook $\{\vec{z}_j, 1 \le j \le L\}$ by replacing each codebook entry by two vectors. Each of the existing clusters belonging to each existing codebook entry must be split into two clusters and

the mean of each of these new clusters will become a new codebook entry. The algorithm for splitting the cluster is as follows [29]:

a) Compute the covariance matrix of the feature vectors in the cluster.

b) Determine the eigenvectors and eigenvalues of the covariance matrix.

c) Find the principal eigenvector, the one that corresponds to the largest eigenvalue.

d) Determine the projection (dot product) of each feature vector in the cluster on the eigenvector from step c. Sort the projections.

e) Split the projections into two clusters where the cutting surface is the median projection.

f) Assign the feature vectors to one of two clusters according to the cluster occupied by the corresponding projection.

Each cluster has therefore been split into two clusters. Perform steps 3 to 5 on each newly split cluster individually in order to determine the two new means for each newly split cluster separately. After the new means of each cluster has been determined separately, repeat steps 3 to 5 over the complete new codebook over all new cluster means and all feature vectors in all clusters.

Step 3: Classification: For each feature vector $\{\vec{x}_i, 1 \le i \le N\}$ determine the nearest vector in the codebook using some distortion measure and associate each vector with the nearest codebook vector. The result is that each codebook entry $\{\vec{z}_j, 1 \le j \le L\}$ will have an associated cluster $\{C_j, 1 \le j \le L\}$ where C_j is the cluster of

feature vectors that are nearer to $\vec{z}_j$ than to any other codebook vector. This rule can be summarized as:

$$\vec{x} \in C_j \text{ iff } d(\vec{x},\vec{z}_j) \le d(\vec{x},\vec{z}_k) \text{ for all } j \neq k \tag{2.23}$$

The distortion measure used is the squared Euclidian distance between the feature vector and codebook entry:

$$d(\vec{x},\vec{z}_j) = (\vec{x}-\vec{z}_j)^T(\vec{x}-\vec{z}_j) \tag{2.24}$$

Repeat the current step until the means cease to change or the step has been repeated 100 times.

Step 4: Codebook updating: For each codebook entry update the entry by setting $\vec{z}_j$ to be the mean of all vectors assigned to the cluster C_j as follows:

$$\vec{z}_j = \frac{1}{K_j}\sum_{\mathbf{x}\in\mathbf{C}_j}\vec{x} \tag{2.25}$$

where K_j is the number of vectors belonging to cluster j.

Step 5: Repetition: Repeat steps 3 to 4 until either 100 repetitions have taken place or the codebook entries no longer change.

Step 6: Repeat steps 2 and 5 until the number of clusters reaches the desired number of clusters.

2.3.3 Auto-Regressive Vector Model

The auto-regressive vector model (ARVM), for speaker recognition, models the evolution of feature vectors over time. It is the vector analog of the scalar LPC model. The LPC model uses speech samples while the ARVM model uses cepstral vectors or filter-bank outputs. It has been used for speaker recognition in, [34]-[40]. The following equation is the basis of the ARVM model [37]:

$$\sum_{k=0}^{p} A_k(\vec{x}_{t-k} - \vec{u}) = \vec{e}_t \tag{2.26}$$

$\{\vec{x}_i, 1 \le i \le N\}$ are the cepstral feature vectors of dimension M. The model order is p. $\{A_k, 0 \le k \le p\}$ are a series of $p+1$ prediction matrices of size (M x M), A_o is the identity matrix. It is set to this value in order to guarantee a unique solution for the prediction matrices [35]. $\vec{e}_t$ is the prediction error, it is an M-dimensional vector. The optimal model order p is 2 [35], [37]-[40].

In order to solve for the prediction matrices, it is first necessary to compute what are referred to in [37] as the 0^{th} to p^{th} lag covariance matrices $\chi_0 \ldots \chi_p$ of the training speech vectors which are computed as follows [37]:

$$\chi_k = \frac{1}{N} \sum_{t=k+1}^{N} (\vec{x}_t - \vec{\mu})(\vec{x}_{t-k} - \vec{\mu})^T \tag{2.27}$$

The prediction matrices of the AR model can then be determined as follows [35]:

$$[A_1 A_2 ... A_p] = -[\chi_1^T \chi_2^T \cdots \chi_p^T] \begin{bmatrix} \chi_0 & \chi_1^T & \ddots & \chi_{p-1}^T \\ \chi_1 & \chi_0 & \ddots & \chi_{p-2}^T \\ \ddots & \ddots & \ddots & \ddots \\ \chi_{p-1} & \chi_{p-2} & \ddots & \chi_0 \end{bmatrix}^{-1} \tag{2.28}$$

In addition to the predictor coefficient matrices the following block Toeplitz matrix is created and stored as part of each speaker's model [35]:

$$[X] = \begin{bmatrix} \chi_0 & \chi_1^T & \ddots & \chi_p^T \\ \chi_1 & \chi_0 & \ddots & \chi_{p-1}^T \\ \ddots & \ddots & \ddots & \ddots \\ \chi_{p-1} & \chi_{p-2} & \ddots & \chi_0 \end{bmatrix} \tag{2.29}$$

When the test speech is scored against the candidate speaker models, an ARVM must be created for the test utterance in the same way that the speaker specific ARVMs were created from the training speech. The ARVM of a test utterance from an unknown speaker must be compared with the ARVM of known speakers to determine the identity of the unknown speaker. The matrix $[Y]$ is computed for the cepstral vectors of a test utterance $\{\vec{y}_i, 1 \le i \le T\}$ just as the matrix $[X]$ is computed for the cepstral vectors of the training utterance $\{\vec{x}_i, 1 \le i \le N\}$. The matrix $[B] = [B_1 B_2 ... B_p]$ must also be computed for the cepstral vectors of the test utterance using the same method used to compute $[A] = [A_1 A_2 ... A_p]$ for the training utterance. The lagged covariance matrix of the test speech is computed as follows [37]:

$$\gamma_k = \frac{1}{T} \sum_{t=k+1}^{T} (\vec{y}_t - \vec{\mu})(\vec{y}_{t-k} - \vec{\mu})^T \tag{2.30}$$

The prediction matrices of the AR model for the test speech can be determined as follows [35]:

$$[B_1 B_2 ... B_p] = -[\gamma_1^T \gamma_2^T \ ... \gamma_p^T] \begin{bmatrix} \gamma_0 & \gamma_1^T & \ddots & \gamma_{p-1}^T \\ \gamma_1 & \gamma_0 & \ddots & \gamma_{p-2}^T \\ \ddots & \ddots & \ddots & \ddots \\ \gamma_{p-1} & \gamma_{p-2} & \ddots & \gamma_0 \end{bmatrix}^{-1} \tag{2.31}$$

The block Toeplitz matrix $[Y]$ is computed as follows [35]:

$$[Y] = \begin{bmatrix} \gamma_0 & \gamma_1^T & \ddots & \gamma_p^T \\ \gamma_1 & \gamma_0 & \ddots & \gamma_{p-1}^T \\ \ddots & \ddots & \ddots & \ddots \\ \gamma_{p-1} & \gamma_{p-2} & \ddots & \gamma_0 \end{bmatrix} \tag{2.32}$$

More than one measure can be used to determine the distance between a set of speaker models and the model of a test utterance. The non-symmetrical Itakura distance measure [42] is the basis of the first discriminative measure. A symmetrized version of the Itakura distance measure is used for speaker recognition in [35], [40]. This method will be referred to as AR-Itakura throughout this text. The performance of this measure was compared in closed set speaker recognition trials on the KING database to the measures used in [29], [38] and [39]. It gave equal if not better performance when trials were performed with un-reverberated or reverberated test speech. The Itakura distance is measured between the model of the test utterance having block-Toeplitz matrix $[Y]$ and prediction coefficient matrices $[B] = [B_0 B_1 B_2]$ and a speaker model having block-Toeplitz matrix $[X]$ and

prediction coefficient matrices $[A]=[A_0 A_1 A_2]$. The non-symmetric Itakura distance is defined as follows [35]:

$$\mu(y,x)=\log(tr\left[\frac{A[Y]A^T}{B[Y]B^T}\right]) \tag{2.33}$$

Here y is the speaker that spoke the test utterance modeled by $[B]$ and $[Y]$. x is the speaker that spoke the training speech modeled in part by $[A]$. The term within the square brackets is the covariance of the prediction error of $\{\vec{y}_i, 1 \le i \le T\}$ filtered by $[A]$ then normalized by the covariance of the prediction error of $\{\vec{y}_i, 1 \le i \le T\}$ filtered by $[B]$.

The symmetrized Itakura distance that is a superior measure to the Itakura distance [35], [38] is used instead of the non-symmetrical Itakura distance.

The symmetrized Itakura distance measure extracted from [35] and [40] is as follows:

$$\mu_{sym}(y,x)=\frac{1}{2}\log(tr\left[\frac{A[Y]A^T}{B[Y]B^T}\right] x\, tr\left[\frac{B[X]B^T}{A[X]A^T}\right]) \tag{2.34}$$

The most likely speaker to have uttered a speech segment $\{\vec{y}_i, 1 \le i \le T\}$ is determined by computing the symmetrized Itakura distance between each speaker model and the test speech then selecting the speaker x whose distance from the model of the test utterance is minimum. A note must be made that the denominators of each term

within the $tr()s$ of eq. 2.34 can be exchanged resulting in the following distance measure:

$$\mu_{sym}(y,x) = \frac{1}{2}\log(tr\left[\frac{A[Y]A^T}{A[X]A^T}\right] x\ tr\left[\frac{B[X]B^T}{B[Y]B^T}\right]) \qquad (2.35)$$

The measures used in eq. 2.34 and 2.35 perform similarly under conditions of no reverberation but the measure in eq. 2.35 has a recognition accuracy of only half that of eq. 2.34 when reverberation is introduced, for this reason the second measure was not used. The auto-regressive method of speaker recognition using the distance measure of eq. 2.34 will be referred to as AR-Itakura.

A second discriminative measure was also used during scoring. It is part of a speaker recognition method that is similar to AR-Itakura except for three differences, one in training and two in scoring. This method, which will be referred to as AR-AGS (as it uses the arithmetic, geometric sphericity measure), was the best performing method from [37] where different AR vector measures were compared. The method will now be described. First the feature vectors are sorted in a random order before they are used to compute $[A]$ and the block Toeplitz matrix $[X]$. The computation of these matrices is the same as for AR-Itakura. Before scoring is performed the test vectors must also be randomized. The distance measure which is the arithmetic-geometric sphericity measure [37] tests the proportionality of two covariance matrices $E_X^{(A)}$ and $E_Y^{(A)}$ which are respectively the covariance matrix of

the residual of $\{\vec{x}_i, 1 \le i \le N\}$ filtered by $[A]$ and $\{\vec{y}_i, 1 \le i \le T\}$ filtered by $[A]$. The distance measure is computed as follows:

$$D = \log\left(\frac{\frac{1}{M}tr(\Gamma)}{[\det(\Gamma)]^{\frac{1}{M}}}\right) \tag{2.36}$$

where

$$\Gamma = (E_X^A)^{-\frac{1}{2}} E_Y^A (E_X^A)^{-\frac{1}{2}} \tag{2.37}$$

The most likely speaker is the one for whom the distance D to his model is minimum.

2.3.4 Covariance Matrix Modeling

Covariance matrices can be used to measure the similarity between a test utterance and a training utterance. This is a relatively simple method conceptually, is relatively computationally efficient, and has been used in [50], [51], [52], [53], [54], [55], [56], for the tasks of closed set speaker recognition or verification.

Let Y represent the covariance matrix of a test utterance $\{y_t, 1 \le t \le T\}$ from an unknown speaker and let X represent the covariance matrix of a training utterance $\{x_t, 1 \le t \le N\}$. X and Y are computed in the same manner as follows:

$$Y = \frac{1}{T}\sum_{t=1}^{T}(\vec{y}_t - \bar{y})^T(\vec{y}_t - \bar{y}) \tag{2.38}$$

$$X = \frac{1}{N}\sum_{t=1}^{N}(\vec{y}_t - \bar{y})^T(\vec{y}_t - \bar{y}) \tag{2.39}$$

The following property exists [49]:

$$\frac{1}{T}\sum_{t=1}^{T}(\vec{y}_t - \bar{y})^T X^{-1}(\vec{y}_t - \bar{y}) = tr(YX^{-1}) \tag{2.40}$$

A symmetrical measure of distance between a training utterance and test utterance is simply [51]:

$$D_i = \log[tr(YX_i^{-1})tr(X_iY^{-1})] - 2\log(P) \tag{2.41}$$

P is the length of the feature vector. This is the arithmetic harmonic sphericity measure or SM. A second method, the divergence shape or DS also exists it is computed as [51]:

$$D_i = \log[tr(Y - X_i)(X_i^{-1} - Y^{-1})]. \tag{2.42}$$

Each speaker in a set of S speakers is modeled by a single covariance matrix X_i. A test utterance from an unknown speaker is modeled by a covariance matrix Y. D_i is computed for all speaker models. The most likely model to have produced the test speech is $i^* = \arg\min_{1\le i\le S} D_i$. The speaker recognition using the SM will be referred to as SM and speaker recognition using the DS will be referred to as DS.

Speaker recognition/verification measures that rely solely on the covariance matrix to compute either the arithmetic harmonic sphericity measure or divergence

shape, measure the differences in the shape of the probability density functions of the training and test speech [57].

2.3.5 Multi-Layer Perceptron (MLP) Neural Network

Neural networks are commonly used for classification problems such as speaker recognition example of this are [58]-[62]. The commonly used MLP was chosen as one of the closed-set speaker recognition algorithms to implement due to the fact that it was markedly different from the other methods already attempted such as GMM, ARVM and covariance models. Elliptical basis functions [61] and radial basis functions [59], which are themselves neural networks, have as their kernel a structure that is similar to the GMM for this reason they are not used. Time delay neural networks require that the feature vectors used maintain their time sequence. When training networks this can be problematic, as it does not allow for the reduction of the number of feature vectors used that is accomplished by discarding vectors. It was for these reasons that the MLP was the neural network configuration that was selected for use in the speaker recognition task. Also there exist in the open literature clear specifications for the implementation of MLPs for speaker recognition namely [60], [62]. The topology of the networks used in each paper is very similar. Each uses a single three-layer MLP neural network to recognize the speech for a single speaker. Each network is composed of an input layer, a hidden layer and an output layer. Each MLP has in its input layer a single neuron for each cepstral coefficient in the input

vector. Coincidentally [60] and [62] both use cepstral vectors of length 12 where in [60] the cepstral vector is an LPCC vector while in [62] it is an MFCC vector. Both papers have a single neuron in the output layer. Both papers differ on the size of the hidden layer; in [60] the hidden layer size is 16 while in [62] it is 12, the same length as the input vector. In this thesis the hidden layer size was set to equal the number of cepstral coefficients in the input vector. The output of each of the input layer neurons is connected to the input of all of the hidden layer neurons. As is the case in [60] the activation function used in the hidden layer is a 'hyperbolic tangent sigmoid' and in the output layer is 'log-sigmoid', these are illustrated in figure 2.8 and figure 2.9 respectively. Both the middle and output layers have a bias.

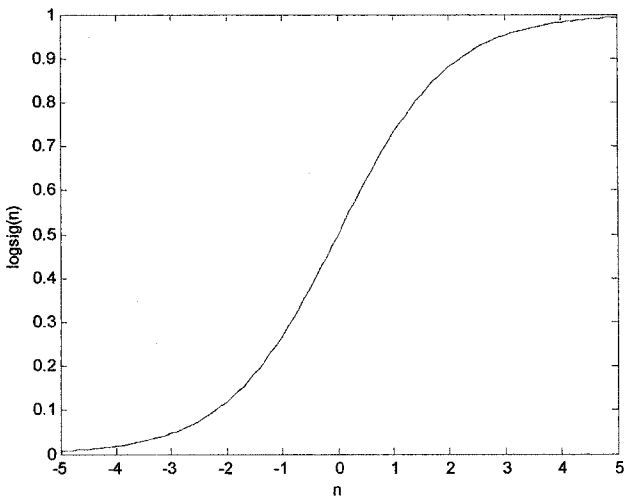

Figure 2.8: The log-sigmoid function

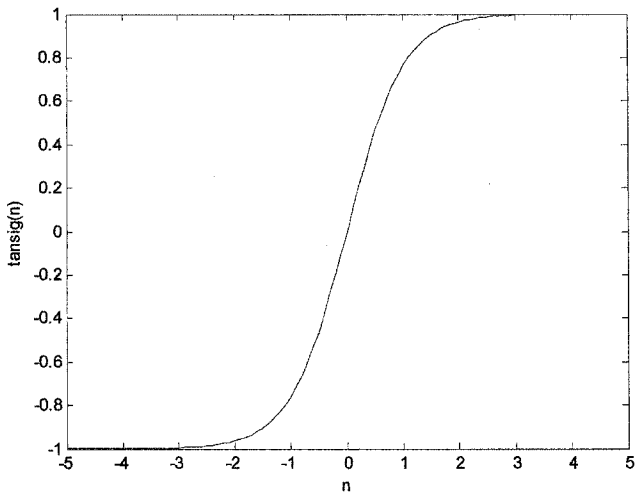

Figure 2.9: The hyperbolic tangent sigmoid function

For an S-speaker closed set speaker recognition task where S is the number of speakers, each speaker in the set has an associated neural network. The weights and biases of the neural network were initialized using the Nguyen-Widrow algorithm; this method introduces randomness into the process of creating the neural network. Both [60] and [62] use the Levenberg-Marquardt algorithm to train each network, for this reason this was the training algorithm chosen. The performance function used during training was the Matlab default, the mean squared error (MSE). Each of the S networks was trained to output a 1 when presented with a feature vector belonging to the speaker associated with the network and a 0 when a feature vector belonging to any of the other S-1 speakers was presented to the network. The vectors belonging to the speaker associated with the network are termed excitatory vectors, those belong to the other speaker (the anti-speakers) are inhibitory vectors.

To make available training frames that will train a network to recognize a 1 when presented with the correct speaker's speech is straightforward. The excitatory vectors are presented to the network one vector at a time and the network is trained to output a 1. Inhibitory vectors are feature vectors from speakers other than the speaker whose model is being trained. On the other hand the selection of the inhibitory vectors to supply to the network required some imagination. The number of inhibitory vectors is much greater than the number of excitatory vectors. Training with such a disparity in the number of vectors can lead to the network being trained to always output a 0. In order to combat this problem one solution is to use an equal number of inhibitory and excitatory vectors to train each network. In [60] and [62] this problem is solved by compressing the number of inhibitory and excitatory vectors. Two codebooks of equal size can be created, one containing inhibitory vector means and the other excitatory vector means. Another alternative exists, to avoid having to compress the large number of inhibitory vectors that can be a time consuming process that must be repeated for each network. The alternative is to retain all the excitatory vectors but to only retain every N^{th} inhibitory vector. Here N is the ratio of inhibitory to excitatory vectors.

2.3.6 Hidden Markov Model and Vector Quantization

Hidden Markov Model's or HMMs are commonly used in speaker recognition. [43]. In this work HMMs are not used for explicit phonetic classification of speech

segments but rather any phonetic segmentation that may result is due to implicit clustering of speech sounds in a similar manner as is the case for a GMM. Ergodic HMMs like ARVM and GMM are considered to be prevalent text independent speaker recognition methods as evidenced by their ability to produce good results [14]. HMMs have been used for text independent speaker recognition in [44], [45], [46], and [47]. It has been found during the experimental work of this thesis that ergodic HMM performance is very similar to that of GMMs. The HMM in addition to the variance, component density and mean information also includes a state transition matrix. It has been found that the addition of the state transition matrix does not result in the HMM performing differently from the GMM when the test speech is reverberant or not. For this reason the HMM will not be discussed further. This is due to the fact that in the computation of the probability of the feature vector sequence, the predominant factor is the probability of the feature vector relative to the Gaussian mixture in the state while the transitional probabilities have significantly less influence on the outcome. This would be different in the case if some transitions had zero probability which may be the case in speech recognition or text dependent speaker recognition. The addition of state duration modeling into the HMM would incorporate information into the HMM that is not present in the GMM. The GMM is a simplification of an HMM, that it is a single state HMM.

Vector quantization (VQ) has also been used for speaker recognition; it is effectively a simplification of the GMM. It has been found during the work of this

thesis that VQ performs similarly to the GMM under conditions of no reverberation and degrades similarly under reverberation, for this reason they are not pursued further. The difference in performance that was observed indicates that a larger codebook is required for the VQ system to obtain similar performance to the GMM.

2.4 Related Work

Speaker recognition in a hands free environment has been addressed in publications [3]-[8]. In reference [3] the effect of reverberation on an 8 speaker closed set text-independent speaker recognition task was studied. The recognition method used was a multiple binary classifier model neural network. The measure used to quantify the recognition accuracy was the number of correctly classified speech frames. In this paper the feature vectors compared were line spectrum pairs (LSP), reflection coefficients and Mel-cepstrum coefficients. Reverberation was simulated using the image method. The following experiments were performed:

1) The test speech was reverberated by increasing the separation between the source and receiver between 0.05m and 0.4m. 4 different speaker positions were used. The receiver position was kept constant (mounted on the wall) and the room size was 2x10x4m. The training speech remained un-reverberated. Three feature vectors were used, (LSP), reflection coefficients and Mel-cepstrum coefficients. The results indicate that Mel-cepstral and LSP feature vectors degrade less than reflection coefficients.

2) Experiment 1 was repeated except that only (LSP) were used. Training speech was reverberated using the same room model as that of the test speech. Results improved considerably.

3) This experiment was similar to experiment 1 with the exceptions that the separation between speaker and microphone was varied between 0.1 and 1.6m, 6 different positions were used, and the training speech was reverberated using the same impulse response used to reverberate the test speech. The results indicates that as long as the impulse response used during training matches that used during testing, the recognition accuracy does not necessarily degrade as the speaker to microphone separation increases.

4) An experiment was conducted where two different room sizes were used. In the first room, the larger of the two, speaker to microphone separation was kept constant but the position of the source and receiver were changed in the room. For each pair of source and receiver positions the reverberation time was increased, probably by increasing the wall reflection coefficients. In the second room the speaker to microphone separation was set to the same value as that in the first room, the reflection coefficients were also set to allow for the same reverberation time as the first room. The results indicate that so long as the speaker to microphone separation is constant, the exact positions in the room of the speaker and microphone do not matter as the recognition

accuracy degrades evenly for all positions; the important fact is the reverberation time. Both rooms have equal reverberation time but the smaller room has lower recognition accuracy due to the fact that its impulse response is denser.

5) In this final experiment the speaker position was varied in a room while the position of the receiver was kept constant. Training was performed with speech that was convolved with an impulse response for a speaker at the center of the room. This same experiment was repeated but with training speech for a speaker at the corner of the room as well as the center of the room. What was found was that using training speech simulated to originate from two positions, the room center and corner, did not reduce the degradation in recognition more than only using training speech simulated to originate from the center of the room.

In the second paper [4], text-dependent speaker recognition was performed on a set of 15 speakers. Here the training and test speech is the same for all speakers, only one sentence is used and it is always the same. One of either 8ms or 64ms frames were extracted from the speech and the magnitude spectrum of each frame was computed. The average of the magnitude spectra of the frames was computed for each speaker. Each speaker's average magnitude spectrum was normalized by the overall energy of the frames. This process was repeated for test speech. The result was a normalized average spectrum x for the training speech of each speaker and a normalized average

spectrum y of each test utterance. The recognition method used was very simple. The normalized average spectrum of the test speech was compared to that of the training speech of each speaker. The speaker whose training speech normalized average spectrum x was nearest to the test speech normalized average spectrum y was selected as the speaker who uttered the test speech. The distance between x and y was the city-block distance and is computed as follows: $D = \sum_{i=1}^{N} |x_i - y_i|$ where N was 64 if 8ms frames are used and 512 if 64ms frames are used. Initially training speech was spoken directly into the microphone and test speech was recorded at 4 different combinations of speaker and microphone positions all in the same room. Here the direction that the speaker faced was also manipulated so that the ratio of reverberant to direct speech reaching the microphone was successively increased. The results indicated that there was degradation in recognition performance as the ratio of reverberant to direct speech increased. The author claims that the transfer function of the room for any of the 4 combinations of speaker and microphone position were nearly identical. The author averages the spectrum of speech uttered by different speakers at 2 of the 4 combinations of speaker and microphone position. This average reverberated spectrum is referred to as $T(f)$. The author also averages the spectrum for different speakers spoken in the un-reverberated training configuration. The average non-reverberated spectrum is referred to as $S(f)$. From

these the author computed that average transfer function $H(f)$ of the room using the relation $H(f) = T(f)/S(f)$. He used the average transfer function to weight the reference template used for training and repeated the recognition experiments with reverberant speech. The results improved considerably. Training with speech from the same room but from a different position than that from which the test speech was spoken from also improves accuracy.

In [5], a VQ codebook was used for closed set speaker recognition on 38 speakers from the TIMIT database. LPCC coefficients were used. A reverberant room impulse response was simulated using the image method. A number of beam former structures were used to counteract the reverberation. These were a 2 dimensional matched filter array, 2 one-dimensional line arrays, and a single one dimensional line array. The matched-filter array inverts the room impulse response before the speech is summed by the array. Delay and sum beam forming was used. In addition to the simulated microphone arrays, a single microphone system was simulated. In the first experiment the reverberant room impulse response was used to reverberate both training and test speech. The identification accuracy of the different microphone array systems and the single microphone system were compared to each other. Included were the results where the speech was not reverberated in training or testing. It was found that a two-dimensional matched-filter array gave the best results and the single microphone gave the worst results. In the second experiment a noise generator was simulated to corrupt the training and test speech. The corruption was

in addition to the reverberation. Once again training and test speech were reverberated using the same impulse response. The performance of the two-dimensional matched-filter array was compared to the single microphone performance. It was found that the two dimensional matched filter arrays gave consistent performance as the noise power was increased. The single microphone system's performance increased as SNR was increased. The identification accuracy of the 2-D matched filter array consistently outperformed that of the single microphone. In the third experiment reverberation was only used to corrupt the test speech and not the training speech. A noise generator was simulated and used to corrupt the test speech. The performance of the 2-D matched-filter array was compared to a single microphone and the 2-D matched filter array gave superior performance. In the first experiments it was explicitly stated in [5] that for each of the microphone systems used, that training and test speech were transduced using the same system. This means that for the single microphone system a single microphone was used for test and training speech. For the 2-D matched filter array, a 2-D matched filter array was used to transducer training and test speech.

In [6], a simulated room reverberation model was used to simulate reverberation. The speech was corrupted using white Gaussian noise and real fan noise extracted from a computing system. The different noise types were not combined but rather were used separately in different experiments. The closet set speaker recognition system was HMM based and LPCC vectors were used. Closed set speaker

recognition was performed on a 25-speaker set. The training speech was not corrupted. The two methods of speech enhancement were compared to the baseline that used no enhancement. In the first the LMS algorithm was used for adaptive noise cancellation. Here the objective was to cancel the effect of the fan or Gaussian noise. The second enhancement method made use of a microphone array. The results indicate that the adaptive noise cancellation technique performed best. The microphone array performed second best. Both enhancement techniques outperformed the case where no enhancement took place.

In [7], a 32 mixture GMM was used for closet set speaker recognition on a population of 25 speakers. LPCC cepstral coefficients were the feature vector type used. Reverberation was simulated using the image method. In order to counteract the reverberation a microphone array was used with a simple delay and sum beamformer. In addition a second system combines the delay and sum beam former with an LMS adaptive filter. A white noise source was used to corrupt the speech. The performance of the single microphone, 4-microphone delay and sum beamformer and the delay and sum beamformer combined with the adaptive filter were compared. The effect of cepstral mean subtraction and relative spectral filtering was investigated. It was found that the delay and sum beamformer combined with the adaptive filter gave the best results. Cepstral mean subtraction and relative spectral filtering gave similar performance and gave better performance that if they were not used.

In [8], speaker verification using a GMM in a reverberant room using a near field adaptive beamformer, augmented with a post filter, with 9 microphones was investigated. The reverberant room impulse response was obtained using a maximum length sequence (MLS) measurement of an office environment room. The speaker position was 70cm from the mircrophone. The verification task consisted of verifying the identity of 112 male speakers from the TIMIT database. MFCC features appended with delta coefficients were used. A diffuse noise source consisting of computer, air conditioner noise and a variable level of background conversation was added to the test speech. A localized noise source was placed 270cm from the microphone, this consisted of noise from the NOISEX database. Comparisons were made of the single microphone system and beamformer system in two experiments where in the first the test speech was reverberated and corrupted by ambient noise, in the second the test speech was reverberated and corrupted by localized noise. The results of the single microphone and beamformer were compared as the noise SNR was decreased. In both experiments the beamformer outperformed the single microphone. This was more apparent in high SNR conditions.

Chapter Three

Simulation and Experimental Setup

This chapter will describe the simulation and experimental setup used for the evaluation of the speaker recognition methods. Section 3.1 describes the KING speech database that is used. Section 3.2 outlines the processing that was performed on this speech database before recognition trials are performed. The reverberation model used to simulate reverberation is described in section 3.4. An actual experiment was conducted in a real reverberant room. The setup of that experiment will be described in section 3.3. Section 3.6 will describe the hardware and software used to conduct the speaker recognition simulations.

3.1 Speech Database Used

The KING [65], [66] speaker recognition database was used for the speaker recognition trials. The KING database was developed specifically for speaker

recognition [65]. It consists of speech from 51 speakers. Each speaker with the exception of two, spoke 10 utterances. The length of the speech files vary in length between 30s and 60s. The two exceptional speakers only speak for 5 utterances. Each utterance is different as the speaker is prompted to speak about one of 10 assigned topics. The speaker does not read from a specified text. The speech is sampled at 8 KHz, 16-bit [66].

3.2 Speech Database Processing

The KING database underwent processing prior to its being used to train and test the speaker models. The processing resulted in three different classes of the KING database with each class of the KING database containing different versions of the database. The three classes are as follows:

1) Non-reverberated KING database processed in order to resemble telephone line speech. This process is illustrated in figure 3.1.

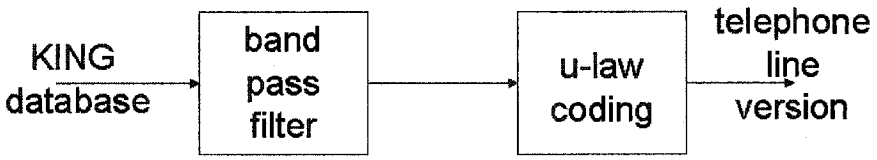

Figure 3.1: Processing of KING database to simulate telephone line speech.

2) Reverberated KING database produced by filtering the KING database using a simulated reverberant room impulse response. After filtering

with the reverberant room impulse response, the reverberated speech is processed in order to resemble telephone line speech. Multiple versions of this class of the KING database resulted. This is because different simulated reverberant room impulse responses were used, this process is illustrated in figure 3.2.

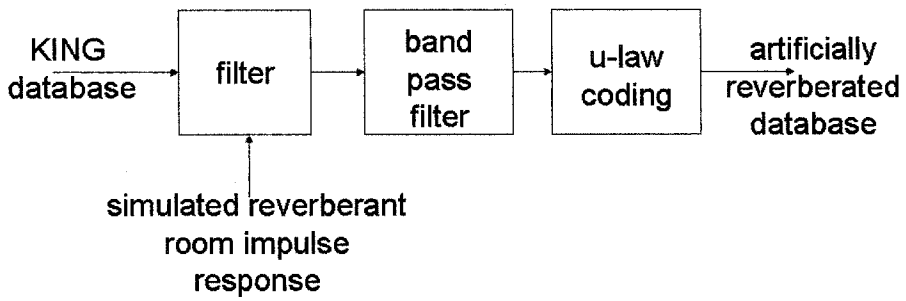

Figure 3.2: Processing of KING database to produce artificially reverberated speech.

3) The KING database was stored on a portable computer and taken to a real reverberant room. The speech database was played using a PC speaker in the reverberant room and simultaneously recorded using a microphone. The result is that the speech was reverberated by the room before it was recorded. The same process was repeated except that instead of playing the speech into the room, the speech was played inside a small box lined with absorptive material. The same speaker and microphone were used to play the speech and simultaneously record it as was done when the speech was played directly into the room. Two different versions of the database therefore exist in this class. Each one was processed after being recorded to resemble

telephone line speech. The database was recorded in the reverberant room in order to reverberate the speech. The database was recorded in an anechoic box in order to have an un-reverberated version of the database that was transduced using the same speaker and microphone combination as the reverberated speech. These processes are illustrated in figure 3.3 and figure 3.4.

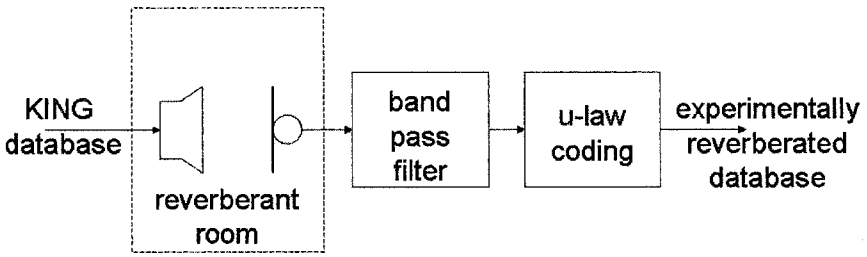

Figure 3.3: Processing of KING database to produce speech reverberated by a real room

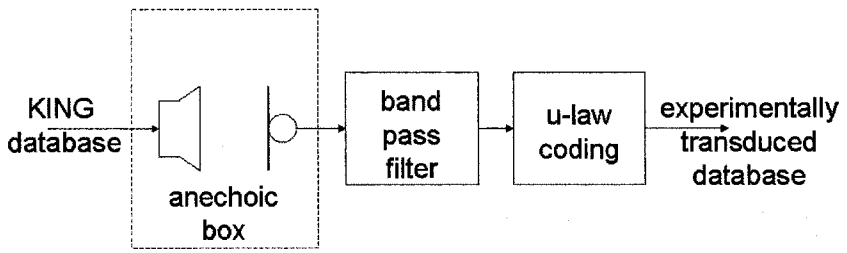

Figure 3.4: Processing of KING database to produce speech transduced in an anechoic box.

The frequency response of the filter used to limit the speech spectrum to the 300 to 3400 band is shown in figure 3.5 where the sampling frequency is 8Khz.

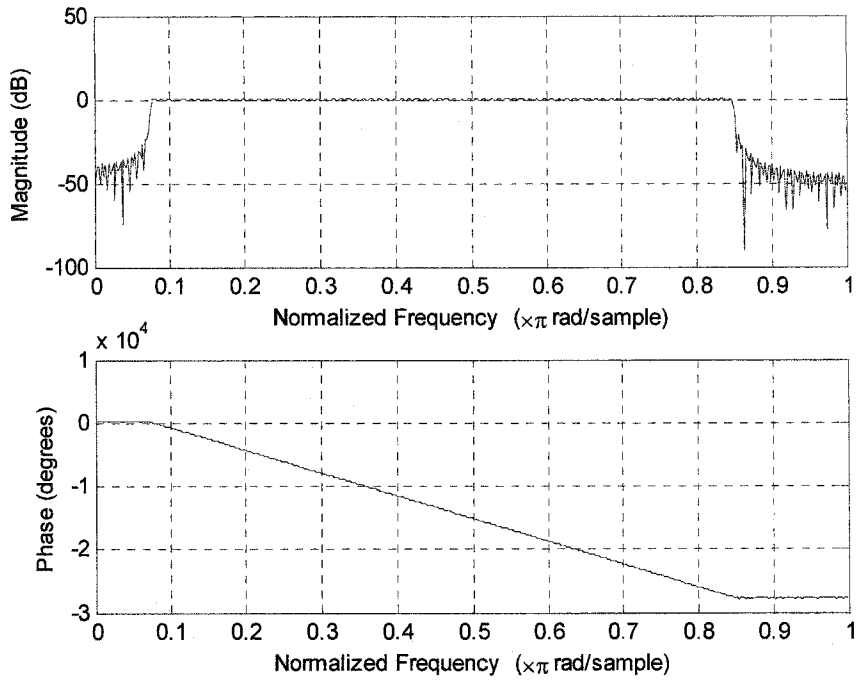

Figure 3.5: Frequency response of telephone band filter.

3.3 Setup of experiment in Reverberant Room

In order to play the KING database a Dell model A-215 speaker was used to play the speech files. The speech was recorded using a P-9970 electret microphone manufactured by Digikey Electronics. The recorded signal was amplified by a propriety pre-amplifier build by the NRC based on the Texas Instrument INA171 low noise instrumentation amplifier before being recoded using an M-Audio Delta 1010 sound card. The speech database was played and recorded in two separate experiments. The room used for both experiments is illustrated in figure 3.6. This room is a room at Carleton University situated in the Mackenzie building. The room number is ME4359.

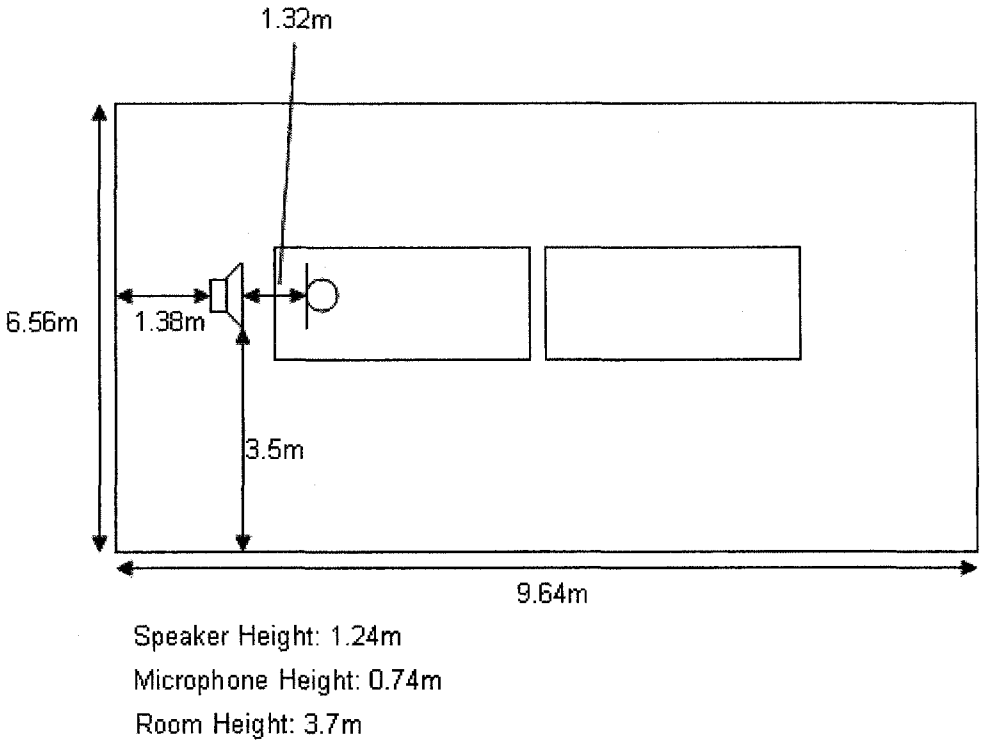

Figure 3.6: Illustration of experimental setup in reverberant room.

The walls in this room are composed of painted concrete blocks. Two of the walls have installed on them plastic drawing surfaces. In the first experiment the speech database was played through the speaker in the setup illustrated in figure 3.6 reverberated by the room and recorded by the microphone. In the second experiment the speaker and microphone were placed in an anechoic box padded on the interior and exterior by foam. Their separation was 15cm. The speech database was once again played and recorded in this box. The objective here was to duplicate the effect of playing the speech using the same speaker and recording it using the same microphone but not to allow it to be reverberated by the room since it was played within an anechoic box. It is crucial that this process of transducing the non-

reverberant speech using the same speaker and microphone take place, as the speaker will change the speech. If training was performed using speech that is not played using the same speaker as that in the reverberant room and testing is performed using speech played using the speaker, a degradation will occur in the classification performance that is not due to the reverberation but rather is due to the experimental setup. This is not desirable as only the effect of the reverberation should be measured. An illustration of the box is shown in figure 3.7.

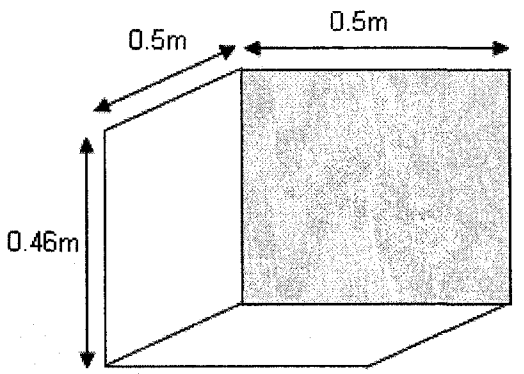

Figure 3.7: Illustration of anechoic box within which speech database was played.

3.4 Simulation of Reverberant Room

The image method [56] for simulating small room acoustics was used to simulate the effects of reverberation in a small rectangular room. The simulation program takes as its input four sets of values or dimensions. The first set is the dimensions of the room, length, width and height, the second set is the location of the speaker in the room, the third is the location of the receiver or microphone and the last is a set of 6 reflection coefficients, 1 for each surface in the room (4 walls, floor

and ceiling). The reflection coefficients specify the ratio of the reflection to absorption of a sound wave incident on each surface.

Multiple paths exist between the speaker and microphone. There is the direct path between the speaker and microphone as well as the path of the signal traveling from the speaker then reaching the microphone after having been reflected by a wall. The latter is a first order reflection. Multiple order reflections are considered i.e. signals that are reflected off walls can be reflected again before they arrive at the speaker. Each time a sound wave is reflected off a wall its amplitude decreases accorded to the reflection coefficient of that wall.

The remainder of this section describes how the image method is used to simulate a reverberant room.

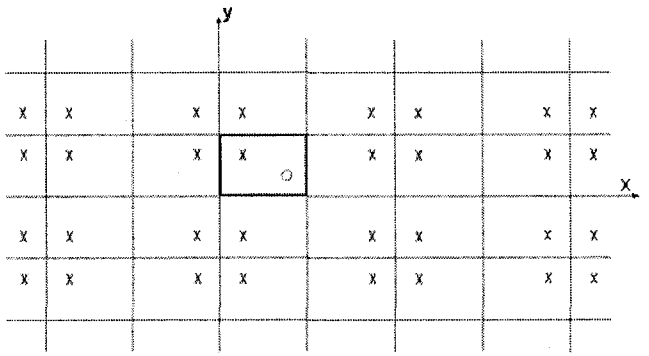

Figure 3.8: Spatial arrangement of image sources [59]

In the center of the illustration is a box in bold print that represents the room. Within the box is an x that designates the position of the sound source; a circle designates the position of the receiver. The boxes that surround the bold print box

contain x's that are also point sources. These simulate the contribution of reflected waves. The model computes the impulse response given the room dimensions and microphone and speaker locations. The simulation software creates a virtual arrangement of speakers outside of the room where the real speaker and microphone are located and simulates an impulse traveling from each of these sources towards the receiver. It attenuates the traveling impulses according to the distance they must travel from their respective virtual source to the microphone. Since each virtual source produces a traveling impulse that simulates the real impulse after it is reflected off walls, the software multiplies the amplitude of the traveling virtual impulses by a reflection coefficient each time it is reflected off a wall.

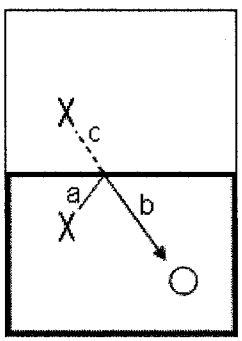

Figure 3.9: Point source acting as the source of a reflected wave

Figure 3.9 illustrates how the simulation software accounts for the contribution of an impulse that is reflected once. In reality the impulse in the illustration travels from the speaker toward the adjacent wall, this path is indicated by the line segment a, and is reflected by that wall towards the microphone, this path is illustrated by the line segment b. The simulation software must determine the distance traveled by this

wave in order to attenuate it correctly. The length of the path illustrated by the solid line is equal to the length of a + b, this is the true distance traveled by the reflected impulse. The position of the x above the bold box (virtual source) is chosen such that c is equal to a. Determining the total distance traveled by the reflected wave is performed by computing the length of c + b which can be determined if the position of the microphone and the position of the x outside the bold box is known.

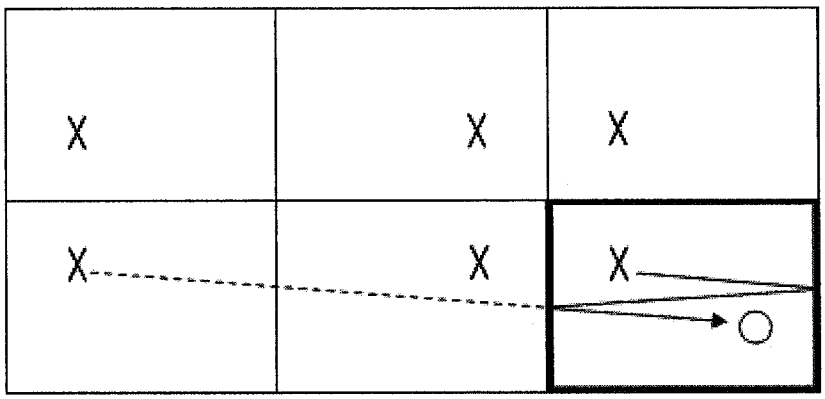

Figure 3.10: An example of multiple reflections.

Figure 3.10 illustrates how the distance traveled by an impulse that is reflected twice is determined. The number of times the impulse is reflected is equal to the number of times the impulse from the virtual source passes through a wall (either virtual or real) in figure 3.10 before it reaches the microphone. As can be seen the dashed line passes through a virtual wall then passes through a real wall (bold box) when it ceases to be a dashed line. The solid line path within the room is the path whose distance is being calculated. 2 reflections occur within the bold box. The distance traveled by the impulse arriving from the simulated source to the receiver

and the distance traveled by the true source to the receiver after having being reflected are the same.

3.5 Reverberation Specifications

Five different reverberant room configurations were used; they are referred to as reverb1 through reverb5. The amount of reverberation increased from reverb1 to reverb4 as is evidenced by the consistent decrease in recognition performance when testing is performed on reverb1 through reverb4. This is of course when training is performed on non-reverberant speech. Under the same training conditions just mentioned, testing on reverb4 and reverb5 leads to similar recognition results. The amount of reverberation is increased by changing the size of the room, the speaker to microphone separation and wall and floor reflection coefficients so that the result is increasing reverberation time (RT60) and decreased recognition performance. Table 3.1 and table 3.2 contain the specifications of the rooms for which the impulse responses were simulated.

Table 3.1: Reverberation Specifications

Specification	Reverb1	Reberb2	Reverb3
Room Dimensions	3.6x4.2x3 m	3x6x2.5 m	7x5x3.5 m
Room volume	45.4 m^3	45 m^3	122.5 m^3
Source position	1.2,2.1,1.2 m	2.2,3.1,1.2 m	3.5,2.5,1.5 m
Receiver position	1.7,2.4,0.73 m	2.45,3.0,0.73 m	3.2,2.98,1.19 m
Source- Receiver separation	0.75 m	0.54 m	0.65m
Wall reflection coeffs.	0.9	0.93	0.93
Ceiling & floor reflection coeffs.	0.7	0.8	0.85
RT60	0.51s	1.08s	1.15s

Table 3.2: Reverberation Specifications continued

Specification	Reberb4	Reverb5
Room Dimensions	8x6x4 m	6x8x4
Room volume	192 m^3	192 m^3
Source position	4,3,1.4m	4,3,1.4m
Receiver position	3.28,3,1.2m	3.28,3,1.2m
Source- Receiver separation	0.75m	0.75m
Wall reflection coeffs.	0.93	0.935
Ceiling & floor reflection coeffs.	0.90	0.90
RT60	1.37s	1.50s

The impulse response for each configuration is shown in figures 3.16 to 3.20. The energy decay curve for each impulse response was computed using the formula from [67]:

$$EDC(t) = \int^{\infty} h^2(\tau) d\tau \tag{3.1}$$

The values for RT60 (reverberation time) shown in table 3.1 are determined as the time required for the energy decay curve to decay 60dB. This value is obtained by performing the integration in equation 3.1 on the impulse response. The energy in the decay curve is computed for each value of t. The ratio of these values to the energy for t=0 is computed on a dB scale. The value of t yielding a ratio at 60dB indicates the RT60. The energy decay curve yields information about the significance of the samples in the tail of the impulse response as well it indicates the RT60 the reverberation time. Table 3.2 indicates that the values for RT60 are between 1 and 2 seconds for reverb2 to reverb5 and less than 1 second for reverb1. For this reason the impulse response for reverb1 was simulated for an impulse response length of 1 second. The other impulse responses, reverb2 to reverb5, were simulated for 2 seconds. Figure 3.11 to figure 3.15 illustrate the first 2000 samples of each of the impulse responses; all impulse responses were simulated using a sampling frequency of 8 KHz. Figure 3.16 to figure 3.17 illustrate the energy decay curves for the reverb1 and reverb2 configurations.

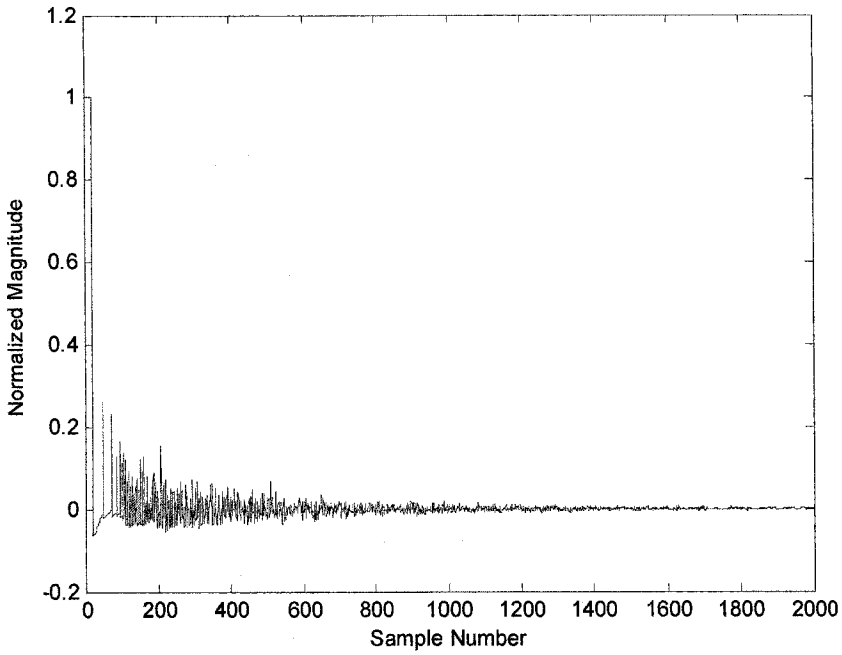

Figure 3.11: Reverberated room impulse response for configuration reverb1

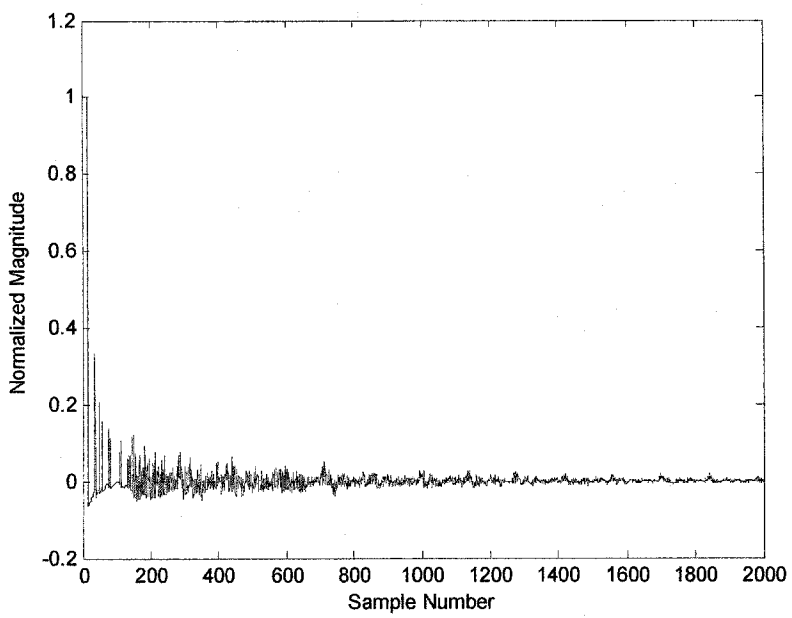

Figure 3.12: Reverberated room impulse response for configuration reverb2

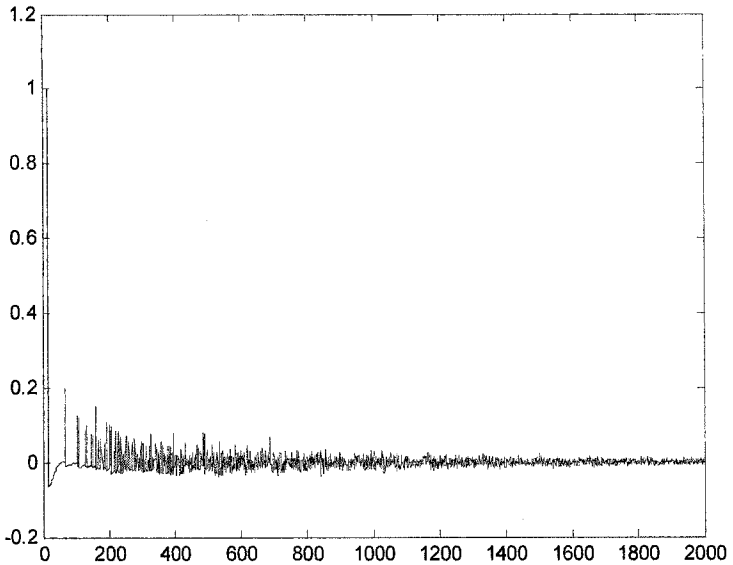

Figure 3.13: Reverberated room impulse response for configuration reverb3

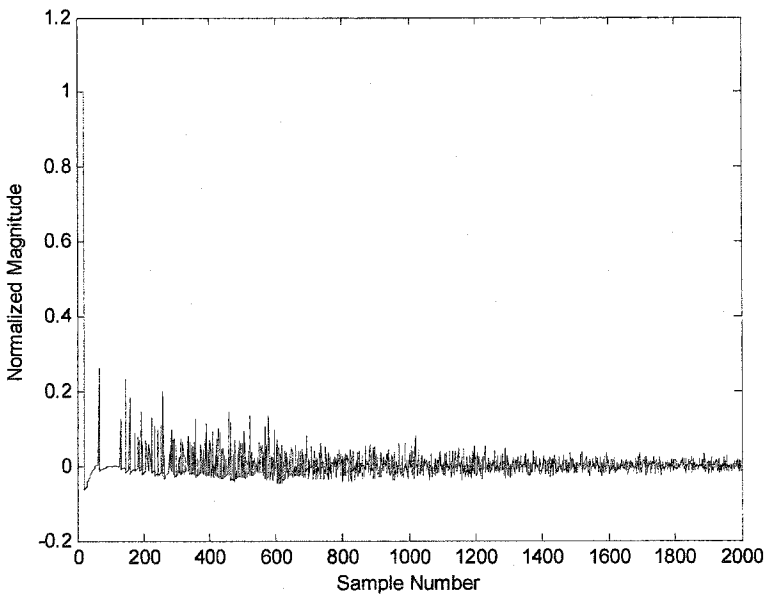

Figure 3.14: Reverberated room impulse response for configuration reverb4

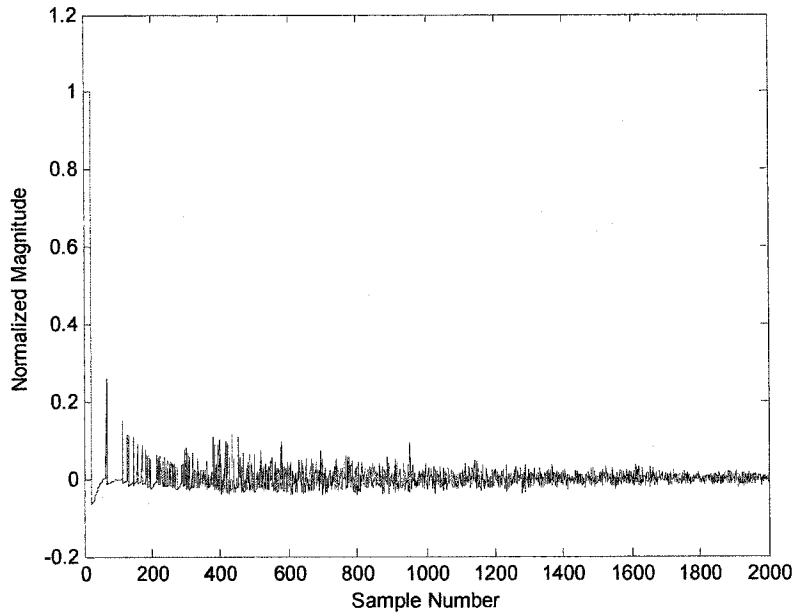

Figure 3.15: Reverberated room impulse response for configuration reverb5

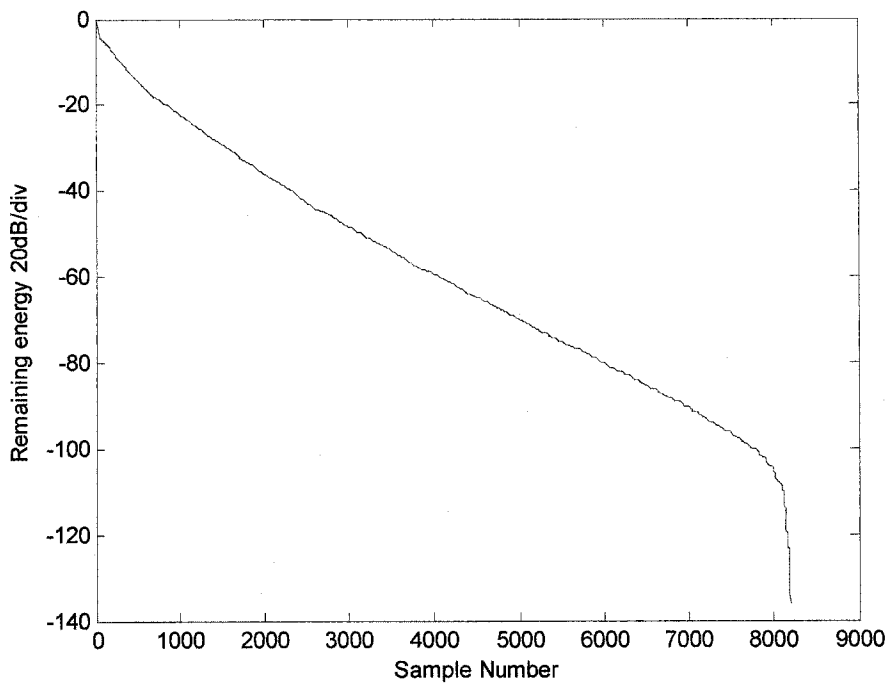

Figure 3.16: Energy decay curve for reverb1

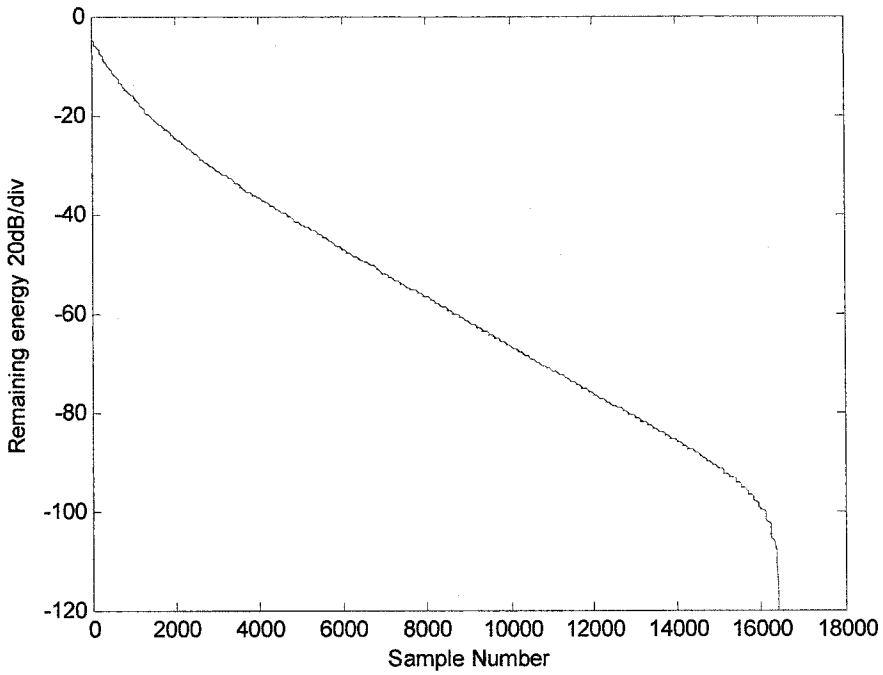

Figure 3.17: Energy decay curve for reverb2

3.6 Data Processing Computers

A total of 14 desktop PCs were used to perform the processing required in this thesis.

CPU: Pentium IV 2.60 GHz (all PCs)

RAM: 1 GB (5 of the PCs), 500MB (remaining 9 PCs)

OS: Windows XP

Simulation environment: Matlab 6.5

3.7 Parameter Selection for Front End Processing

For the front end processing described in section 2.2, pre-emphasis factor, the frame length, percentage of frame overlap and VAD energy threshold were selected.

The pre-emphasis factor α referred to in section 2.2.1 was set to 0.95 [35], [52], [61]. The frame overlap percentage was set to 50% [17], [35], [50]. An energy threshold of 0.5% of the average frame energy was used in the VAD for all speech files except the ones recorded in the reverberant room where due to background noise, the energy threshold was raised to 2% of the average frame energy. The energy thresholds for the VAD were computed empirically. The thresholds where set such that no non-speech information will be retained from the speech files therefore removing all the silence periods. The frame length used is 20ms [17], [35], [40]. The size of the filter bank used is 19 Filters [9]. The filters are placed across the 300-3400 Hz frequency band.

Chapter Four

Comparison of Speaker Recognition Methods

The following chapter contains the results of simulations where the performance of speaker recognition methods were compared without the presence of reverb in either the training or test speech. The speaker recognition methods used were those described in chapter 2.3. The purpose of this chapter is to determine how well each method works in comparison to the others. Each method was compared based on its performance in closed set speaker recognition and speaker verification trials. Each speaker recognition method was implemented using each feature vector type. The recognition methods used were the GMM, SM, DS, AR-ITAKURA, AR-AGS, and MLP. These were the methods outlined in section 2.3. The feature vectors used were LPCC, MFCC, LPCC+Δ, MFCC+Δ, outlined in section 2.2.

4.1 Baseline Method Comparison

The performance of the methods was compared when training was performed on non-reverberant speech and testing was performed on non-reverberant speech. For each feature vector trials were performed where delta-cepstral features were used and trials were performed where delta cepstral coefficients were not used. The exception is the MLP where delta-cepstral coefficients were not used. Figure 4.1 will compare the closed set speaker recognition performance when each type of feature vector was used. Figure 4.2 compares the verification performance of the methods. The same results displayed in figure 4.1 are transformed to an error rate simply by subtracting the scores from 100. These are shown in figure 4.3. Training was performed using the first 3 sessions for each speaker from KING database. Testing was performed using the remaining 7 sessions. Each trial was performed separately using each of the 7 test sessions separately.

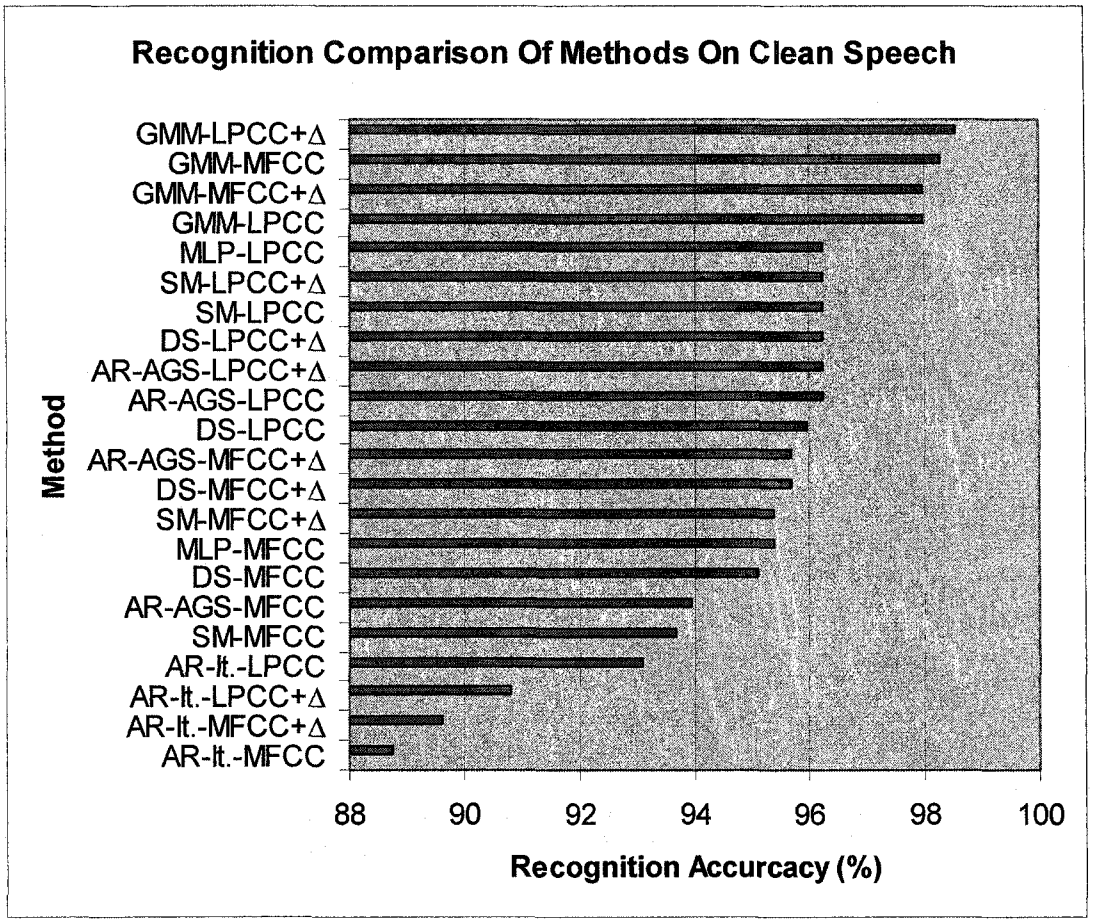

Figure 4.1: Recognition performance of different methods.

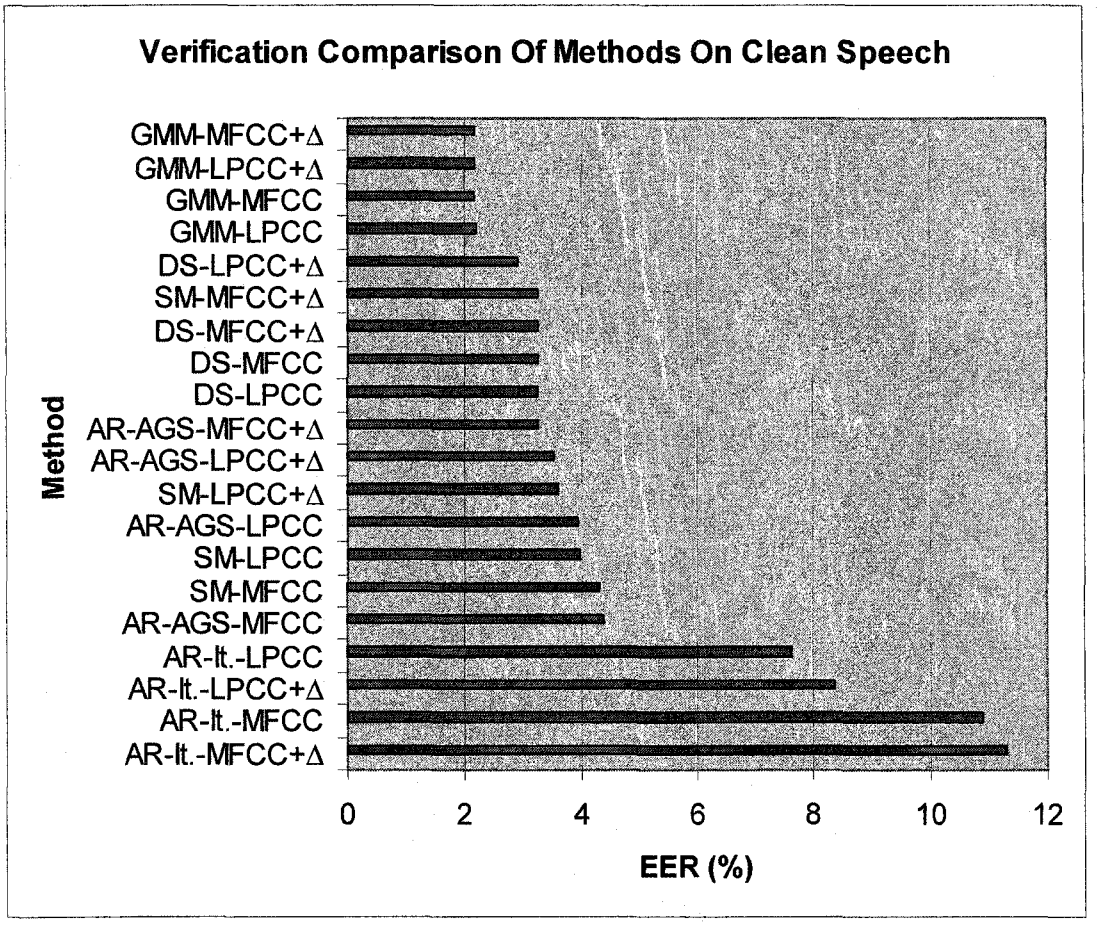

Figure 4.2: Verification performance of different methods

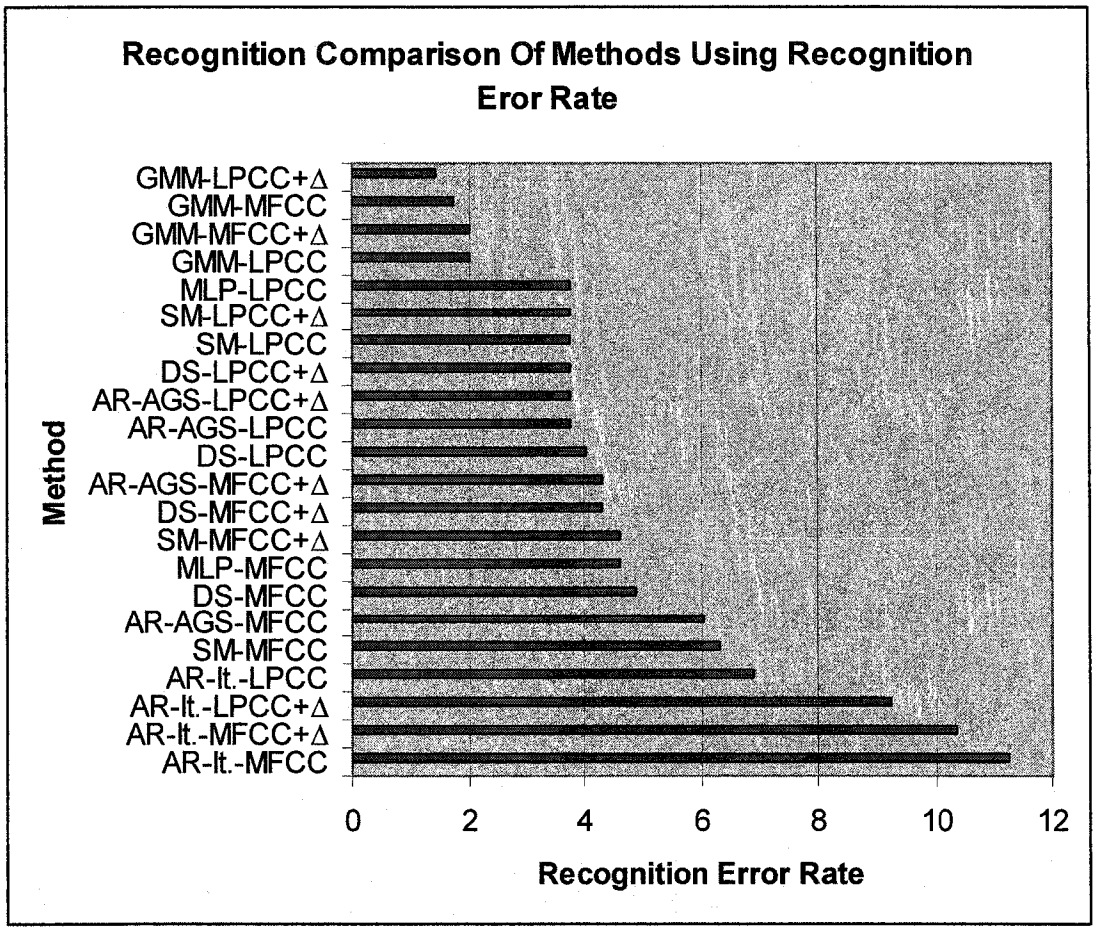

Figure 4.3: Recognition performance of methods using recognition error rate.

From the results in figure 4.1 and figure 4.2 it is clear that the GMM outperformed all the methods in both verification and recognition trials. For verification and recognition AR-Itakura method produced the worst results. The MLP produces good results for recognition. In fact it is the second best recognition method when LPCC features are used. Both the covariance-based methods, SM and DS, produce similar results for both recognition and verification. Also their performance was similar to the AR-AGS method. Without the use of delta cepstral

coefficients, LPCC vectors tend to outperform MFCC vectors, that was the case for all methods except the GMM and MLP. Delta cepstral coefficients improved performance slightly for most methods with the exception of the GMM where little improvement was realized. For AR-Itakura using LPCC vectors, delta cepstral coefficients degrade performance by 2%.

4.2 Comparison of Systems

Closed set speaker recognition performance is judged using a single metric that is recognition accuracy. This is simply the number of times the identity of the unknown speaker was determined by the recognition system to be the correct speaker divided by the total number of tests:

$$recognition\ accuracy = \frac{\#\ correct\ trials}{total\ \#\ of\ trials} x100\% \tag{4.1}$$

The total number of trials used for speaker recognition is 347. This results from the fact that there are a total of 51 speakers and 7 test utterances for each speaker with the exception of 2 speakers that only have 2 test utterances.

For speaker verification, testing involves evaluating an utterance of unknown speech against the model of the claimant speaker. The result for a method such as the GMM is a probability that the claimant uttered the test speech. For all other recognition methods the result is a distance between the claimant speaker model and the test speech. This distance or probability can then be compared to a threshold and

the speaker can be accepted or rejected. A process known as score normalization takes place before the comparison to the threshold. Normalization is a process that allows for the use of a single verification threshold for all tests. Test normalization [9] is the score normalization method that is used in this thesis because of its ease of implementation and effectiveness. The details of test normalized will be explained in section 4.3.

One of the metrics used to judge verification accuracy is equal error rate, EER. It is computed using the probability of false alarm and the probability of misses. The probability of false alarm is the probability that an imposter will be accepted. The probability of a miss is the probability that a true speaker will be rejected. Both of these probabilities depend on the threshold that is used. EER is determined by varying the threshold until the probability of a false alarm is equal to the probability of a miss. The following section describes how speaker verification trials are carried out.

4.3 Speaker Verification with Test Normalization

In order to perform test normalization a set of speaker models were required for use as normalization models. The process of speaker verification using Test normalization will now be outlined. The KING database contains 51 speakers; each speaker has a number of test utterances. All test utterances were used in a single verification trial. The following process was carried out for each test utterance.

1) Each test utterance was scored against all 51-speaker models. The result was 51 distances or probabilities. If the speaker recognition method used was one of the SM, DS, AR-Itakura or AGS then the negative of the distances was calculated. These will be referred to as scores in the remainder of the explanation. This is because a distance is a measure of unlikeness between the test utterance and the model. The remainder of this process requires a measure of likeness. If the speaker recognition method used was the GMM, the probabilities remained unchanged, as they were a measure of likeness between the test utterance and the model. These will also be referred to as scores in the remainder of the explanation.
2) The database was divided into 2 parts. The first part contains 41 speakers and was used to generate true target and imposter scores. The second part contains 10 speakers and was used to generate normalization scores.
3) The following process was performed for each test utterance. Using the first part of the database, consisting of 41 speakers, each score for a speaker's test segment relative to his own model is a true target score. Each score for a speaker's test segment on the other 40 models in the set is an imposter score. These two types of scores are not normalized. They are the un-normalized true target and imposter scores. The normalization scores were obtained from the second part of the database consisting of 10 speakers. The score for each speaker speaker's utterance in the 41 speaker set against the 10 models in the 10 speaker set are the

normalization scores. The mean and standard deviation of the normalization scores was computed. The un-normalized true target scores from the test utterance and the un-normalized imposter scores from the test-utterance were normalized using the following equation [9]:

$$S^{Norm} = \frac{S^{un-norm} - u}{\sigma} \tag{4.2}$$

where $S^{un-norm}$ is an un-normalized imposter or un-normalized true target score and μ and σ are the mean and standard deviation of the normalization scores.

4) All the normalized true target scores were agglomerated into one set that will be referred to as the true target scores. All normalized imposter scores were agglomerated into a set that will be referred to as the imposter scores.

5) Each of the scores in the imposter set and true target set were tested successively using the following test: A score was selected and used as the verification threshold. This means that the true target and imposter scores were compared to this threshold. A proportion of the true target scores was be greater than this threshold and a proportion was less. If 95% of the true target scores were greater than the threshold then the false rejection rate is 5% because 5% of the true target scores were rejected even though they should have be accepted. The imposter scores were then compared to the threshold. A proportion of the imposter scores were be greater than this threshold and a proportion were less. If the proportion that was less is 90% than the false acceptance rate is 10% because 10% of the

imposter scores were accepted but should have been rejected. The false rejection and false acceptance rate for each of the scores form a pair.

6) After step 5 was performed for all the scores, the result was a set of pairs. The pair where the difference between the false acceptance rate and false rejection rate was minimum is the EER pair. The average of the false acceptance rate and false rejection rate in this pair is the EER.

7) Steps two to six were performed with the imposter and true target scores coming from the first 41 speakers, and the normalization scores coming from the last 10 speakers. Steps two to six were repeated except that the imposter and true target scores come from the last 41 speakers and the normalization scores came from the first 10 speakers. A new EER was computed. The larger of the EER value from these trials was retained as a metric for later comparison.

4.4 Determination of Model Parameters

Six methods for closed set speaker recognition and verification were introduced, GMM, AR-ITAKURA, AR-AGS, SM, DS and MLP. These methods were the methods outlined in section 2.3. The MLP is exceptional because it can only be used for closed set speaker recognition. For each combination of speaker recognition method and feature vector used, the feature vector length that yields the best performance were determined before comparison could take place. This issue is elaborated upon in the next section.

For the GMM the number of Gaussians no use is usually between 32 and 128 [14]. Preliminary tests have revealed that there was very little difference between the performances of GMMs of these sizes. The size of the GMM was set to 64 Gaussians as this will offer significant reduction in simulation time relative to a GMM of size 128 also GMM sizes are almost always a power of two and 64 is the power of two between 32 and 128 so it is a logical 'middle of the acceptable range' value.

4.4.1 GMM Variance Limits

In [17] and [33] the issue was raised that some of the clusters in a GMM may be trained with a relatively small amount of data. This may manifest itself when the number of Gaussians in the mixture is large relative to the length of the training speech or if the data is noisy. This may result in the under-estimation of the variance of some of the clusters. The recommended remedy is to use a minimum variance limit. This is an empirically determined constant that depends on the speech database, the feature set used and the number of Gaussians in the model. The variance limit is compared to each element of the variance vector, in the case that an element is less than the minimum limit; the minimum limit will replace the element. Variance limiting is applied after each iteration of the EM algorithm [17]. This method of variance limiting uses a single variance limit all elements.

In general the variances of elements in MFCC and LPCC vectors decrease as the element index increases. For this reason it seems counter intuitive to use the same

variance limit for all elements as was done in [17]. Another method by which to perform variance limiting is to take the average of all the variance vectors for a single speaker. This average variance vector can then be multiplied with a constant between 0.1 and 1. The elements in this scaled variance vector can be used as minimum variance limits. Each element in the scaled vector will be a variance limit for the corresponding element in variance vectors generated during training.

4.4.2 Selection of Feature Vector Lengths

In the literature different feature vector lengths have been used. For GMMs, filter bank based cepstral vectors of length 12 [9], 14 [68] 16 [69] 19 [70], have been used. For AR-Vector methods and covariance-based methods the length of the MFCC cepstral vectors vary between 15 and 18 coefficients. LPCC vectors lengths of 12, 16 and 20 coefficients have been used for covariance-based methods [54], [55]. AR-Vector models using LPCC coefficients have used vectors of length 16 and 20 [35], [34], [39]. In order to compare the performance of the different methods it was necessary to determine the vector length that resulted in the best performance for each method. The size of the filter bank to use for MFCC vectors was set at 19-filters [9] as it was found that there was no significant affect on performance if the filter bank length was increased or decreased. MFCC vector lengths were varied between 10 and 18 coefficients and LPCC vector lengths were varied between 14 and 23 coefficients. All test segments consisted of the first 30 seconds of speech files extracted from the

4th through 7th sessions of the KING database. The first 3 sessions were retained for training.

The graphs showing the performance of each method as the feature vector lengths are varied are shown in appendix A. The best performing feature vector length for each method was selected from the results in appendix A and used for the comparison of method.

The MLP is a special case. First of all no verification trials are performed using the MLP because the network is trained with knowledge of the imposters and therefore cannot be used for verification. The time required to train an MLP is large. For this reason, the same networks were trained for use when tests were performed on reverberant and non-reverberant speech. It was found that when LPCC vectors where used in reverberant and non-reverberant trials, that performance improved with increasing vector length under both conditions. When MFCC vectors are used in reverberant trials, performance improved with increasing vector length. In non-reverberant trials performance was near its maximum when the longest MFCC feature vector was used. In order to give the LPCC and MFCC MLP networks a fair comparison under reverberation the same feature vector length was used. The feature vector length used was 18 for both MFCC and LPCC. Increasing the length of the MFCC or LPCC feature vector from 18 to 23 did not considerably improve recognition performance in reverberant or clean speech trials. The feature vector

length used for the comparison of the methods on non-reverberant speech are shown in table 4.1.

Table 4.1: Selected feature vector lengths

Method Type	LPCC Vector Length	MFCC Vector Length
GMM	16	15
AR-ITAKURA	17	15
AR-AGS	19	13
SM	21	13
DS	19	15
MLP	18	18

4.4.3 Determination of GMM Variance Limits

Scaled variance limits were used for both MFCC and LPCC vectors. It was found that this method gave the best results. The variance multipliers used for the GMM are shown in table 4.2.

Table 4.2: GMM best minimum variance limit

Feature Vector	Variance limit multiplier
MFCC	0.8
LPCC	0.8

Chapter Five

Comparison of Speaker Recognition Methods Under Reverberation

The following section compares the performance of each of the methods in the presence of reverberation in the test speech and not in the training speech.

5.1 Performance Comparison of Features under Reverberation

The performance of the different features used for each method under reverberant conditions was compared. Training speech was non-reverberant, test speech was reverberant. Each method is looked at independently. For each individual method the performance of each feature vector is plotted against each reverberant condition. In addition, for each method, the average recognition and verification of performance of each feature over the 5 reverberant conditions is plotted.

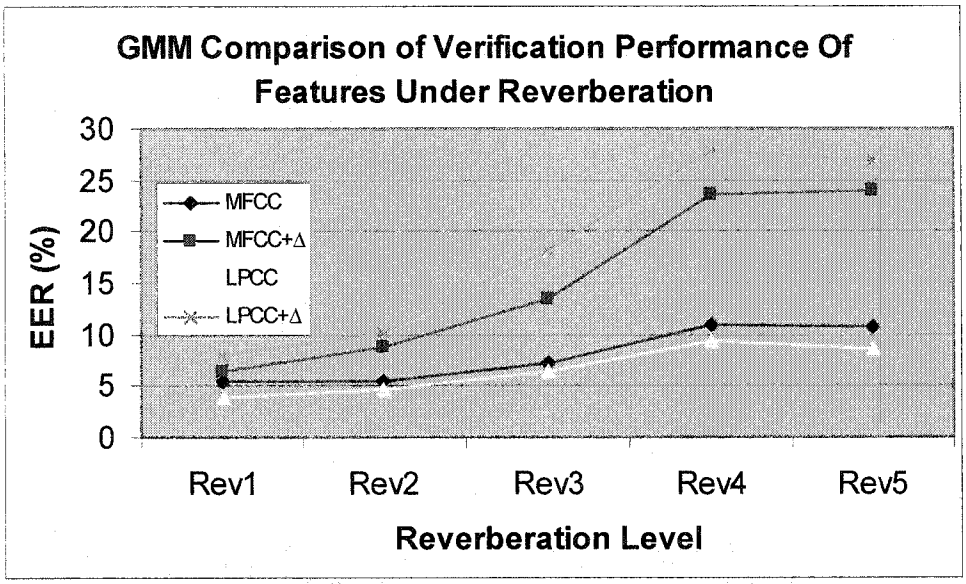

Figure 5.1: Verification performance using different features for GMM under reverberation.

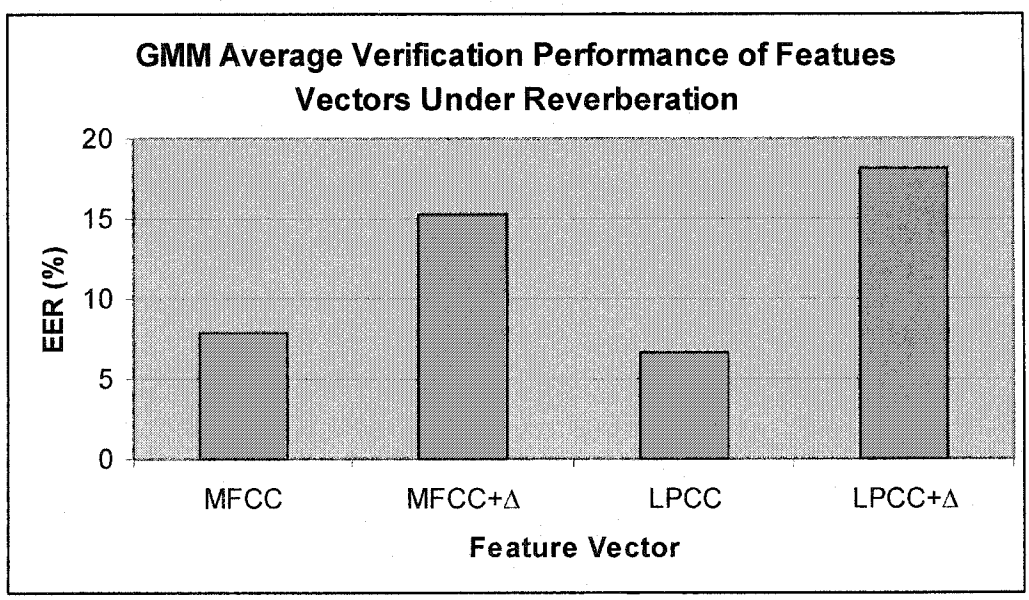

Figure 5.2: Average verification performance using different features for GMM under reverberation.

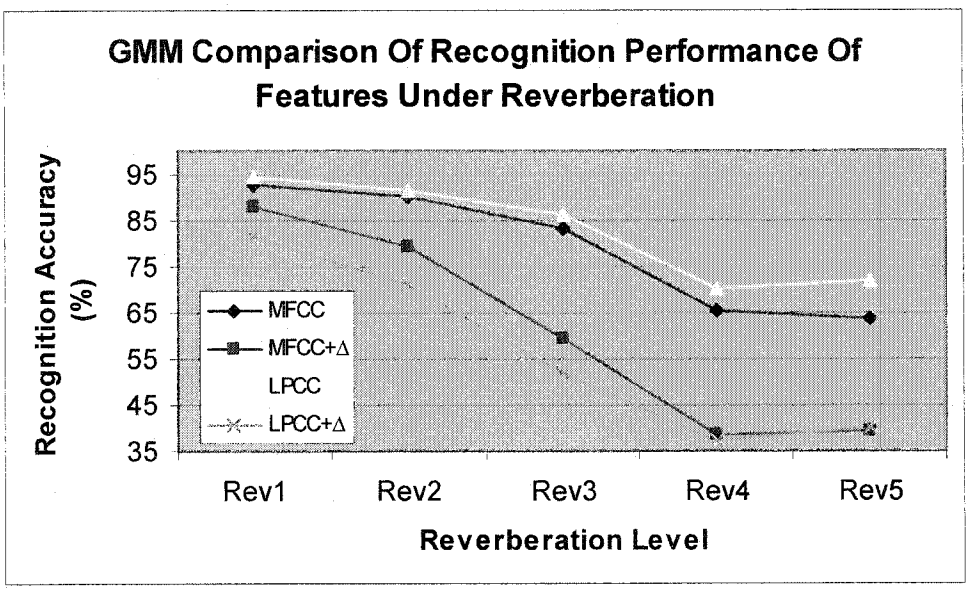

Figure 5.3: Recognition performance using different features for GMM under reverberation.

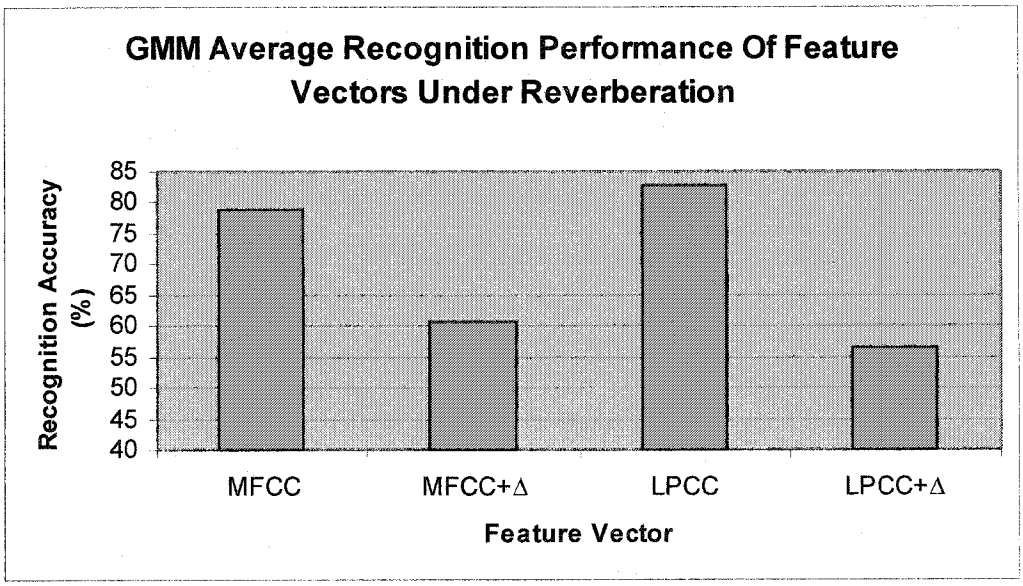

Figure 5.4: Average recognition performance using different features for GMM under reverberation.

With regard to recognition accuracy LPCC and MFCC vectors without delta cepstrum consistently outperformed their delta-appended counterparts. Their

superiority was more acute as the reverberation increases. Even at relatively low levels of reverberation, the disadvantage of using delta-cepstral coefficients when the test speech was reverberant is clear. At higher levels of reverb the spread between the performance of delta and non-delta features increased. LPCC vectors without delta cepstral coefficients were the best performing feature vectors for both recognition and verification. They outperform the MFCC feature vectors. When delta cepstral coefficients were used the MFCC features outperformed the LPCC features.

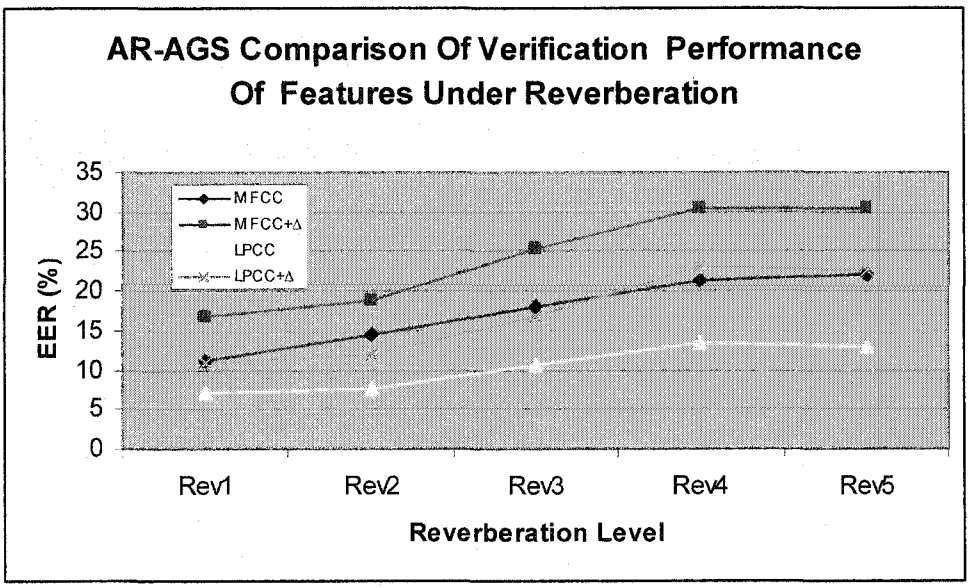

Figure 5.5: Verification performance using different features for AR-AGS under reverberation.

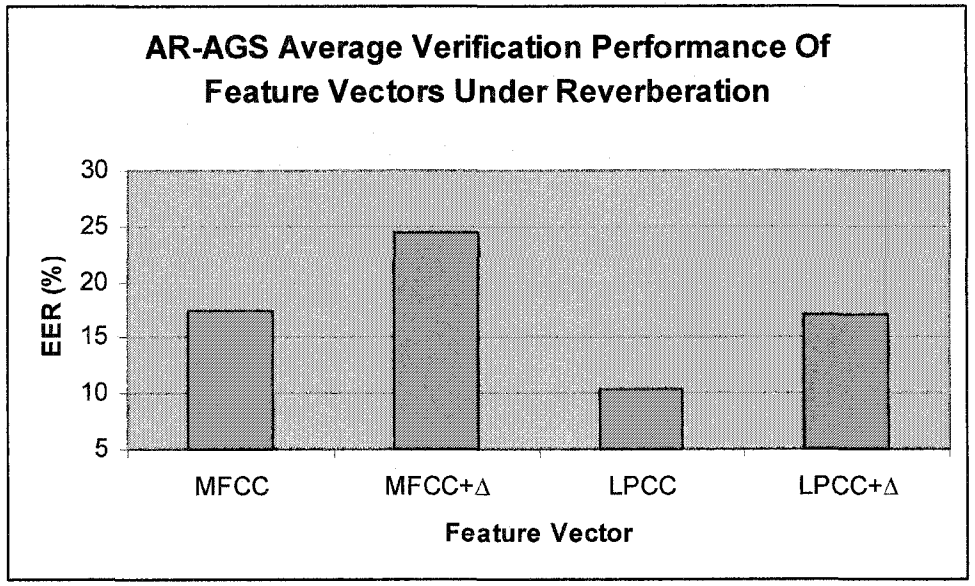

Figure 5.6: Average verification performance using different features for AR-AGS under reverberation.

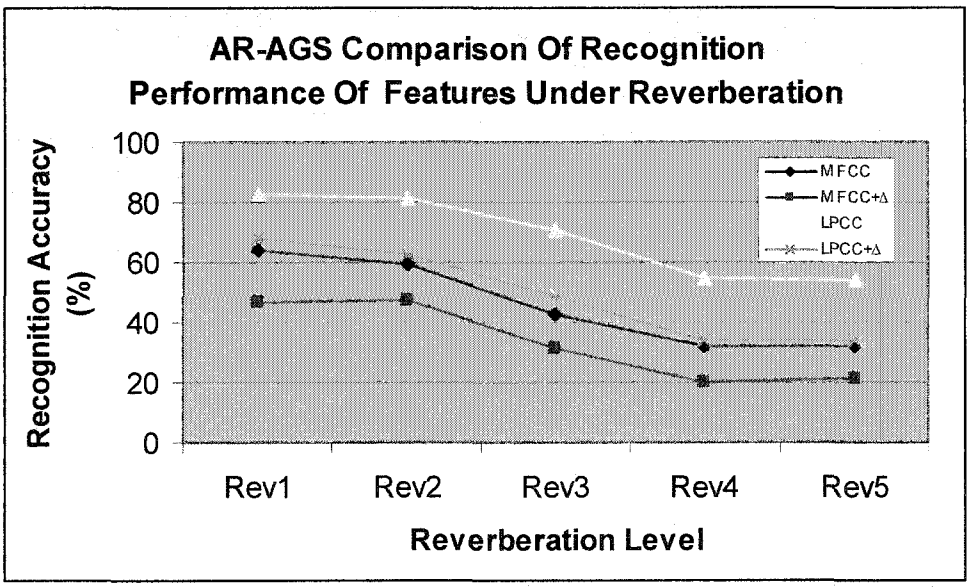

Figure 5.7: Recognition performance using different features for AR-AGS under reverberation.

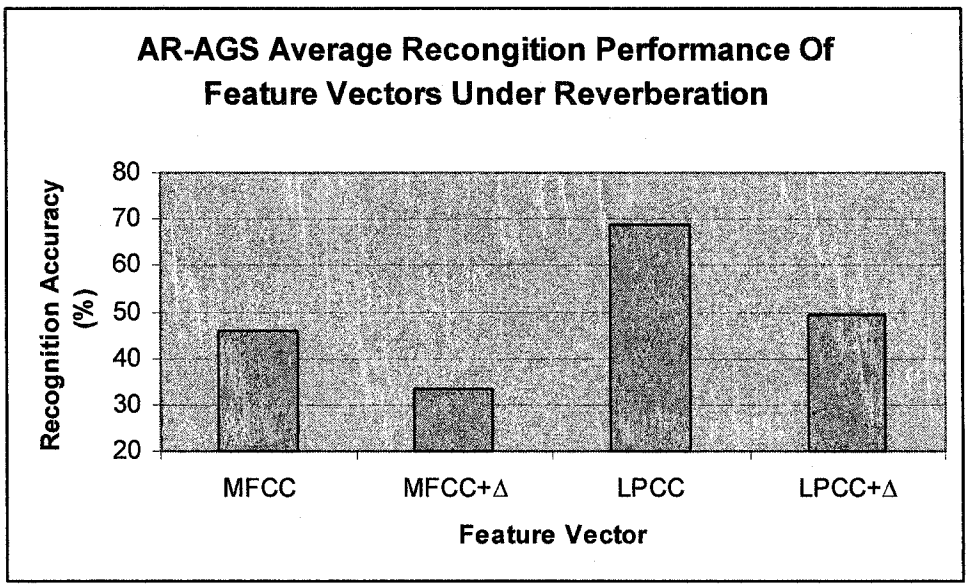

Figure 5.8: Average recognition performance using different features for AR-AGS under reverberation.

With regard to AR-AGS, the given results show clear trends. Using this method, delta cepstral coefficients clearly degraded performance consistently. LPCC vectors outperformed MFCC vectors consistently. Both LPCC parameterizations, with and without delta coefficients outperformed both MFCC vector versions. The best over all performing feature was the LPCC parameterization without the use of delta cepstral coefficients.

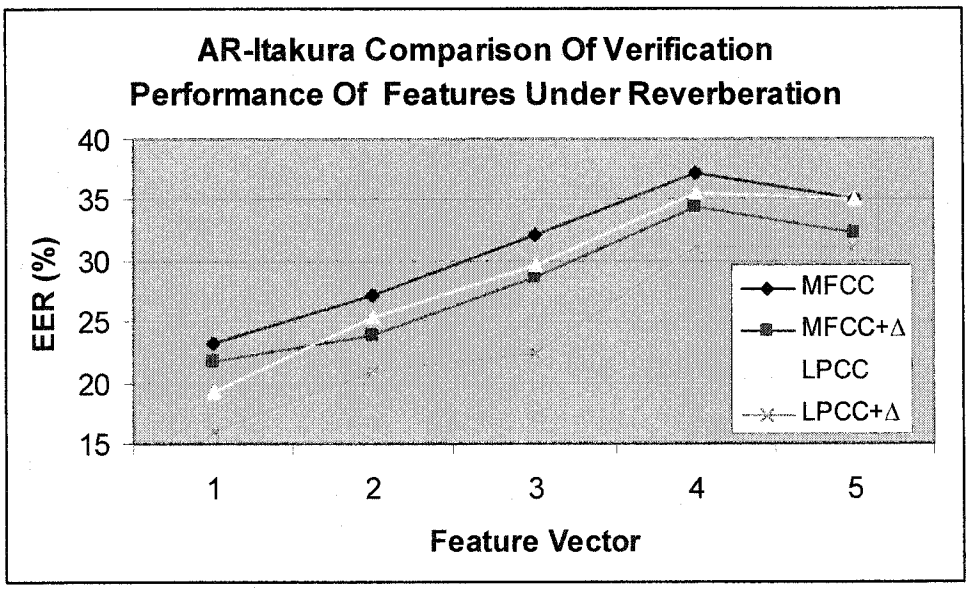

Figure 5.9: Verification performance using different features for AR-Itakura under reverberation.

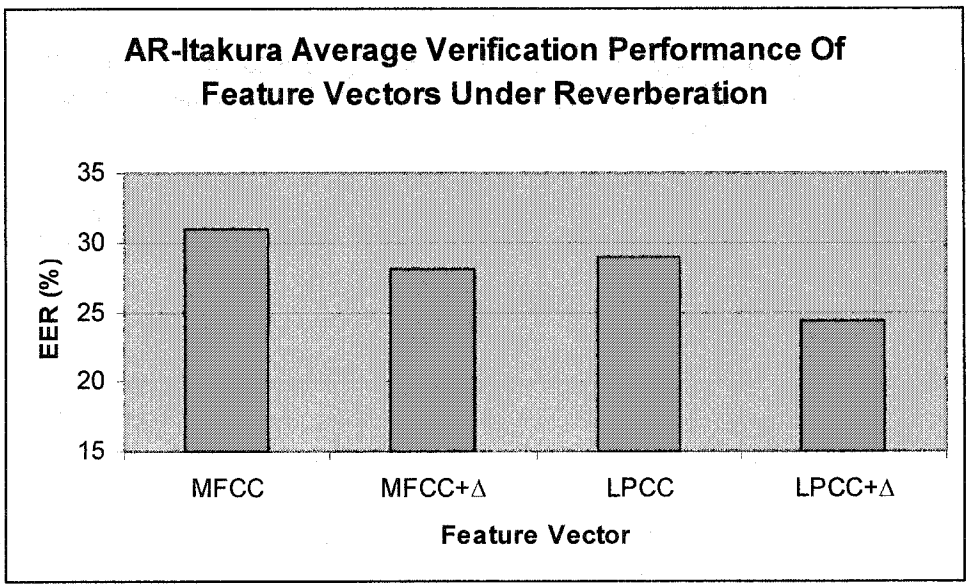

Figure 5.10: Average verification performance using different features for AR-Itakura under reverberation.

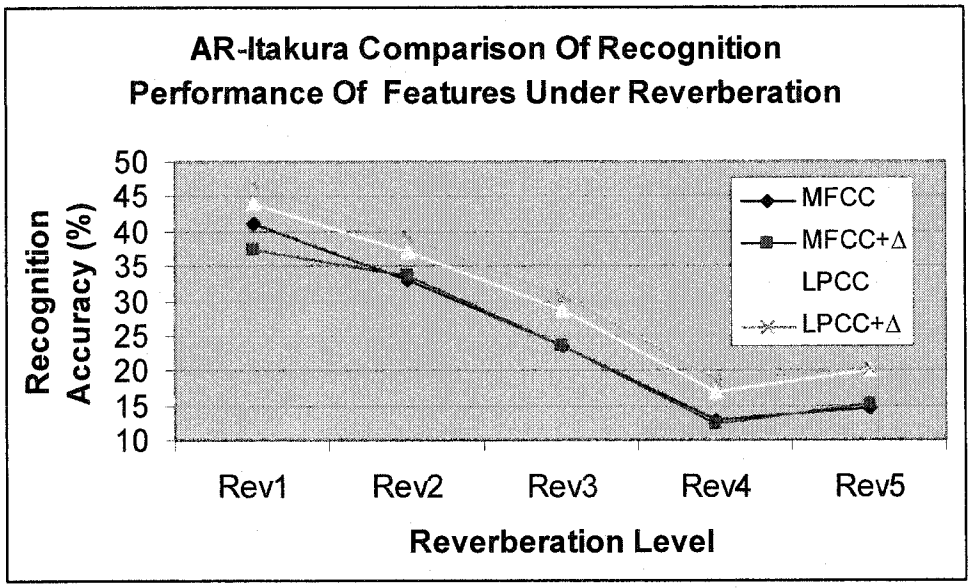

Figure 5.11: Recognition performance using different features for AR-Itakura under reverberation.

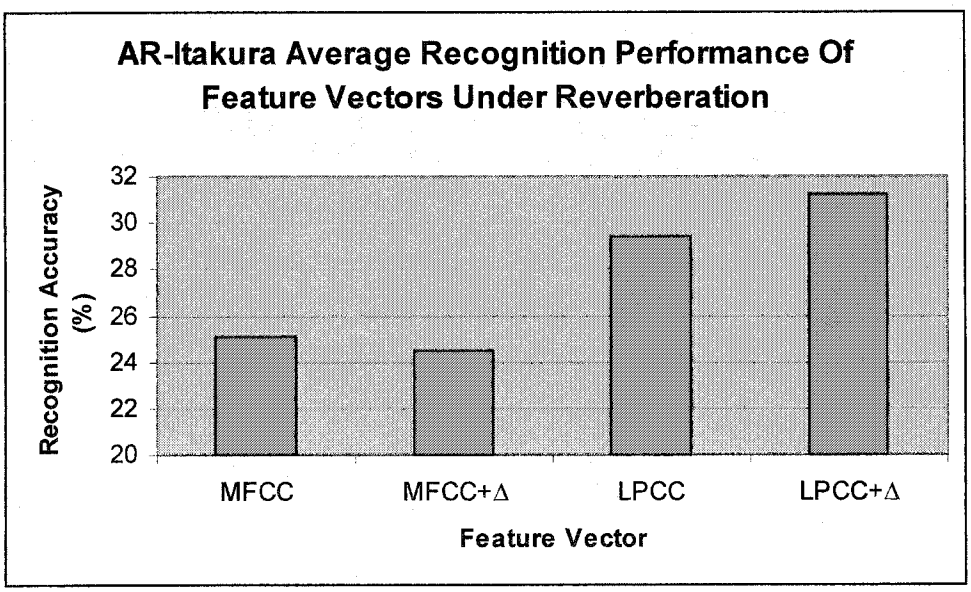

Figure 5.12: Average recognition performance using different features for AR-Itakura under reverberation.

For AR-Itakura the performance was clearly degraded severely by reverberation. This was likely due to the fact that this method looks at the evolution of the spectrum

over a time span of three frames. Within this 40ms period of time there is forward smearing in the spectrum due to reverberation. It is possible that was the cause of the deterioration in performance. A counter argument to this conjecture is the fact that delta cepstral coefficients do not degrade performance and in some cases increase performance for this method. When each feature, LPCC using delta and MFCC using delta are compared to their non-delta counterpart there is an improvement in verification performance for both the LPCC and MFCC vectors. This is against the trend in these results that they degrade performance. For recognition the addition of delta cepstral coefficients for the LPCC vector improved performance as it did for verification. For MFCC vectors, the addition of delta cepstral coefficients decreases recognition performance by 0.6% but improved verification performance by 2.8%. LPCC coefficients appended with delta cepstrum gave the best results consistently both in regard to overall average and when viewed in each reverberant condition. This was true for both recognition and verification.

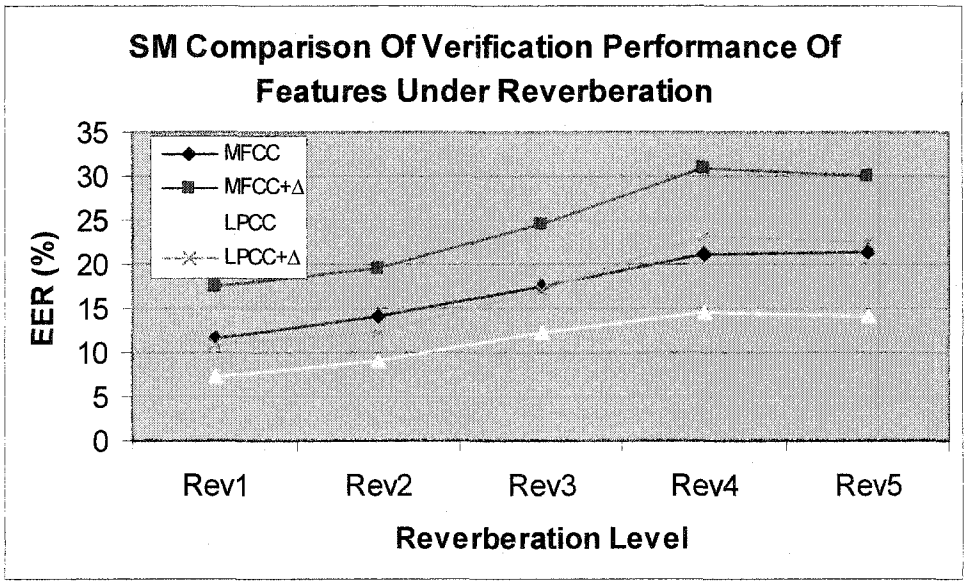

Figure 5.13: Verification performance using different features for SM under reverberation.

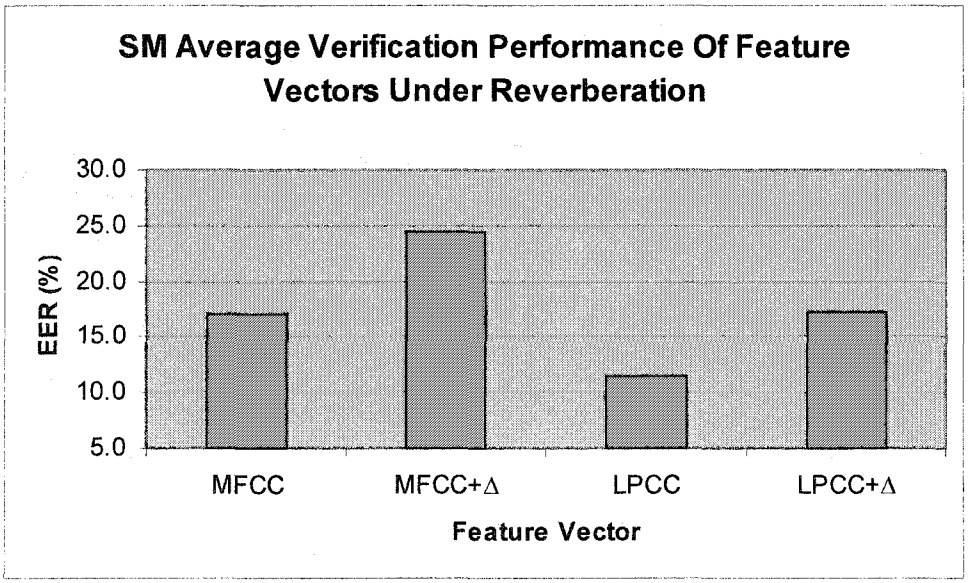

Figure 5.14: Average verification performance using different features for SM under reverberation.

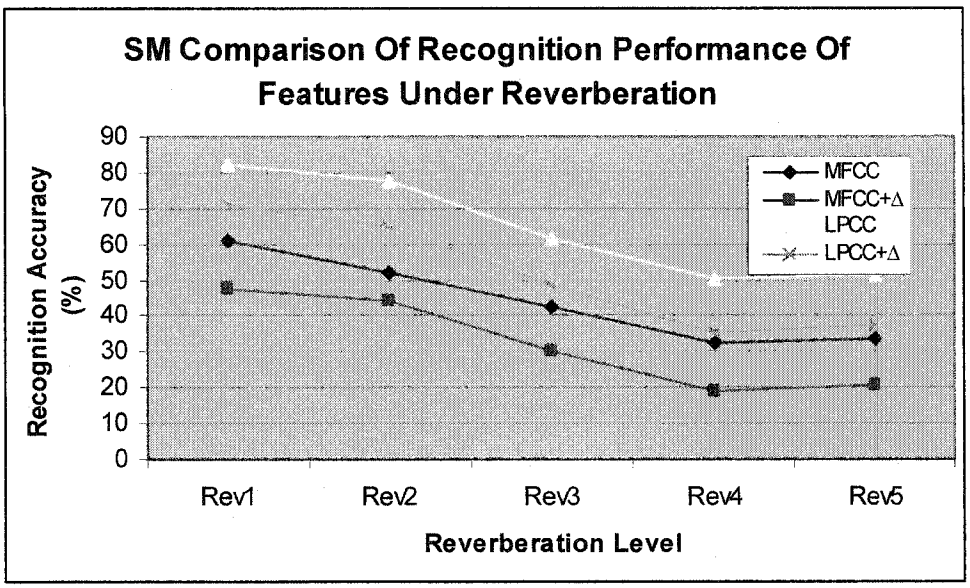

Figure 5.15: Recognition performance for SM using different features under reverberation.

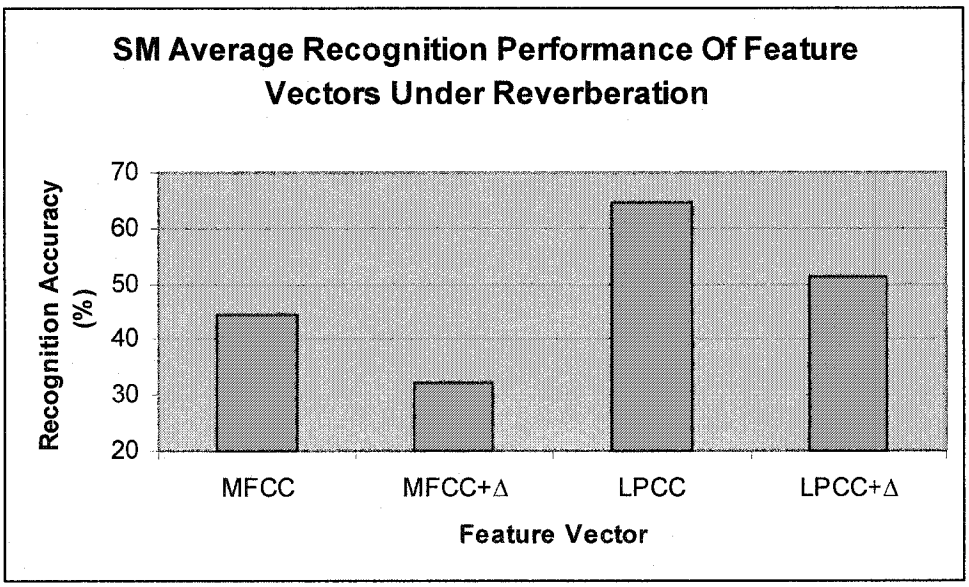

Figure 5.16: Average recognition performance using different features for SM under reverberation.

For SM, LPCC vectors were clearly the best performing feature vector over all conditions. For each feature vector, delta cepstral coefficients degraded performance.

The best performing feature for recognition and verification was the same. The LPCC vector without delta cepstral coefficients were the best and the worst feature vector was the MFCC vector appended with delta cepstral coefficients. For recognition performance LPCC vectors appended with delta cepstral coefficients outperformed MFCC vectors without delta cepstral coefficients. For verification the LPCC vectors using delta cepstral coefficients gave similar performance to the MFCC vectors without delta cepstral coefficients.

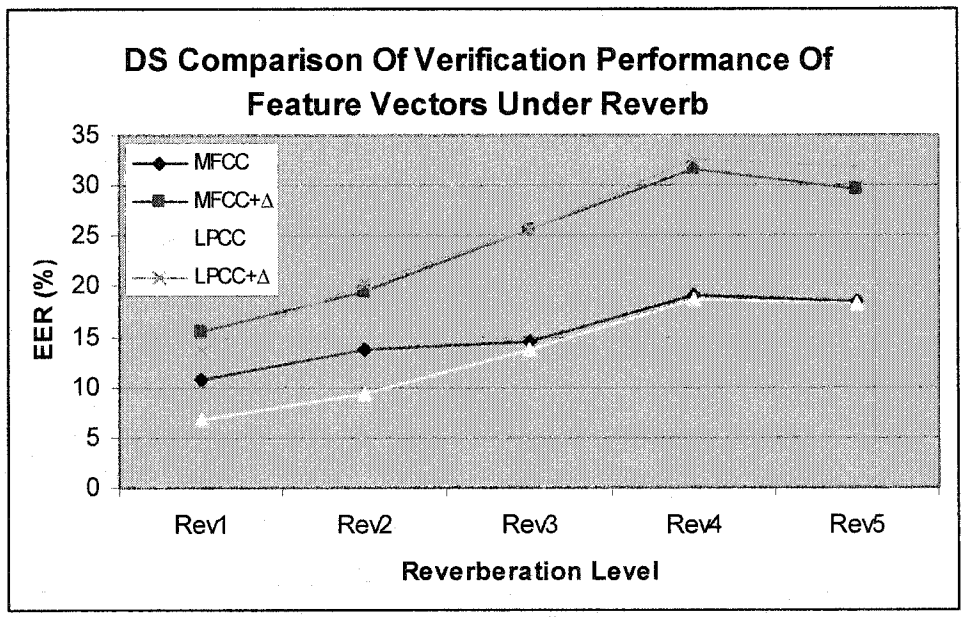

Figure 5.17: Verification performance using different features for DS under reverberation.

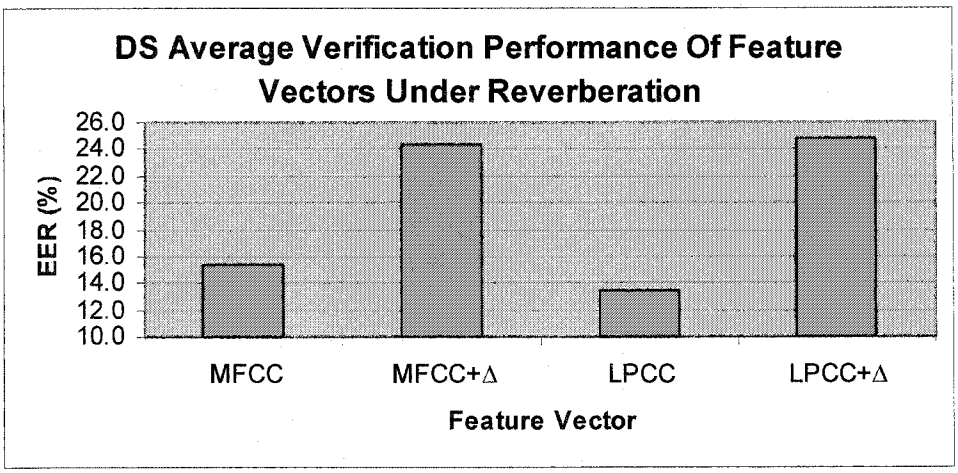

Figure 5.18: Average verification performance using different features for DS under reverberation.

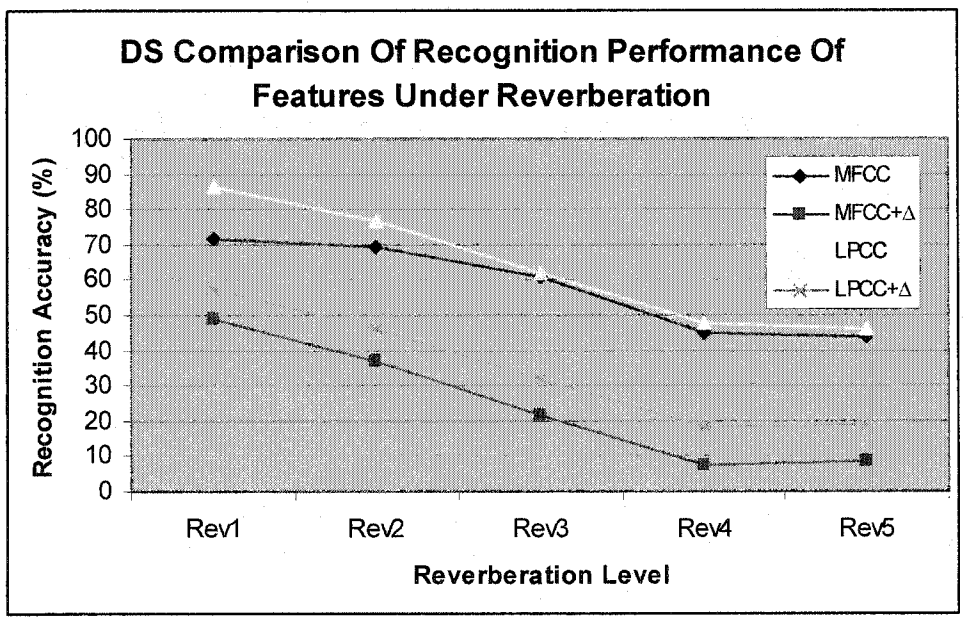

Figure 5.19: Recognition performance using different features for DS under reverberation.

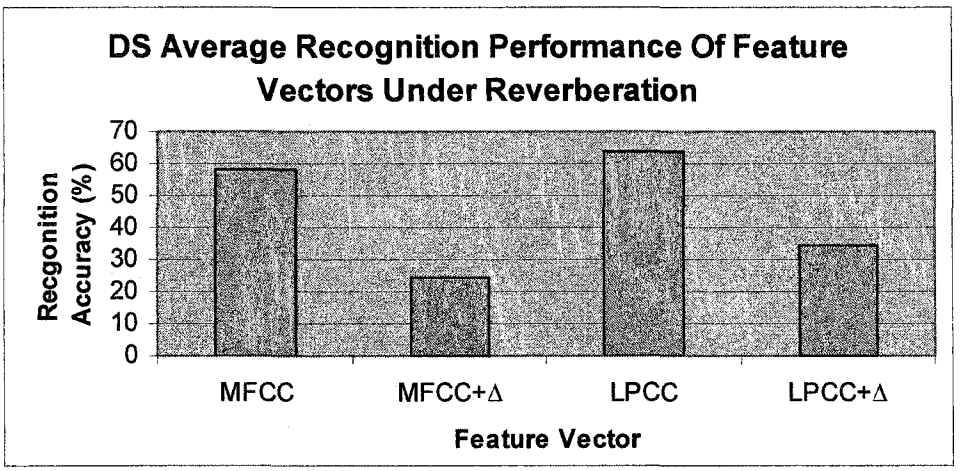

Figure 5.20: Average recognition performance using different features for DS under reverberation.

Using DS, LPCC vectors had a slight average performance advantage over MFCC vectors when the results given the current simulations are reviewed. This was for the case when delta cepstral coefficients were not used. This advantage was only slight in both recognition and verification. For low levels of reverberation, the LPCC vectors outperformed the MFCC vectors in both recognition and verification. At higher levels of reverberation the performance of LPCC and MFCC vectors without delta cepstral coefficients was similar. As was the case with AR-AGS and SM the delta-cepstral coefficients degraded accuracy. When delta-cepstral coefficients were included MFCC vectors and LPCC vectors both give similar verification performances, but for recognition the LPCC version yielded slightly better results.

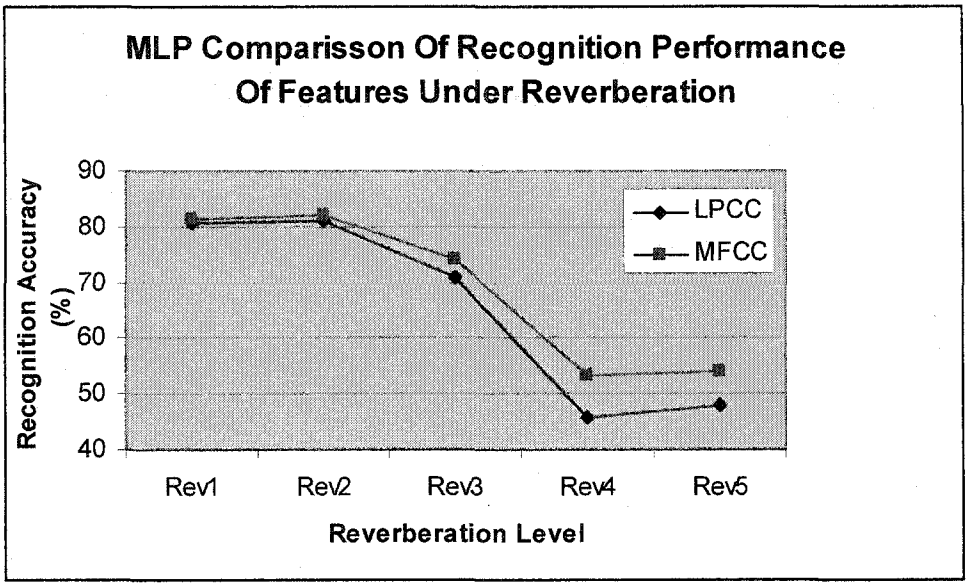

Figure 5.21: Recognition performance using different features for MLP under reverberation.

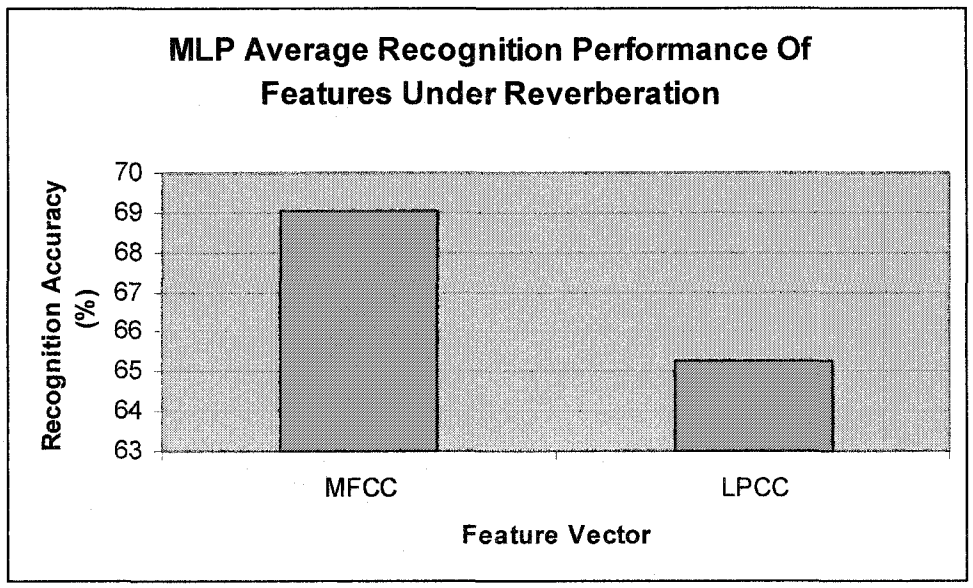

Figure 5.22: Average recognition performance using different features for MLP under reverberation.

For the MLP MFCC vectors gave better results under conditions of reverberation than did LPCC vectors.

5.2 Comparison of Methods

A comparison will now be made between methods. For all methods the best performing verification and recognition results under reverberation were produced using the same feature vector. For example for AR-AGS, LPCC features outperformed the other features for this method in both recognition and verification trials under reverberation. For AR-Itakura LPCC vectors with delta cepstrum outperformed all other features for this method in both recognition and verification trials. For all methods except AR-Itakura and the MLP, the performance using LPCC vectors is compared. For AR-Itakura the performance using LPCC vectors appended with delta cepstral coefficients is compared and for the MLP the performance using MFCC vectors is compared. The average performance over the 5 reverberant conditions for each method is what is displayed in the upcoming graphs.

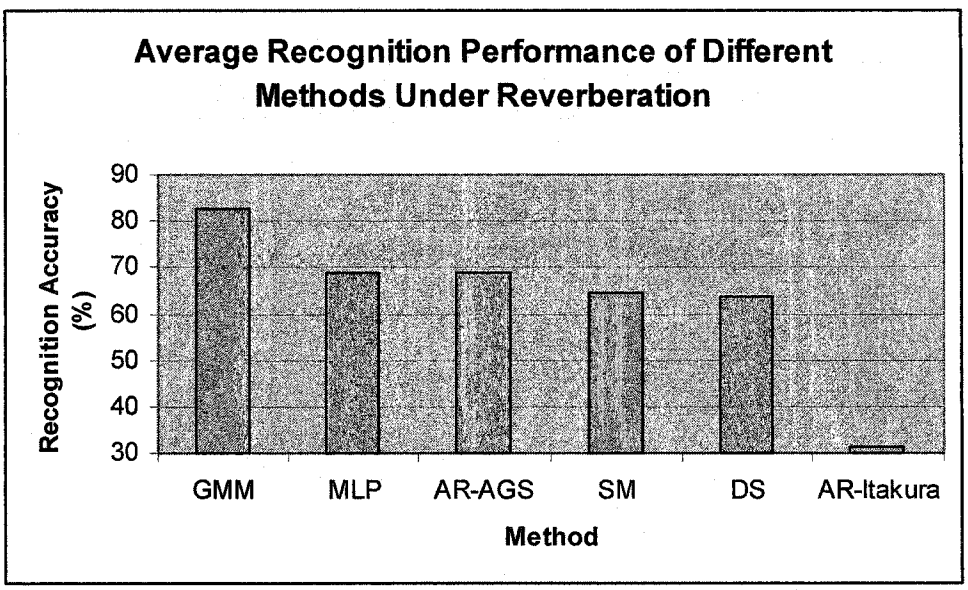

Figure 5.23: Average recognition performance of all methods under increasing reverberation.

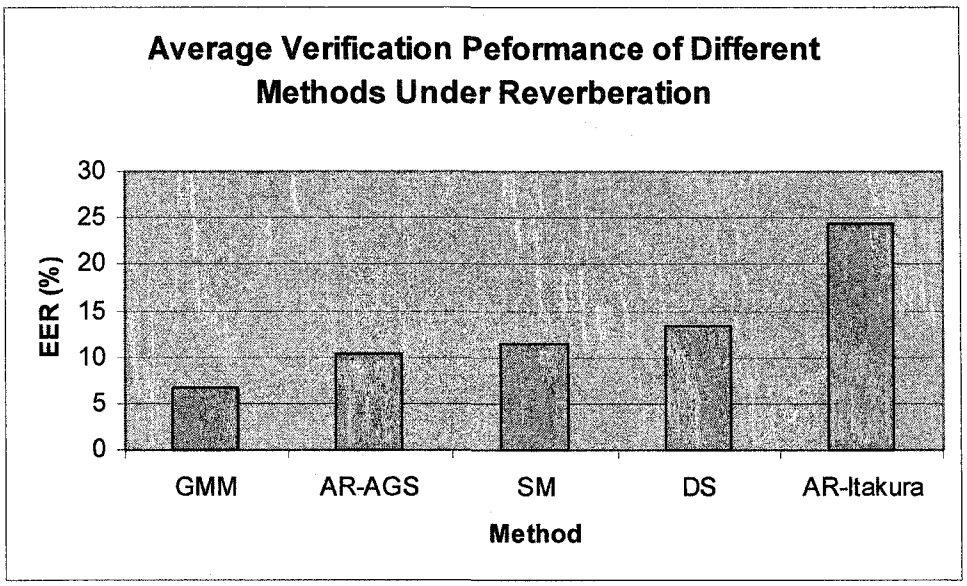

Figure 5.24: Average verification performance of all methods under increasing reverberation.

It is clear from both the verification and recognition results that the GMM was the best performing method. The worst performing method using both performance metrics was AR-Itakura. The two covariance based methods gave similar recognition performance but the verification performance of the sphericity measure as better.

5.3 Parameter Selection

The feature vector length that yielded the best results when testing was performed on reverberant speech was determined and used before the methods and features were compared in sections 5.1 and 5.2. This prevented a non-objective comparison of the methods that would have occurred if some methods were compared using the best performing vector length and others were not. The average recognition

accuracy of each method and features was calculated over the 5 reverberant conditions. The performance of each method and feature vector combination was plotted against the feature vector length. These illustrations of average performance can be found in Appendix B.

The parameter settings that give the best results were determined. Parameter settings that resulted in good closed set recognition accuracy also resulted in good verification accuracy. The parameters that were used for the simulation results shown in this chapter are shown in the following two tables.

Table 5.1: Selected parameters for use under reverberant conditions.

Method Type	LPCC Vector Length	MFCC Vector Length
GMM	15	12
AR-ITAKURA	20	18
AR-AGS	20	11
SM	18	12
DS	20	14
MLP	18	18

Table 5.2: GMM variance multiplier

Feature Vector	Variance limit multiplier
MFCC	0.95
LPCC	0.95

Chapter 6

Experimental Verification of Results

The purpose of this chapter is to determine the effect on performance of speaker recognition methods when the reverberation is that of a real room.

6.1 Experiment Description

In order to reverberate the speech in a real room it was necessary to play the speech in a room and record it as it was being played. This has the effect of adding microphone and speaker effects to the speech in addition to the room reverberation effects. It is not possible to train with the KING database in a form that is not transduced by these speaker and microphone effects and test with speech having these

speaker and microphone effects. In order to obtain training speech that was not reverberated in the room but still had the same speaker and microphone characteristics as the speech transduced in the room it was necessary to play the speech and record it using the same speaker and microphone combination but in a non-reverberant enclosure. For this reason the speech was played in an anechoic box using the same speaker and recorded using the same microphone. The specifics of the equipment and the setup of the experiments used to record the reverberant and non-reverberant speech was outlined in section 3.3.

6.2 Experimental Results

For each method the performance of each vector will be compared where the training speech was that from an anechoic box and the test speech was recorded with the microphone and speaker placed apart in the reverberant room. The parameters used are those shown in table 5.1 and table 5.2. The following graphs contain the recognition and verification results.

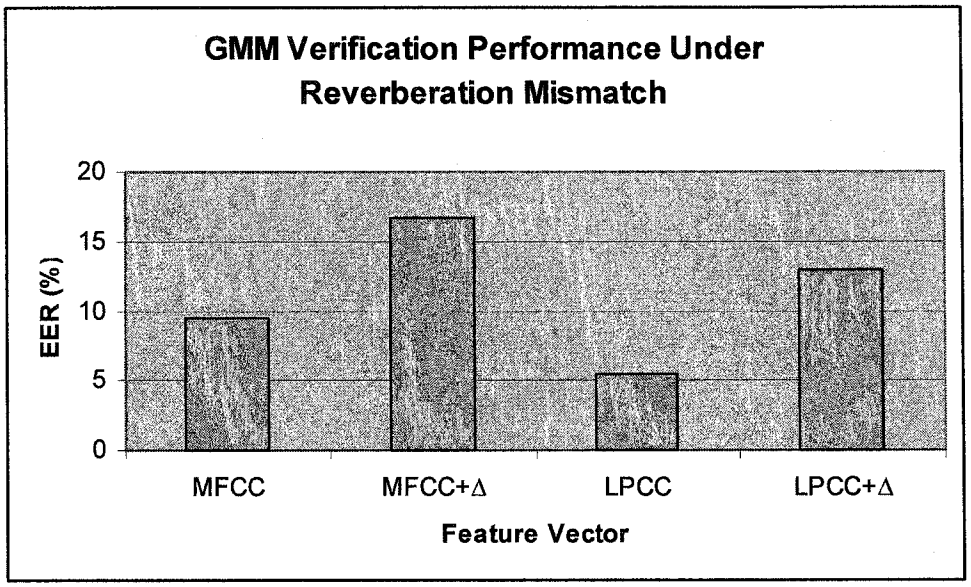

Figure 6.1: Verification performance for GMM under reverberation mismatch.

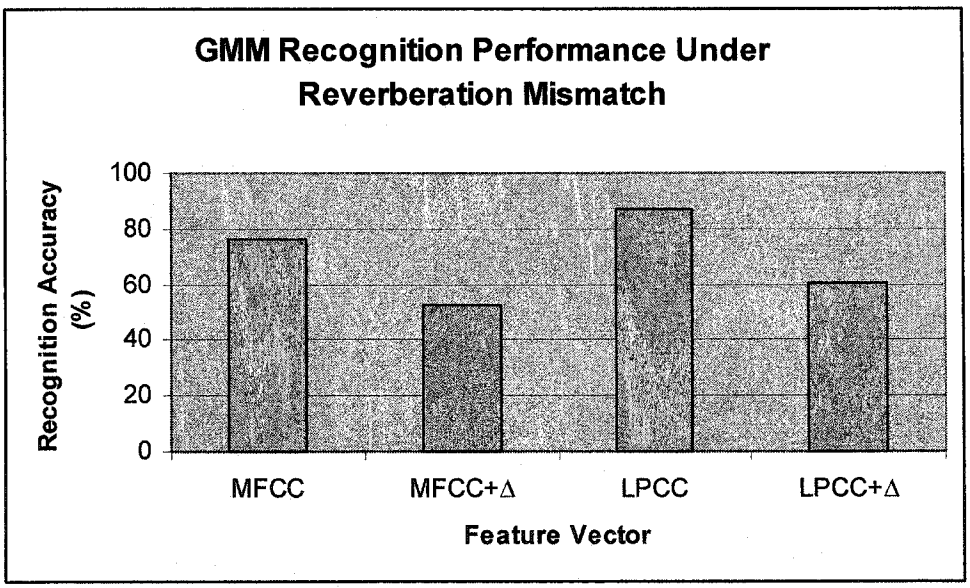

Figure 6.2: Recognition performance for GMM under reverberation mismatch.

The recognition and verification results for the GMM for the experimental results shown above have some similarities and differences from the simulated results in chapter 5. As far as the similarities are concerned, LPCC vectors without delta

cepstral coefficients gave better performance than MFCC vectors for both recognition and verification. Delta cepstral coefficients degraded accuracy for both feature vector types in both the simulated and experimental results. The difference between the simulated and experimental results is that the LPCC vectors with delta cepstral coefficients gave better performance in the experimental results than the MFCC vectors with delta cepstral coefficients. In the simulated results, the LPCC vectors with delta cepstral coefficients gave slightly worse performance than the MFCC vectors with delta cepstral coefficients.

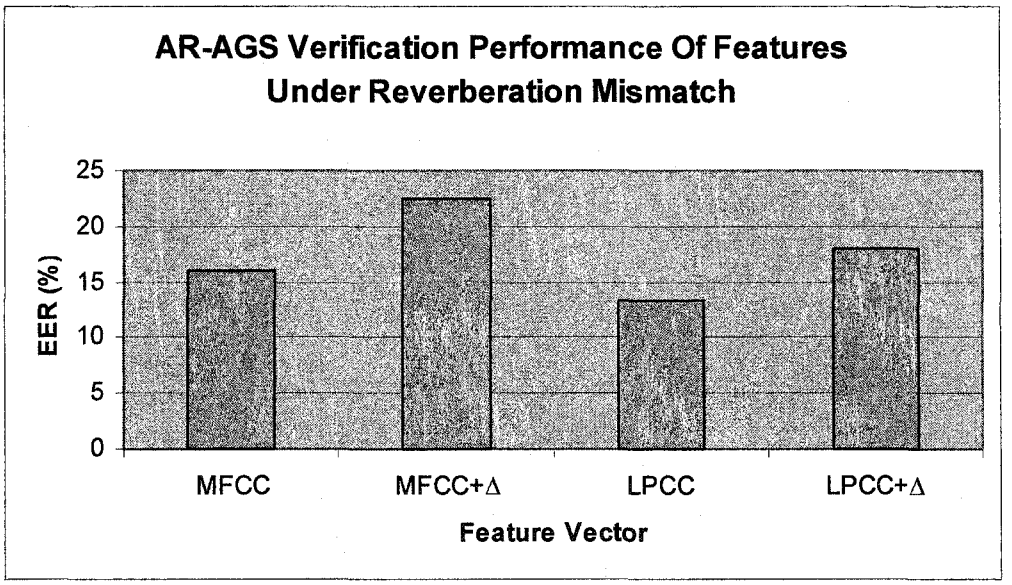

Figure 6.3: Verification performance for AR-AGS under reverberation mismatch.

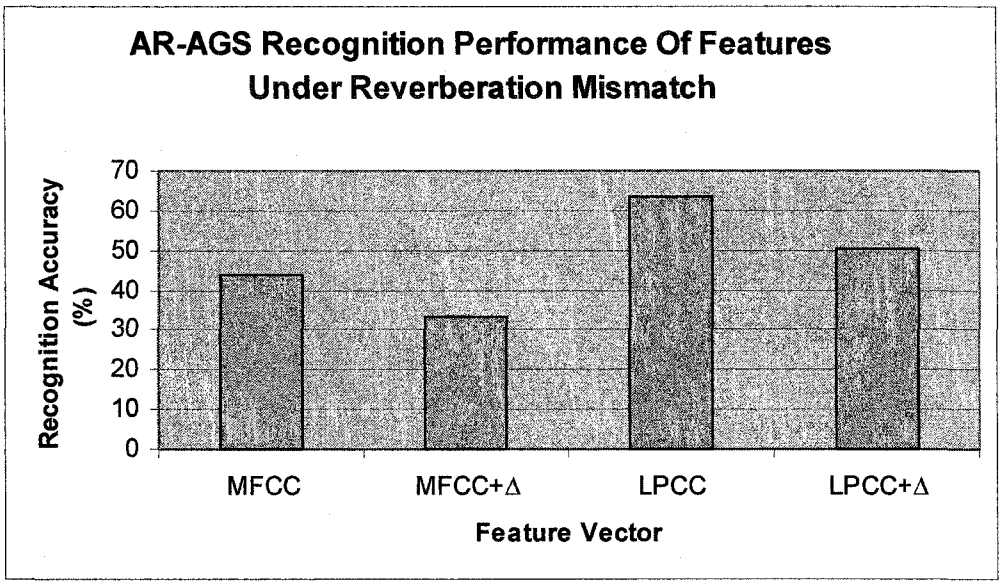

Figure 6.4: Recognition performance for AR-AGS under reverberation mismatch.

For AR-AGS the experimental results from figure 6.3 and figure 6.4 are consistent with simulated results as far as the relative performance of the feature vectors is concerned. The ranking of the features vectors with regard to their recognition and verification performance is the same as the simulated results. In the simulated results LPCC vectors with delta cepstral coefficients outperformed MFCC vectors without delta cepstral coefficients in recognition but gave similar verification performance. In the experimental results the relative recognition performance for these two parameterizations was the same as the simulated results, there was however a difference in the verification results. The difference was that in the experimental results, LPCC vectors with delta cepstral coefficients gave worse performance than MFCC vector without delta cepstral coefficients in the simulated results they gave the same performance. In the simulated another difference between the simulated and

experimental results was that the verification performance in the experimental results of the LPCC vectors did not exceed that of the MFCC vectors by the same margin as in the simulated results.

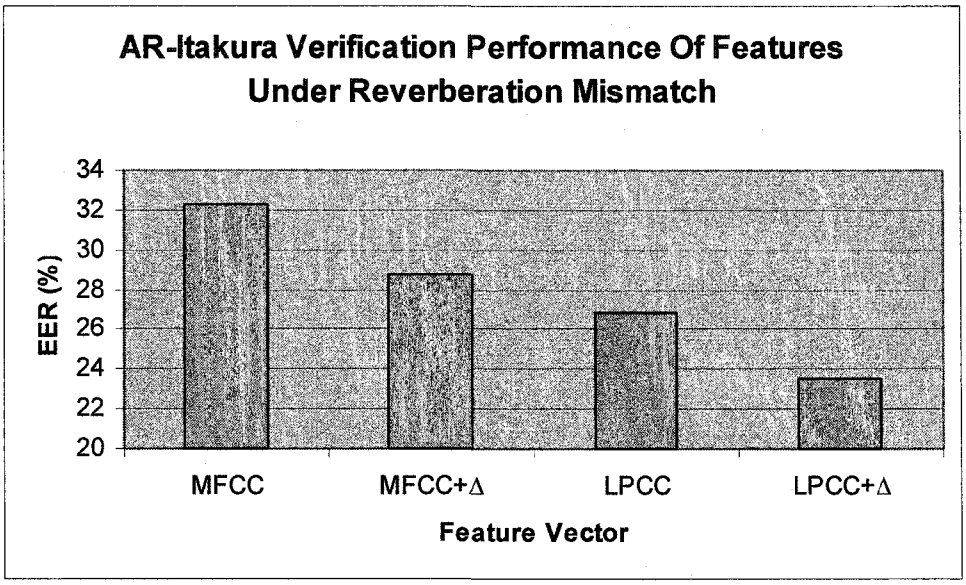

Figure 6.5: Verification performance for AR-Itakura under reverberation mismatch.

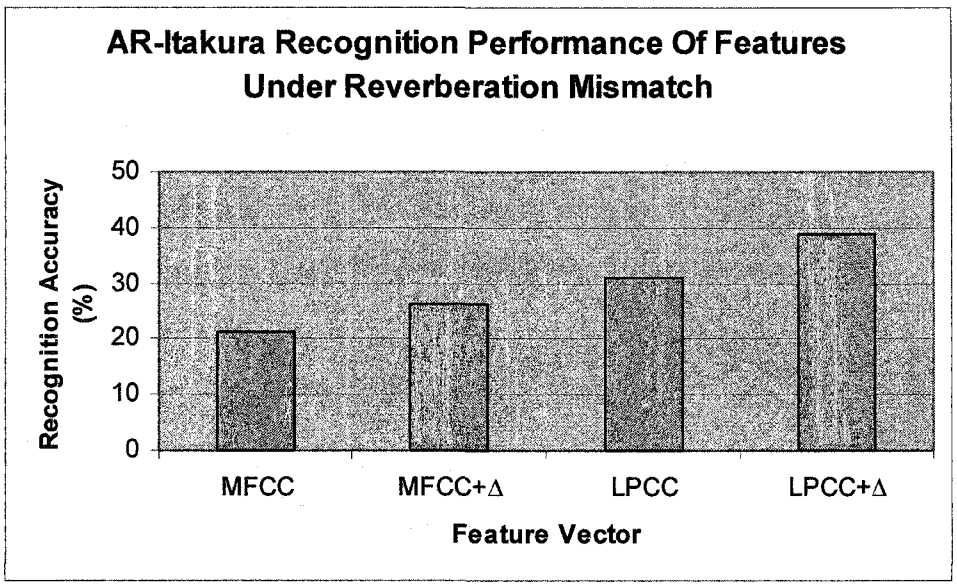

Figure 6.6: Recognition performance for AR-Itakura under reverberation mismatch.

For AR-Itakura there were some differences between the simulated and experimental results. In the experimental results the ranking of the methods in verification performance and recognition performance was the same. In the simulated results there was a slight discrepancy in the performance ranking between the verification and recognition results. In the experimental results, delta cepstral coefficients improved performance for both the MFCC and LPCC vectors. In the simulated results, the use of delta cepstral coefficients did not improve the recognition performance for the MFCC vectors. In both, the simulated and experimental results, LPCC vectors with delta cepstral coefficients gave the best performance. Overall, LPCC vectors outperformed MFCC vectors in both simulated and experimental results.

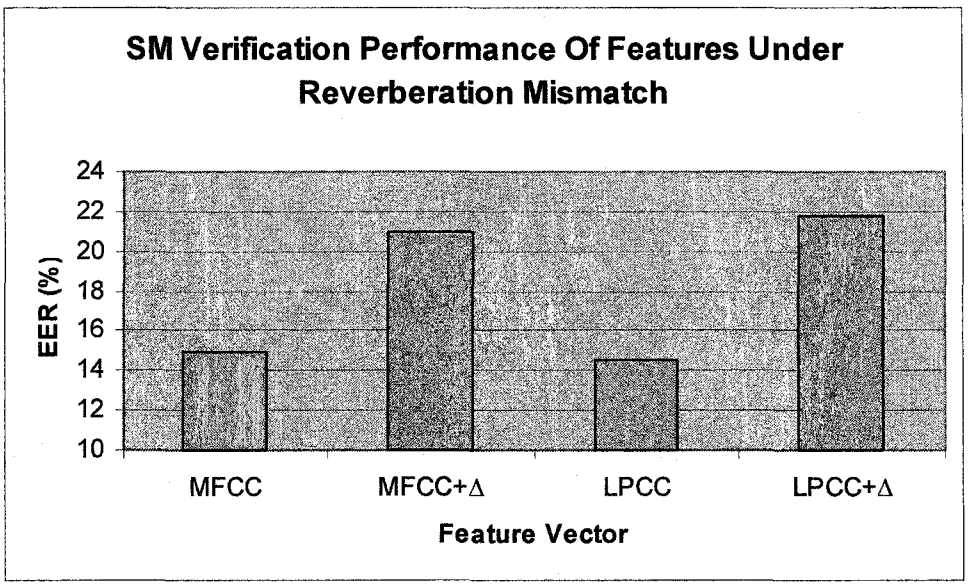

Figure 6.7: Verification performance for SM under reverberation mismatch.

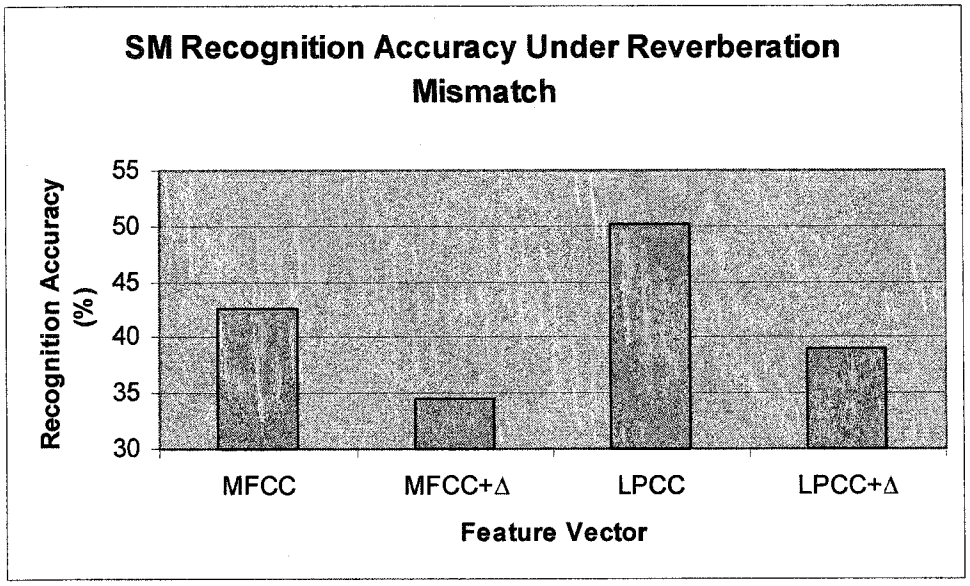

Figure 6.8: Recognition performance for SM under reverberation mismatch.

There were signification differences between the simulated and experimental results for this method. With regard to the recognition results the main difference was that LPCC features with delta cepstral coefficients do not perform as well in the experimental results as they did in the simulated results. In the simulated results they outperform MFCC vectors, in the experimental results this was not the case, they actually performed worse. With regard to verification a similar phenomenon was observed. The LPCC vectors with delta coefficients performed better in the simulated results than they did in the experimental results. Another difference was that the LPCC and MFCC vectors without delta cepstral coefficients performed similarly in verification performance in the experimental results, in the simulated results however the LPCC vectors outperformed the MFCC vectors.

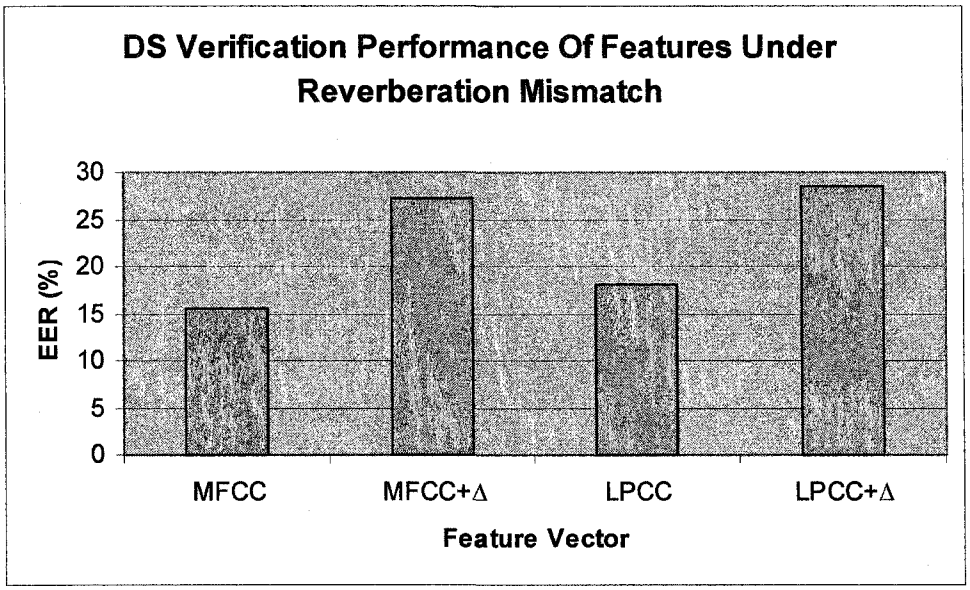

Figure 6.9: Verification performance for DS under reverberation mismatch.

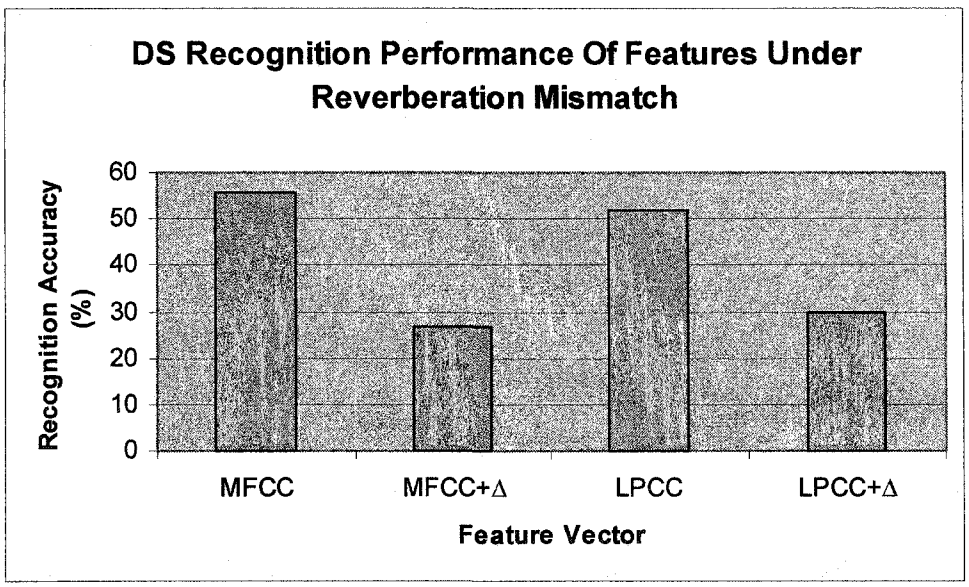

Figure 6.10: Recognition performance for DS under reverberation mismatch.

In the simulated results for DS an interesting point to is that the MFCC and LPCC vectors without delta cepstral coefficients performed similarly with a smaller advantage for LPCC vectors over MFCC vectors than with other methods. In fact in

the simulated results there is an approximately 2% performance advantage in verification for using LPCC vectors over MFCC vectors and a 4% advantage in recognition. In the experimental results the MFCC vectors actually perform better than the LPCC vectors by the same margin that the LPCC vectors outperformed the MFCC vectors in the simulated results. It is noted however that for this method the MFCC vectors under reverberation perform well when compared to LPCC vectors than for the previous methods. This is shown in the experimental and simulated results.

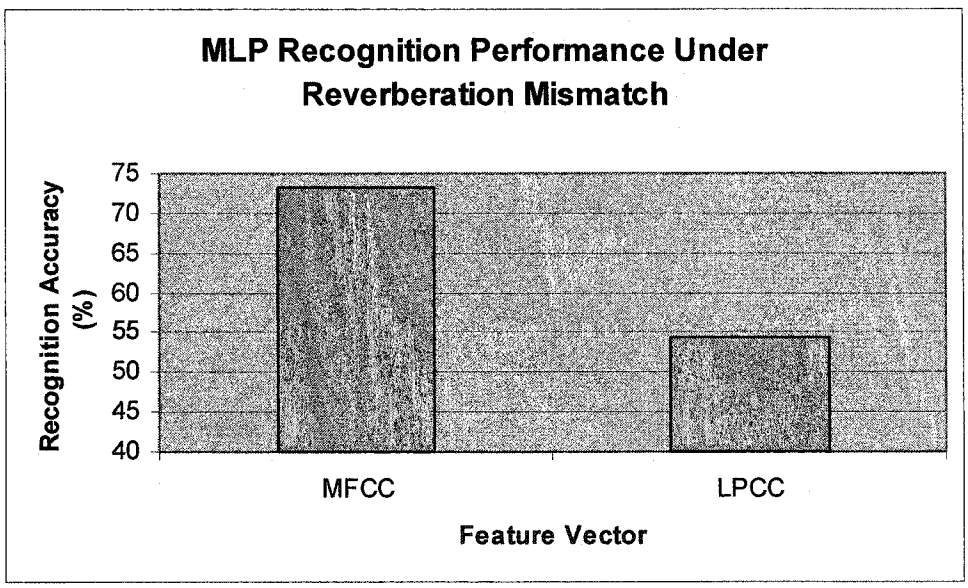

Figure 6.11: Recognition performance for MLP under reverberation mismatch.

With regard to the MLP, the simulated results agree with the experimental results. The MFCC vectors for the MLP outperformed the LPCC vectors. In the experimental results however the advantage of using MFCC results is significantly

greater than in the simulated results. The difference in performance in the simulated results is approximately 4% while in the experimental results it is close to 15%.

The performance of the methods will now be compared to each other. The same feature vectors used for the comparison of the methods in section 5.2 will be compared here.

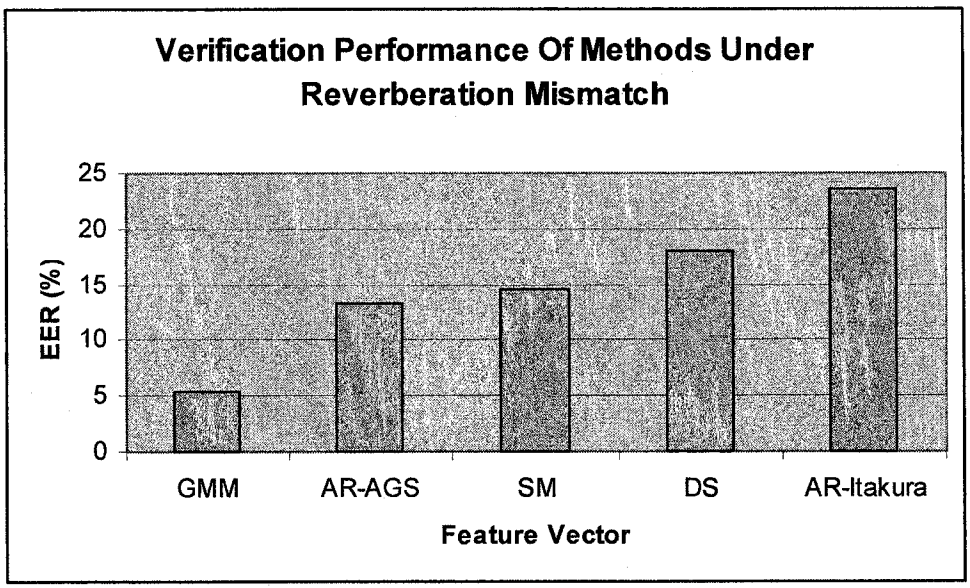

Figure 6.12: Comparison of verification performance of methods in real reverberation conditions.

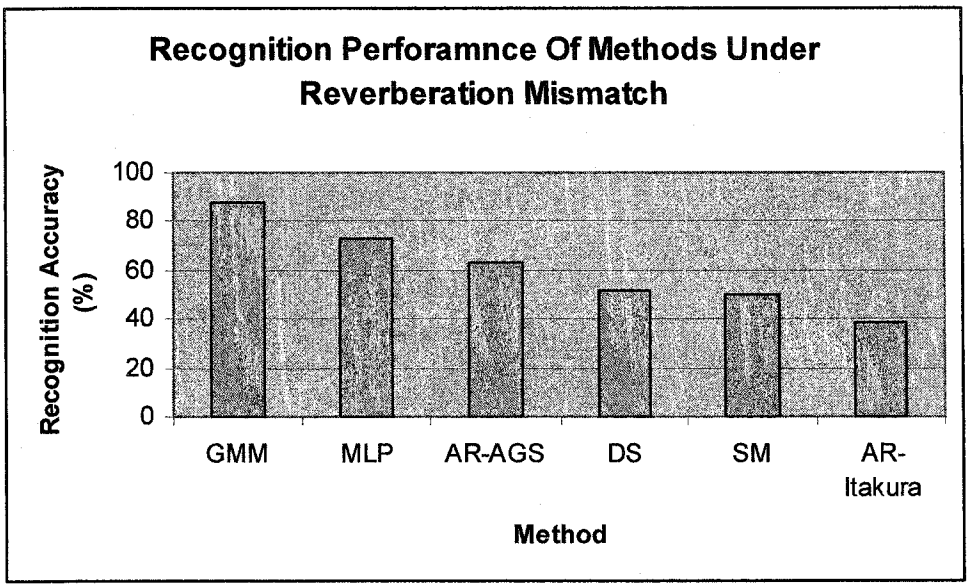

Figure 6.13: Comparison of recognition performance of methods in real reverberation conditions.

First the experimentally derived recognition results in figure 6.13 will be compared to the results in figure 5.23 which were derived through simulation. It can be seen that the best and worst performing methods in both graphs are the same. The GMM is still the best method and AR-Itakura is the worst. The MLP performs better in the experimental results relative to the other methods than it does in the simulated results. The two covariance based methods gave similar performance in the experimental results, this is consistent with the simulated results where they gave similar performance. AR-Itakura performed better in the experimental results relative to the other methods than it did in the simulated results. AR-AGS outperformed the covariance based methods by a similar margin in the simulated results as it did in the experimental results. In the experimental results it did not perform as well relative to the GMM as it did in the simulated results. Neither do the covariance based

methods. The relative performance of AR-Itakura and the MLP compared to the GMM is consistent in the experimental results with the simulated results.

The experimentally derived verification results in figure 6.12 will be compared to those in figure 5.22 which were derived through simulation. The ranking of the methods in the experimental results is the exact same as in the simulated results. The relative performance of AR-AGS and the covariance based methods in the experimental results degraded relative to the other methods compared to the simulated results. The degree of degradation was similar for AR-AGS and the covariance based methods, it was in the region of 4%. This means that the performance of AR-Itakura and the GMM improved in the experimental results compared to the other methods. The amount of improvement was small, in the region of 1%.

6.3 Parameter Selection

The parameters such as vector length and GMM variance limits used were the same as those used in chapter 5. The results shown in Appendix B were inspected and it was determined which vector lengths gave the best performance the parameters used were the same as those listed in table 5.1 and table 5.2.

Chapter Seven
Cross Training

Training with non-reverberant speech and testing with reverberant speech causes degradation in the closed set recognition and verification accuracy. The question arises: what will happen if before testing with reverberant speech, the models are trained with reverberant speech? Two distinct cases of this type of compensated training exist. In the first case the reverberation that is used to train the speech with, matches the reverberation used to test the speech. This scenario may occur in the application of building access secured using a speaker verification system where the speaker being verified must stand at a specific location before uttering the test speech. In this scenario the speaker to microphone impulse response may be known, and training may be performed using this known impulse response. In the second case, the impulse response that reverberates the test speech may not be known. To what degree can the impulse response that is used for testing compensate for or fail to compensate for the reverberation that affects the test speech. An exhaustive set of tests that make use of the 5 existing impulse responses were carried out to gain some

perspective on this matter. The test speech was reverberated using any of the 5 mentioned impulse responses or may not have been reverberated at all. The training speech can likewise be reverberated by any of the 5 impulse responses or may not be reverberated. Each combination of training and test condition was attempted. This was attempted for each of the 5 methods and the feature vector that gave the best average results. The results will be displayed in the remainder of this section. The parameters used are those listed in table 5.1 and table 5.2. Table 7.1 to table 7.10 contain the results from simulations where the models were trained using speech from each of the 5 reverberant room models and test speech originated from each of the 5 reverberant room models. Each combination of training speech model and test speech model were attempted.

Table 7.1: Verification performance of GMM with LPCC vectors when training and testing is performed in exhaustive combinations.

Train / Test	No Reverb	Reverb 1	Reverb2	Reverb3	Reverb4	Reverb5
No Reverb	2.6%	5.1%	6.5%	8.0%	10.5%	10.9%
Reverb 1	4.0	3.9	4.3	5.4	8.0	8.0
Reverb 2	4.7	4.4	4.3	4.3	6.5	6.9
Reverb 3	6.5	4.4	4.3	4.0	5.8	6.5
Reverb 4	9.4	5.4	5.5	4.7	5.5	5.8
Reverb 5	8.7	5.4	5.1	4.3	5.7	5.8
Average Rev. 1 - 5	6.7%	4.7%	4.7%	4.6%	6.3%	6.6%

Table 7.2: Recognition performance of GMM with LPCC vectors when training and testing is performed in exhaustive combinations.

Train / Test	No Reverb	Reverb 1	Reverb2	Reverb3	Reverb4	Reverb5
No Reverb	97.1%	95.1%	92.2%	85.6%	72.0%	73.8%
Reverb 1	94.5	96.5	95.4	94.8	85.9	87.3
Reverb 2	91.4	95.1	95.4	94.8	91.9	91.6
Reverb 3	86.2	94.2	94.5	96.0	93.4	93.1
Reverb 4	70.0	86.7	91.4	93.7	93.7	94.2
Reverb 5	72.0	85.3	91.1	94.2	92.2	93.4
Average Rev. 1 - 5	82.8%	91.6%	93.5%	94.7%	91.4%	91.9%

Table 7.3: Recognition performance of AR-AGS with LPCC vectors when training and testing is performed in exhaustive combinations.

Train / Test	No Reverb	Reverb 1	Reverb2	Reverb3	Reverb4	Reverb5
No Reverb	96.3%	85.9%	79.0%	68.0%	58.2%	55.3%
Reverb 1	82.4	93.1	89.0	86.7	78.1	78.1
Reverb 2	81.6	87.0	93.1	86.5	80.1	81.0
Reverb 3	70.6	82.1	81.6	90.8	85.3	85.3
Reverb 4	54.8	74.4	74.6	86.7	87.6	87.3
Reverb 5	53.9	74.6	73.8	87.6	85.3	87.0
Average Rev. 1 - 5	68.6%	82.2%	82.4%	87.7%	83.3%	83.7%

Table 7.4: Verification performance of AR-AGS with LPCC vectors when training and testing is performed in exhaustive combinations.

Train / Test	No Reverb	Reverb 1	Reverb2	Reverb3	Reverb4	Reverb5
No Reverb	3.9%	7.2%	8.3%	11.2%	15.3%	15.9%
Reverb 1	7.2	5.9	7.3	8.3	9.4	9.3
Reverb 2	7.5	6.1	6.1	8.3	9.4	9.4
Reverb 3	10.8	6.9	8.9	6.8	7.6	7.5
Reverb 4	13.4	8.7	10.9	7.6	6.8	7.5
Reverb 5	13.1	8.7	10.5	7.2	8.0	7.2
Average Rev. 1 - 5	10.4%	7.2%	8.7%	7.6%	8.3%	8.2%

Table 7.5: Recognition performance of AR-Itakura with LPCC+Δ vectors when training and testing is performed in exhaustive combinations.

Train / Test	No Reverb	Reverb 1	Reverb2	Reverb3	Reverb4	Reverb5
No Reverb	92.5%	80.4%	77.8%	74.4%	52.7%	59.9%
Reverb 1	47.0	93.4	87.3	89.6	70.9	77.5
Reverb 2	39.2	87.3	91.9	89.3	80.7	84.4
Reverb 3	30.8	82.1	85.3	92.8	83.9	85.6
Reverb 4	18.7	63.7	71.8	82.4	89.9	89.9
Reverb 5	20.5	67.1	76.4	81.6	85.3	88.8
Average Rev. 1 - 5	31.2%	78.7%	82.5%	87.1%	82.1%	85.2%

Table 7.6: Verification performance of AR-Itakura with LPCC+Δ vectors when training and testing is performed in exhaustive combinations.

Train / Test	No Reverb	Reverb 1	Reverb2	Reverb3	Reverb4	Reverb5
No Reverb	8.8%	10.5%	11.6%	12.3%	16.3%	14.2%
Reverb 1	15.9	6.2	8.7	7.7	11.9	11.6
Reverb 2	21.0	6.5	5.8	6.9	9.4	8.7
Reverb 3	22.4	7.2	7.6	6.1	10.1	9.8
Reverb 4	31.1	11.5	9.7	7.6	7.7	7.2
Reverb 5	31.1	10.1	8.3	7.2	8.7	7.5
Average Rev. 1 - 5	24.3%	8.3%	8.0%	7.1%	9.6%	9.0%

Table 7.7: Recognition performance of SM with LPCC vectors when training and testing is performed in exhaustive combinations.

Train / Test	No Reverb	Reverb 1	Reverb2	Reverb3	Reverb4	Reverb5
No Reverb	95.7%	86.7%	79.3%	70.3%	59.9%	57.3%
Reverb 1	82.1	93.1	87.0	86.7	78.7	77.2
Reverb 2	77.8	87.9	91.9	85.9	81.6	81.3
Reverb 3	61.7	83.0	82.4	89.0	84.4	85.9
Reverb 4	50.4	72.9	75.2	86.7	86.7	87.0
Reverb 5	51.6	71.5	75.2	85.3	85.0	86.2
Average Rev. 1 - 5	64.7%	81.7%	82.4%	86.7%	83.3%	83.5%

Table 7.8: Verification performance of SM with LPCC vectors when training and testing is performed in exhaustive combinations.

Train / Test	No Reverb	Reverb 1	Reverb2	Reverb3	Reverb4	Reverb5
No Reverb	4.0%	6.9%	7.9%	9.7%	12.3%	12.6%
Reverb 1	7.2	5.4	6.8	7.2	8.6	9.0
Reverb 2	9.0	6.9	6.1	8.0	8.7	9.0
Reverb 3	12.2	7.6	8.3	6.5	7.2	7.2
Reverb 4	14.5	8.7	10.1	7.3	6.9	7.2
Reverb 5	14.2	8.6	9.1	7.2	7.6	7.2
Average Rev. 1 - 5	11.4%	7.4%	8.1%	7.2%	7.8%	7.9%

Table 7.9: Recognition performance of DS with LPCC vectors when training and testing is performed in exhaustive combinations.

Train / Test	No Reverb	Reverb 1	Reverb2	Reverb3	Reverb4	Reverb5
No Reverb	95.7%	88.8%	80.7%	70.6%	50.7%	51.9%
Reverb 1	86.2	92.5	87.9	85.0	72.0	74.1
Reverb 2	76.9	85.6	92.5	87.0	81.8	81.8
Reverb 3	61.7	76.7	81.0	91.4	85.3	87.3
Reverb 4	47.8	64.6	69.7	84.1	88.2	87.6
Reverb 5	46.1	63.4	69.5	83.6	86.7	86.5
Average Rev. 1 - 5	63.7%	76.5%	80.1%	86.2%	82.8%	83.5%

Table 7.10: Verification performance of DS with LPCC vectors when training and testing is performed in exhaustive combinations.

Train / Test	No Reverb	Reverb 1	Reverb2	Reverb3	Reverb4	Reverb5
No Reverb	4.0%	6.1%	8.3%	11.3%	15.2%	15.6%
Reverb 1	6.9	5.8	7.2	7.5	9.7	9.7
Reverb 2	9.4	7.2	5.4	7.2	8.7	8.7
Reverb 3	13.7	8.7	8.6	5.8	6.5	6.4
Reverb 4	18.7	12.2	12.3	7.9	6.8	7.2
Reverb 5	18.4	11.2	10.8	7.6	7.1	7.2
Average Rev. 1 - 5	13.4%	9.0%	8.9%	7.2%	7.8%	7.9%

The results presented are for the best performing feature vector types of each method. There was a significant improvement when testing with reverberant speech, when the training data was reverberant rather than non-reverberant. This was the case even if the training room specifications were different than the test room specifications. There are cases however where training with reverberant speech consistently reduced performance. This usually occurred when the test speech was from Reverb1 or Reverb2 which are mildly reverberated conditions and the training speech was from Reverb4 or Reverb5, which contain the most reverberation. For this reason based on these results it would seem that if reverberant training speech is used the reverberation should not exceed that of the test speech. In general, the best performance was achieved if the training and test speech used were from the same reverberant room. This is of course intuitive. The cases where training was performed using the same reverberation model as the test speech are those on the diagonal elements of the above tables. As results in each row of the tables are reviewed, the performance tends to worsen as one inspects results further to the left of right of the diagonal elements. Moving across a row in one of the tables towards the left is the case where the reverberation in the training speech becomes successively less than that in the test speech. Moving towards the right is the opposite case where the reverberation in the training speech becomes successively more than that in the training speech.

When training speech came from the same room as the test speech there was always an improvement in recognition and reverberation performance over the case where the test speech from that room was scored against un-reverberated training speech models. Training with synthetic reverberation and testing with real reverberation from a real room may not yeild results that are as good as those when training and testing reverberation are both synthetic or training and testing reverberation are both real. The reason is that the synthetic reverberation characteristics differ from those in real rooms. This is due to some approximations of the image model such as angle independent reflection coefficients as well as reflection coefficients that are constant for different frequencies.

Chapter Eight

Classification of Reverberation Source

In the previous chapter it was revealed that training with reverberant speech can give improved performance over training with non-reverberant speech when the test speech is reverberant. It was found that reverberating the training speech using the same impulse response as the test speech gave the best results. Using significantly more or less reverberation in the training speech than the test speech gave worse performance than using training speech with only slightly more or less reverberation than the test speech. From these results it would appear that when testing with reverberant speech from an unknown room configuration, performance advantages can be gained if the reverberant training speech used is close to the reverberant test speech used. If a set of models trained with reverberant test speech exist, then the reverberant model trained with the speech reverberated using the impulse response closest to the test speech should be used for classification.

The following section will describe a method that can be used to determine given a set of models trained with speech reverberated using different impulse responses, which model should be used. This method makes a decision based on the test speech to determine which of the training models was trained with speech that is closest to the test speech. The objective is to determine, given a set of reverberant room impulses, and a reverberant speech segment, which of the reverberant room impulse responses best characterizes the reverberation in the reverberant speech segment. Speaker recognition methods were adapted to perform reverberation recognition. The reverberation classification methods can be used to determine, first of all, whether the input test speech is reverberant or not. Secondly, if the speech is reverberant it can give information about which training speech is most suitable to perform speaker recognition or verification. Experiments were performed to quantify the ability of speaker recognition methods using different feature vectors to perform reverberation classification.

8.1 Reverberation Classification Method

For each speaker it is proposed that a set of models be present. Each of the models in a speaker's set are trained with speech reverberated using a different room impulse response. Successive models in the set will have been trained used speech with successively greater reverberation. At test time the model that was trained using speech that is closest to the reverberant test speech should be used. The same training

utterance will still be used to train all the models for the same speaker. Each time it is used to train one of the speaker's models, it will be reverberated before hand using a reverberant room impulse response. If there are 5 room impulse responses, then each speaker will have five models. Each speaker will have one model corresponding to each one of the reverberant impulse responses. Each speaker will also have one model trained from non-reverberant speech to be used if the test speech is non-reverberant.

At test time, for either closed set recognition or for verification it must be determined which model must be used for each speaker. If it is a closed set recognition task then one model will be used to represent each speaker. The model to be used for each speaker will have a common characteristic with the model selected for the other speakers. These selected models will have been trained using speech reverberated using the same impulse response. In the case of speaker verification, the normalization models will be selected such that they are the ones trained using speech reverberated using the same impulse response as that used for the target verification model. In either case the models selected will have been trained with speech that is as close as possible in its reverberation characteristics to the test speech.

The challenge in this method is to determine which of a speaker's models should be used in tests with the reverberant test speech. The proposed solution is as follows:

1) Select one of the speakers at random.

2) Create for this speaker a set of models; each model should be trained with the same training utterance except that the utterance will be reverberated using different room impulse responses before training is performed. One model will be created using non-reverberant speech the others with reverberant speech. The models will be referred to as "reverberation classification models" (RCMs). The models can be created using any one of the speaker recognition models i.e. GMM, covariance based or auto-regressive models.
3) At test time the test speech will be scored using a speaker recognition method against each of the models from step 2. The closest model will indicate that its associated training speech is nearest to the test utterance. The reverberant impulse response used to reverberate this training speech is therefore nearest to the reverberation that corrupted the test speech.
4) The models that will be used for performing recognition or verification on the test speech will be the ones trained using the impulse response selected in step 3.

8.2 Reverberation Classification Results

In this section experiments were conducted to determine if it is possible to take a reverberated segment of speech and determine from a set of room impulse responses, the room impulse response that reverberated the test speech. In order to perform this task, the training speech for a single speaker was reverberated using the

5 room impulse responses. 6 models were created, one for each of the 5 impulse responses and one for the un-reverberated speech. Test segments of un-reverberated speech and speech reverberated using the 5 impulse responses were than scored against the 6 models to determine which of the 6 speech types the test speech was nearest to. The test segments came from all of the speakers in the database while the training speech for the 6 models came from only one of the speakers in the database.

Experiments were conducted using different feature vectors and reverberation classification methods. It has been found that delta cepstral coefficients degrade recognition performance for most methods because they are affected by reverberation. When reverberant test speech is used the delta-features created from the reverberant test speech are different than those created from non-reverberant speech. In effect they capture some of the characteristics of the reverberant speech. It has also been found that filter-bank outputs that are not de-correlated using a DCT were also greatly affected by reverberation. For this reason reverberation classification were be performed using both LPCC and MFCC vectors in addition to filter-bank outputs that are not de-correlated using a DCT. Experiments were performed where delta-cepstral were included in the feature vector as well as experiments where they were not included.

As was stated in the last section, all the reverberation classification models are created using the training speech from the same speaker. The RCMs were created using the training speech for speaker 1. Sessions 1 to 3 were used. One model was

trained using the non-reverberant training speech and 5 were created using the same training speech after it had been reverberated with the 5 reverberant room impulse responses. Test segments to be classified consisted of the first 30 seconds extracted from sessions 4 to 10 of all speakers. An un-reverberated version of each test segment was used in addition to 5 reverberated versions of each test segment. The reverberation used was that of the 5 room impulse responses described in section 3.5.

The classification experiments consisted of determining which of the RCMs is nearest to each test segment. A successful trial occurs if the correct RCM is selected. The remainder of this section presents the results. The feature vector lengths used where 18 for both MFCC and LPCC vectors. When filter bank outputs where used the length of the feature vector was equal to the number of filters in the filter bank (19). The test segments consisted of the first 30s segments of each the last 7 sessions of the KING database. Table 8.1 to table 8.5 contain the result of these classification experiments. In table 8.1, the models used to classify the test segments were created using a GMM. One model was created for each of the room impulse responses and one was created using un-reverberated speech. The test segments were then scored against each of the 6 models and the nearest model was determined. A successful classification occurred if the nearest model was reverberated using the same impulse response used to reverberate the test speech.

This experiment was repeated for each of the speaker recognition methods. For each method, LPCC, MFCC vectors and filter bank outputs were the feature vectors used. Experiments were performed with and without delta coefficients

Table 8.1: Reverberation classification results for GMM

Feature Vector	Classification Accuracy
LPCC	57.4%
LPCC+Δ	62.4%
MFCC	73.0%
MFCC +Δ	78.7%
Average Performance	67.9%

Table 8.2: Reverberation classification results for AR-Itakura

Feature Vector	Classification Accuracy
LPCC	80.6%
LPCC+Δ	89.8%
MFCC	84.9%
MFCC+Δ	90.3%
Filter bank	93.0%
Filter bank +Δ	98.4%
Average Performance	89.5%

Table 8.3: Reverberation classification results for AR-AGS

Feature Vector	Classification Accuracy
LPCC	59.9%
LPCC+Δ	83.4%
MFCC	76.8%
MFCC+Δ	93.9%
Filter bank	82.6%
Filter bank +Δ	98.7%
Average Performance	82.5%

Table 8.4: Reverberation classification results for SM

Feature Vector	Classification Accuracy
LPCC	54.4%
LPCC+Δ	81.3%
MFCC	69.2%
MFCC+Δ	91.3%
Filter bank	75.7%
Filter bank +Δ	97.5%
Average Performance	78.2%

Table 8.5: Reverberation classification results for DS

Feature Vector	Classification Accuracy
LPCC	54.1%
LPCC+Δ	79.9%
MFCC	71.0%
MFCC+Δ	91.6%
Filter bank	76.7%
Filter bank +Δ	98.1%
Average Performance	78.6%

It is clear that based on overall average performance that AR-Itakura performed the best as a reverberation classification method. This probably stems from the fact that this method suffers the most compared to all other methods when training speech is reverberant and test speech is non- reverberant. This is due to the fact that the AR vector model captures information regarding the reverberation in its AR model. The AR model trained from clean speech has a significant mismatch from the AR model trained at test time from reverberant speech. The AR models trained from different reverberant speech are therefore different enough from each other that they can be classified.

Filter bank outputs appended with delta cepstral coefficients perform better than any of the other features for reverberation classification. In fact any feature vector

appended with delta cepstral coefficients outperforms the same feature vector without delta cepstral coefficients. This further illustrates the fact that delta cepstral coefficients capture information from reverberation. Apparently filter bank outputs that are not de-correlated capture information about reverberation. It is clear from the results that the GMM is the least capable of differentiating between the reverberant speech sources. It may be that this is because the GMM is the least sensitive method to reverberation.

Chapter Nine
Conclusion and Future Work

This thesis investigated the effects of reverberation on speaker recognition and verification techniques. The most commonly used feature vectors, namely LPCC and MFCC vectors were tested. It was found in the majority of methods with the exception of the MLP, that LPCC vectors gave the better performance. The effect of delta cepstral coefficients on the speaker recognition and verification with reverberant training speech and non-reverberant test speech was investigated and it was found that for all methods except AR-Itakura that these degraded performance. The reason for this is most likely that delta cepstral coefficients measure the slope in time of the spectrum. This quantity is different in reverberant and non-reverberant speech due to the forward smearing in time of the spectrum of reverberant speech.

Different methods were compared and it was found that the best performing method under conditions of reverberation is the GMM. The worst method was AR-Itakura. AR-AGS which is similar to AR-Itakura in the sense that it is an auto

regressive vectors method did not suffer a performance degradation as severe as AR-Itakura due to the fact that the feature vectors are sorted in random order before training and testing does not model the evolution of the spectrum in time, as does AR-Itakura. A possible cause for the bad performance of the AR-Itakura method is the fact that the evolution in time of a spectrum that has been smeared in time, as is the case for reverberant speech will be different from one that is not smeared in time.

In order to combat the existence of reverberation in the test speech, training with reverberant speech was attempted. It was found that offered significant performance improvements. It was found however that when training with speech that was more reverberant than the test speech that some deterioration in performance may occur.

In order to determine the amount of reverberation in the training speech so that the trained models used should contain similar or less reverberation a method was devised to classify reverberant speech. It was found that using AR-Itakura as a reverberation classification method gave good performance. Filter bank outputs appended delta features were found to be the feature vector that could best classify reverberant speech.

With regard to possible future work, first of all it would be interesting to compare the methods on speech that has been recorded over telephone lines after having been spoken in a hands free environment using a real hands free phone in a reverberant room. It would be interesting to see how speech de-reverberation

algorithms perform when applied to reverberant test speech before speaker recognition or verification takes place.

Finally the use of different normalization techniques such as world background models could be attempted.

References

[1] G. Doddington, "Speaker recognition-identifying people by their voices", Proc. of the IEEE, Vol. 73, no. 11, pp. 1651-1664, Nov. 1985

[2] D. O'Shaughnessy, "Speaker recognition", IEEE ASSP Magazine, Vol. 3, No. 4, pp. 4-17, October 1986

[3] P. Castellano, S. Sridharan and D. Cole, "Speaker recognition in reverberant enclosures", in Proc. ICASSP, pp. 117-120, May 1996

[4] H. G. Hirsch, "Automatic speaker and speech recognition in rooms", 7th FASE symposium, pp. 897-903, 1988

[5] Q. Lin, E. Jan, J. Flanagan, "Microphone arrays and speaker identification", IEEE Transactions on Speech and Audio Processing, Vol. 2, Issue: 4, pp. 622-629, Oct. 1994

[6] J. Ortega-Garcia, J. Gonzalez Rodriquez, "Overview of speech enhancement techniques for automatic speaker recognition", in Proc. ICSLP, Vol. 2, pp. 929-932, Oct 1996

[7] J. Gonzalez Rodriquez, J. Ortega-Garcia, "Increasing robustness in GMM speaker recognition systems for noisy and reverberant speech with low complexity microphone arrays", in Proc. ICSLP, Vol. 3, pp. 1333-1336, Oct 1996

[8] I. McCowan, J. Pelecanos, S. Sridharan, "Robust speaker recognition using microphone arrays", in Proc. 2001: A Speaker Odyssey, pp. 101-106, June 2001

[9] R. Auchenthaler, M. Carey, H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems", Digital Signal Processing, Vol. 10, pp. 42-54, 2000

[10] D. Reynolds, "An overview of automatic speaker recognition technology", in Proc. ICASSP, Vol. 4, pp. 13-17, May 2002

[11] D. Reynolds, T. Quatieri, and R. Dunn, "Speaker verification using adapted Gaussian mixture models", in Digital Signal Processing, Vol. 10, Academic Press, pp. 19-41, 2000

[12] L.P. Cordella, P. Foggia, C. Sansone, M. Vento, "A real-time text-independent speaker identification system", in Proc. ICIAP, Sept. 2003

[13] S. Furui, "Recent advances in speaker recognition", Audio-and Video-based Biometric Person Authentication, AVBPA, pp. 237-252, March 1997

[14] G. Doddington, Mark A. Prybocki, Alvin F. Martin, Douglas A Reynolds, "The NIST speaker recognition evaluation-overview, methodology,

systems, results, perspective", Speech Communication, Vol. 31, No. 2-3, pp. 225-254, 2000

[15] Nikos Tsourakis, "Speech Recognition", July 2002, http://www.telecom.tuc.gr/~ntsourak/tutorial_sr.htm

[16] F. Bimbot, J. F. Bonastre, C. Fredouille, G. Gravier, I. M. Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia., D. Petrovska-Delacretaz and D. Reynolds, "A Tutorial on text-independent speaker verification", EURASIP Journal on Applied Signal Processing, Vol. 4, pp. 430-451, 2004

[17] D. Reynolds, R. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models", IEEE Transactions on Speech and Audio Processing, Vol. 3, Issue: 1, pp. 72-83, Jan 1995

[18] M. Greenwood, A Kinghorn, "SUVing: automatic silence/unvoiced/voiced classification of speech", http://www.dcs.shef.ac.uk/~mark/uni/speech1.pdf

[19] S. Davis, P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences", IEEE Transactions on Signal Processing, Vol. 28, Issue 4, pp. 357-366, Aug. 1980

[20] S. Young et al, "The HTK (for HTK Version 3.2)", Dec. 2002; http://htk.eng.cam.ac.uk/prot-docs/HTKBook/htkbook.html

[21] E.Wong and S.Sridharan, "Comparison of linear prediction cepstrum coefficients and mel-frequency cepstrum coefficients for language

identification", 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing, pp. 95-98, Hong Kong, May 2001

[22] M. Hunt, "Spectral signal processing for ASR", in Proceedings of the International Workshop on Automatic Speech Recognition and Understanding, Colorado, 1999

[23] J. Koolwaaij, iSpeak, "Speech Processing", May 2001, http://www.ispeak.nl/start.html?url=http://www.ispeak.nl/prfhtm/node12.html&n=1&ref=http://www.google.ca/search

[24] R. J. Mammone, X. Zhang, R. P. Ramachandran, "Robust speaker recognition a feature-based approach", Proc of IEEE Signal Processing Magazine, pp. 58-71, Sept. 1996

[25] S. Saito, K. Nakata, *Fundamentals of Speech Signal Processing*, Montreal: Academic Press, 1985.

[26] F. K. Soong, A. E. Rosenberg, "On the use of instantaneous and transitional spectral information in speaker recognition", IEEE Trans. ASSP, Vol. 36, pp. 871-879, June 1988,

[27] X. D. Huang, Y. Ariki, M. A. Jack, *Hidden Markov Models for Speech Recognition*, Melksham: Redwood Press, 1990

[28] Y. Linde, A. Buzo, R. M. Gray, "An algorithm for vector quantizer design", IEEE Trans. on Communications, Vol., No. 1, pp. 84-95, Jan 1980

[29] X. Wu, K. Zhang, "A better tree-Structured vector quantizer", in Proc. IEEE Data Compression Conference, pp. 392-401, April 1991

[30] F. K. Soong, A. E. Rosenberg, L. R. Rabiner, B. H. Juang, "A vector quantization approach to speaker recognition", in Proc. ICASSP, pp. 387-390, March 1985

[31] R. Stapert, J. S. Mason, R. Auckenthaler, "Optimization of GMM in speaker recognition", in Proc. ICSLP'2000, Oct. 2000

[32] J. Hollmen, "Principal component analysis", Mar 8 1996, http://www.cis.hut.fi/~jhollmen/dippa/node30.html

[33] J, N. Holmes, N. C. Sedgwick, "Noise compensation for speech recognition using probabilistic models", in Proc. ICASSP, Vol. 11, pp. 741-744, April 1986

[34] C. Griffin, T. Matsui, S. Furui, "Distance measures for text-independent speaker recognition based on MAR model", in Proc. ICASSP, Vol. 1, pp. 309-312, April 1994

[35] F. Bimbot, L. Mathan, A De Lima, G. Chollet, "Standard and target driven AR-vector models for speech analysis and speaker recognition", in Proc. ICASSP, Vol. 2, pp. 5-8, March 1992

[36] Y. Bennani, "Text-Independent talker identification system combining connectionist and conventional models", in Proc. IEEE Signal Processing Workshop, pp. 131-138, Sept. 1992

[37] I. Magrin-Chagnolleau, J. Wilke, F. Bimbot, "A further investigation on AR-vector models for text-independent speaker identification", in Proc. ICASSP, pp. 401-404, Jan. 1996

[38] C. Montacie, P. Deleglise, F. Bimbot, M. Caraty, "Cinematic techniques for speech processing: temporal decomposition and multivariate linear prediction", in Proc. ICASSP, pp. 153-156, March 1992

[39] C. Montacie, J. Le Floch, "AR-Vector models for free-text speaker recognition", in Proc. ICSLP, Vol. 1, pp. 611-614, October 1992

[40] C. de Lima, D. da Silva, A. Alcaim, J. Apolinario Jr., "AR-Vector Using CMS for Robust Text Independent Speaker Verification". in Proc. 14th International Conference on DSP, vol. 2, pp. 1073-1076, July 2002

[41] Besacier, J. F. Bonastre, "Time and frequency pruning for speaker identification", in Proc. On Speaker Recognition and its Commercial Forensic Applications (RLA2C), April 1998

[42] F. Itakura, "Minimum prediction residual principle applied to speech recognition", IEEE Transactions on Signal Processing, Vol. 23, no 1, pp. 67-72, Feb. 1975

[43] H. Bourlard, N. Morgan, "Speaker verification a quick overview", Dalle Molle Institute for Perceptual Artificial Intelligence, Martigny, Switzerland, Research Report: IDIAP-RR 98-12, August 1998

[44] N. Tishby, "On the application of mixture AR hidden Markov models to text independent speaker recognition", IEEE Trans. On Signal Processing, Vol. 39, No. 3, pp. 563-570, Mar. 1991

[45] K. Yu, J. Mason, J. Oglesby, "Speaker recognition using hidden Markov models, dynamic time warping and vector quantization", in IEE Proceedings Vision, Image and Signal Processing, Vol. 142, No. 5, pp. 323-318, Oct 1995

[46] T. Matsui, S. Furui, "Comparison of text-independent speaker recognition methods using VQ-distortion and discrete/continuous HMM's", IEEE Transaction on Speech and Audio Processing, Vol. 2. No. 3. July 1994

[47] M. Savic, S. K. Gupta, "Variable parameter speaker verification system based on hidden Markov modeling", in Proc. ICASSP, Vol. 1, April 1990, pp. 281-284

[48] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition", Proc. of the IEEE, Vol. 77, No. 2. pp. 256-286, 1989

[49] F. Bimbot, I. Magrin-Chagnolleau, L. Mathan, "Second-order statistical measures for text-independent speaker identification", Speech Communication, Vol. 17, No. 1-2, pp. 177-192, August 1995

[50] R. Zilca, "Text-independent speaker verification using covariance modeling", IEEE Signal Processing Leters, Vol. 8, No. 4, pp. 97-99, April 2001

[51] Zilca, Ran D. (2001): "Using second order statistics for text independent speaker verification", in ODYSSEY-2001, pp. 45-49, June 2001

[52] C. Alonso-Martinez, M. Faundez-Zanuy, "Speaker identification in mismatch training and testing conditions", in Proc. ICASSP, Vol. 2, pp. 1184-1185, June 2000

[53] M. Faundez-Zanuy, "A combination between VQ and Covariance matrices for speaker recognition", in Proc. ICASSP, Vol. 1., pp. 453-456, May 2001

[54] M. Faundez-Zanuy, "A comparative study of several parameterizations for speaker recognition", in Proc. EUSIPCO, Vol. 1, pp. 445-448, Sept. 2000

[55] A. Satué, M. Faúndez-Zanuy, "On the relevance of language in speaker recognition", in Proc. EUROSPEECH, Vol. 3 pp. 1231-1234, Sept. 1999

[56] S. van Vuuren, "Comparison of Text-Independent Speaker Recognition Methods on Telephone Speech with Acoustic Mismatch", in Proc. ICSLP, pp. 1784-1787, Oct. 1996

[57] Campbell, J.P., Jr., "Speaker recognition: a tutorial", in Proceedings of the IEEE, Vol. 85, No. 9, pp. 1437-1462, Sept. 1997

[58] Bennani, Y. Gallinari, P. "On the use of TDNN-extracted features information in talker identification", in Proc. ICASSP, Vol. 1, pp. 385-388, April 1991

[59] Mak, M.W., Allen W. G., and G. G. Sexton, "Speaker Identification using Radial Basis Functions," The 3rd IEE Int. Conf. on Artificial Neural Networks, pp. 138-142, May 1993

[60] D. Rodriquez-Porcheron, M. Faundez-Zanuy, "Speaker recognition with a MLP classifier and LPCC codebook", in Proc. ICASSP, Vol. 2, pp. 1005-1008, March 1992

[61] M. W. Mak, S. Y. Kung, "Estimation of Elliptical Basis Function Parameters by the EM Algorithm with Application to Speaker Verification", IEEE Transactions on Neural Networks, Vol. 11, no.4, pp. 961-969, July 2000

[62] U. Ig-Tae, R. Jong-Hei, K. Moon-Hyun, "Comparison of clustering methods for MLP-based speaker verification", in Proc. 15th International Conference on Pattern Recognition, Vol. 2., pp. 475-478, Sep. 2000

[63] T. Masters, "Advanced algorithms for neural networks: A C++ Sourcebook", John Wiley & Sons, New York, pp. 47-71

[64] J. B. Allen, D. A. Berkley, "Image method for efficiently simulating small-room acoustics", Journal of the Acoustical Society of America, Vol. 65. No. 4., April 1979, pp. 943-950, April 1979

[65] J. P. Campbell, D. A. Reynolds, "Corpora for the evaluation of speaker recognition systems", in Proc. ICASSP, pp. 829-832, March 1999

[66] National Institute of Standards and Technology (NIST), *Brief Description of the KING Speech Data Base*, NIST, 1992

[67] M. Kahrs, K. Brandenburg, *Applications of Digital Signal Processing to Audio and Acoustics,* Boston: Kluwer Academic Publishers, 1998

[68] N. W. D. Evans, J. S. Mason, R. Auckenthaler and R. Stapert, "Assessment of speaker verification due to packet loss in the context of wireless mobile devices", in Proc. COST 275 Workshop - The Advent of Biometrics on the Internet, pp. 47-50, Nov. 2002

[69] I. M. Changnoleau, G. Gravier, R. Blouet, "Overview of the 2000-2001 ELISA consortium research activities", http://citeseer.ist.psu.edu/562444.html

[70] J. McLaughlin, D. A. Reynolds and T. Gleason, "A Study of Computation Speed-Ups of the GMM-UBM Speaker Recognition System", in Proc. EUROSPEECH, Vol. 3, pp. 1215-1218, Sept. 1999

[71] J. Gammal, R. Goubran, "Speaker recognition in reverberant environments", in Proc., Canadian Acoustics, Vol. 32, pp. 134-135, September 2004

[72] J. Gammal, R. Goubran, "Combating reverberation in speaker verification", submitted to IMTC 2005

Appendix A
Simulations for Determination of Vector Length for Non-Reverberant Speech

As the vector length was varied the performance of each method in recognition on clean speech varies. It was necessary to compare the methods in such a fashion that each method was using parameters that give the best results. This allows for the comparison of all methods at their maximum performance. The following graphs illustrate closed set recognition performance as the vector length was varied. Training was performed with the first 3 sessions and testing was performed using 30s segments of the final 7 sessions.

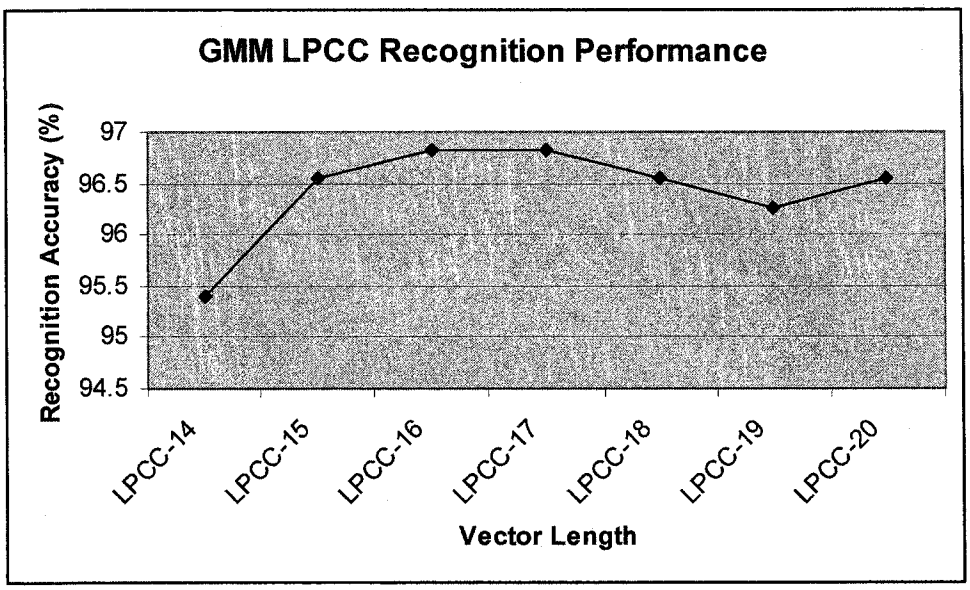

Figure A.1: Closed set recognition performance for GMM with LPCC vector.

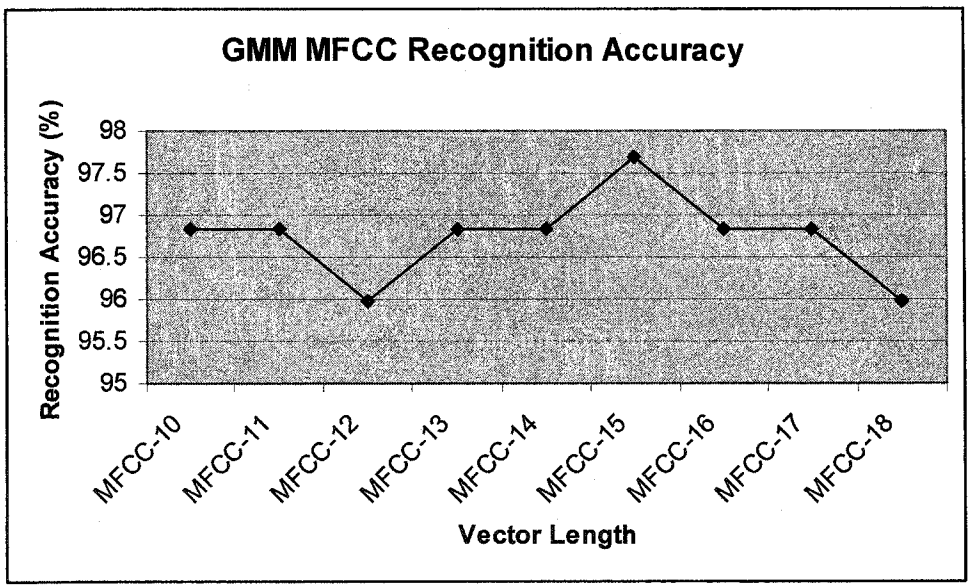

Figure A.2: Closed set recognition performance for GMM with MFCC vector.

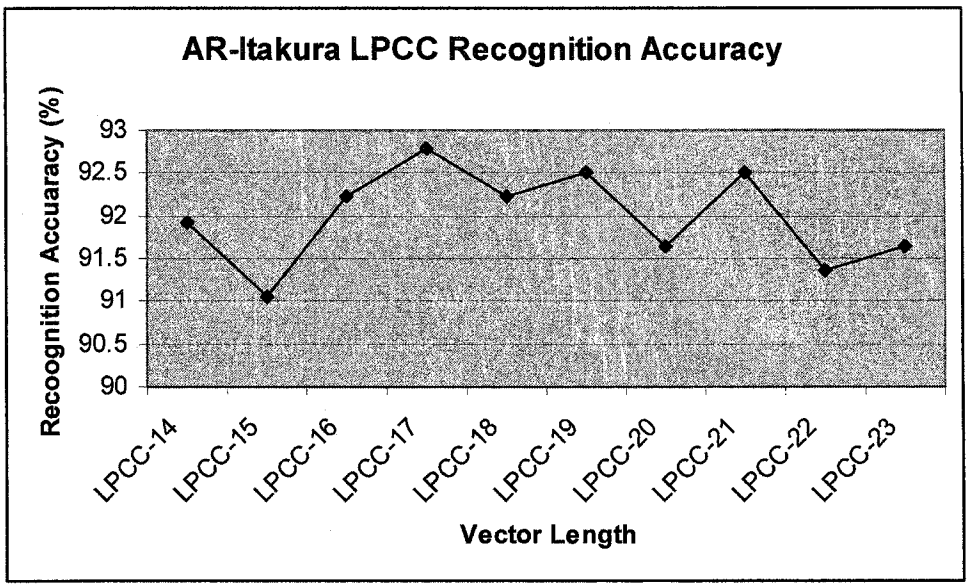

Figure A3: Closed set recognition performance for AR-ITAKURA with LPCC vector.

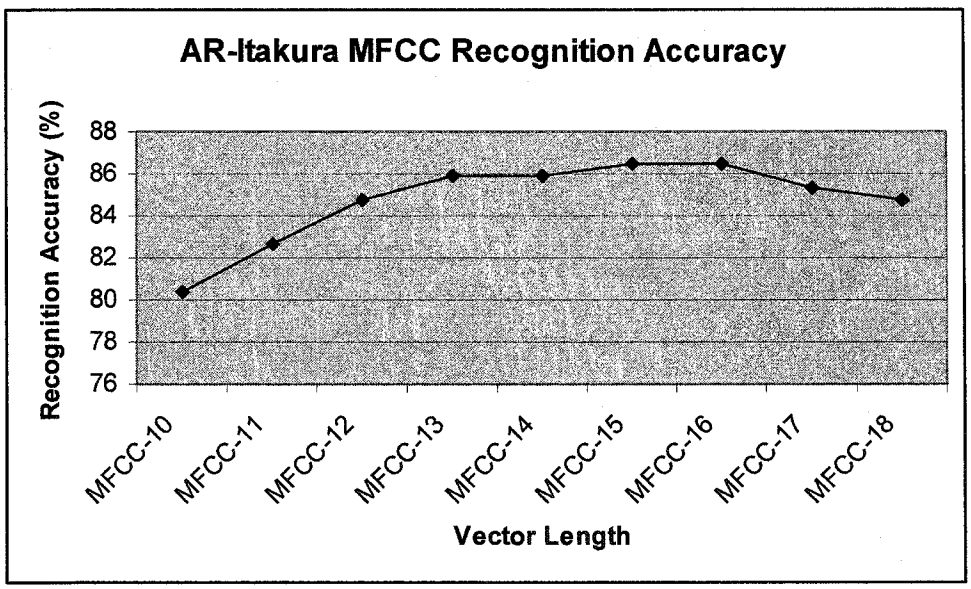

Figure A4: Closed set recognition performance for AR-ITAKURA with MFCC vector.

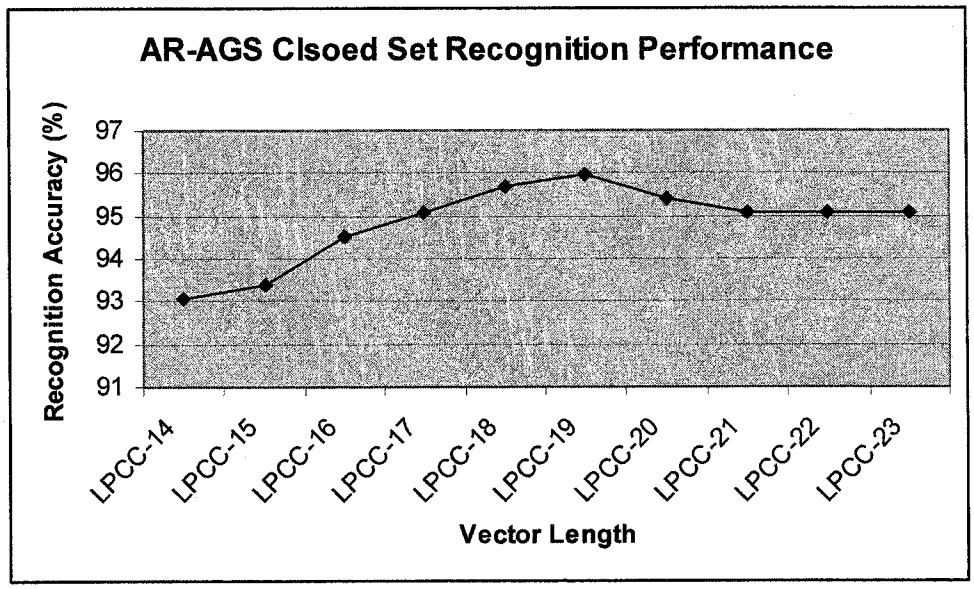

Figure A5: Closed set recognition performance for AR-AGS with LPCC vector.

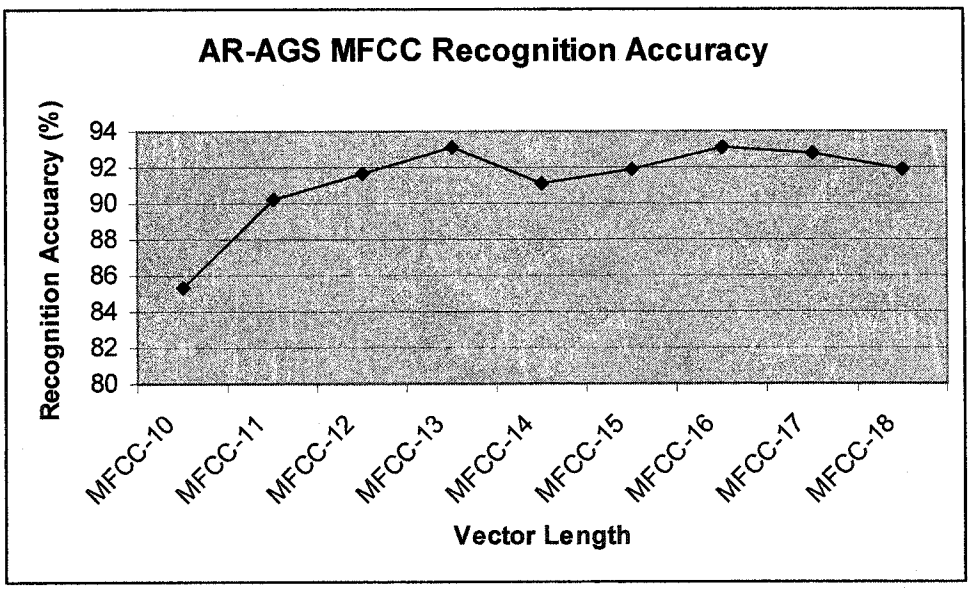

Figure A.6: Closed set recognition performance for AR-AGS with MFCC vector.

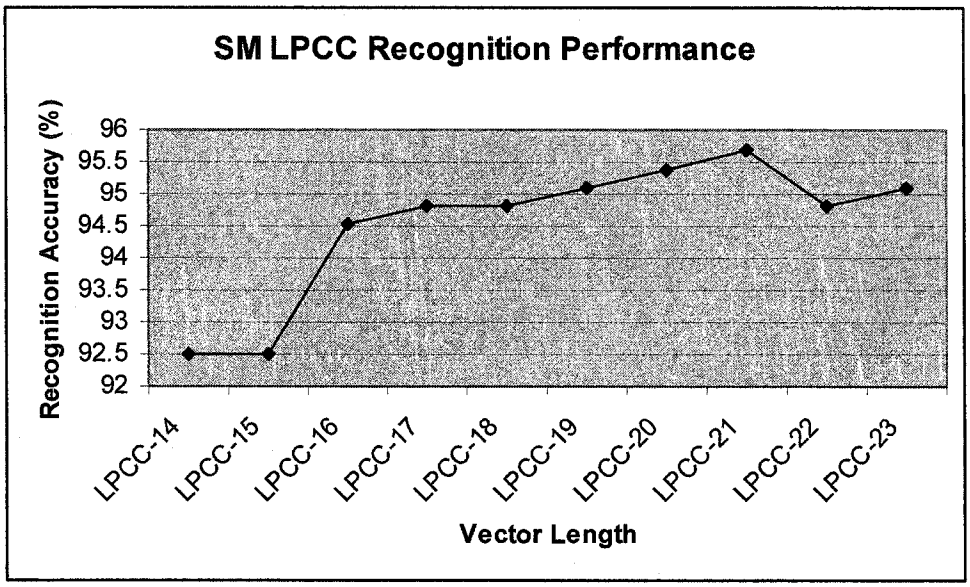

Figure A.7: Closed set recognition performance for SM with LPCC vector.

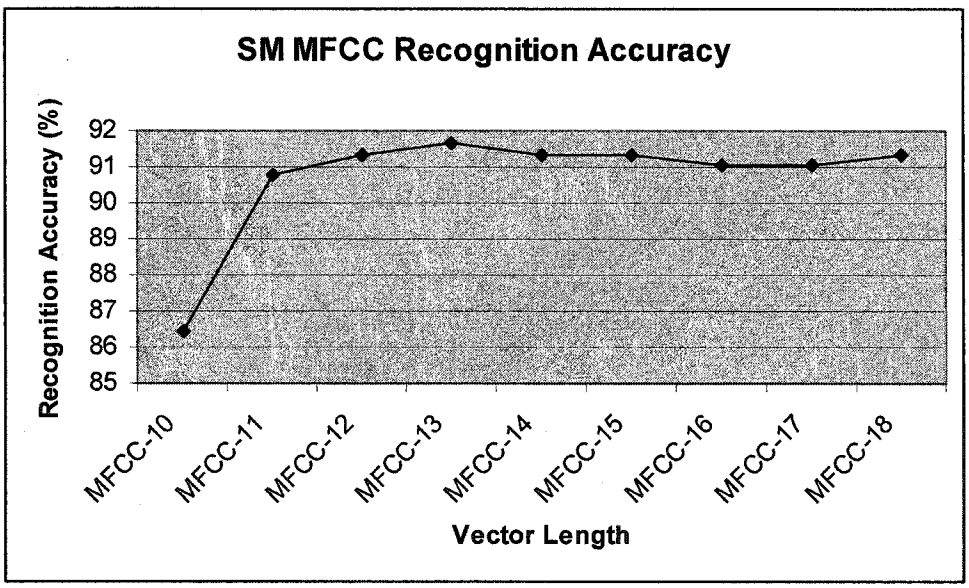

Figure A.8: Closed set recognition performance for SM with MFCC vector.

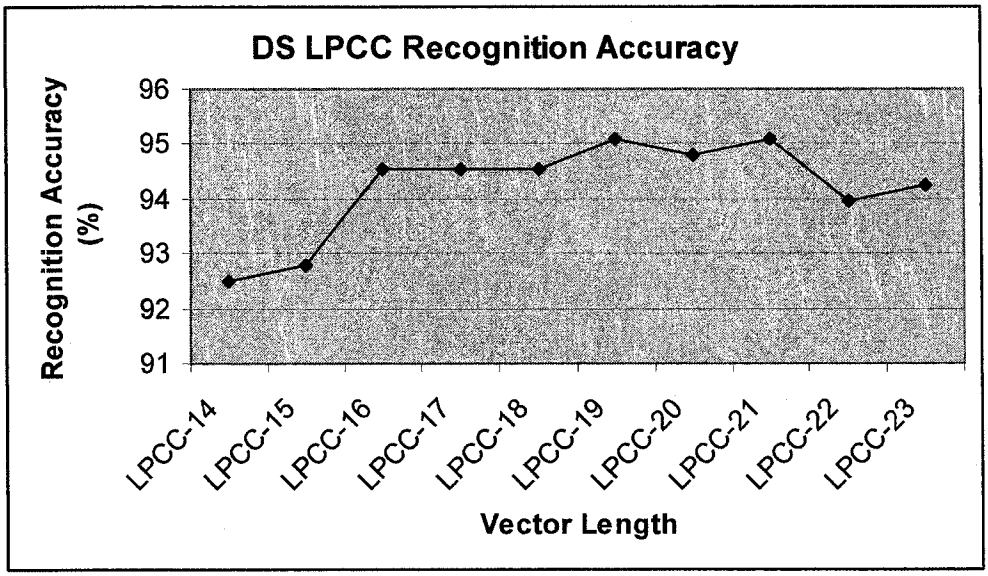

Figure A.9: Closed set recognition performance for DS with LPCC vector.

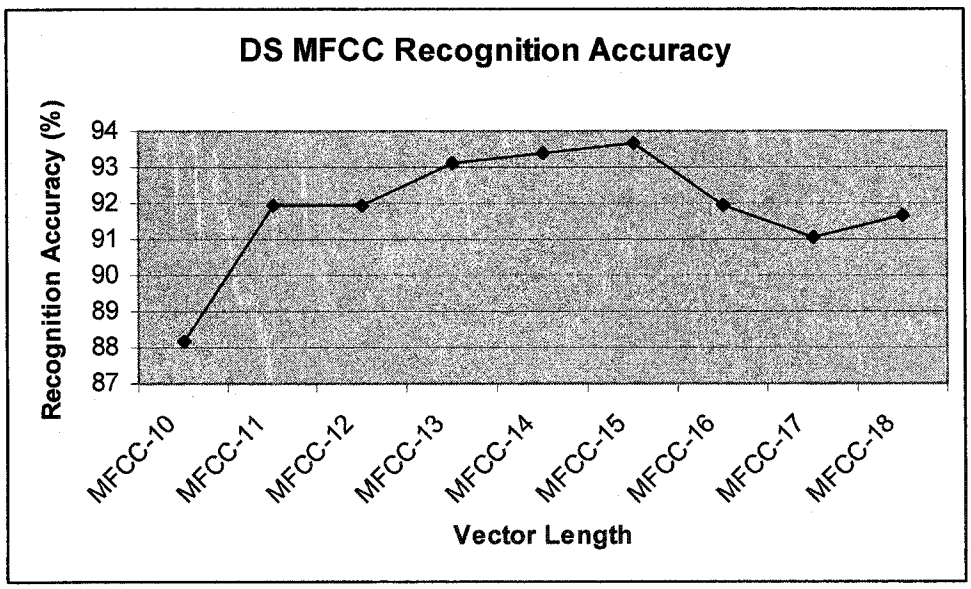

Figure A.10: Closed set recognition performance for DS with MFCC vector.

Appendix B
Simulations for Determination of Vector Length for Reverberant Speech

Simulations were conducted for each method and feature vector combination to determine which vector length gives the best closed set recognition performance when the training speech is non-reverberant and the test speech is reverberant. The following section illustrates the performance of each method and feature vector as the vector length was varied. The statistics shown are the closed set recognition accuracy computed over the 5 reverberant conditions. Training was performed with the first 3 sessions and testing was performed using 30s segments of the final 7 sessions.

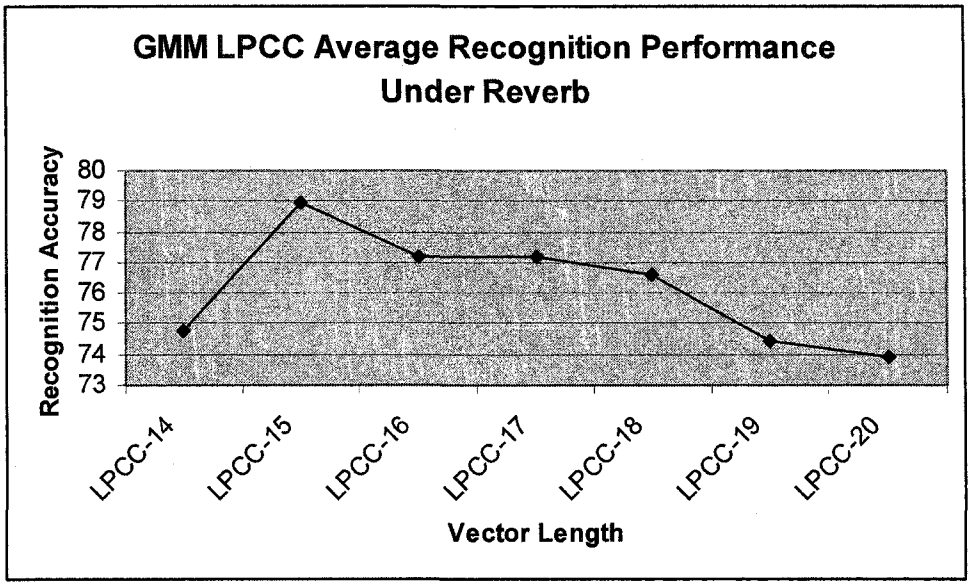

Figure B.1: GMM LPCC average closed set recognition performance over 5 reverberant room conditions.

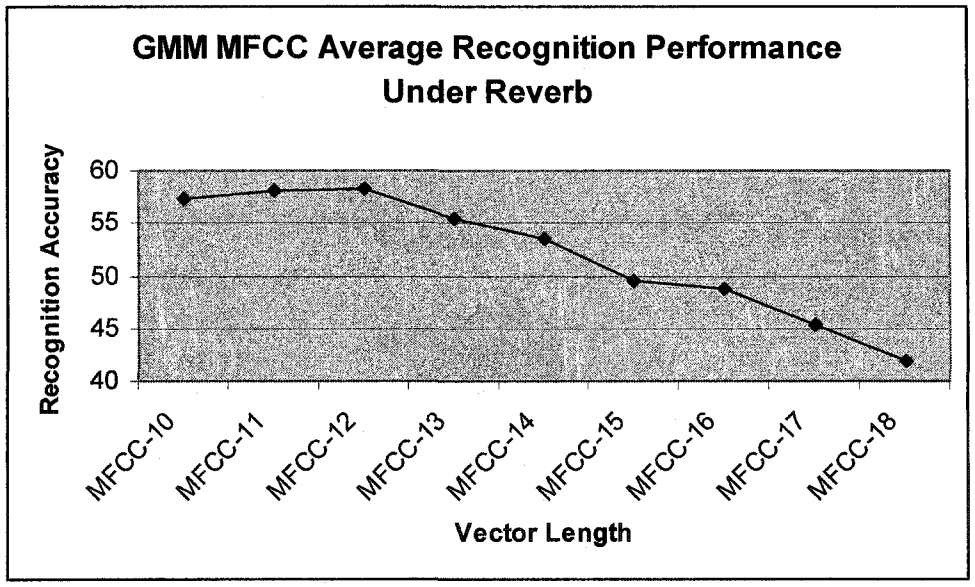

Figure B.2: GMM MFCC average closed set recognition performance over 5 reverberant room conditions.

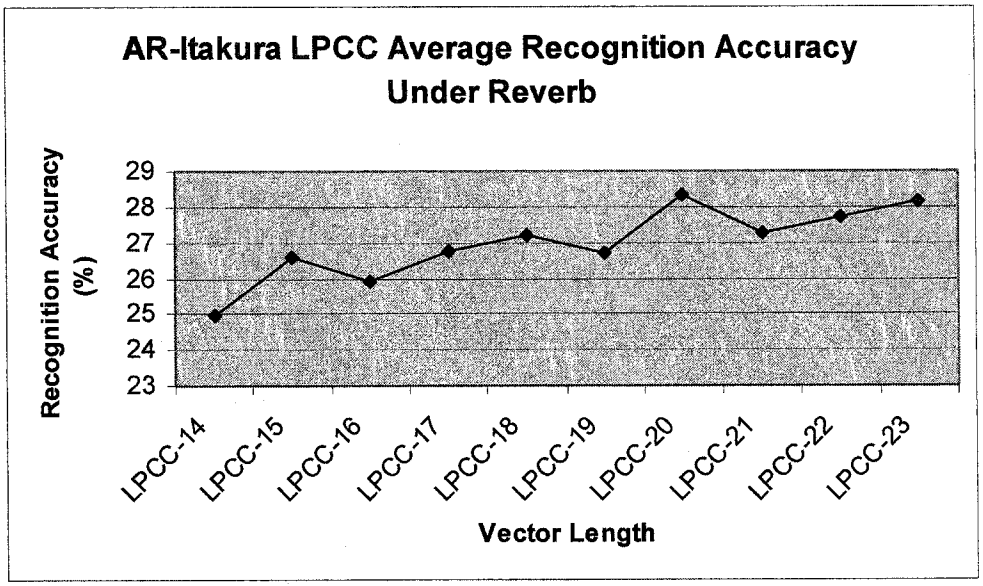

Figure B.3: AR-Itakura LPCC average closed set recognition performance over 5 reverberant room conditions.

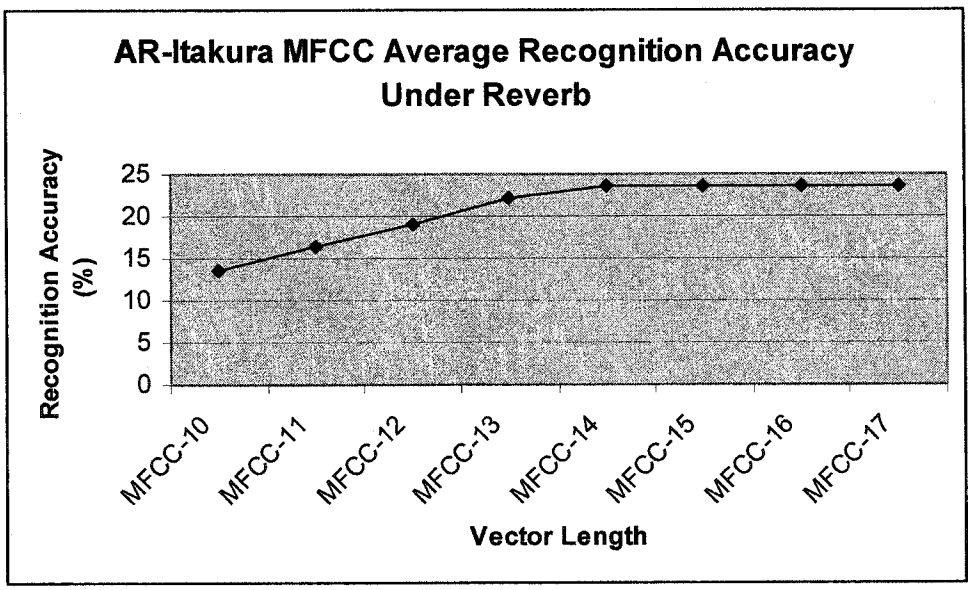

Figure B.4: AR-Itakura MFCC average closed set recognition performance over 5 reverberant room conditions.

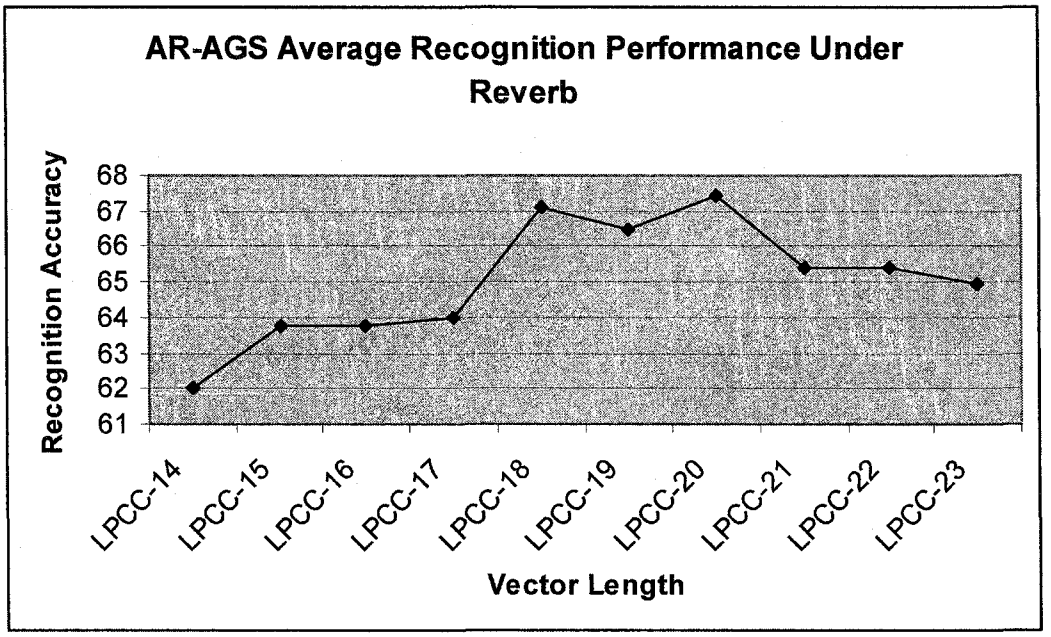

Figure B.5: AR-AGS LPCC average closed set recognition performance over 5 reverberant room conditions.

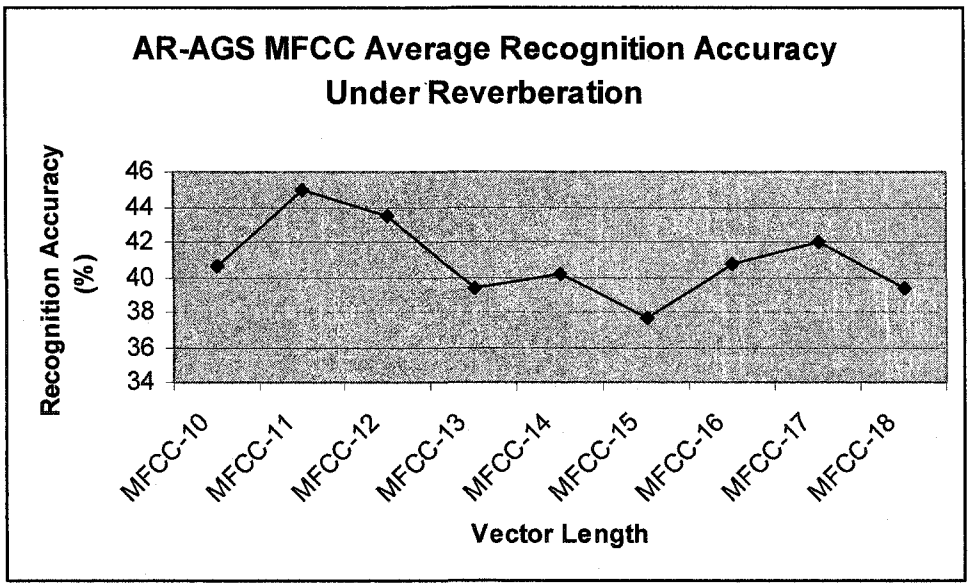

Figure B.6: AR-AGS MFCC average recognition performance over 5 reverberant room conditions.

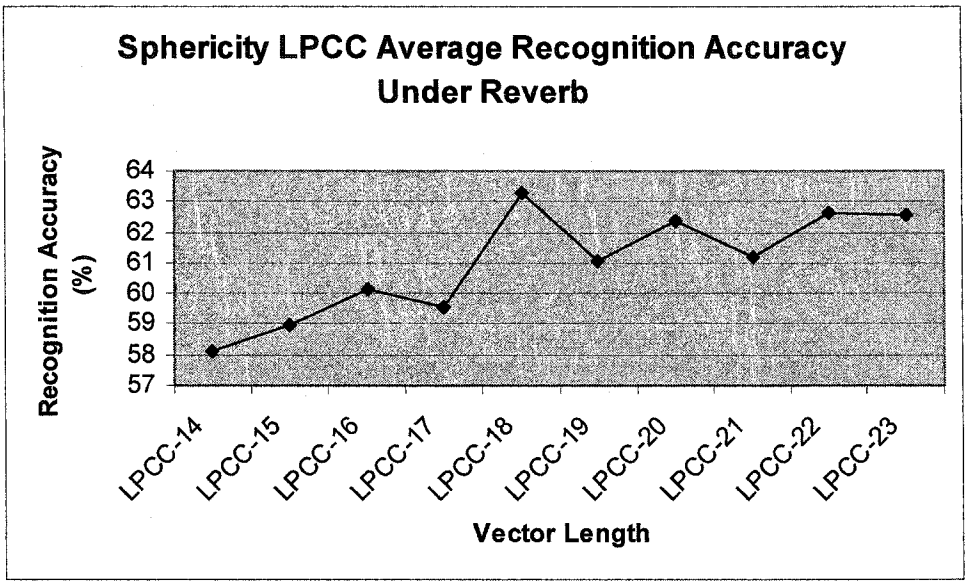

Figure B7: SM LPCC average closed set recognition performance over 5 reverberant room conditions.

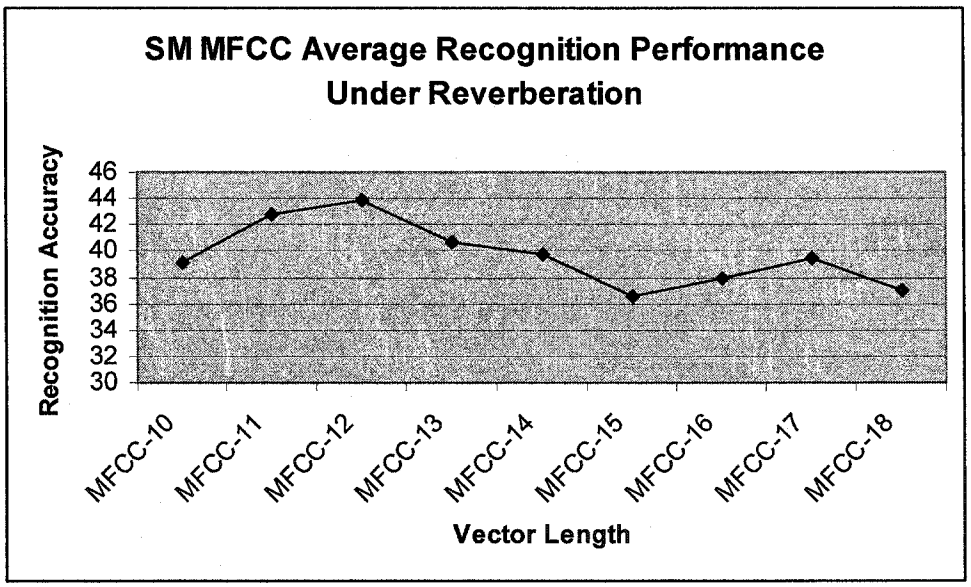

Figure B.8: SM MFCC average closed set recognition performance over 5 reverberant room conditions.

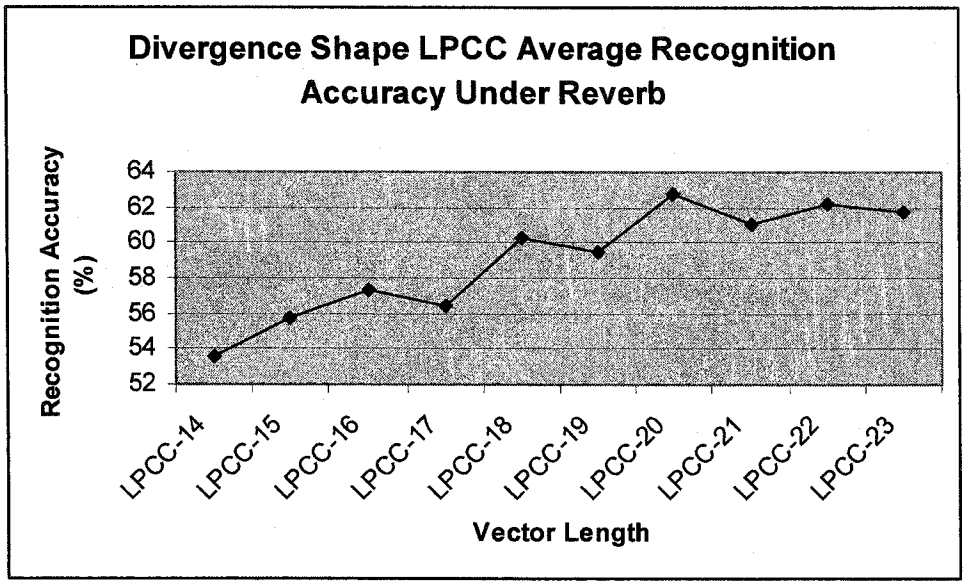

Figure B.9: DS LPCC average closed set recognition performance over 5 reverberant room conditions.

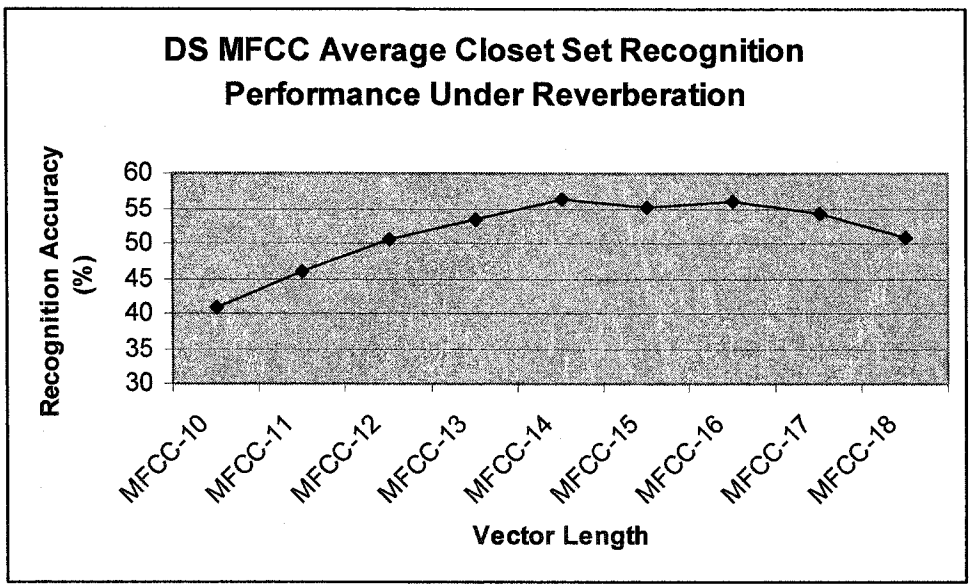

Figure B.10: DS MFCC average closed set recognition performance over 5 reverberant room conditions.